Integrated view and comparison of alignment semantics

Pascal Hitzler, Jérôme Euzenat, Markus Krötzsch, Luciano Serafini, Heiner Stuckenschmidt, Holger Wache, Antoine Zimmermann

D2.2.5 Integrated view and comparison of alignment semantics

Coordinator: Pascal Hitzler (UKARL)
Jérôme Euzenat (INRIA)
Markus Krötzsch (UKARL)
Luciano Serafini (U. Trento)
Heiner Stuckenschmidt (VU Amsterdam)
Holger Wache (VU Amsterdam)
Antoine Zimmermann (INRIA)

Abstract.
We take a general perspective on alignment in order to develop common theoretical foundations for the subject. The deliverable comprises a comparative study of different mapping languages by means of distributed first-order logic, and a study on category-theoretical modelling of alignment and merging by means of pushout-combinations.

Keyword list: semantic alignment, data integration, mappings, merging, description logics, category theory

Document Identifier	KWEB/2005/D2.2.5
Project	KWEB EU-IST-2004-507482
Version	v1.0
Date	January 26, 2006
State	final
Distribution	public

Knowledge Web Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2004-507482.

University of Innsbruck (UIBK) - Coordinator
Institute of Computer Science
Technikerstrasse 13
A-6020 Innsbruck
Austria
Contact person: Dieter Fensel
E-mail address: dieter.fensel@uibk.ac.at

École Polytechnique Fédérale de Lausanne (EPFL)
Computer Science Department
Swiss Federal Institute of Technology
IN (Ecublens), CH-1015 Lausanne
Switzerland
Contact person: Boi Faltings
E-mail address: boi.faltings@epfl.ch

France Telecom (FT)
4 Rue du Clos Courtel
35512 Cesson Sévigné
France. PO Box 91226
Contact person : Alain Leger
E-mail address: alain.leger@rd.francetelecom.com

Freie Universität Berlin (FU Berlin)
Takustrasse 9
14195 Berlin
Germany
Contact person: Robert Tolksdorf
E-mail address: tolk@inf.fu-berlin.de

Free University of Bozen-Bolzano (FUB)
Piazza Domenicani 3
39100 Bolzano
Italy
Contact person: Enrico Franconi
E-mail address: franconi@inf.unibz.it

Institut National de Recherche en Informatique et en Automatique (INRIA)
ZIRST - 655 avenue de l'Europe -
Montbonnot Saint Martin
38334 Saint-Ismier
France
Contact person: Jérôme Euzenat
E-mail address: Jerome.Euzenat@inrialpes.fr

Centre for Research and Technology Hellas / Informatics and Telematics Institute (ITI-CERTH)
1st km Thermi - Panorama road
57001 Thermi-Thessaloniki
Greece. Po Box 361
Contact person: Michael G. Strintzis
E-mail address: strintzi@iti.gr

Learning Lab Lower Saxony (L3S)
Expo Plaza 1
30539 Hannover
Germany
Contact person: Wolfgang Nejdl
E-mail address: nejdl@learninglab.de

National University of Ireland Galway (NUIG)
National University of Ireland
Science and Technology Building
University Road
Galway
Ireland
Contact person: Christoph Bussler
E-mail address: chris.bussler@deri.ie

The Open University (OU)
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA
United Kingdom
Contact person: Enrico Motta
E-mail address: e.motta@open.ac.uk

Universidad Politécnica de Madrid (UPM)
Campus de Montegancedo sn
28660 Boadilla del Monte
Spain
Contact person: Asunción Gómez Pérez
E-mail address: asun@fi.upm.es

University of Karlsruhe (UKARL)
Institut für Angewandte Informatik und Formale
Beschreibungsverfahren - AIFB
Universität Karlsruhe
D-76128 Karlsruhe
Germany
Contact person: Rudi Studer
E-mail address: studer@aifb.uni-karlsruhe.de

University of Liverpool (UniLiv)
Chadwick Building, Peach Street
L697ZF Liverpool
United Kingdom
Contact person: Michael Wooldridge
E-mail address: M.J.Wooldridge@csc.liv.ac.uk

University of Manchester (UoM)
Room 2.32. Kilburn Building, Department of Computer
Science, University of Manchester, Oxford Road
Manchester, M13 9PL
United Kingdom
Contact person: Carole Goble
E-mail address: carole@cs.man.ac.uk

University of Sheffield (USFD)
Regent Court, 211 Portobello street
S14DP Sheffield
United Kingdom
Contact person: Hamish Cunningham
E-mail address: hamish@dcs.shef.ac.uk

University of Trento (UniTn)
Via Sommarive 14
38050 Trento
Italy
Contact person: Fausto Giunchiglia
E-mail address: fausto@dit.unitn.it

Vrije Universiteit Amsterdam (VUA)
De Boelelaan 1081a
1081HV. Amsterdam
The Netherlands
Contact person: Frank van Harmelen
E-mail address: Frank.van.Harmelen@cs.vu.nl

Vrije Universiteit Brussel (VUB)
Pleinlaan 2, Building G10
1050 Brussels
Belgium
Contact person: Robert Meersman
E-mail address: robert.meersman@vub.ac.be

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to writing parts of this document:

Centre for Research and Technology Hellas
École Polytechnique Fédérale de Lausanne
Free University of Bozen-Bolzano
Institut National de Recherche en Informatique et en Automatique
National University of Ireland Galway
Universidad Politécnica de Madrid
University of Innsbruck
University of Karlsruhe
University of Manchester
University of Sheffield
University of Trento
Vrije Universiteit Amsterdam
Vrije Universiteit Brussel

Changes

Version	Date	Author	Changes
0.1	10-07-2005	Pascal Hitzler (UKARL)	creation
0.2	09-08-2005	Pascal Hitzler (UKARL)	input chaps 2 and 3
0.3	29-10-2005	Pascal Hitzler (UKARL)	revisions chaps 2 and 3
			abstract, summary, intro & conclusions added
0.4	19-11-2005	Pascal Hitzler (UKARL)	minor revisions and delivered to quality control
1.0	26-01-2006	Pascal Hitzler (UKARL)	revisions according to quality control

Executive Summary

For the currently evolving semantic web, a considerable number of different formalisms for alignment and merging of heterogeneous distributed data have been proposed. They differ in form and semantics due to different intuitions, application domains and use cases for which they were developed. These different formalisms are obviously related, but they have rarely been studied in a comparative manner, and attempts to unify them into a single framework are largely missing. Both the researcher and the practitioner concerned with ontology alignment, mapping, and merging are thus faced with a plethora of different choices, without having access to guidelines or comparative work which can help them locate methods within a larger context, compare them, and make sensible decisions as to the formalisms to be applied or further investigated.

In this deliverable, we therefore provide two comparative studies of ontology alignment and merging. Both aim at establishing a unifying perspective on alignment formalisms, but differ in the level of abstraction taken.

The first study provides a concrete unification of different alignment formalisms by means of distributed first-order logic (DFOL). The approach covers C-OWL, the Ontology Integration Framework (OIS), DL for Information Integration (DLII), and e-connections. By representing all of these approaches by means of DFOL, their formal differences are exposed. This work can provide guidelines for the use and for the further development of different alignment formalisms.

The second study is more abstract in nature. It provides a general perspective on ontology alignment and merging by means of category-theory, more precisely by the modelling of merging by categorical pushouts. Simple so-called V-alignments can be expressed by a single pushout, while more complex W-alignments consist of a combination of three pushouts. Algebraic operations for alignment are presented, with a focus on combining alignments. This work can provide the foundations for an in-depth abstract analysis of semantic alignment techniques.

The two studies are compatible in the sense that the first can be considered an instance of the second.

The studies presented in this deliverable thus advance the state of the art on ontology alignment by providing bird's eye perspectives, and can serve as a foundation for further studies into the subject.

Contents

Chapter 1

Introduction

For the currently evolving Semantic Web, a considerable number of different formalisms for alignment and merging of heterogeneous distributed data have been proposed. They differ in form and semantics due to different intuitions, application domains and use cases for which they were developed. These different formalisms are obviously related, but they have rarely been studied in a comparative manner, and attempts to unify them into a single framework are largely missing. Both the researcher and the practitioner concerned with ontology alignment, mapping, and merging are thus faced with a plethora of different choices, without having access to guidelines or comparative work which can help them locate methods within a larger context, compare them, and make sensible decisions as to the formalisms to be applied or further investigated.

This deliverable is a general continuation of the theme started in the deliverables D2.2.1v1 and D2.2.1v2. We provide two comparative studies of ontology alignment and merging. They shall

- aid the community to focus on compatible and meaningful approaches,
- help researchers and practitioners to navigate the research area,
- lay the foundations to identify conceptually sound and strong work,
- and serve as a foundation for grounding alignment research in a sound theory.

Both studies presented in this deliverable aim at establishing a unifying perspective on alignment formalisms, but differ in the level of abstraction taken. The first study, presented in Chapter 2 provides a concrete unification of different alignment formalisms by means of distributed first-order logic. It covers distributed description logics/C-OWL, the ontology integration framework OIS, Description Logics for Information Integration DLII, and ϵ-connections, and compares these by means of logical expressiveness.

The second study, presented in Chapter 3 is more abstract in nature and provides a general perspective on ontology alignment and merging by means of category-theory, more precisely by the modelling of merging by combining categorical pushouts. The study abstracts from specific ontology languages and describes simple alignments by so-called V-alignments, consisting of a single pushout, and complex alignments by so-called W-alignments, which consist of the combination of several pushouts. This part of the deliverable is a direct extension of a part of deliverable D2.2.1v2.

The two studies presented are compatible in the sense that alignments in distributed first-order logic can be expressed by W-alignments, i.e. alignments as in the language-dependent first study are specific instances of W-alignments as in the more abstract and language-independent second study.

The deliverable will close with conclusions in Chapter 4.

Chapter 2

A Formal Investigation of Mapping Languages for Terminological Knowledge

The benefits of using ontologies as explicit models of the conceptualization underlying information sources has widely been recognized. Meanwhile, a number of logical languages for representing and reasoning about ontologies have been proposed and there are even language standards now that guarantee stability and homogeneity on the language level. At the same time, the need to represent ontology alignment by means of mappings between different ontologies has been recognized as a result of the fact that different ontologies may partially overlap or even represent the same domain from different points of view [Bouquet *et al.*, 2004a]. As a result, a number of proposals have been made for extending ontology languages with notions of mappings between different models. Unlike for the case of ontology languages, work on languages to represent ontology mappings has not yet reached a state where a common understanding of the basic principles exists. As a consequence, existing proposals show major differences concerning almost all possible aspects of such languages. This makes it difficult to compare approaches and to make a decision about the usefulness of a particular approach in a given situation.

The purpose of this work is to provide a better understanding of the commonalities and differences of existing proposals for ontology mapping languages. We restrict our attention to logic-based approaches that have been defined as extensions of existing formalisms for representing Terminological Knowledge. In particular, we chose approaches that extend description logics (DL) with notions of mappings between different T-boxes. The rationale for this choice is the fact that DLs are a widely agreed standard for describing terminological knowledge. In particular, DLs have gained a lot of attention as a standardized way of representing ontologies on the Semantic Web [Horrocks *et al.*, 2003].

Approach and Contributions

We encode the different mapping languages in an extended version of distributed first-order logic (DFOL), a logical framework for representing distributed knowledge systems [Ghidini and Serafini, 2000]. DFOL consists of two components: a family of first order theories and a set of axioms describing the relations between these theories. As most proposals for mapping languages are based on a subset of first-order logic for describing local models and mappings with a particular semantics for the connections between models, these mapping languages can be expressed in distributed first order logic in the following way:

- restrictions on the use of first order sentences for describing domain models
- the form of axioms that can be used for describing relations between domain models
- axioms describing the assumptions that are encoded in the specific semantics of mappings

Encoding the different mapping approaches in first-order logic in this way has several advantages for an analysis and comparison of existing work. In particular it allows us to do a formal analysis and comparison of different approaches in a uniform logical framework. In the course of the investigations, we make the following contributions to the state of the art in distributed knowledge representation and reasoning:

- we show how the DFOL formalism can be used to model relations between heterogeneous domains
- we encode existing mapping approaches in a common framework, making them more comparable
- we make hidden assumptions explicit in terms of distributed first order logic axioms
- we provide first results on the relative expressiveness of the approaches and identify shared fragments

The chapter is structured as follows. In section 2.1 we introduce distributed first order logic as a general model for describing distributed knowledge systems. We explain the intuition of the logic and introduce its syntax and semantics. In section 2.2 we describe how the different mapping approaches can be encoded in distributed first order language. Here we will focus on the representation of mappings and the encoding of hidden assumptions. In section 2.3 we compare the different approaches based on their encoding in DFOL and discuss issues such as relative expressiveness and compatibility of the different approaches and conclude with a summary of our findings and open questions.

2.1 Distributed First-Order Logic

This section introduces distributed first order logic as a basis for modeling distributed knowledge bases. More details about the language including a sound and complete calculus can be found in [Ghidini and Serafini, 2005].

Let $\{L_i\}_{i\in I}$ (in the following $\{L_i\}$) be a family of first order languages with equality defined over a non-empty set I of indexes. Each language L_i is the language used by the i-th knowledge base (ontology). $\{L_i\}$ may intersect — but do not need to intersect. The signature of L_i is extended with a new set of symbols used to denote objects which are related with other objects in different ontologies. For each variable x, and each index $j \in I$ with $j \neq i$ we have two new symbols $x^{\to j}$ and $x^{j\to}$, called *arrow variables*. Terms and formulas of L_i, also called *i-terms* and *i-formulas* are defined in the usual way. Quantification on arrow variables is not permitted. The notation $\phi(\mathbf{x})$ is used to denote the formula ϕ and the fact that the free variables of ϕ are $\mathbf{x} = \{x_1, \ldots, x_n\}$. In order to distinguish occurrences of terms and formulas in different languages we label them with their index. The expression $i:\phi$ denotes the formula ϕ of the i-th knowledge base.

The semantics of DFOL is an extension of Local Models Semantics defined in [Ghidini and Giunchiglia, 2001]. Local models are defined in terms of first order models. To capture the fact that certain predicates are completely known by the i-th sub-system we select a sub-language of L_i containing the equality predicate, denoted as L_i^c, which we call the *complete fragment* of L_i. *Complete terms* and *complete formulas* are terms and formulas of L_i^c and vice versa.

Definition 2.1.1 (Set of local Models) A *set of local models* of L_i is a set of first order interpretations of L_i, on a domain $\mathbf{dom}_i$, which agree on the interpretation of L_i^c, the complete fragment of L_i.

As noted in [Franconi and Tessaris, 2004] there is a foundational difference between approaches that use epistemic states and approaches that use a classical model theoretic semantics. The two approaches differ as long as there is more than one model m. Using the notion of complete sublanguage L_c, however, we can force that the set of local models is either a singleton or the empty set by enforcing that $L^c = L$. Under this assumption the two ways of defining the semantics are equivalent. Using this assumption, we are therefore able to simulate both kinds of semantics in DFOL.

Two or more models can carry information about the same portion of the world. In this case we say that they *semantically overlap*. Overlapping is unrelated to the fact that the same constant appears in two languages, as from the local semantics we have that the interpretation of a constant c in L_i is independent from the interpretation of the very same constant in L_j, with $i \neq j$. Overlapping is also unrelated to the intersection between the interpretation domains of two or more contexts. Namely if $\mathbf{dom}_1 \cap \mathbf{dom}_2 \neq \emptyset$, it does not mean that L_1 and L_2 overlap. Instead, DFOL explicitly represents semantic overlapping via a domain relation.

Definition 2.1.2 (Domain relation) A *domain relation* from $\mathbf{dom}_i$ to $\mathbf{dom}_j$ is a binary relation $r_{ij} \subseteq \mathbf{dom}_i \times \mathbf{dom}_j$.

A domain relation from i to j represents the capability of the j-th sub-system to represent in its domain the domain of the i-th subsystem. A pair $\langle d, d' \rangle$ being in r_{ij} means that, from the point of view of j, d in $\mathbf{dom}_i$ is the representation of d' in $\mathbf{dom}_j$. We use the functional notation $r_{ij}(d)$ to denote the set $\{d' \in \mathbf{dom}_j | \langle d, d' \rangle \in r_{ij}\}$. The domain relation r_{ij} formalizes j's subjective point of view on the relation between $\mathbf{dom}_i$ and $\mathbf{dom}_j$ and not an absolute objective point of view. Or in other words $r_{ij} \neq r_{ji}$ because of the non-symmetrical nature of mappings. Therefore $\langle d, d' \rangle \in r_{ij}$ must not be read as if d and d' were the same object in a domain shared by i and j. This fact would indeed be formalized by some observer which is external (above, meta) to both i and j who will state that d and d' corresponds to the same real world object. Using the notion of domain relation, we can define the notion of a DFOL model for a set of local models.

Definition 2.1.3 (DFOL Model) A DFOL *model* $\mathcal{M}$ is a pair $\left\langle \{\mathcal{M}_i\}, \{r_{ij}\} \right\rangle$ where, for each $i \neq j \in I$: $\mathcal{M}_i$ is a set of local models for L_i, and r_{ij} is a domain relation from $\mathbf{dom}_i$ to $\mathbf{dom}_j$.

In the following we will sometimes need to specify the set of tuples of objects that belongs to the interpretation of a predicate P in all the local models in S_i. We use the notation $|P(\mathbf{x})|_i$ (if it is not necessary we can omit the variables, using the simpler notation $|P|_i$) to indicate such a set. Formally $|P(\mathbf{x})|_i = \bigcap_{m=\langle \mathbf{dom}, \mathcal{I} \rangle \in S_i} P^{\mathcal{I}}$. $|P|_i$ intuitively indicates the set of objects that are known to be P by the sub-system i.

We extend the classical notion of assignment (e.g., the one given for first order logic) to deal with arrow variables using domain relations. In particular, an assignment a, provides for each system i, an interpretation for all the variables, and for *some* (but not necessarily all) arrow variables, as the domain relations might be such that there is no consistent way to assign arrow variables. For instance if $a_i(x) = d$ and $r_{ij}(d) = \emptyset$, then a_j cannot assign anything to $x^{i\rightarrow}$.

Definition 2.1.4 (Assignment) Let $\mathcal{M} = \left\langle \{\mathcal{M}_i\}, \{r_{ij}\} \right\rangle$ be a model for $\{L_i\}$. An *assignment* a is a family $\{a_i\}$ of partial functions from the set of variables and arrow variables to $\mathbf{dom}_i$, such that for each variable x and all $i \neq j$:

1. $a_i(x) \in \mathbf{dom}_i$;
2. $a_i(x^{j\rightarrow}) \in r_{ji}(a_j(x))$;
3. $a_j(x) \in r_{ij}(a_i(x^{\rightarrow j}))$;

An assignment a is *admissible for a formula* $i : \phi$ if a_i assigns all the arrow variables occurring in ϕ. Furthermore, a is admissible for a set of formulas Γ if it is admissible for any $j : \phi \in \Gamma$. An assignment a is *strictly admissible* for a set of formulas Γ if it is admissible for Γ and assigns only the arrow variables that occur in Γ.

The intuition on how arrow variables are assigned is the following. If the variable x occurring in $i : \phi$ is thought as a placeholder for a generic element $d \in \mathbf{dom}_i$, the arrow variable $x^{\rightarrow i}$ occurring in $j : \psi$ is a placeholder for an element $d' \in \mathbf{dom}_j$ which is a pre-image (via r_{ji}) of d. Analogously the extended variable $x^{i\rightarrow}$ occurring in $k : \psi$ is a placeholder for any element $d'' \in \mathbf{dom}_k$ which is an image (via r_{ik}) of d. This situation is illustrated in the following drawing:

$$\begin{array}{rccccc}
\text{Languages} & L_j & & L_i & & L_k \\
 & & & & & \\
mboxSymbols & x^{\rightarrow i} & & x & & x^{i\rightarrow} \\
 & \downarrow a_j & & \downarrow a_i & & \downarrow a_k \\
\text{Semantics} & d' & \xrightarrow{r_{ji}} & d & \xrightarrow{r_{ik}} & d''
\end{array}$$

Using the notion of an admissible assignment given above, satisfiability in distributed first order logic is defined as follows:

Definition 2.1.5 (Satisfiability) Let $\mathcal{M} = \left\langle \{\mathcal{M}_i\}, \{r_{ij}\} \right\rangle$ be a model for $\{L_i\}$, $m \in \mathcal{M}_i$, and a an assignment. An i-formula ϕ is *satisfied* by m, w.r.t, a, in symbols $m \models_D \phi[a]$ if

1. a is admissible for $i : \phi$ and
2. $m \models \phi[a_i]$, according to the definition of satisfiability for first order logic.

$\mathcal{M} \models \Gamma[a]$ if for all $i : \phi \in \Gamma$ and $m \in \mathcal{M}_i$, $m \models_D \phi[a_i]$[1].

Mappings between different knowledge bases are formalized in DFOL by a new form of constraints that involves more than one knowledge base. These formulas that will be the basis for describing different mapping approaches are called interpretation constraints and are defined as follows:

[1]Since it will be clear from the context, in the remainder we will use the classical satisfiability symbol $\models$ instead of $\models_D$ and we will write $m \models \phi[a]$ to mean that an i-formula ϕ is satisfied by m. In writing $m \models \phi[a]$ we always mean that a is admissible for $i : \phi$ (in addition to the fact that m classically satisfies ϕ under the assignment a).

a)	$\mathcal{M} \models i\!:\!P(x^{\to j}) \to j\!:\!Q(x)$	iff	For all $d \in \|P\|_i$ and $\underline{\text{for all } d' \in r_{ij}(d)}$, $d' \in \|Q\|_j$
b)	$\mathcal{M} \models i\!:\!P(x) \to j\!:\!Q(x^{i\to})$	iff	For all $d \in \|P\|_i$ $\underline{\text{there is a } d' \in r_{ij}(d)}$, s.t., $d' \in \|Q\|_j$
c)	$\mathcal{M} \models j\!:\!Q(x^{i\to}) \to i\!:\!P(x)$	iff	For all $d \in \|Q\|_j$ and $\underline{\text{for all } d'}$ $\underline{\text{with } d \in r_{ij}(d')}$, $d' \in \|P\|_i$
d)	$\mathcal{M} \models j\!:\!Q(x) \to i\!:\!P(x^{\to j})$	iff	For all $d \in \|Q\|_i$ $\underline{\text{there is a } d'}$ $\underline{\text{with } d \in r_{ij}(d')}$, s.t., $d' \in \|P\|_i$

Figure 2.1: Implicit Quantification of Arrow Variables in Interpretation Constraints

Definition 2.1.6 (Interpretation constraint) An *interpretation constraint from* $i_1, \ldots, i_n$ *to* i with $i_k \neq i$ for $1 \leq k \leq n$ is an expression of the form

$$i_1\!:\!\phi_1, \ldots, i_n\!:\!\phi_n \to i\!:\!\phi \tag{2.1}$$

The interpretation constraint (2.1) can be considered as an axiom that restricts the set of possible DFOL models to those which satisfy it. Therefore we need to define when a DFOL model satisfies an interpretation constraint.

Definition 2.1.7 (Satisfiability of interpretation constraints) A model $\mathcal{M}$ *satisfies* the interpretation constraint (2.1), in symbols $\mathcal{M} \models i_1 : \phi_1, \ldots, i_n : \phi_n \to i : \phi$ if for any assignment a strictly admissible for $\{i_1 : \phi_1, \ldots, i_n : \phi_n\}$, if $\mathcal{M} \models i_k : \phi_k[a]$ for $1 \leq k \leq n$, then a can be extended to an assignment a' admissible for $i\!:\!\phi$ and such that $\mathcal{M} \models i\!:\!\phi[a']$.

Notice that, depending on whether an arrow variable $x^{\to}$ occurs on the left or on the right side of the constraint, $x^{\to}$ has a universal or an existential reading. Figure 2.1 summarizes the different possible readings that will reoccur later. Notationally, for any predicate P, $|P|_i = \bigcap_{m \in \mathcal{M}_i} m(P)$, where $m(P)$ is the interpretation of P in m.

We like to explain these four kind of interpretation constraints in terms of the two ontologies of Figure 3.1. The first ontology $\mathbf{O_1}$ is designed by an expert of vehicles but still have some knowledge about houses. The second ontology $\mathbf{O_2}$ is the opposite: it has detailed knowledge about house but only fundamental knowledge about vehicles. In a set of mappings we may want to express for example that a *Volant* which is copied to $\mathbf{O_2}$ is an *Vehicle*. It can be expressed by the following interpretation constraint a): $1 : Volant(x^{\to 2}) \to 2 : Vehicle(x)$. An instantiation of interpretation constraint b) is that every *Maison* in ontology $\mathbf{O_1}$ has to be copied to $\mathbf{O_2}$ and is an *Domicile*: $1\!:\!Maison(x) \to 2\!:\!Domicile(x^{1\to})$. Also the opposite direction can be expressed. Every *seaplane* which is copied to $\mathbf{O_1}$ has to be a *Hydravion*. In an interpretation constraint c) it can be expressed as $2\!:\!Seaplane(x^{1\to}) \to 1\!:\!Hydravion(x)$. The last type of interpretation constraint d) express with $2\!:\!Vehicle(x) \to 1\!:\!Vehicule(x^{\to 2})$ that every *Vehicle* is a *Vehicule*.

By means of interpretation constraints on equality, we can formalize possible relations between heterogeneous domains.

$$\begin{array}{rcl}
\mathsf{F}_{ij} & = & \left\{ i\!:\!x^{\to j} = y^{\to j} \to j\!:\!x = y \right\} \\
\mathsf{INV}_{ij} & = & \left\{ \begin{array}{l} i\!:\!x = y^{j\to} \to j\!:\!x^{i\to} = y \\ j\!:\!x = y^{i\to} \to i\!:\!x^{j\to} = y \end{array} \right\} \\
\mathsf{OD}_{ij} & = & \mathsf{F}_{ij} \cup \mathsf{F}_{ji} \cup \mathsf{INV}_{ij} \\
\mathsf{ED}_{ij} & = & \mathsf{OD}_{ij} \cup \{ i\!:\!x = x \to j\!:\!x^{i\to} = x^{i\to} \} \\
\mathsf{ID}_{ij} & = & \mathsf{ED}_{ij} \cup \mathsf{ED}_{ji} \\
\mathsf{RD}_{ij} & = & \left\{ \begin{array}{l} i\!:\!x = c \to j\!:\!x^{i\to} = c \\ j\!:\!x = c \to i\!:\!x^{j\to} = c \end{array} \middle| \, c \in L_i \cap L_j \right\} \\
\mathsf{IP}_{ij} & = & i\!:\!\bot \to j\!:\!\bot
\end{array}$$

Proposition 2.1.8 Let $\mathcal{M}$ be a DFOL model and $i \neq j \in I$.

1. $\mathcal{M} \models \mathsf{F}_{ij}$ iff r_{ij} is a partial function.
2. $\mathcal{M} \models \mathsf{INV}_{ij}$ iff r_{ij} is the inverse of r_{ji}.
3. $\mathcal{M} \models \mathsf{OD}_{ij}$ if $r_{ij}(= r_{ji}^{-1})$ is an isomorphism between a subset of $\mathbf{dom}_i$ and a subset of $\mathbf{dom}_j$. I.e., $\mathbf{dom}_i$ and $\mathbf{dom}_j$ overlap.
4. $\mathcal{M} \models \mathsf{ED}_{ij}$ iff $r_{ij}(= r_{ji}^{-1})$ is an isomorphism between $\mathbf{dom}_i$ and a subset of $\mathbf{dom}_j$. I.e., $\mathbf{dom}_i$ is (isomorphically) embedded in $\mathbf{dom}_j$
5. $\mathcal{M} \models \mathsf{ID}_{ij}$ iff $r_{ij}(= r_{ji}^{-1})$ is an isomorphism between $\mathbf{dom}_i$ and $\mathbf{dom}_j$. I.e., $\mathbf{dom}_i$ is isomorphic to $\mathbf{dom}_j$.
6. $\mathcal{M} \models \mathsf{RD}$ implies for every constant c of L_i and L_j, if c is interpreted in d for all $m \in \mathcal{M}_i$ then c is interpreted in $r_{ij}(d)$ for all models of $m \in \mathcal{M}_j$, and vice-versa. I.e., the constant c is rigidly interpreted by i and j in (two) corresponding objects.
7. Finally $\mathcal{M} \models \mathsf{IP}_{ij}$ iff $\mathcal{M}_i = \emptyset$ implies that $\mathcal{M}_j = \emptyset$. I.e., inconsistency propagates from i to j.

2.2 Modeling Mapping Languages in DFOL

Formalisms for mapping languages are based on four main parameters: local languages and local semantics used to specify the local knowledge, and mapping languages and semantics for mappings, used to specify the semantic relations between the local knowledge. In this section we focus on the second pairs and as far as local languages and local semantics it is enough to notice that

Local languages In all approaches local knowledge is expressed by a suitable fragment of first order languages.

Local semantics with the notable exception of [Franconi and Tessaris, 2004], where authors propose an *epistemic approach* to information integration (see chapter 5 in deliverable D2.2.1 "Specification of a common framework for characterizing alignment"), all the other formalisms for ontology mapping assume that each local knowledge is interpreted in a (partial) state of the world and not into an epistemic state. This formally corresponds to the fact that each local knowledge base is associated with *at most one* FOL interpretation. The case of incomplete local knowledge will be described in the future.

The first assumption is naturally captured in DFOL, by simply considering L_i to be an adequately restricted FOL language. Concerning the local semantics, in DFOL models each L_i is associated with a *set of interpretations*. To simulate the single local model assumption, in DFOL it is enough to declare each L_i to be a *complete* language. This implies that all the $m \in M_i$ have to agree on the interpretation of L_i-symbols.

Notationally, $\phi, \psi, \ldots$ will be used to denote both DL expressions and FOL open formulas. If ϕ is a DL concept, $\phi(x)$ (or $\phi(x_1, \ldots, x_n)$) will denote the corresponding translation of ϕ in FOL as described in [Borgida, 1996]. If ϕ is a role R then $\phi(x, y)$ denotes its translation $R(x, y)$, and if ϕ is a constant c, then $\phi(x)$ denote its translation $x = c$. Finally we use $\mathbf{x}$ to denote a vector $x_1, \ldots, x_n$ of variables.

2.2.1 Distributed Description Logics/C-OWL

The approach presented in [Borgida and Serafini, 2003] extends DL with a local model semantics similar to the one introduced above and so-called bridge rules to define semantic relations between different T-Boxes. A distributed interpretation for DDL on a family of DL languages $\{L_i\}$, is a family $\{\mathcal{I}_i\}$ of interpretations, one for each L_i plus a family $\{r_{ij}\}_{i \neq j \in I}$ of domain relations. While the original proposal only considered subsumption between concept expressions, the model was extended to a set of five semantic relations discussed below. The semantics of the five semantic relations defined in C-OWL is the following:

Definition 2.2.1 ([Bouquet *et al.*, 2004a]) Let ϕ and ψ be either concepts, or individuals, or roles of the descriptive languages L_i and L_j respectively[2].

1. $\mathfrak{I} \models i\!:\!\phi \xrightarrow{\sqsubseteq} j\!:\!\psi$ if $r_{ij}(\phi^{\mathcal{I}_i}) \subseteq \psi^{\mathcal{I}_j}$;
2. $\mathfrak{I} \models i\!:\!\phi \xrightarrow{\sqsupseteq} j\!:\!\psi$ if $r_{ij}(\phi^{\mathcal{I}_i}) \supseteq \psi^{\mathcal{I}_j}$;
3. $\mathfrak{I} \models i\!:\!\phi \xrightarrow{\equiv} j\!:\!\psi$ if $r_{ij}(\phi^{\mathcal{I}_i}) = \psi^{\mathcal{I}_j}$;
4. $\mathfrak{I} \models i\!:\!\phi \xrightarrow{\perp} j\!:\!\psi$ if $r_{ij}(\phi^{\mathcal{I}_i}) \cap \psi^{\mathcal{I}_j} = \emptyset$;
5. $\mathfrak{I} \models i\!:\!\phi \xrightarrow{*} j\!:\!\psi$ if $r_{ij}(\phi^{\mathcal{I}_i}) \cap \psi^{\mathcal{I}_j} \neq \emptyset$;

An interpretation for a context space is a model for it if all the bridge rules are satisfied.

From the above satisfiability condition one can see that the mapping $i : \phi \xrightarrow{\equiv} j : \psi$ is equivalent to the conjunction of the mappings $i : \phi \xrightarrow{\sqsubseteq} j : \psi$ and $i : \phi \xrightarrow{\sqsupseteq} j : \psi$. The mapping $i : \phi \xrightarrow{\perp} j : \psi$ is equivalent to $i : \phi \xrightarrow{\sqsubseteq} j : \neg\psi$. And finally the mapping $i : \phi \xrightarrow{*} j : \psi$ is the negation of the mapping $i : \phi \xrightarrow{\perp} j : \psi$. As the underlying notion of a model is the same as for DFOL, we can directly try to translate bridge rules into interpretation constraints. In particular, there are no additional assumptions about the nature of the domains that have to be modeled. The translation is the following:

C-OWL	DFOL
$i\!:\!\phi \xrightarrow{\sqsubseteq} j\!:\!\psi$	$i\!:\!\phi(x^{\rightarrow j}) \rightarrow j\!:\!\psi(x)$
$i\!:\!\phi \xrightarrow{\sqsupseteq} j\!:\!\psi$	$j\!:\!\psi(x) \rightarrow i\!:\!\phi(x^{\rightarrow j})$
$i\!:\!\phi \xrightarrow{\equiv} j\!:\!\psi$	$i\!:\!\phi(x^{\rightarrow j}) \rightarrow j\!:\!\psi(x)$ and $i\!:\!\phi(x^{\rightarrow j}) \rightarrow j\!:\!\psi(x)$
$i\!:\!\phi \xrightarrow{\perp} j\!:\!\psi$	$i\!:\!\phi(x^{\rightarrow j}) \rightarrow j\!:\!\neg\psi(x)$
$i\!:\!\phi \xrightarrow{*} j\!:\!\psi$	No translation

We see that a bridge rule basically corresponds to the interpretation a) and d) in Figure 2.1. The different semantic relations correspond to the usual readings of implications. Finally negative information about mappings (i.e., $i\!:\!\phi \stackrel{\sqsubseteq}{\not\longrightarrow} j\!:\!\psi$ is not representable by means of DFOL interpretation constraints.

2.2.2 Ontology Integration Framework (OIS)

Calvanese and colleagues in [Calvanese *et al.*, 2002b] propose a framework for mappings between ontologies that generalizes existing work on view-based schema integration [Ullman, 1997] and subsumes other approaches on connecting DL models with rules. In particular, they distinguish global centric, local centric and the combined approach. These approaches differ in the types of

[2] In this definition, to be more homogeneous, we consider the interpretations of individuals to be sets containing a single object rather than the object itself.

expressions connected by mappings. With respect to the semantics of mappings, they do not differ and we therefore treat them as one.

OIS assumes the existence of a global ontology g into which all local models s are mapped. On the semantic level, the domains of the local models are assumed to be embedded in a global domain. Further, in OIS constants are assumed to rigidly designate the same objects across domain. Finally, global inconsistency is assumed, in the sense that the inconsistency of a local knowledge base makes the whole system inconsistent. As shown in Proposition 2.1.8, we can capture these assumptions by the set of interpretation constraints ED_{sg}, RD_{sg}, and IP_{sg}, where s is the index of any source ontology and g the index of the global ontology.

According to these assumptions mappings are described in terms of correspondences between a local and the global model. The interpretation of these correspondences are defined as follows:

Definition 2.2.2 ([Calvanese *et al.*, 2002b]) Correspondences between source ontologies (with interpretation $\mathcal{D}$) and global ontology (with interpretation $\mathcal{I}$) are of the following four forms

1. $\mathcal{I}$ satisfies $\langle\phi, \psi, sound\rangle$ w.r.t. the local interpretation $\mathcal{D}$, if all the tuples satisfying ψ in $\mathcal{D}$ satisfy ϕ in $\mathcal{I}$
2. $\langle\phi, \psi, complete\rangle$ w.r.t. the local interpretation $\mathcal{D}$, if no tuple other than those satisfying ψ in $\mathcal{D}$ satisfies ϕ in $\mathcal{I}$,
3. $\langle\phi, \psi, exact\rangle$ w.r.t. the local interpretation $\mathcal{D}$, if the set of tuples that satisfies ψ in $\mathcal{D}$ is exactly the set of tuples satisfying ϕ in $\mathcal{I}$.

From the above semantic conditions, $\langle\phi, \psi, exact\rangle$ is equivalent to the conjunction of $\langle\phi, \psi, sound\rangle$ and $\langle\phi, \psi, complete\rangle$. It's therefore enough to provide the translation of the first two correspondences. The definitions 1 and 2 above can directly be expressed into interpretation constraints (compare Figure 2.1) resulting in the following translation:

GLAV Correspondence	DFOL
$\langle\phi, \psi, sound\rangle$	$s:\psi(\mathbf{x}) \rightarrow g:\phi(\mathbf{x}^{s\rightarrow})$
$\langle\phi, \psi, complete\rangle$	$g:\phi(\mathbf{x}) \rightarrow s:\psi(\mathbf{x}^{\rightarrow g})$
$\langle\phi, \psi, exact\rangle$	$s:\psi(\mathbf{x}) \rightarrow g:\phi(\mathbf{x}^{s\rightarrow})$ and $g:\phi(\mathbf{x}) \rightarrow s:\psi(\mathbf{x}^{\rightarrow g})$

The translation shows that there is a fundamental difference in the way mappings are interpreted in C-OWL and in OIS. While C-OWL mappings correspond to a universally quantified reading (Figure 1 a), OIS mappings have an existentially quantified readings (Figure 1 b/d). We will come back to this difference later.

2.2.3 DL for Information Integration (DLII)

A slightly different approach to the integration of different DL models is described in [Calvanese *et al.*, 2002a]. This approach assumes a partial overlap between the domains of the models M_i and M_j, rather than a complete embedding of them in a global domain. This is captured by the interpretation constraint OD_{ij}. The other assumptions (rigid designators and global inconsistency) are the same as for OIS.

An interpretation $\mathcal{I}$ associates to each M_i a domain Δ_i. These different models are connected by interschema assertions. Satisfiability of interschema assertions is defined as follows [3]

[3]To simplify the definition we introduce the notation $\top^{\mathcal{I}}_{nij} = \top^{\mathcal{I}}_{ni} \cap \top^{\mathcal{I}}_{nj}$ for any $n \geq 1$. Notice that $\top^{\mathcal{I}}_{nij} = \Delta^n_i \cap \Delta^n_j$.

Definition 2.2.3 (Satisfiability of interschema assertions) If $\mathcal{I}$ is an interpretation for M_i and M_j we say that $\mathcal{I}$ satisfies the interschema assertion

$\phi \sqsubseteq_{ext} \psi$, if $\phi^{\mathcal{I}} \subseteq \psi^{\mathcal{I}}$ $\quad$ $\phi \not\sqsubseteq_{ext} \psi$, if $\phi^{\mathcal{I}} \not\subseteq \psi^{\mathcal{I}}$
$\phi \equiv_{ext} \psi$, if $\phi^{\mathcal{I}} = \psi^{\mathcal{I}}$ $\quad$ $\phi \not\equiv_{ext} \psi$, if $\phi^{\mathcal{I}} \neq \psi^{\mathcal{I}}$
$\phi \sqsubseteq_{int} \psi$, if $\phi^{\mathcal{I}} \cap \top_{nij}^{\mathcal{I}} \subseteq \psi^{\mathcal{I}} \cap \top_{nij}^{\mathcal{I}}$
$\phi \equiv_{int} \psi$, if $\phi^{\mathcal{I}} \cap \top_{nij}^{\mathcal{I}} = \psi^{\mathcal{I}} \cap \top_{nij}^{\mathcal{I}}$
$\phi \not\sqsubseteq_{int} \psi$, if $\phi^{\mathcal{I}} \cap \top_{nij}^{\mathcal{I}} \not\subseteq \psi^{\mathcal{I}} \cap \top_{nij}^{\mathcal{I}}$
$\phi \not\equiv_{int} \psi$, if $\phi^{\mathcal{I}} \cap \top_{nij}^{\mathcal{I}} \neq \psi^{\mathcal{I}} \cap \top_{nij}^{\mathcal{I}}$

As before $\equiv_{ext}$ and $\equiv_{int}$ are definable as conjunctions of $\sqsubseteq_{ext}$ and $\sqsubseteq_{int}$, so we can ignore them for the DFOL translation. Furthermore, a distinction is made between extensional and intentional interpretation of interschema assertions, which leads to different translations into DFOL.

interschema assertions	**DFOL**
$\phi \sqsubseteq_{ext} \psi$	$i:\phi(\mathbf{x}) \rightarrow j:\psi(\mathbf{x}^{i\rightarrow})$
$\phi \equiv_{ext} \psi$	$i:\phi(\mathbf{x}) \rightarrow j:\psi(\mathbf{x}^{i\rightarrow})$ and $j:\psi(\mathbf{x}) \rightarrow i:\phi(\mathbf{x}^{j\rightarrow})$
$\phi \not\sqsubseteq_{ext} \psi, \phi \not\equiv_{ext} \psi$	No translation
$\phi \sqsubseteq_{int} \psi$	$i:\phi(\mathbf{x}^{\rightarrow j}) \rightarrow j:\psi(\mathbf{x})$
$\phi \equiv_{int} \psi$	$i:\phi(\mathbf{x}^{\rightarrow j}) \rightarrow j:\psi(\mathbf{x})$ and $j:\psi(\mathbf{x}^{\rightarrow i}) \rightarrow i:\phi(\mathbf{x})$
$\phi \not\sqsubseteq_{int} \psi, \phi \not\equiv_{int} \psi$	No translation

While the extensional interpretation corresponds to the semantics of mappings in OIS, the intentional interpretation corresponds to the semantics of mappings in C-OWL. Thus using the distinction made in this approach we get an explanation of different conceptualizations underlying the semantics of C-OWL and OIS that use an extensional and an intentional interpretation, respectively.

2.2.4 ϵ-connections

A different approach for defining relations between DL knowledge bases has emerged from the investigation of so-called ϵ-connections between abstract description systems [Kutz *et al.*, 2004]. Originally intended to extend the decidability of DL models by partitioning them into a set of models that use a weaker logic, the approach has recently been proposed as a framework for defining mappings between ontologies [Grau *et al.*, 2004].

In the ϵ-connections framework, for every pair of ontologies ij there is a set ϵ_{ij} of *links*, which represents binary relations between the domain of the i-th ontology and the domain of the j-th ontology. Links from i to j can be used to define i-concepts, in a way that is analogous to how roles are used to define concepts. In the following table we report the syntax and the semantics of i-concept definitions based on links. (E denotes a link from i to j. The only assumption about the relation between domains is global inconsistency, see above).

In DFOL we have only one single relation from i to j, while in ϵ-connection there are many possible relations. However, we can use a similar trick as used in [Borgida and Serafini, 2003] to map relations to interschema relations: each of the relations in ϵ_{ij} acts as a r_{ij}. To represent ϵ-connections it is therefore enough to label each arrow variable with the proper link name. The arrow variable $x^{\xrightarrow{E} i}$ is read as the arrow variable $x^{\rightarrow i}$ where r_{ij} is intended to be the interpretation of relation E_{ij}. With this syntactic extension of DFOL, concept definitions based on links (denoted as E) can be codified in DFOL as follows:

ϵ-connections	DFOL
$\phi \sqsubseteq \exists E.\psi$	$i:\phi(x) \rightarrow j:\psi(x^{i\xrightarrow{E}})$
$\phi \sqsubseteq \forall E.\psi$	$i:\phi(x^{\xrightarrow{E}j}) \rightarrow j:\psi(x)$
$\phi \sqsubseteq \geq nE.\psi$	$i:\bigwedge_{k=1}^{n}\phi(x_1) \rightarrow$ $j:\bigwedge_{k\neq h=1}^{n}\psi(x_k^{i\xrightarrow{E}}) \wedge x_k \neq x_h$
$\phi \sqsubseteq \leq nE.\psi$	$i:\phi(x) \wedge \bigwedge_{k=1}^{n+1} x = x_k^{\xrightarrow{E}j} \rightarrow$ $j:\bigvee_{k=1}^{n+1}(\psi(x_k) \supset \bigvee_{h\neq k} x_h = x_k)$

We see that like OIS, links in the ϵ-connections framework have an extensional interpretation. The fact, that the framework distinguishes between different types of domain relations, however, makes it different from all other approaches.

Another difference to the previous approaches is that new links can be defined on the basis of existing links, similar to complex roles in DL. Syntax and semantics for link constructors is defined in the usual way: $\left(E^{-1}\right)^{\mathcal{I}} = \left(E^{\mathcal{I}}\right)^{-1}$ (Inverse), $(E \sqcap F)^{\mathcal{I}} = E^{\mathcal{I}} \cap F^{\mathcal{I}}$ (Conjunction), $(E \sqcup F)^{\mathcal{I}} = E^{\mathcal{I}} \cup F^{\mathcal{I}}$ (Disjunction), and $(\neg E)^{\mathcal{I}} = (\Delta_i \times \Delta_j) \setminus E^{\mathcal{I}}$ (Complement). Notice that, by means of inverse links we can define mappings of the b and d type. E.g., the ϵ-connection statement $\phi \sqsubseteq \exists E^{-1}\psi$ corresponds to the DFOL bridge rule $i : \phi(x) \rightarrow j : \psi(x^{i\rightarrow})$ which is of type b). Similarly the ϵ-connection $\phi \sqsubseteq \forall E^{-1}\psi$ corresponds to a mapping of type d).

As the distinctions between different types of links is only made on the model theoretic level, it is not possible to model Boolean combinations of links. Inverse links, however, can be represented by the following axiom:

$$i:y = x^{\xrightarrow{E}j} \rightarrow j:y^{\xrightarrow{E^{-1}}i} = x$$
$$j:y^{\xrightarrow{E^{-1}}i} = x \rightarrow i:y = x^{\xrightarrow{E}j}$$

Finally the inclusion axioms between links, i.e., axioms of the form $E \sqsubseteq F$ where E and F are homogeneous links, i.e., links of the same ϵ_{ij}, can be translated in DFOL as follows:

$$i:x = y^{\xrightarrow{E}j} \rightarrow j:x^{i\xrightarrow{F}} = y$$

We can say that the ϵ-connections framework significantly differs from the other approaches in terms of the possibilities to define and combine mappings of different types.

2.3 Discussion and Chapter Conclusions

The encoding of different mapping approaches in a common framework has two immediate advantages. The first one is the ability to reason across the different frameworks. This can be done on the basis of the DFOL translation of the different approaches using the sound and complete calculus for DFOL [Ghidini and Serafini, 2000]. As there are not always complete translations, this approach does not cover all aspects of the different approaches, but as shown above, we can capture most aspects. There are only two aspects which cannot be represented in DFOL, namely "non mappings" ($i:\phi \xrightarrow{*} j:\psi$ in C-OWL, $\phi \not\sqsubseteq_{int} \psi$ etc. in DLII) and "complex mappings" such as complex links in ϵ-connection. The second benefit is the possibility to compare the expressiveness of the approaches. We have several dimensions along which the framework can differ:

Arity of mapped items[4] C-OWL allows only to align constants, concepts and roles (2-arity relations), ϵ-connections allow to align only 1-arity items, i.e., concepts, while DLII and OIS allow to integrate n-arity items.

Positive/negative mappings Most approaches state positive facts about mapping, e.g that two elements are equivalent. The DLII and C-OWL frameworks also allow to state that two elements do not map ($\phi \not\equiv \psi$).

Domain relations The approaches make different assumptions about the nature of the domain. While C-OWL and ϵ-connections do not assume any relation between the domains, DLII assumes overlapping domains and OIS assumes local domains that are embedded in a global domain.

Multiple mappings Approaches with multiple mappings allow different kind of mapping relations. Only the ϵ-connection approach supports the definition of different types of mappings between ontologies. In general multiple mappings need for more than one domain relation.

Local inconsistency Some approaches provide a consistent semantics also in the case in which some of the ontologies or mappings are inconsistent.

We summarize the comparison in the following table.

	Int. constr. (cf. fig. 2.1)				Mapping type			Domain	Arity	Local
	a)	b)	c)	d)	Pos.	Neg.	Mult.	relation		$\perp$
C-OWL	×			×	×	×		Het.	2	×
OIS		×		×	×			Incl.	n	
DLII	×	×			×	×		Emb.	n	
ϵ-Conn.	×	×	×	×	×	×	×	Het.	1	

We conclude that existing approaches make choices along a number of dimensions. These choices are obviously influenced by the intended use. Approaches intended for database integration for example will support the mapping of n-ary items that correspond to tuples in the relational model. Despite this fact, almost no work has been done on charting the landscape of choices to be made when designing a mapping approach, and for adapting the approach to the requirements of an application. The work reported in this chapter provides the basis for this kind of work by identifying the possible choices on a formal level. An important topic of future work is to identify possible combinations of features for mapping languages on a formal level in order to get a more complete picture of the design space of mapping languages.

Chapter 3

A Language-Independent Perspective on Alignment and Merging

In its most general form, the term "ontology alignment" can refer to almost any formal description of the (semantic) relationship between ontologies. Deliverable D2.2.1 [Bouquet *et al.*, 2004c] discussed a more restricted conception of the term, that conceived alignments as pairs of elements of the ontologies[1], together with meta-information on the type of the relation and the confidence in its correctness.

An ontology alignment is thus given a set theoretic definition as a set of mappings. Though perfectly acceptable in many applications, it has the disadvantage of using entities—something local—as the basis of ontology alignment—something global. Indeed, the complexity of ontology languages like OWL (see [Antoniou and van Harmelen, 2004] for an introduction) enables us to express more complicated relationships between elements even without the use of meta-logical constructions. For example, consider an ontology with a concept *Professor* and one with the concepts *Woman* and *Man*. A comprehensive alignment should contain the information that *Professor* is a subconcept of the union of *Woman* and *Man*. On the one hand, this statement has a non-local flavor in the sense that it cannot be expressed in terms of relations between pairs of (atomic) elements. On the other hand, we deal with a relationship between concept expressions which is not a mere equality of two statements. In contrast, most of todays alignment algorithms focus on equality or similarity mappings between pairs of elements.

Here, however, we want to discuss an approach that emphasizes alignment information which can be represented in terms of the ontology language. This has the advantage that alignments are more closely related to the ontological formalism under consideration, thereby allowing for simple and concrete descriptions of the merging of aligned ontologies. On the other hand, while the relationships between ontologies can be structurally more complex, meta-level information like confidence values are not included in the framework[2]. Our presentation will be simplified by using a coherent notation that employs elementary *category theory* as a concise meta-language.

After presenting some related work, we start our investigation of so-called *V-alignments* in Section 3.1. We give a formal definition, introduce merging, and present an algebra for working with these simple alignments. Thereafter, in Section 3.2, we extend our approach to a more com-

[1] We speak of "elements of an ontology" to refer to arbitrary semantic entities of a given ontology language, like e.g. concepts, relations, or instances. Comparisons of elements in different ontologies are restricted to elements of similar type, i.e. we would never compare instances with relations etc.

[2] Though they could be added "on top" if desired.

plex type of alignments called *W-alignments*. Again we present suitable formalizations of merging and composition within this framework. Section 3.3 shows how alignment in the disjunctive first-order logic from Chapter 2 can be expressed by means of W-alignments. Finally, Section 3.4 gives some concluding remarks.

Related work

In [Verheijen *et al.*, 1998], the categorical approach is mentioned but not well formalized. [Bench-Capon and Malcolm, 1999] use morphisms of algebraic specifications (see [Guttag and Horning, 1978]) to define morphisms between ontologies and say a *relation* (an alignment in their sense) between ontologies O_1 and O_2 consists of an ontology O and a pair of morphisms $\chi_1 : O \to O_1$ and $\chi_2 : O \to O_2$. This is precisely the definition of alignments we use, but they do not provide any means of representing complex alignments as we do. In [Kent, 2000], a category theoretic approach using the information flow theory of Barwise and Seligman [Barwise and Seligman, 1997] is given, with no concrete representation of alignments. Information Flow is also the basis of an implemented system called IF-Map [Kalfoglou and Schorlemmer, 2002] designed for automated ontology mapping, but they do not have a categorical representation for rich alignments. [Hitzler *et al.*, 2005] gives a concrete example of a representation of an alignment in category theory, but since it is so simplistic, it is hard to see the generality of the approach. Much more details on the categorical approach are given in a survey on ontology mapping [Kalfoglou and Schorlemmer, 2005], in particular in Sections 2.a. and 3.f.

3.1 Simple Alignments

In spite of our declared objective to express alignments in a non-local fashion, we will start our considerations with a very simple relational type of alignments. Probably the most basic way to describe relationships between two ontologies is to identify those elements which represent the same semantic entities. This can be adequately described by a binary relation between the sets of elements.

To simplify our presentation, we employ a unified notation based on elementary category theory. Category theory describes objects (in our case: ontologies) and their relationships in a leveled fashion:

- On the concrete level, one starts with a given collection of objects and defines directed relationships (called *morphisms*) between them. The latter requires to refer to the details of the objects structure. For example, in Section 3.1.1, we introduce *ontology refinements*—a special type of functions—as morphisms between ontologies.
- On the abstract level, one works solely on the structure of objects and morphisms that are defined in each concrete case. Properties and constructions on the abstract level can thereby be carried out without detailed knowledge about the concrete level, and yet one can produce meaningful results. In our given case, this enables us to define alignments without restricting to any concrete ontology language.

Thus category theory introduces a kind of "object-oriented" paradigm, that enable us to work on interfaces (to the unspecified concrete level) without knowing the implementation details (i.e.

the ontology language that is in use). To see how morphisms can be used to externalize structural properties of ontologies, consider the sketchy description of simple alignments given above. The binary relation that describes a simple alignment is a set of pairs of elements, and we can assign to any such pair one element in the first and one element of the second ontology. These operations are called *projections* and can be visualized as follows:

$$\begin{tikzcd} O_1 & & O_2 \\ & A \arrow[ul, "\pi_1"] \arrow[ur, "\pi_2"'] & \end{tikzcd}$$

Here π_1 and π_2 are the according projection functions. Now we observe that any set A with two such projections can be interpreted as a relation, since the projections assign to any element of A a pair of elements. Thus the essence of being a binary relation is captured in the above diagram, even if we do not have any details about the sets A, O_1, and O_2. This view represents the abstract level of category theory: the shape of the diagram carries meaning, even if no further details about the concrete definitions are known.

3.1.1 Category theoretic alignments

In the rest of this work, some familiarity with category theory will be useful. However, we will usually provide examples that refer to problems on a concrete level, and the reader might be content with taking the categorical language as a somewhat more general formal description of the exemplified ideas. For more details on the basics of category theory, see [Pierce, 1991] for an easy yet good introduction. [Adámek *et al.*, 2004] gives something more elaborated.

Categories of ontologies

Most of our later considerations can nicely be done on the abstract level of category theory, but it is sensible first to introduce some underlying concrete levels that are meaningful in the context of ontology research. The objects that we will deal with here naturally are ontological descriptions, like description logic knowledge bases, first-order theories, or simple hierarchical taxonomies. However, we shall define morphisms only for ontology languages with a logical semantics that assigns to any knowledge base a collection of models (i.e. formal interpretations that respect the axioms and constraints of the knowledge base)[3].

Thus we can view ontologies as logical theories and define morphisms between them based on well-known studies in these fields. In particular, *theory morphisms* were discussed in *institution theory* [Goguen and Burstall, 1992] as suitable "translation functions" between logical theories. Basically, a theory morphism is a function from one ontology (regarded as a set of axioms) into another ontology, such that all of the axioms of the first ontology are mapped to statements that are satisfied in the second ontology. However, we only want to consider functions that respect the logical structure of the ontology language. This somewhat complicates the formal definition, since the ontologies that we compare might be based on different signatures (i.e. on different underlying

[3]This is not very precise yet. However, all of the mentioned paradigms (which can be viewed as fragments of first-order logic), extensions with equality, and some typical higher order paradigms fall into the scope of this discussion. Nonmonotonic paradigms might require a different definition.

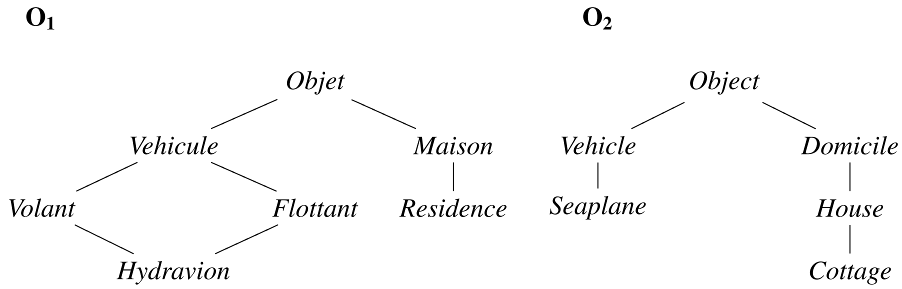


Figure 3.1: Two ontologies modelling the same domain

alphabets). A formal definition employs the notion of *signature morphisms* known in institution theory. It can for example be found in [Goguen and Burstall, 1992, Definition 5].

For this work, it suffices to know that the intuition behind theory morphism is that they describe translation functions that "embed" logical theories into more specific theories. Thus any theory morphism describes a way to view some ontology as a generalization of another ontology, which is why the name "ontology refinement" is also appropriate for this type of morphism. In the following, we will work on the abstract level, with ontologies and their refinements as the concrete category that we have in mind.

V-alignments of ontologies

Assuming that a suitable concrete category of ontologies is available, we can now define simple ontology alignments on the abstract level. Due to the shape of the underlying diagram, we dub these alignments "V-alignments" in order not to confuse them with the informal idea of alignments in general.

Definition 3.1.1 (Ontology alignment) A *V-alignment* between two ontologies O_1 and O_2 is a triple $\langle O, p_1, p_2 \rangle$ such that:

- O is an ontology (i.e. *an object* in the concrete category)
- $p_1 : O \to O_1$ and $p_2 : O \to O_2$ are ontology refinements (i.e. morphisms in the concrete category)

For any V-alignment A, the object O in the definition is written $|A|$.

This very basic definition has already been used by several of the aforementioned works, e.g. [Bench-Capon and Malcolm, 1999; Kent, 2000; Kalfoglou and Schorlemmer, 2002; Hitzler *et al.*, 2005], except that the terms "V-alignment" and "ontology refinement" are non-standard, and sometimes, the two morphisms are not explicitly described as theory morphisms in an institution.

Example 3.1.2 In order to illustrate the definition, consider the ontologies in Figure 3.1, which will also serve us as a running example later on. The two ontologies describe simple taxonomies of concepts, but one is written in English while the other is in French.

The ontologies given in Figure 3.1 are clearly related. For instance, one would like to express the fact that *Objet* is equivalent to class *Object*, as well as *Vehicule* is equivalent to *Vehicle*. In order to do so, let us define an alignment ontology O with concept names $\{$*Objet-Object*, *Vehicule-Vehicle*, *Hydravion-Seaplane*, *Maison-House*, *Residence-Domicile*$\}$ and no further axioms.[4]

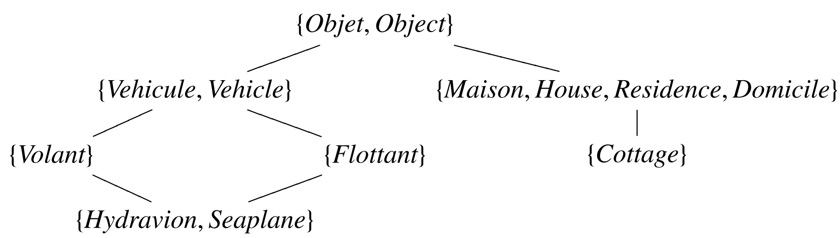
To obtain the intended V-alignment, let $f_1 : O \to O_1$ and $f_2 : O \to O_2$ be the obvious projection mappings that map each concept of O to the concept given in the first respectively second part of its concept name. Taxonomies form a simple description logic, so they are expressed as theories in a concrete institution. In such institution, theory morphisms are exactly order-preserving functions. Thus, it is routine to check that f_1 and f_2 are indeed theory morphisms. For further reference, we label this V-alignment $A = \langle O, f_1, f_2 \rangle$. Note that the names of the concepts in O are arbitrary and were just chosen to simplify the presentation. In particular, they do not affect the content of the alignment.

Merging with simple alignments

Once a V-alignment between two ontologies is known, it is desirable to integrate the aligned ontologies into a combined knowledge base. This operation, called *ontology merging*, aims at uniting heterogeneous specifications into a bigger, more precise one which allows to share information easily. The categorical formalization of V-alignments allows for a rather simple description of the merge, and it is well-known that this can be described in terms of the category theoretic *pushout* construction. This topic has already been discussed in several places, and it is not the goal of this deliverable to repeat the respective argumentation (see [Kent, 2000], [Kalfoglou and Schorlemmer, 2002] or [Hitzler *et al.*, 2005; Bouquet *et al.*, 2004b] for more details). However, we provide the following example.

Example 3.1.3 The pushout of the V-alignment A from Example 3.1.2 is the triple $\langle M, g_1, g_2 \rangle$ with the merge M given by the following object:

M



The ontology refinements g_1 and g_2 are the obvious functions that are indicated by the chosen concept names.

3.1.2 An algebra for simple alignments

The need for ontology alignment naturally arises when information from many ontologies is relevant to a given task. However, since the task of constructing alignments is not an easy one and can hardly be accomplished in a fully automatic fashion, it is reasonable to store and reuse known alignments. Any application like the Semantic Web will offer published ontologies as well as ontology alignments—sometimes partial, weak, or inconsistent—and there will be a need to integrate

[4]Note that ontologies that play the role of binary relations in V-alignments will usually not have any additional axioms.

several alignments in a meaningful way. The purpose of this section is to introduce a sound algebra of V-alignments that allows for essential operations that enable us to compose, add, and intersect alignments.

Composing alignments

Composition is a central operation for the reuse of alignments: if we have an alignment between ontologies O_1 and O_2, and an alignment between O_2 and O_3, then it should be possible to obtain an alignment of O_1 and O_3. To formalize this operation, we employ a well-known categorical construction called *pullback*.

Definition 3.1.4 (Composition of alignments) Consider V-alignments $A = \langle |A|, p_1^A, p_2^A \rangle$ and $B = \langle |B|, p_2^B, p_3^B \rangle$ between ontologies O_1 and O_2, and O_2 and O_3, respectively. The *composition* $B \circ A$ of A and B is a V-alignment $C = \langle |C|, p_1^C, p_3^C \rangle$, defined as follows:

- $|C| =_{\mathsf{def}} O$,
- $p_1^C =_{\mathsf{def}} p_1^A \circ f_A$,
- $p_3^C =_{\mathsf{def}} p_3^B \circ f_B$,

where $\langle O, f_A, f_B \rangle$ is the category-theoretic pullback of p_2^A and p_2^B.

This definition can be visualized by the following commutative diagram:

$$\begin{tikzcd}
O_1 & & O_2 & & O_3 \\
& {|A|} \arrow[ul, "p_1^A"'] \arrow[ur, "p_2^A"] & & {|B|} \arrow[ul, "p_2^B"'] \arrow[ur, "p_3^B"] & \\
& & {|C|} \arrow[ul, "f_A"'] \arrow[ur, "f_B"] \arrow[uull, bend left, "p_1^C"] \arrow[uurr, bend right, "p_3^C"'] & &
\end{tikzcd}$$

The V-alignment C is written in the same way as a composition of morphism: $C = B \circ A$.

For those knowledgeable in category theory this definition is quite obvious, but an intuition can also be given in a purely set-theoretical terms. Viewing V-alignments A and B as binary relations, the pullback C is just the well-known composition of relations (though care must be taken not to confuse our categorical $\circ$ with the symbol for relational composition, since the latter is usually read in a forward fashion). Though the pullback is closely related to composition of relations, it acts on (structured) ontologies instead of mere sets of pairs. Thus the pullback will also have ontological structure, which plays only a minor role in our investigations.

Example 3.1.5 In order to give an example of composition of V-alignments, we need to extend our setting from Example 3.1.2 with a third ontology, shown in Figure 3.2. It models the same domain as before, but the labeling is in Spanish.

This ontology is aligned with O_2 according to V-alignment B that we sketch with the following relation: $\{(Vehicle, Vehiculo), (Seaplane, Hydroavion)\}$.

The expected composition alignment $B \circ A$ then is given by the relation $\{(Vehicule, Vehiculo), (Hydravion, Hydroavion)\}$.

$\mathbf{O_3}$

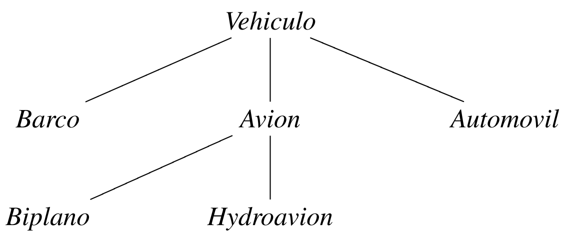


Figure 3.2: A Spanish ontology.

Property 3.1.6 Consider ontologies O_1, O_2, O_3, O_4 which are aligned by V-alignments A (O_1 and O_2), B (O_2 and O_3), and C (O_3 and O_4). The following properties hold:

- If $A \circ B = \langle X, f, g \rangle$ then $B \circ A = \langle X, g, f \rangle$. (Symmetry)
- $(C \circ B) \circ A = C \circ (B \circ A)$. (Associativity)
- $\langle |B \circ A|, f_A, f_B \rangle$ is a V-alignment for $|A|$ and $|B|$.

The required proofs are straightforward. In our current setting, the above properties confirm that the proposed operation is well-behaved as a composition of alignments.

Intersection and union of alignments

Two V-alignments A and B for the same pair of ontologies O_1 and O_2 may give different information about the correspondences between O_1 and O_2. Given two V-alignments, one should be able to extract the consensual alignment, based on the agreed correspondences. This is called the *intersection* of alignments. Besides combining alignments in the appropriate way, there should exist an alignment which is more precise than both A and B and which gathers everything expressed in A and B. This new alignment should satisfy a kind of minimality property, i.e. it should not contain more information than that inside A or B. We call the result of this operation *union* of alignments, denoted by $A \cup B$.

These operations are indeed very useful in the context of the Semantic Web since they allow some kind of modularization of the alignments. In this respect, one can give a partial alignment with only a dozen or less correspondences and expect to retrieve more on the Web when needed.

Definition 3.1.7 (Intersection) Consider V-alignments $A = \langle |A|, f_1, f_2 \rangle$ and $B = \langle |B|, g_1, g_2 \rangle$ between ontologies O_1 and O_2. Let C be the limit of the diagram composed of alignments A and B (see Figure 3.3) associated to the morphisms $k_A : C \to A$, $k_B : C \to B$, $h_1 = f_1 \circ k_A$ and $h_2 = f_2 \circ k_A$. The *intersection* of A and B is a V-alignment $A \cap B = \langle C, h_1, h_2 \rangle$.

Union is defined via the intersection. In order to unify two alignments, one has to know what is common to both of them. Then the union is the disjoint union of this common part and the non-common parts. Formally, it is equivalent to a pushout of the intersection.

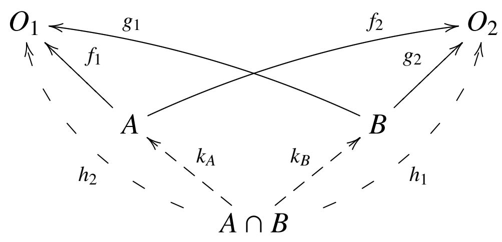


Figure 3.3: Intersection of alignments.

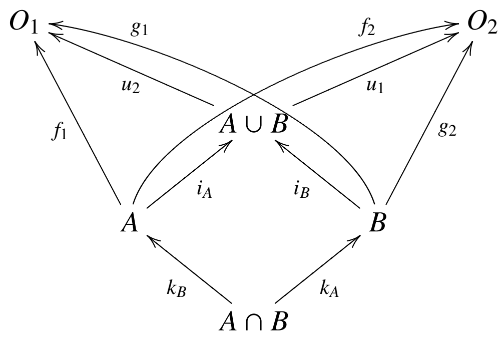


Figure 3.4: Union of alignments.

Definition 3.1.8 (Union of alignments) The union is the pushout of the intersection. Using the former notations, let $\langle C, i_A, i_B\rangle$ be the pushout of $\langle k_A, k_B\rangle$. The union of A and B is a V-alignment $\langle |A \cup B|, u_1, u_2\rangle$ such that $|A \cup B| = C$ and u_1 is the factorization of f_1 through i_A and u_2 is the factorization of f_2 through i_B.

The definition is visualized in Figure 3.4.

The following example shows how union allows the definition of better alignments based on several weak or partial ones.

Example 3.1.9 Let us assume there is a V-alignment C between O_1 and O_3 given by the relation $\{(Volant, Avion)\}$. Then, the intersection is just an empty alignment (i.e. alignment with an empty underlying ontology). Union is the alignment given by relation {(*Vehicle*, *Vehiculo*), (*Seaplane*, *Hydroavion*), (*Volant*, *Avion*)}.

Intersection and union have the following properties:

Property 3.1.10 Consider ontologies O_1 and O_2 and A, B, C V-alignments between them. The following properties hold:

- $A \cap B = B \cap A$ and $A \cup B = B \cup A$ (commutativity).
- $(A \cap B) \cap C = A \cap (B \cap C)$ and $(A \cup B) \cup C = A \cup (B \cup C)$ (associativity).

Remark: In the general case, the properties of union and intersection do not coincide with those of set-theoretic union and intersection. For example, take the alignment E with underlying

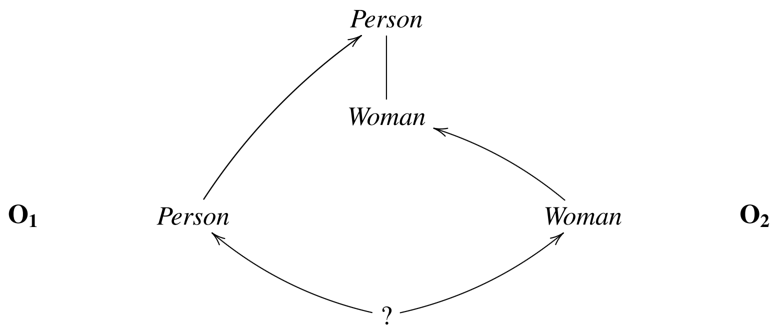


Figure 3.5: The expressivity of V-alignments is limited.

set $|E|$ = {*Objet-Object*,*Objet'-Object'*} and obvious associated projections. Then $E \cap E \neq E$. This is because E is a non-canonical representation of relation {(*Objet*,*Object*)}.

3.1.3 The expressivity of simple alignments

In the previous section, we showed examples where the only relation existing between entities was equivalence of primitive concepts. In many other cases, the two ontologies to align are designed in such a way that a concept does not have its exact equivalent in the other ontology, although several concepts are closely related. For instance, one may find that concept *Woman* in ontology O_1 is a subclass of concept *Person* in ontology O_2. Let us look at this example more closely. In this case, the merge should obviously contain a concept *Person* and a concept *Woman* with a subsumption relation between them (see Figure 3.5). However, assuming this is the result of a pushout operation, it is not clear what the alignment should be.

Indeed, if morphisms are mere functions, then this simply cannot be a pushout. To the best of our knowledge, this problem has not really been investigated yet. Consequently, we have to consider other possible approaches to solve the problem:

1. Find more complex categories, where objects still are ontologies, but with morphisms able to express other relations;

2. Keep the category simple, and complexify the definition of an alignment using more elaborate structure;

3. Change the definition of the merge, for example by using a different type of colimit.

Although, it has not been established by formal proof, it seems not possible to design a concrete category of ontologies that would, at the same time, preserve the merge-as-pushout property and be able to express complex alignments (with subsumption and other non-symmetrical relations). Examples of categories of ontologies found in the literature are categories of theories or presentations in an institution [Goguen and Burstall, 1992]; or categories of local logic in the information flow theory [Barwise and Seligman, 1997]. They only offer the capability to express equivalence relations in our V-alignment framework. Their lack of expressivity with regard to V-alignments is a consequence of their being mere functions. However, by using relations or even sets of relations as morphisms, expressiveness increases at the cost of losing the merge-as-pushout principle.

However, these failed attempts do not prove the non-existence of a category in which most conceivable alignments are expressible as V-alignments. Indeed, if we consider ontology alignments as a (generalized) relation between valid interpretations of the ontologies, then one can define a highly non-practical category that is able to express any kind of alignments. Nonetheless, the lesson learned from the aforementioned investigations is that describing explicitly (syntactically) complex alignments with V-alignments is likely not feasible in practice. As a result, the second approach was examined and offered reasonable advantages among which the possibility to use already known concrete categories of ontologies. In the next section, a more complex structure, called W-alignment, is defined to improve on the expressivity of V-alignments. Under this extension, we also have to modify the formalization of merging, such that this approach also encompasses item (3) above.

3.2 Complex Alignments

As discussed in the previous section, the simplicity of V-shaped alignments comes at the price of a reduced expressivity of these constructions. In principle, they can only be used to formalize equivalences between syntactical expressions of the two ontologies.[5] In practical applications, ontologies can be related in much more complicated ways, that may involve complex, in general non-symmetrical relationships between their elements. In order to overcome this apparent restriction of our approach, this section introduces an extended formulation for alignments that we will suggestively dub *W-alignments*.

3.2.1 Bridge axioms for ontology alignment

Let us start with the simple example from Section 3.1.3: Consider two OWL-ontologies O_1 and O_2 that contain the atomic concepts *Woman* and *Person*, respectively. Assuming that none of the ontologies contains both concepts, it is not possible to express the intended subsumption of *Woman* and *Person* with V-alignments.[6]

As a solution, one might consider introducing additional, possibly non-symmetric, types of relations between ontological concepts, thereby leaving the purely categorical formulation. This idea resembles the framework for alignments introduced in Deliverable 2.2.1 [Bouquet *et al.*, 2004c]. However, the obvious idea of extending correspondence relations between elements of an ontology (inside or outside a categorical setting) has strong limitations. While subsumption appears to be an important type of relationship, it is by far not the only one: disjointness of concepts, incomparability, or relationships between more than two concepts are examples of further relationships that are interesting in practice.

In effect, it is easy to see that there are infinitely many relevant relationships that could be considered in this context. For instance *1:WW2*, and instance of concept *1:HistoricalEvent* in one ontology, may be related to *2:TwentiethCentury*, an instance of concept *2:PeriodOfTime* in another ontology, via relation *occursDuring*. Explicitly introducing types for such relations seems

[5] To be more precise, it is possible to express complex alignments, but this will in general imply that pushout and merge will no longer coincide. The coincidence of pushout and merge, however, is a desirable property in many, though not all, ontology alignment manipulations.

[6] But note footnote 5.

$$\begin{tikzcd}
O_1 & & B & & O_2 \\
& A_1 \arrow[ul] \arrow[ur] & & A_2 \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Figure 3.6: A W-alignment between two ontologies O_1 and O_2.

not to be feasible, since the required relations depend on the structure of the alignment that one wants to express. We therefore prefer another well-known approach towards the formalization of ontological alignments, namely the introduction of *bridge axioms*.

Bridge axioms are arbitrary (onto)logical statements that describe the relationship between ontological concepts. A very simple case of bridge axiom for the above example is the statement "$\mathit{Woman} \sqsubseteq \mathit{Person}$" which expresses exactly the intended connection between the concepts. In general, this technique amounts to providing another set of ontological axioms, i.e. a third ontology, that describes how two ontologies are related. Clearly, this allows to formalize many types of relationships, but only within the borders of the ontology language under consideration. On the other hand, merging ontologies that are aligned in such a way becomes rather easy, since all connections are already available as expressions of the ontology language.

3.2.2 A categorical formulation of bridge axioms

It is not too complicated to cast the informal description of the previous section into a definition of more complex alignments along the lines of the categorical approach described earlier. The way to do this is to represent bridge axioms in form of an additional *bridge ontology*. The fact that certain concepts of the aligned ontologies occur within the bridge ontology is captured by V-alignments between the bridge ontology and each of the aligned ontologies. We thus arrive at the following definition:

Definition 3.2.1 A *W-alignment* between two ontologies O_1 and O_2 consists of a *bridge ontology* B together with two V-alignments between O_1 and B and between O_2 and B, respectively.

The situation is depicted in Figure 3.6, which also serves to illustrate why the above terminology was chosen. Note also that we do not impose any restrictions on the bridge ontology B. In particular B could contain axioms that are related to neither O_1 nor O_2. While this could be excluded by further specification, we shall continue working with the general definition above, and discuss possible restrictions later on.

Based on this categorical formulation, it is now quite easy to give a suitable definition for merging of ontologies that are aligned with a W-alignment.

Definition 3.2.2 Given two ontologies O_1 and O_2 and a W-alignment between them, the *merge* of O_1 and O_2 is defined to be the colimit of the respective alignment diagram.

More explicitly, this colimit M is computed by successive pushouts as in Figure 3.7.

The ontologies O_1^+ and O_2^+ play an interesting role in ontology merging. Intuitively, they represent the original ontologies O_1 and O_2 extended with certain axioms and elements that enable us to express their alignment as a simple V-alignment. This idea is not entirely new, and elsewhere O_1^+

$$
\begin{tikzcd}
 & & M & & \\
 & O_1^+ \arrow[ur, dashed] & & O_2^+ \arrow[ul, dashed] & \\
O_1 \arrow[ur, dashed] & & B \arrow[ul, dashed] \arrow[ur, dashed] & & O_2 \arrow[ul, dashed] \\
 & A_1 \arrow[ul] \arrow[ur] & & A_2 \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

Figure 3.7: Merging with W-alignments.

and O_2^+ have been called *portal ontologies*, referring to their specific role in making the knowledge of each of the ontologies accessible to the other one [Kent, 2000].

Due to our consistent categorical treatment of alignments, we can re-use earlier results to describe the merge in concrete cases. It is easy to see that merging in logic-based ontology languages like OWL can be achieved by just joining the axioms of both ontologies and the bridge ontology and identifying elements that are equivalent according to the V-alignments A_1 and A_2. Alternatively, one can compute the merge stepwise by constructing three pushouts, which will yield essentially the same result.

3.2.3 Composing W-alignments

Due to the increased complexity of W-alignments, a full-featured algebra along the lines of Section 3.1.2 would be more complicated than in the case of V-alignments. Nonetheless, we can easily describe a useful operation for composing W-alignments.

Definition 3.2.3 Consider ontologies O_1, O_2, and O_3 with W-alignments as in Figure 3.8. The *composition* of the W-alignments of Figure 3.8 is described as follows:

- The bridge ontology B is obtained as the merge of the bridge ontologies B_1 and B_2, according to the W-alignment $\langle O_2, A_2, A_3 \rangle$,
- the V-alignment of O_1 and B is $\langle A_1, f_1, b_1 \circ g_1 \rangle$, and
- the V-alignment of O_3 and B is $\langle A_4, g_4, b_1 \circ f_4 \rangle$.

The idea behind this definition is quite obvious: we know that there is a relation of O_1 and O_3, given by means of an intermediate ontology O_2. In order to describe this by a single bridge ontology, we just integrate both of the involved bridges with O_2. This construction has the advantage that it faithfully captures all information that is available about the composed alignment.

However, there is a major problem with the above definition: by deriving bridge axioms from the ontologies B_1, B_2, and O_2, we incorporate all the information in these knowledge bases into the new bridge ontology. But this set of bridge axioms might be highly redundant for the given purpose. For example, it may involve axioms of O_2 that are neither related to O_1 nor to O_3. Another pathological case is when O_2 is just the (disjoint) union of O_1 and O_3, while O_1 and O_3 are not

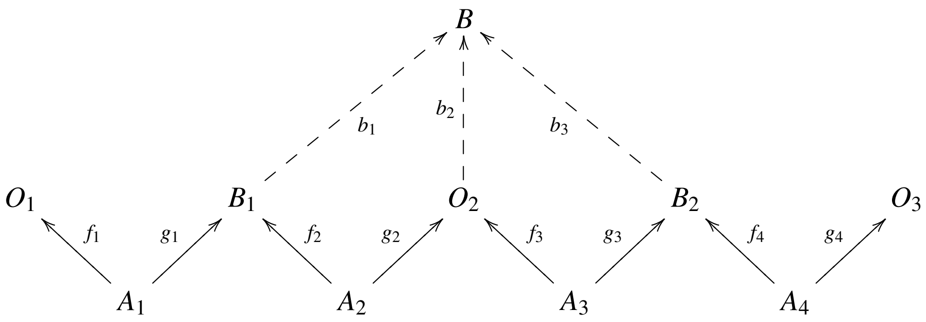


Figure 3.8: Computing the composition of W-alignments.

related at all. In this case, we would rather wish the composed alignment to be empty, instead of containing the whole information of all involved ontologies.

Overcoming this difficulty relates to the problem of finding a minimal non-redundant set of axioms that yields a given set of desired (or relevant) conclusions. Unfortunately, logical languages tend to be highly non-local in this respect and a naïve syntactical approach of casting out irrelevant information is not feasible. One such overly simple idea would be to ignore bridge axioms that do not involve elements from the languages of both of the aligned ontologies. However, as the above example with concepts *Woman* and *Person* shows, it might be the case that meaningful bridge axioms relate elements exclusively to entities of the bridge ontology, which in turn carries the meaningful interrelation between its elements. Detecting whether some axiom eventually contributes to a relationship of elements from the aligned ontologies involves complex reasoning tasks, and we therefore do not attempt to provide a concise definition of a minimal bridge ontology here—this may be subject to further research.

3.3 Expressing DFOL Mappings

In this section, we describe how to model in our category-theoretic setting the formalisms specified in Chapter 2. In particular, we define a translation of the four interpretation constraints presented in Figure 2.1 into bridge axioms, thus proving a case in point that the W-alignments introduced above are indeed expressive enough to capture most of the previously designed approaches. Then a short informal discussion of soundness and completeness is given. The primary role of this section is to emphasize the generality of the categorical abstraction by embedding an example approach into our framework.

To achieve this goal, we will rely on a category of theories and theory morphisms as defined by institution theory [Goguen and Burstall, 1992]. In short, such a category has ontologies as objects, and its morphisms are just functions between the signatures (i.e. the vocabularies of the languages used to express the ontologies) of the ontologies. Additionally, these morphisms should preserve semantics, that is, they preserve semantical consequences between translated formulas. Since we are trying to model mappings in distributed first-order logic DFOL, we are only considering ontologies written in a fragment of first-order logic. So an object (ontology) O_i is a pair (L_i, A_i) where L_i is a first-order language with equality and A_i is a set of axioms in this language.

Now let us consider a set of interpretation constraints $\mathcal{IC}$ in DFOL, expressing an ontology alignment between O_1 and O_2. Its translation in the category-theoretic framework will be the W-alignment defined as follows (see Figure 3.6 for notation):

- O_1 and O_2 are the two ontologies to align, having signatures L_1 and L_2 respectively; for the sake of simplicity, we will assume $L_1 \cap L_2 = \emptyset$, which is not restrictive since renaming the elements of the vocabularies would result in an isomorphic language with the same semantics;
- B is the bridge ontology, written in the language $L_1 \cup L_2 \cup \{R\}$ with $R \notin L_1 \cup L_2$ being a binary predicate;
- A_1 and A_2 are ontologies with signatures L_1 and L_2, respectively, which do not contain any axiom;
- $f_1 : A_1 \to O_1, g_1 : A_1 \to B, f_2 : A_2 \to O_2, g_2 : A_2 \to B$ are ontology morphisms such that for each $e_i \in L_i$ we have that $f_i(e_i) = g_i(e_i) = e_i$.

The bridge ontology is built in the following way. For each constraint $c \in C$, the following axioms are in B:

1. if c is of the form $i : P(x^{\to j}) \to j : Q(x)$ then add axiom $\forall x(P(x) \to \forall y(R(x, y) \to Q(y)))$;
2. if c is of the form $i : P(x) \to j : Q(x^{i\to})$ then add axiom $\forall x(P(x) \to \exists y(R(x, y) \wedge Q(y)))$;
3. if c is of the form $j : Q(x^{i\to}) \to i : P(x)$ then add axiom $\forall y(Q(y) \to \forall x(R(x, y) \to P(x)))$;
4. if c is of the form $j : Q(x) \to i : P(x^{\to j})$ then add axiom $\forall y(Q(y) \to \exists x(R(x, y) \wedge P(x)))$.

The axioms can be rewritten in a more standard way as follows:

1. $\forall x \forall y \neg P(x) \vee Q(y) \vee \neg R(x, y)$;
2. $\forall x \exists y \neg P(x) \vee Q(y) \wedge R(x, y)$;
3. $\forall y \forall x \neg Q(y) \vee P(x) \vee \neg R(x, y)$;
4. $\forall y \exists x \neg Q(y) \vee P(x) \wedge R(x, y)$.

3.3.1 Semantic considerations

The soundness and completeness of the translation just given depends on the semantics given to W-alignments. This semantics could be adequately described using e.g. institution theory, but it is not our goal to present this theory here. However, we will give an insight into the semantics by restricting ourselves to interpreting only W-alignments built with the translation given above. Moreover, in any case the semantics of the W-alignments relies on the underlying semantics of the language of the ontology. So we will use *interpretations* or *models* of an ontology to denote *local* interpretations or models.

In each interpretation $\mathcal{I}$ of B, the predicate R will be interpreted as a binary relation r_{12}. Moreover, since L_1 and L_2 are disjoint and do not contain the predicate R, the union $\mathcal{I} = m_1 \cup m_2 \cup \{r_{12}\}$ of interpretations m_1 and m_2 of L_1 and L_2, respectively, and the relation r_{12}, is an interpretation of the ontology B. Note that, $\langle\{m_1, m_2\}, \{r_{12}\}\rangle$ is a DFOL model of the distributed knowledge base $\{O_1, O_2\}$. With these notations, we have the following property:

Proposition 3.3.1 $\langle\{m_1, m_2\}, \{r_{12}\}\rangle$ satisfies the interpretation constraints $\mathcal{IC}$ iff $\mathcal{I}$ is a (first-order) model of B. This generalizes easily to a set of local models $\mathcal{M}_1$ and $\mathcal{M}_2$.

So the translation is at least complete. Additionally, if we restrict to this type of W-alignment, with interpretations $\mathcal{I}$ defined above, then the translation is also sound. A generalization is possible but we omit details here as this section aims at exemplifying the way concrete approaches can be embedded in our theoretical framework. Once such an embedding is done, the concrete approach automatically inherits the algebra as well as properties obtained in the abstraction.

Example 3.3.2 We exemplify the previous definitions by translating the example given in Chapter 2, Section 2.1, where the following mappings are given:

1. $1\!:\!Volant(x^{\rightarrow 2}) \rightarrow 2\!:\!Vehicle(x)$;
2. $1\!:\!Maison(x) \rightarrow 2\!:\!Domicile(x^{1\rightarrow})$;
3. $2\!:\!Seaplane(x^{1\rightarrow}) \rightarrow 1\!:\!Hydravion(x)$;
4. $2\!:\!Vehicle(x) \rightarrow 1\!:\!Vehicule(x^{\rightarrow 2})$.

We first need to work within a concrete category of ontologies. DFOL assumes ontologies are written in a fragment of first-order logic, for which institution theory [Goguen and Burstall, 1992] provides a category of first-order theories. In this theory, ontologies are pairs $\langle\Sigma, A\rangle$ where Σ is a signature, i.e. the symbols used to write the ontology (predicates, constants, functions, etc.), and A is a set of axioms. Simply said, ontology morphisms are signature morphisms such that the axioms of the first theory (ontology) are mapped to theorems of the second theory. So ontology O_1 signature is the set $\{Objet, Maison, Residence, Vehicule, Volant, Flottant, Hydravion\}$, and its axioms are $\forall x Residence(x) \rightarrow Maison(x)$, $\forall x Maison(x) \rightarrow Objet(x)$, etc. O_2 is defined in a similar way. We build the bridge ontology as follows: its signature is $\{Volant, Vehicle, Maison, Domicile, Seaplane, Hydravion, Vehicule, R\}$ and its axioms are $\forall x\forall y \neg Volant(x) \vee Vehicle(y) \vee \neg R(x,y)$, $\forall x\exists y \neg Maison(x) \vee Domicile(y) \wedge R(x,y)$, $\forall y\forall x \neg Seaplane(y) \vee Hydravion(x) \vee \neg R(x,y)$, $\forall y\exists x \neg Vehicle(y) \vee Vehicule(x) \wedge R(x,y)$. The two V-alignments just relate predicate symbols of B to those of O_1 and O_2 when they have the same name, with no additional axioms.

3.4 Chapter Conclusions

In this chapter, we discussed general approaches for modelling alignments for a broad range of ontology languages. The basic concept of V-alignments is quite close in spirit to the alignment framework that was discussed in Deliverable D2.2.1, but it is more restricted in its expressivity. Building on these investigations, aligning with bridge axioms was formalized using W-alignments. This concept of alignment allows for global statements about the relationship between two ontologies, and is thus in a way more expressive than the framework based on binary relations. W-alignments appear to be a flexible and useful approach for the category-theoretical modelling, and we suggest that further investigations on category-theoretic aspects of alignment should build on these.

In general, the categorical viewpoint on alignments also allows to abstract from the details of a particular ontology language, and, in this general setting, to identify principle strengths of available alignment expression languages. In this sense, when further developed into a categorical framework for various concrete alignment formalisms, the presented approach can help potential users to have the required knowledge when choosing such a language.

On the other hand, it must be admitted that to-date non-local alignments with bridge axioms cannot easily be created in an automated fashion. The reason is that they express relationships that are very complex, and the search for such alignments imposes huge conceptional challenges. However, bridge axioms still have their use as an elegant and efficient formulation of alignments. Without introducing any meta-logical formalisms, they can already express complicated relations based on the underlying ontology language. This suggests them for use by human ontology engineers: anybody who is familiar with the given ontology language can also specify the relationships between the according ontologies easily. It thus might be beneficial to investigate ways of employing both paradigms for alignment in practical applications. Future research may furthermore yield efficient and practical algorithms for automated or semi-automated complex alignments based on W-alignments.

Chapter 4

Conclusions

We have advanced the state of the art by providing a general perspective on ontology alignment and merging. In Chapter 3, W-alignments have been identified as the preferable means of studying aligments from an abstract category-theoretic perspective. Concrete instances of W-alignments are alignments expressed in distributed first-order logic DFOL. In turn, as shown in Chapter 2 DFOL alignments generalize and unify a variety of alignment frameworks proposed in the literature including ϵ-connections.

The studies reported in this deliverable form a foundation for further comparative work on ontology alignment and merging, with the ultimate goal of arriving at clearly defined set of sound and well-understood frameworks together with their relationships and guidelines for their application in practical settings.

Bibliography

[Adámek *et al.*, 2004] Jiří Adámek, Horst Herrlich, and George E. Strecker. *Abstract and Concrete Categorie (The Joy of Cats)*. 2004. Corrected version of the 1990 book of the same name, available online at `http://katmat.math.uni-bremen.de/acc/`.

[Antoniou and van Harmelen, 2004] G. Antoniou and F. van Harmelen. Web Ontology Language: OWL. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 67–92. Springer, 2004.

[Barwise and Seligman, 1997] John Barwise and Jerry Seligman. *Information flow: the logic of distributed systems*. Cambridge University Press, 1997.

[Bench-Capon and Malcolm, 1999] Trevor J. M. Bench-Capon and Grant Malcolm. Formalising ontologies and their relations. In *Database and Expert Systems Applications*, pages 250–259, 1999.

[Borgida and Serafini, 2003] A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, 1:153–184, 2003.

[Borgida, 1996] A. Borgida. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82:353–367, 1996. Research Note.

[Bouquet *et al.*, 2004a] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. Contextualizing ontologies. *Journal on Web Semantics*, 1(4):1–24, 2004.

[Bouquet *et al.*, 2004b] Paolo Bouquet, Marc Ehrig, Jérôme Euzenat, Enrico Franconi, Pascal Hitzler, Markus Krötzsch, Luciano Serafini, Giorgos Stamou, York Sure, and Sergio Tessaris. Specification of a common framework for characterizing alignment. Deliverable D2.2.1v2, Knowledge Web, December 2004.

[Bouquet *et al.*, 2004c] Paolo Bouquet, Jérôme Euzenat, Enrico Franconi, Luciano Serafini, Giorgos Stamou, and Sergio Tessaris. Specification of a common framework for characterizing alignment. Deliverable D2.2.1v1, Knowledge Web, August 2004.

[Calvanese *et al.*, 2002a] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Description logics for information integration. In A. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond*, volume 2408 of *Lecture Notes in Computer Science*, pages 41–60. Springer, 2002.

[Calvanese *et al.*, 2002b] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. A framework for ontology integration. In Isabel Cruz, Stefan Decker, Jerome Euzenat, and Deborah McGuinness, editors, *The Emerging Semantic Web*, pages 201–214. IOS Press, 2002.

[Franconi and Tessaris, 2004] E. Franconi and S. Tessaris. Rules and queries with ontologies: a unified logical framework. In *Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'04)*, 2004.

[Ghidini and Giunchiglia, 2001] C. Ghidini and F. Giunchiglia. Local model semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

[Ghidini and Serafini, 2000] C. Ghidini and L. Serafini. Distributed first order logics. In D.M. Gabbay and M. De Rijke, editors, *Frontiers of Combining Systems 2*, pages 121–139. Research Studies Press Ltd., 2000.

[Ghidini and Serafini, 2005] Chiara Ghidini and Luciano Serafini. Distributed first order logic - revised semantics. Technical report, ITC-irst, January 2005.

[Goguen and Burstall, 1992] J. Goguen and R. Burstall. Institutions: abstract model theory for specification and programming. *Journal of the ACM*, 39, 1992.

[Grau *et al.*, 2004] Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Working with multiple ontologies on the semantic web. In *Proceedings of the Third Internatonal Semantic Web Conference (ISWC2004)*, volume 3298 of *Lecture Notes in Computer Science*, 2004.

[Guttag and Horning, 1978] John V. Guttag and James J. Horning. The algebraic specification of abstract data types. *Acta Informatica*, 10:27–52, 1978.

[Hitzler *et al.*, 2005] Pascal Hitzler, Markus Krötzsch, Marc Ehrig, and York Sure. What is ontology merging? – A category-theoretical perspective using pushouts. In Pavel Shvaiko, Jerome Euzenat, Alain Leger, Deborah L. McGuinness, and Holger Wache, editors, *Proceedings of the First International Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&0). Workshop at the 20th National Conference on Artificial Intelligence, AAAI-05, Pittsburgh, Pennsylvania*, pages 104–107. AAAI Press, Menlo Park, California, July 2005. Technical Report WS-05-01.

[Horrocks *et al.*, 2003] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From shiq and rdf to owl: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[Kalfoglou and Schorlemmer, 2002] Yannis Kalfoglou and Marco Schorlemmer. Information-flow-based ontology mapping. *Lecture Notes on Computer Science*, 2519, 2002.

[Kalfoglou and Schorlemmer, 2005] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: The state of the art. In Yannis Kalfoglou, Marco Schorlemmer, Amit Sheth, Stephen Staab, and Michael Uschold, editors, *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005.

[Kent, 2000] Robert E. Kent. The information flow foundation for conceptual knowledge organization. In *Proceedings of the 6th International Conference of the International Society for Knowledge Organization (ISKO)*, Toronto, Canada, 2000.

[Kutz *et al.*, 2004] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.

[Pierce, 1991] Benjamin C. Pierce. *Basic Category Theory for Computer Scientists*. Foundations of Computing. The MIT Press, Cambridge, MA, 1991.

[Ullman, 1997] Jeffrey D. Ullman. Information integration using logical views. In *Proceedings of the 6th International Conference on Database Theory*, volume 1186 of *Lecture Notes In Computer Science*, pages 19 – 40, 1997.

[Verheijen *et al.*, 1998] Danladi Verheijen, Gio Wiederhold, Jan Jannink, and Srinivasan Pichai. Encapsulation and composition of ontologies. In *The Fifteenth National Conference on Artificial Intelligence, AAAI'98, Madison, Wisconsin (USA)*, July 1998.

Related deliverables

A number of Knowledge web deliverable are clearly related to this one, and they are listed in the following.

Project	Number	**Title** and relationship
KW	D2.1.1	**State of the art on the scalability of ontology-based technology** studies the use of modularity for the purpose of scalability. The composition of modules can raise heterogeneity problems that are naturally solved by using alignment results. The techniques for this are found in the present deliverable.
KW	D2.1.2.2v1	**Report on realizing practical approximate and distributed reasoning for ontologies** elaborates on D2.1.1 and extends the study of modularity for the purpose of scalability.
KW	D2.1.3.1	**Report on modularization of ontologies** elaborates on D2.1.1 and studies partitioning algorithms for large ontologies into smaller modules, distributed reasoning, engineering approaches to ontology modularization, and composition, i.e. the inverse to modularization.
KW	D2.2.1v1	**Specification of a common framework for characterizing alignment** defines a common semantic framework for characterizing alignment of heterogeneous information.
KW	D2.2.1v2	**Specification of a common framework for characterizing alignment** extends D2.2.1v1 and parts of it are the base for the studies reported on in D2.2.5.
KW	D2.2.2	**Specification of a benchmarking methodology for alignment techniques** considers potential strategies for evaluating ontology alignment algorithms. It complements the foundational studies undertaken in D2.2.5.
KW	D2.2.3	**State of the art on ontology alignment** presents various pracical ways to find alignments and thus maps out the diverse ways of doing this, for which unifying perspectives are examined in D2.2.5.
KW	D2.2.4	**Description of alignment implementation and benchmarking results** is a practical comparative study of competing alignment techniques and tools.
KW	D2.2.6	**Specification of delivery alignment format** presents the general alignment format to be delivered in WP2.2.