

Shapes of Alignments

Construction, Combination, and Computation

Oliver Kutz¹, Till Mossakowski², and Mihai Codrescu²

¹ SFB/TR 8 Spatial Cognition, Bremen, Germany
okutz@informatik.uni-bremen.de

² DFKI Lab Bremen, Germany
mihai.codrescu@dfki.de
till.mossakowski@dfki.de

Abstract. We present a general approach for representing and combining alignments and computing these combinations, based on the category theoretic notions of diagram, pushout, and colimit. This generalises the possible ‘shapes’ of alignments that have been introduced previously in similar approaches.

We use the theory of institutions to represent heterogeneous ontologies, and show how the tool HETS can be employed to compute the colimit ontology of an alignment diagram.

1 Introduction

The problem of aligning, or matching, ontologies can be essentially broken down into three sub-problems: (1) the problem of discovery, i.e., the problem of finding adequate relationships or mappings between the syntactical material of different ontologies; (2) the problem of representing such possibly rather heterogeneous ‘theory connections’; and (3) the problem of computing or constructing a new super-ontology realising the intended integration.

While the first problem can be of an empirical, heuristic or statistical nature, and often requires the intervention of human experts to be adequately solved (see Euzenat and Shvaiko [2007] for a survey), (2) and (3) are purely theoretical or logical problems of adequate representation, construction, and computation.

We concentrate on the latter problems and propose a general framework for representing, combining, and computing complex alignments building on the category theoretic notions of diagram and colimit and the theory of institutions. This generalises earlier work of a similar spirit, briefly introduced and discussed in Section 3, most notably the ‘semantic integrations’ of Schorlemmer and Kalfoglou [2008] (that we call Λ -alignments), and the V- and W-alignments of Zimmermann et al. [2006].

Our approach also gives an elegant and simple solution to the problem of combining alignments, compare Section 3.5, and easily covers and unifies standard alignment problems of identifying symbols from different ontologies or keeping symbols from different ontologies with the same name apart. However, it also

covers more elaborate integration scenarios, for instance those based on \mathcal{E} -connections or DDLs, where not a simple identity but a more complex relationship between symbols is established—this is presented in Section 6.

Moreover, we briefly discuss how heterogeneous ontology alignments can be represented as diagrams using the heterogeneous specification language HET-CASL, and demonstrate how the tool HETS can be used to compute a colimit ontology of such a diagram, i.e., the required integrated super-ontology of the alignment.

2 Institutions

The study of modularity principles can be carried out to a quite large extent independently of the details of the underlying logical system that is used. The notion of **institution** was introduced by Goguen and Burstall in the late 1970s exactly for this purpose (see [Goguen and Burstall, 1992]). Institutions capture in a very abstract and flexible way the notion of a logical system by leaving open the details of signatures, models, sentences (axioms) and satisfaction (of sentences in models).

The only condition governing the behaviour of institutions is the *satisfaction condition*, stating that *truth is invariant under change of notation* (or enlargement of context):

$$M' \models_{\Sigma'} \sigma(\varphi) \Leftrightarrow M'|_{\sigma} \models_{\Sigma} \varphi$$

Here, $\sigma: \Sigma \rightarrow \Sigma'$ is a **signature morphism**, relating different signatures (or module interfaces), $\sigma(\varphi)$ is the translation of the Σ -sentence φ along σ , and $M'|_{\sigma}$ is the reduction of the Σ' -model M' to a Σ -model.

The importance of the notion of institutions lies in the fact that a surprisingly large body of logical notions and results can be developed in a way that is completely independent of the specific nature of the underlying institution—all that is needed is captured by the satisfaction condition. We refer the reader to the literature, see Goguen and Burstall [1992]; Diaconescu [2008], for full formal details.

A **theory** in an institution is a pair $T = (\Sigma, \Gamma)$ consisting of a **signature** $\text{Sig}(T) = \Sigma$ and a set of Σ -sentences $\text{Ax}(T) = \Gamma$, the **axioms** of the theory. If $T = (\Sigma, \Gamma)$ is a theory and Σ' (resp. Γ') a signature (resp. set of sentences), we write $\Sigma' \preceq T$ (resp. $\Gamma' \preceq T$) shorthand for $\Sigma' \subseteq \text{Sig}(T) = \Sigma$ (resp. $\Gamma' \subseteq \text{Ax}(T) = \Gamma$).

The models of a theory T are those $\text{Sig}(T)$ -models that satisfy all axioms in $\text{Ax}(T)$. Logical consequence is defined as usual: $T \models \varphi$ if all T -models satisfy φ . **Theory morphisms** are signature morphisms that map axioms to logical consequences.

Example 1. First-order Logic. In the institution $\text{FOL}^{\text{ms=}}$ of many-sorted first-order logic with equality, signatures are many-sorted first-order signatures, consisting of sorts and typed function and predicate symbols. Signature morphisms map symbols such that typing is preserved. Models are many-sorted first-order

structures. Sentences are first-order formulas. Sentence translation means replacement of the translated symbols. Model reduct means reassembling the model's components according to the signature morphism. Satisfaction is the usual satisfaction of a first-order sentence in a first-order structure. \square

Example 2. Relational Schemes. A signature consists of a set of sorts and a set of relation symbols, where each relation symbol is indexed with a string of sorted field names. Signature morphisms map sorts, relation symbols and field names. A model consists of a carrier set for each sort, and an n -ary relation for each relation symbol with n fields. A model reduction just forgets the parts of a model that are not needed. A sentence is a link (integrity constraint) between two field names of two relation symbols. Sentence translation is just renaming. A link is satisfied in a model if for each element occurring in the source field component of a tuple in the source relation, the same element also occurs in the target field component of a tuple in the target relation. \square

Example 3. Description Logics. Signatures of the description logic \mathcal{ALC} consist of a set B of atomic concepts and a set R of roles, while signature morphisms provide respective mappings. Models are single-sorted first-order structures that interpret concepts as unary and roles as binary predicates. Sentences are subsumption relations $C_1 \sqsubseteq C_2$ between concepts, where concepts follow the grammar

$$C ::= B \mid \top \mid \perp \mid C_1 \sqcup C_2 \mid C_1 \sqcap C_2 \mid \neg C \mid \forall R.C \mid \exists R.C$$

Sentence translation and reduct is defined similarly as in $\mathbf{FOL}^=$. Satisfaction is the standard satisfaction of description logics. $\mathcal{ALC}^{\text{ms}}$ is the many-sorted variant of \mathcal{ALC} . The description logic \mathcal{EL} restricts \mathcal{ALC} as follows: $C ::= B \mid \top \mid C_1 \sqcap C_2 \mid \exists R.C$. \mathcal{SHOIN} extends \mathcal{ALC} with role hierarchies, transitive and inverse roles, (unqualified) number restrictions, and nominals. \square

Example 4. Quantified Modal Logic. (Constant-domain) first-order modal logic $\mathbf{QS5}$ has signatures similar to $\mathbf{FOL}^=$, including variables, constants, and predicate symbols, but extended by modal operators and leaving out equality. Models are constant-domain first-order Kripke structures. Sentences follow the grammar for \mathbf{FOL} -sentences while adding the \Box operator, and satisfaction is standard modal satisfaction. \square

3 Alignments as Diagrams

3.1 A-Alignments

In the approach of Schorlemmer and Kalfoglou [2008], two ontologies O_1 and O_2 are aligned by mapping them into a common reference ontology O as follows: theories O_1 and O_2 are said to be **semantically integrated** with respect to a theory O if (1) there exist *theory interpretations* $\alpha_1: O_1 \longrightarrow O, \alpha_2: O_2 \longrightarrow O$; (2) there exist *structure reducts* $\beta_1: \mathbf{Mod}(O_1) \longrightarrow \mathbf{Mod}(O), \beta_2: \mathbf{Mod}(O_2) \longrightarrow \mathbf{Mod}(O)$; and (3) O is consistent.

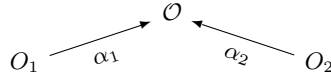


Fig. 1. A -alignment: integration into reference ontology

Example 5. [From Schorlemmer and Kalfoglou, 2008, abridged] Suppose that O_1 is a relational scheme. It contains `author_of(person,paper)` and `person(id,name)` with a relationship from `person` to `id`. O_2 is a description logic theory. It contains `Article` $\sqsubseteq \exists author.\top \sqcap \exists title.\top$, etc.

The reference ontology O is a first-order theory. It contains, among others:

$$\begin{aligned} \forall x.(Working_Person(x) \rightarrow (Tangible_Thing(x) \wedge \exists y.(String(y) \wedge Name(x,y)))) \\ \forall x.(Researcher(x) \rightarrow Working_Person(x)) \end{aligned}$$

Theory interpretations α_1, α_2 can be given as follows:

$$\begin{aligned} \alpha_1(\mathbf{person}(p,n)) &= Researcher(p) \wedge String(n) \wedge Name(p,n) \\ \alpha_1(\mathbf{author_of}(p,a)) &= Researcher(p) \wedge Article(a) \wedge Author(a,p) \wedge \\ &\quad \wedge \exists j.(Journal(j) \wedge Has_Article(j,a)) \\ \alpha_2(Article(x)) &= Publication(x) \end{aligned}$$

We will reformulate this example as a (general) heterogeneous alignment in Section 4. \square

We see the following problems with this approach³

- Allowing for arbitrary sentence maps α_i is too liberal: for example, α_i could map every sentence to *true*.⁴ It makes more sense to use *signature morphisms* and their induced sentence translation maps instead. This approach is less flexible in one aspect: with the approach of Schorlemmer and Kalfoglou [2008], e.g. in first-order logic, a predicate symbol p may be mapped to a formula φ . However, this is usually better captured by allowing for *derived signature morphisms* (see Sannella and Burstall [1983]), which here are just signature morphisms into a conservative extension (e.g. an extension by the definition $p(x) \Leftrightarrow \varphi$).
- More importantly, perhaps, there may be no suitable common reference ontology at hand. Rather, the common super-ontology should be constructed via a union of O_1 and O_2 , identifying certain concepts, while keeping others distinct. This leads to V -Alignments, discussed in the next section.

³ The aspect of logic change is ignored here, but further discussed in Section 4.

⁴ Schorlemmer and Kalfoglou [2008] suggest to solve this problem by a possible restriction to conservative translations; however, even then the translation mapping every theorem in O_i to *true* and every non-theorem to *false* still is a valid but useless example.

3.2 V-Alignments

Zimmermann et al. [2006] address the problem of alignment without a common reference ontology. Given ontologies O_1 and O_2 , an **interface** (for O_1, O_2)

$$\langle \Sigma, \sigma_1: \Sigma \longrightarrow \text{Sig}(O_1), \sigma_2: \Sigma \longrightarrow \text{Sig}(O_2) \rangle$$

specifies that (using informal but suggestive notation)

- concepts $\sigma_1(c)$ in O_1 and $\sigma_2(c)$ in O_2 are identified for each concept c in Σ , regardless of whether the concepts have the same name or not, and
- concepts in $O_1 \setminus \sigma(\Sigma_1)$ and $O_2 \setminus \sigma(\Sigma_2)$ are kept distinct, again regardless of whether they have the same name or not.

The resulting common ontology \mathcal{O} is not given a priori, but rather it is computed from the aligned ontologies via the interface. This computation is a pushout in the sense of category theory, which in this case is just a disjoint union with identification of specific parts (namely those given through $\langle \Sigma, \sigma_1, \sigma_2 \rangle$).

V-alignments can thus deal with basic alignment problems, such as *synonymy* (identifying different symbols with the same meaning) and *homonymy* (separating (accidentally) identical symbols with different meaning)—see Figure 2.

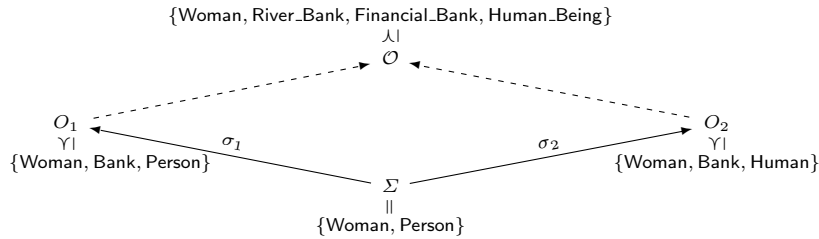


Fig. 2. V-alignment: integration through interface

Example 6. In Figure 2, the interface $\langle \Sigma, \sigma_1, \sigma_2 \rangle$ specifies that the two instances of the concept **Woman** as well as **Person** and **Human** are to be identified. This yields two concepts **Woman** and **Human_Being** in the push-out ontology \mathcal{O} obtained along the dashed arrows. It also determines that the two instances of **Bank** are to be understood as homonyms, and thus generates two new distinct concepts. \square

However, notion such as *polysemy* are typically understood to relate terms that have a different, but *related* meaning, and can thus not be dealt with by simply identifying symbols or keeping them apart. We will come back to this when discussing \mathcal{E} -connections as alignments in Section 5. Similarly, Zimmermann et al. [2006] themselves raise the criticism that V-Alignments do not cover the case where a concept **Woman** in O_1 is aligned with a concept **Person** in O_2 : here, the resulting ontology should turn **Woman** into a subconcept of **Person**. This is not directly possible with the pushout approach.

3.3 W-Alignments

In order to solve this problem of V-Alignments, Zimmermann et al. [2006] introduce W-Alignments. They consist of two V-Alignments, using an intermediate bridge ontology B . The latter can be used to specify subconcept relationships like $\text{Woman} \sqsubseteq \text{Person}$ as mentioned above.

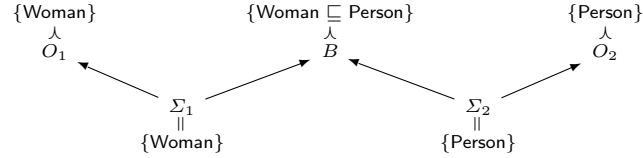


Fig. 3. W-alignment: integration through bridge ontology

Zimmermann et al. [2006] list the behaviour of compositions as a weak point of this approach. However, we see as the main weak point the rather loose coupling of O_1 and O_2 ; indeed, the bridge ontology is something like a super-ontology of a sub-ontology and hence can be anything.

3.4 M-Alignments

Given two ontologies O_1 and O_2 , let us assume that we want to align them using an interface Σ . We assume that $O_1^\#$ and $O_2^\#$ are extensions (typically conservative extensions) of O_1 and O_2 , respectively, taking into account the possible requirements to (1) define new symbols (in order to emulate a derived theory morphism), and (2) introduce new subconcept relationships, such as $\text{Woman} \sqsubseteq \text{Person}$, as discussed above. We thus arrive at the concept of M-alignment:

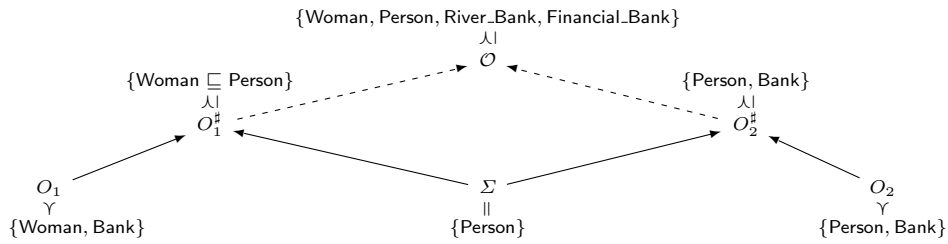


Fig. 4. M-alignment: integration through bridge along extensions

\mathcal{E} -connections as a kind of extended (and heterogeneous) M-alignment will be discussed in Section 5. Compare also Example 8 below.

3.5 General Alignments and Their Combination

Zimmermann et al. [2006] note that the composition (or better: combination) of W -alignments via pushouts resp. colimits leads to the unpleasant phenomenon that the bridge ontology of the resulting W -alignment includes the whole of one of the aligned ontologies. We think that this problem arises because colimits are used for the wrong purpose: they should be used for the computation of an integrated overall ontology, but not for the combination of alignments. Instead, the complete diagram structure of the alignments should be kept intact. This means that combination generally changes shapes of diagrams, and we hence need to generalise the notion of a (diagrammatic) alignment.

The notion of diagram is formalised in category theory. It generalises the different shapes of alignments that we have seen so far. Diagrams map an index category (via a functor) to a given category of interest. They can be thought of as graphs in the category. For details, see Adámek et al. [1990].

Definition 7. *A **general alignment** of ontologies is a diagram of theories such that the nodes are subdivided into ontology nodes and interface nodes.*

Now, combination of alignments is basically union of the diagrams. Further details may be found in Kutz and Mossakowski [2007], where also the problem of proof-theoretic and model-theoretic conservativity in diagrams is studied, a problem area that is extremely important when considering ontologies as ‘modules’ of other ontologies, cf. Lutz et al. [2007]; Cuenca Grau et al. [2008].

4 Heterogeneous Alignments

As Schorlemmer and Kalfoglou [2008] argue convincingly, since ontologies are being written in many different formalisms, like relation schemata, description logics, first-order logic, and modal (first-order) logics, alignments of ontologies need to be constructed across different institutions.

Heterogeneous specification is based on some graph of logics and logic translations, formalised as institutions and so-called institution comorphisms, see Goguen and Roşu [2002]. The latter are again governed by the satisfaction condition, this time expressing that truth is invariant also under change of notation across different logical formalisms:

$$M' \models_{\Phi(\Sigma)}^J \alpha_\Sigma(\varphi) \Leftrightarrow \beta_\Sigma(M') \models_\Sigma^I \varphi.$$

Here, $\Phi(\Sigma)$ is the translation of signature Σ from institution I to institution J , $\alpha_\Sigma(\varphi)$ is the translation of the Σ -sentence φ to a $\Phi(\Sigma)$ -sentence, and $\beta_\Sigma(M')$ is the translation (or perhaps: reduction) of the $\Phi(\Sigma)$ -model M' to a Σ -model.

The so-called **Grothendieck institution** is a technical device for giving a semantics to heterogeneous theories involving several institution [see Diaconescu, 2002; Mossakowski, 2002]. The Grothendieck institution is basically a flattening, or disjoint union, of the logic graph. A signature in the Grothendieck institution consists of a pair (L, Σ) where L is a logic (institution) and Σ is a signature in

the logic L . Similarly, a Grothendieck signature morphism $(\rho, \sigma) : (L_1, \Sigma_1) \rightarrow (L_2, \Sigma_2)$ consists of a logic translation $\rho = (\Phi, \alpha, \beta) : L_1 \rightarrow L_2$ plus an L_2 -signature morphism $\sigma : \Phi(\Sigma_1) \rightarrow \Sigma_2$. Sentences, models and satisfaction in the Grothendieck institution are defined in a componentwise manner.

Example 8. Recall Example 5 of a Λ -alignment. The kind of integration required here can be dealt with much more elegantly as a heterogeneous general alignment. For illustrative purposes, we include the full heterogeneous theory.

```

logic DL

spec Biblio_DL =
  Class: Researcher
    SubclassOf: name some Thing
  Class: Article
    SubclassOf: author some Thing, title some Thing
  Class: Journal
    SubclassOf: name some Thing, hasArticle some Thing, impactFactor some Thing
end

logic Rel

spec Biblio_RS =
  Tables
    person(key id:integer, name:string)
    author_of(person, paper:integer)
    paper(key id:integer, title:string, published_in:integer)
    journal(key id:integer, name:string, impact_factor:float)
  Relationships
    author_of[person]      -> person[id]   one_to_many
    author_of[paper]      -> paper[id]    one_to_many
    paper[published_in]   -> journal[id]  one_to_many
end

logic CASL

view Biblio_RS_in_DL : Biblio_RS to
{ Biblio_DL with logic DL -> CASL
  then %def
    preds
      journal(j,n,f:Thing) <=>
        Journal(j) /\ name(j,n) /\ impactFactor(j,f);
      paper(a,t,j:Thing) <=>
        Article(a) /\ Journal(j) /\ hasArticle(j,a) /\ title(a,t);
      author_of(p,a:Thing) <=>
        Researcher(p) /\ Article(a) /\ author(p,a);
      person(p,n:Thing) <=> Researcher(p) /\ name(p,n)
    } = logic RelationalScheme -> CASL
end

```

Here, `Biblio_DL` is a DL ontology about bibliographical information, and `Biblio_RS` is the scheme of a related relational database. The view

`Biblio_RS_in_DL` states that the ontology satisfies the relational scheme axioms (referential integrity constraints). Of course, this is not possible literally, but rather the ontology is mapped to first-order logic (CASL) and then extended definitionally to `Biblio_DL'` with a definition of the database tables in terms of the ontology classes and properties (compare the specification above after `%def`). Also, `Biblio_RS` is translated to first-order logic, yielding `Biblio_RS'`, and so the view shown in Fig. 5 as a dotted line expresses a theory morphism from `Biblio_RS'` to `Biblio_DL'`.



Fig. 5. A heterogeneous general alignment

The involved signature and theory morphisms live in the Grothendieck institution. Thus, we can avoid the use of arbitrary maps α_i as in Schorlemmer and Kalfoglou [2008] and instead rely entirely on (Grothendieck) signature morphisms.

In fact, note that the above view is *not* provable. However, it becomes provable if an inverse of the role `hasArticle` is introduced and used to restrict the class `Article`. \square

5 \mathcal{E} -Connections as Heterogeneous Alignments

In this section, we show how the integration of ontologies via ‘modular languages’ can be conceived of as specific alignments. We concentrate on \mathcal{E} -connections, but note here that DDLs [Borgida and Serafini, 2003] can be treated in exactly the same way [Kutz et al., 2004].

Originally conceived as a versatile and computationally well-behaved technique for combining logics [Kutz et al., 2004], \mathcal{E} -connections have also been adopted as a framework for the integration of ontologies in the Semantic Web [Cuenca-Grau et al., 2006]. The general idea behind this combination method is that the interpretation domains of the connected logics are interpreted by disjoint (or sorted) vocabulary and interconnected by means of *link relations*. The language of the \mathcal{E} -connection is then the union of the original languages enriched with operators capable of talking about the link relations.

\mathcal{E} -connections, just as DLs themselves, offer an appealing compromise between expressive power and computational complexity: although powerful enough to express many interesting concepts, the coupling between the combined logics is sufficiently loose for proving general results about the transfer of decidability: if the connected logics are decidable, then their connection will also

be decidable. We here introduce \mathcal{E} -connections only by way of an informal but suggestive example, for full details refer to [Kutz et al., 2004].

Given interpretations $\mathfrak{W}_i = (W_i, \cdot^{\mathfrak{W}_i})$, $i \in \{1, 2\}$, of \mathcal{S}_i , a model of the \mathcal{E} -connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2)$, where $\mathcal{E} = \{E\}$, is a structure of the form

$$\mathfrak{M} = \langle \mathfrak{W}_1, \mathfrak{W}_2, E^{\mathfrak{M}} \rangle,$$

where $E^{\mathfrak{M}} \subseteq W_1 \times W_2$. The **extension** $C^{\mathfrak{M}} \subseteq W_i$ of an i -concept C is defined by simultaneous induction. For concept names C of \mathcal{S}_i , we put $C^{\mathfrak{M}} = C^{\mathfrak{W}_i}$; the inductive steps for the Booleans and function symbols of \mathcal{S}_i are standard; finally,

$$\begin{aligned} \langle E_j \rangle^1 C^{\mathfrak{M}} &= \{x \in W_1 \mid \exists y \in C^{\mathfrak{M}} (x, y) \in E_j^{\mathfrak{M}}\}, \\ \langle E_j \rangle^2 D^{\mathfrak{M}} &= \{x \in W_2 \mid \exists y \in D^{\mathfrak{M}} (y, x) \in E_j^{\mathfrak{M}}\}. \end{aligned}$$

Example 9. Suppose two ontologies O_1 and O_2 , possibly formulated in different DLs \mathcal{S}_1 and \mathcal{S}_2 , contain the concept *Window*. Now, ontology O_1 might formalise functionalities of objects found in buildings, while ontology O_2 might be about the properties of materials of such objects. The intended relation between the two instances of *Window* might now be one of *polysemy* (meaning variation), i.e., *Window* in O_1 involves ‘something with views that can be open or closed’:

$$\text{Window} \sqsubseteq \exists \text{has_state.}(\text{Open} \sqcup \text{Closed}) \sqcap \exists \text{offers.Views},$$

while the meaning of *Window* in O_2 might be ‘something that is bulletproof glass’:

$$\text{Window} \equiv \text{Glass} \sqcap \exists \text{has_feature.Bulletproof}.$$

A systematic integration of these two ontologies could now require a mapping of objects in O_1 to the material they are made from, using a link relation ‘*consists_of*’. A concept of the form $\langle \text{consists_of} \rangle^1 C$ then collects all objects of O_1 that are made from something in C , while a concept $\langle \text{consists_of} \rangle^2 D$ collects the materials in O_2 some object in D consists of. A sensible alignment between the two instances of *Window* could now be formalised in \mathcal{E} -connections as:

$$\begin{aligned} \langle \text{consists_of} \rangle^2 \text{Window}_1 &\sqsubseteq \exists \text{has_feature.Transparent} \\ \langle \text{consists_of} \rangle^1 \text{Window}_2 &\sqsubseteq \text{Window}_1 \sqcap \exists \text{provides_security.Inhabitant} \end{aligned}$$

assuming that windows in O_1 might also be made of plastic, etc. □

As should be clear from the discussion so far, \mathcal{E} -connections can essentially be considered as many-sorted heterogeneous theories: component ontologies can be formulated in different logics, but have to be build from many-sorted vocabulary, and link relations are interpreted as relations connecting the sorts of the component logics (compare Baader and Ghilardi [2007] who note that this is an instance of a more general co-comma construction). The main difference between DDLs and various \mathcal{E} -connections now lies in the expressivity of the ‘link language’ \mathcal{L} connecting the different ontologies. While the link language of DDL

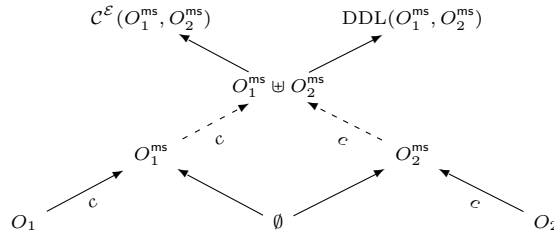


Fig. 6. \mathcal{E} -connections many-sorted: extension of an M-alignment

is a certain sub-Boolean fragment of many sorted \mathcal{ALC} , the basic link language of \mathcal{E} -connections is \mathcal{ALCT}^{ms} .⁵

Such many-sorted theories can easily be represented in a diagram as shown in Figure 6, showing an extension of an M-alignment.

6 Computation of Alignments

The Heterogeneous Tool Set HETS [Mossakowski et al., 2007a,b] provides analysis and reasoning tools for the specification language HETCASL, a heterogeneous extension of CASL supporting a wide variety of logics [CoFI (The Common Framework Initiative), 2004; Bidoit and Mosses, 2004]. In particular, OWL-DL (with *SHOIN* and its sublogics \mathcal{EL} and \mathcal{ALC} , also supporting Manchester syntax), relational schemes, as well as **FOL**^{ms} and **QS5** (using syntax of the CASL language). See the extended example in Sect. 4 for the look-and-feel of HETCASL specifications.

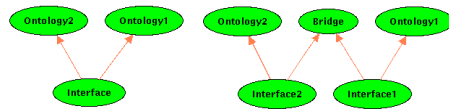


Fig. 7. Alignment diagrams in HETS.

Heterogeneous theories grouped inside a library of specifications are represented in HETS as graphs which can be displayed in a GUI window. Thus, by specifying ontologies and the mappings between them in a HETCASL library, we can visualise the diagram of the ontology alignment. Figure 7 shows the diagrams of a V- and a

W-alignment obtained with HETS.

HETS also offers an algorithmic method for computing colimits of theories in various logics, based on an implementation for computing colimits of arbitrary sets, which is further applied to sets of signature symbols, like sorts, operation and predicate symbols (the latter two divided according to profiles). As a general

⁵ But can be weakened to \mathcal{ALC}^{ms} or the link language of DDLs, or strengthened to more expressive many-sorted DLs such as \mathcal{ALCQT}^{ms} .

strategy, names are kept identical to their original as far as possible (see the example below). If this is not possible, the common origin of symbols is indicated by a (shared) number appended to their name.

Example 10. Considering the V-alignment introduced in Example 6, Figure 8 presents the HETS concept graphs of the theories combining it, as well as the one of the push-out ontology obtained with HETS (the top one). \square

The construction of colimits for heterogeneous diagrams is considerably more difficult. We refer the reader to Mossakowski [2006]; Codescu and Mossakowski [2008] for a detailed analysis of sufficient conditions for obtaining colimits of heterogeneous theories, and for a discussion of weaker notions that are useful in cases where heterogeneous colimits do not exist.

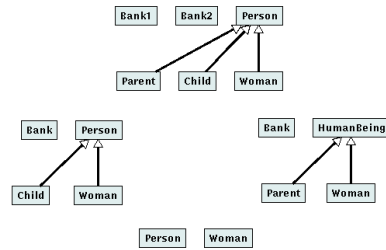


Fig. 8. Colimit of a V-alignment in HETS.

7 Discussion and Outlook

We have introduced an abstract framework of general alignments that remedies shortcomings of similar frameworks that have been discussed in the literature. The framework allows for a systematic and conceptual analysis of approaches that were previously considered rather disparate. More importantly, it makes possible generic algorithms for heterogeneous alignment problems, as have been implemented in the Heterogeneous Tool Set.

An essential prerequisite for the representation of alignments as diagrams is of course the discovery of alignment mappings of various kinds. While this was not the subject of this paper, we work on integrating a tool for finding theory morphisms into the Heterogeneous Tool Set. This tool, together with other known alignment tools, could then be used as a basis for finding alignment diagrams.

Acknowledgements

Work on this paper has been supported by the Vigoni program of the DAAD, by the DFG-funded collaborative research center SFB/TR 8 Spatial Cognition, and by the German Federal Ministry of Education and Research (Project 01 IW 07002 FormalSafe).

We thank John Bateman, Joana Hois, and Lutz Schröder for fruitful discussions, Dominik Lücke for implementing relational schemes and DL in HETS, and Erwin R. Catesbeiana for singling out an inconsistent alignment.

Bibliography

- ADÁMEK, J., HERRLICH, H. AND STRECKER, G. (1990). *Abstract and Concrete Categories*. Wiley, New York.
- BAADER, F. AND GHILARDI, S. (2007). Connecting Many-Sorted Theories. *The Journal of Symbolic Logic* 72, 535–583.
- BIDOIT, M. AND MOSSES, P. D. (2004). *CASL User Manual*. LNCS Vol. 2900 (IFIP Series). Springer. Freely available at <http://www.cofi.info>.
- BORGIDA, A. AND SERAFINI, L. (2003). Distributed Description Logics: Assimilating Information from Peer Sources. *Journal of Data Semantics* 1, 153–184.
- CODESCU, M. AND MOSSAKOWSKI, T. (2008). Heterogeneous colimits. In *Workshop on Modeling, Validation and Heterogeneity (MoVaH-08)*. Lillehammer, Norway, April 9 2008.
- COFI (THE COMMON FRAMEWORK INITIATIVE) (2004). *CASL Reference Manual*. LNCS Vol. 2960 (IFIP Series). Springer. Freely available at <http://www.cofi.info>.
- CUENCA GRAU, B., HORROCKS, I., KAZAKOV, Y. AND SATTLER, U. (2008). Modular Reuse of Ontologies: Theory and Practice. *JAIR* 31, 273–318.
- CUENCA-GRAU, B., PARSIA, B. AND SIRIN, E. (2006). Combining OWL Ontologies Using \mathcal{E} -Connections. *Journal of Web Semantics* 4, 40–59.
- DIACONESCU, R. (2002). Grothendieck institutions. *Applied Categorical Structures* 10, 383–402.
- DIACONESCU, R. (2008). *Institution-independent Model Theory*. Birkhäuser. To appear.
- EUZENAT, J. AND SHVAIKO, P. (2007). *Ontology Matching*. Heidelberg: Springer.
- GOGUEN, J. AND ROŞU, G. (2002). Institution morphisms. *Formal aspects of computing* 13, 274–307.
- GOGUEN, J. A. AND BURSTALL, R. M. (1992). Institutions: Abstract Model Theory for Specification and Programming. *Journal of the ACM* 39, 95–146.
- KUTZ, O., LUTZ, C., WOLTER, F. AND ZAKHARYASCHEV, M. (2004). \mathcal{E} -Connections of Abstract Description Systems. *Artificial Intelligence* 156, 1–73.
- KUTZ, O. AND MOSSAKOWSKI, T. (2007). Modules in Transition: Conservativity, Composition, and Colimits. In *Proc. of WoMO-07, Whistler, Canada*.
- LUTZ, C., WALTHER, D. AND WOLTER, F. (2007). Conservative Extensions in Expressive Description Logics. In *Proceedings of IJCAI-07*. AAAI Press.
- MOSSAKOWSKI, T. (2002). Comorphism-based Grothendieck logics. In K. Diks and W. Rytter, eds., *Mathematical Foundations of Computer Science*, volume 2420 of *LNCS*. Springer, pages 593–604.
- MOSSAKOWSKI, T. (2006). Institutional 2-cells and Grothendieck institutions. In K. Futatsugi, J.-P. Jouannaud and J. Meseguer, eds., *Algebra, Meaning and Computation. Essays Dedicated to Joseph A. Goguen*, LNCS 4060. Springer.
- MOSSAKOWSKI, T., MAEDER, C. AND LÜTTICH, K. (2007a). The Heterogeneous Tool Set. In *TACAS 2007*, volume 4424 of *LNCS*. Springer.
- MOSSAKOWSKI, T., MAEDER, C. AND LÜTTICH, K. (2007b). The Heterogeneous Tool Set. In B. Beckert, ed., *VERIFY 2007*, volume 259. CEUR-WS.
- SANNELLA, D. AND BURSTALL, R. (1983). Structured theories in LCF. In *Proc. 8th Colloq. on Trees in Algebra and Programming*, volume 159 of *Lecture Notes in Computer Science*. Springer.
- SCHORLEMMER, W. M. AND KALFOGLOU, Y. (2008). Institutionalising Ontology-Based Semantic Integration. *Journal of Applied Ontology*. To appear.
- ZIMMERMANN, A., KRÖTZSCH, M., EUZENAT, J. AND HITZLER, P. (2006). Formalizing Ontology Alignment and its Operations with Category Theory. In *Proc. of FOIS*.