Chinese Whispers and Connected Alignments

Oliver Kutz[1], Immanuel Normann[2], Till Mossakowski[3], and Dirk Walther[4]

[1] Research Center on Spatial Cognition (SFB/TR 8), University of Bremen, Germany
okutz@informatik.uni-bremen.de
[2] Department of Linguistics and Literature, University of Bremen, Germany
normann@uni-bremen.de
[3] DFKI GmbH and SFB/TR 8 Spatial Cognition, University of Bremen, Germany
Till.Mossakowski@dfki.de
[4] Faculty of Informatics, Technical University of Madrid (UPM), Spain
dwalther@fi.upm.es

Abstract. This paper investigates the idea to treat repositories of ontologies as interlinked networks of ontologies, formally captured by the notion of a hyperontology. We apply standard matching algorithms to automatically create the linkage structure of the repository by performing pairwise matching. Subsequently, we define a modular workflow to construct combinations of alignments for any finite number of ontologies. This workflow employs and makes interoperable several tools from the ontology engineering world, comprising matching, reasoning, and structuring tools, and supports in particular modular ontology extraction based on alignment, and a study and empirical analysis of (in)consistency propagation in connected alignments (the Chinese Whispers problem).

Keywords: Hyperontologies; Connected Alignments; Modularity; Consistency

1 Introduction and Problem Description

Ontology matching and alignment based on statistical methods is a relatively developed field, with yearly competitions since 2004 comparing the various strengths and weaknesses of existing algorithms.[5] In this paper, we aim at exploring the degrees to which statistical alignment may lead to inconsistency in the merged ontologies. More precisely, we aim at investigating the effects, both theoretically and practically, of **connected alignment**, i.e. aligning several ontologies that match (non-trivially) pairwise. Our general approach is to treat large repositories of ontologies (in the order of hundreds of ontologies) as our starting point to perform pairwise matching in order to obtain an interlinked network of ontologies. Formally, such networks are captured by the notion of a **hyperontology** [6]. Our work in progress is intended to answer questions such as the following:

> *Assuming pairwise alignments are consistent, how, and when, can we align further ontologies (in various orders) before we drift into inconsistency? In particular, how, and when, can we reduce the question of consistency of aligned ontologies and satisfiability of matched concepts to the consistency of aligned sub-ontologies (i.e. modules generated by the matched sub-signatures)?*

[5] See http://oaei.ontologymatching.org/2009/

We here set up the theoretical background and necessary engineering environment to give meaningful answers to such questions. In our related paper [9], we have studied techniques of information hiding to support the visualisation of the linkage structure and to allow a user to explore the complex networks resulting from pairwise matching on large sets of ontologies. We here focus on the last question mentioned above, namely how to reduce the consistency problem in an aligned network of ontologies to the consistency of merged modules generated by respective alignments, and the corresponding interoperability problem between matching, modularity, and structuring tools. We also study the effects of matching ontologies in different orders by looking at some specific examples.

Synonymy and Alignment as Colimit Computation. An essential part of the matching and alignment process is to relate and identify signature elements from different ontologies (possibly formulated in different ontology languages). Formally, this is captured by the notion of a **signature morphism**.[6] In the case of $\mathcal{OWL}$ ontologies, these are type-preserving symbol mappings of the form $\sigma : \mathsf{Sig}(O_1) \to \mathsf{Sig}(O_2)$, i.e. mapping the signature of O_1 $(= \mathsf{Sig}(O_1))$ to that of O_2, i.e. concepts to concepts, individuals to individuals, and roles to roles.[7] For a fixed ontology language, a signature morphism straightforwardly induces a sentence translation map.

V-Alignments [13] abstractly capture the alignment process for synonymous signature elements. Given ontologies O_1 and O_2, an **interface** (for O_1, O_2)

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

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 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 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 Fig. 1.

Example 1. In Fig. 1, 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. ⊣

[6] See e.g. [5] for the general institutional definition.

[7] We use the DL terminology *concept name* and *role* interchangeably with the $\mathcal{OWL}$ terminology *class* and *property*.

Notion such as *polysemy*, however, 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.[8] Similarly, [13] 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 merged ontology should turn Woman into a subconcept of Person.

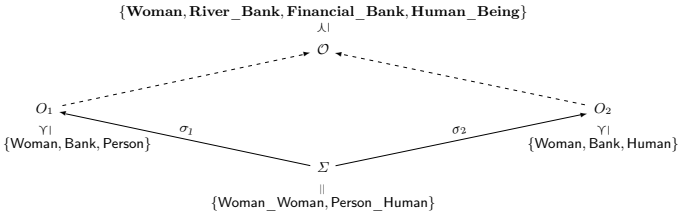

Fig. 1. V-alignment: merge through interface (dashed arrows are automatically computed via colimits)

Whilst this is not directly possible with pushouts, we are here only interested in matching *synonyms across a network of ontologies*, and for this purpose, V-alignments (and their compositions) are sufficient. Studying more complex alignment operations we leave for future work. We next turn to the problem of aligning several ontologies at once.

Consistency and Chinese Whispers. The game of **Chinese Whispers**[9] is played as follows: n persons are arranged in a certain (typically circular) order such that for each person P_i there is a j such that P_i exchanges a message with P_j. The point of the game is to observe the *distortion of the message* as it travels from P_1 along the communication channel. We here are interested in the effects of playing Chinese Whispers with ontologies, where the pairwise matching replaces the transmission of a message, i.e. the messages being exchanged are of the form: *"O_i and O_j agree that concept C of O_i is synonymous with concept D of O_j"*.

We make the following idealisations concerning 'matching' a) we assume that in pairwise matching the order does not matter, i.e. matching O_1 with O_2 yields the same colimit ontology (i.e. alignment) as matching O_2 with O_1[10]; b) matching algorithms are 'not transitive', i.e., matching $\langle O_1, O_2\rangle$ and $\langle O_2, O_3\rangle$ and computing the colimit yields, in general, a different result than matching and aligning $\langle O_1, O_3\rangle$,[11] c) we assume that we do not match ontologies with themselves.[12]

[8] This problem can be addressed by considering $\mathcal{E}$-connections as a general form of alignment (see [6]).

[9] In the United States, "Telephone" is the most common name for the game. The name "Chinese whispers" reflects the former stereotype in Europe of the Chinese language as being incomprehensible. Although it is sometimes considered offensive in the US, it remains the common British English name for the game and is not generally regarded as being offensive.

[10] Whether or not this holds for actual matching systems is an implementational artefact which we ignore; the assumption is certainly reasonable to make as both 'agreement' and 'synonymy' are symmetric.

[11] In other words, the composition of the two alignments (the composition operation is easily seen to be associative via a pullback operation, see [13]) will typically not agree with a matcher's results comparing O_1 and O_3 directly.

[12] Although one would expect to get the identity matching in such a case, actual matching tools behave sometimes rather unpredictable in these cases and often return no results.

With these assumptions in place, given a repository $\mathfrak{R}$ with N ontologies, we start with $l = \frac{N \times (N-1)}{2}$ matching pairs.

Playing chinese whispers on $\mathfrak{R}$ with $k \leq N$ players now means to pick a connected subgraph of the hyperontology graph (to ensure that each ontology 'talks' to at least one other), which we call a **matching configuration** . Note that, by assumption, matching configurations contain no loops (i.e. reflexive vertices), are undirected (because of the assumed symmetry of the matching results), and contain at most one edge between two vertices. Therefore, matching configurations are **connected simple graphs**.

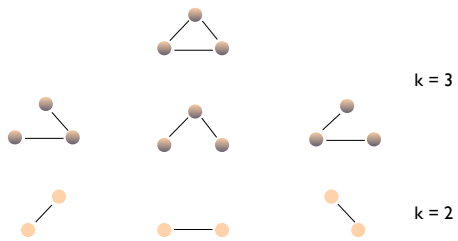

Fig. 2. The number of non-isomorphic matching configurations for $k = 2, 3$

Given a fixed k, the largest possible matching configuration (measured in pairs of matched ontologies, i.e. edges) corresponds to a clique with k nodes, i.e. a complete subgraph of the hyperontology graph with k nodes: such a graph has $\sum_{i=1}^{k-1} i$ edges.

In practise, not every pair of ontologies will match, i.e. a matcher will report no synonyms. Therefore, for fixed $k \leq N$ ontologies $O_1, \ldots, O_k$, the number of non-isomorphic matching configurations containing the O_i $(i = 1, \ldots, k)$ corresponds to the number of connected components of the hyperontology graph with these ontology nodes. The cases of $N = 2, 3$ (assuming a clique) are illustrated in Fig. 2. The alignment operation on a matching configuration, i.e. the computation of the colimit of that graph, we call a **connected alignment**. The **Chinese Whispers Problem** is now the following question:

> *For what kinds (or shapes) of matching configurations and exchanged matching results does the consistency of the input ontologies propagate to the merge (colimit) of the matching configuration?*

We will give some partial answers to this problem in Sec. 2 by showing that the consistency of an aligned matching configuration is reducible to the consistency of the alignment of 'reasonably' large modules talking only about the matched signatures. In Sec. 3, we will discuss in detail an alignment of three ontologies involving the DOLCE ontology, with pairwise consistent alignments, but an overall inconsistent one. Moreover, in Sec. 4, we will describe how the results of a matcher comparing ontologies O_1 and O_2 (and giving rise to a V-alignment) can be rewritten into a structured ontology for further processing with our tool HETS introduced below, and describe the workflow employing standard ontology matching and reasoning tools. Finally, Sec. 5 describes related and future work.

2 Modularity in Hyperontologies

In the following, we will make precise what we mean by a module and define the notion of conservativity. We start with some auxiliary notions. Let Σ be a signature containing concept names and roles. Let $\mathbf{Sen}(\Sigma)$ be the set of sentences formulated using the symbols in Σ in some language. An ontology O in signature Σ is then simply a subset of $\mathbf{Sen}(\Sigma)$. The sentences of course depend on the language the ontologies under consideration are formulated in, e.g. $\mathcal{OWL}$ or some fragment thereof.

We continue with introducing a general notion of a module in the sense that a module of an ontology is not restricted to be a subset of the ontology. It is crucial, however, that the module says everything (expressible in its signature) that is said by the ontology itself (i.e. the ontology is required to be a conservative extension of the module)—see Fig. 3 .

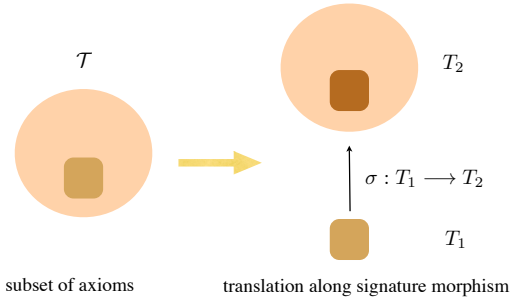

Fig. 3. Modules as subsets vs. modules as image under translation.

Definition 1. *A theory morphism $\sigma\colon O_1 \longrightarrow O_2$ is* ***consequence-theoretically conservative****, if O_2 does not entail anything new w.r.t. O_1, formally, $O_2 \models \sigma(\varphi)$ implies $O_1 \models \varphi$. Moreover, $\sigma\colon O_1 \longrightarrow O_2$ is* ***model-theoretically conservative****, if any O_1-model M_1 has a σ-expansion to O_2, i.e. a O_2-model M_2 with $M_2|_\sigma = M_1$.*

Here, $\models$ as usual denotes logical consequence, whereas $\_|_\sigma$ denotes model reduct for a signature morphism $\sigma\colon \Sigma_1 \to \Sigma_2$, i.e. for a Σ_2-model M_2, $M_2|_\sigma$ is a Σ_1-model that interprets a symbol by first translating it along σ and then interpreting it using M_2.

It is easy to show that conservative theory morphisms compose. Moreover, the notion of model-theoretic conservativity is stronger than consequence-theoretic conservativity. To be precise, the former implies the latter, but not vice versa [7]. The two notions coincide if we define consequence-theoretic conservativity using Σ-theories that contain consequences $\phi \in \mathbf{Sen}(\Sigma)$ formulated in second-order logic.

The computational complexity of deciding conservativity appears to be rather daunting even if the ontologies are formulated in weak logics. For instance, for ontologies formulated in the light-weight Description Logic $\mathcal{EL}$, deciding consequence-theoretic conservativity is ExpTime-complete, and model-theoretic conservativity is undecidable. The former problem also becomes undecidable when adding nominals to $\mathcal{ALCIQ}$, for which

it is still 2-ExpTime-complete [8]. This suggests that, for practical purposes and applications, we often have to live with approximations of these notions, more precisely with sufficient (syntactic) conditions for conservativity that allow to construct non-minimal modules. Indeed, the notion of an ontology module of an ontology O has been defined as any "subontology O' such that O is a conservative extension of O'" [1].

Definition 2 (Module Generator). *Let O be an ontology in some fixed DL, and let $\Sigma \subseteq \mathsf{Sig}(O)$ be a signature. $\mathbf{Sen}(\mathsf{Sig}(O))$ is the set of sentences in $\mathsf{Sig}(O)$. A function*

$$\Pi : \langle O, \Sigma \rangle \mapsto \mathbf{Sen}(\mathsf{Sig}(O))$$

*mapping pairs $\langle O, \Sigma \rangle$ consisting of an ontology O together with a signature Σ to a set of sentences in $\mathsf{Sig}(O)$ is called a **Σ-module generator** if for all O and Σ:*

$$\Pi(\langle O, \Sigma \rangle) \text{ is a model-theoretic } \mathsf{Sig}(O)\text{-conservative extension of } O.$$

*For a Σ-module generator Π, the set $\Pi(\langle O, \Sigma \rangle)$ is called a **Σ-module for** O. $\Pi(\langle O, \Sigma \rangle)$ is called **Σ-covering for** O if:*

$$O \text{ is a model-theoretic } \Sigma\text{-conservative extension of } \Pi(\langle O, \Sigma \rangle).$$

The idea to use (conservative) module generators is to massively reduce the size of a colimit ontology to a merge of modules generated by the matched signatures and preserving the semantics completely. This means that we can check the satisfiability of our matched concepts (and the consistency of the overall merged ontology) already in a rather small fragment of the overall ontology. Indeed, it is not hard to construct (or find) cases where ontologies have a moderate semantic overlap, but where the overall merge will be very hard to process by current tools.

Before coming to our central theoretical result, we need some preparatory notions.

Definition 3. *A **diagram** is a graph D of signatures $(D_i)_{i \in D}$ and signature morphisms $(D_m \colon D_i \to D_j)_{m \colon i \to j \in D}$. Given a diagram D, a family of models $M_i \in \mathbf{Mod}(D_i)_{i \in D}$ is **compatible**, if $M_j|_{D_m} = M_i$ for any $m \colon i \to j \in D$. A logic has the **amalgamation property** w.r.t. a class of diagrams if for any diagram in the class, any compatible family of models can be amalgamated to a unique model of the colimit of the diagram (i.e. such that the reducts along the colimit injections yields the models of the family).*

*A logic is **semi-exact**, if it has the amalgamation property for pushout diagrams.*

Note that colimits of theories can be easily defined in terms of signature colimits and unions of (translated) axioms; the amalgamation property then carries over from signature colimits to theory colimits (see [11], 4.4.17). We formulate the next results in their full generality to cover also ontology languages other than DLs. But note that they apply in particular to all usual DLs as these are semi-exact and moreover have an initial signature (i.e. a signature with a unique signature morphism into any signature) [5].

Theorem 1 (Combination of two modules). *Assume a semi-exact logic. Consider Fig. 4, and assume that $\widetilde{\mathcal{O}}$ and $\mathcal{O}$ are obtained by pushouts (with base Σ). Then $\mathcal{O}$ is a model-theoretic conservative extension of $\widetilde{\mathcal{O}}$. In particular, $\mathcal{O}$ is satisfiable iff $\widetilde{\mathcal{O}}$ is.*

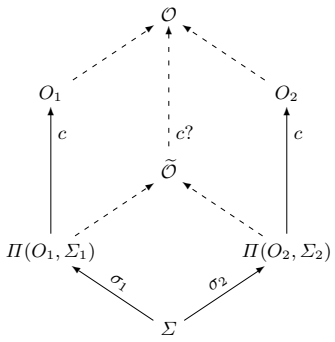

Fig. 4. Propagation of modular structure through one matching (a 'c' denotes conservativity)

Proof. Let $\widetilde{M}$ be a model of $\widetilde{\mathcal{O}}$. For $i = 1, 2$, let M_i be an O_i-expansion of the $\Pi(O_1, \Sigma_i)$-reduct of $\widetilde{M}$ (which exists by conservativity of O_i over $\Pi(O_1, \Sigma_i)$). M_1 and M_2 obviously agree on Σ, hence they have an amalgamation M which is an $\mathcal{O}$-model. Now the $\Pi(O_1, \Sigma_i)$-reducts of M agree with those of $\widetilde{M}$, hence by uniqueness of amalgamation, the $\widetilde{\mathcal{O}}$-reduct of M is $\widetilde{M}$. ⊣

We next generalise this result to the case of arbitrary matching configurations.

Theorem 2 (Combination of multiple modules). *Assume a semi-exact logic with an initial signature. Consider a family of ontologies $(O_i)_{i \in I}$ indexed by a finite non-empty set I and a simple graph $G \subseteq I \times I$, such that for $(i, j) \in G$, O_i and O_j are interfaced by*

$$O_i \xleftarrow{\theta_{i,j}} \Sigma_{i,j} \xrightarrow{\theta_{j,i}} O_j$$

Define

$$\Sigma_i := \bigcup_{j \in I \setminus \{i\}} \theta_{i,j}(\Sigma_{i,j}) \tag{1}$$

and $\sigma_i \colon \Sigma_i \to \Pi(O_i, \Sigma_i)$ the module in O_i for Σ_i. Let $\sigma_{i,j} \colon \Sigma_{i,j} \to \Pi(O_i, \Sigma_i)$ be the restriction of $\theta_{i,j}$, namely $\theta_{i,j} \colon \Sigma_{i,j} \to \Sigma_i$ composed with σ_i.[13] *Assume that $\widetilde{\mathcal{O}}$ (resp. $\mathcal{O}$) is obtained by the colimit of the diagram of all $\sigma_{i,j}$ (resp. all $\sigma_{i,j}$ composed with the inclusion of $\Pi(O_i, \Sigma_i)$ into O_i). Then $\mathcal{O}$ is a model-theoretic conservative extension of $\widetilde{\mathcal{O}}$. In particular, $\mathcal{O}$ is satisfiable iff $\widetilde{\mathcal{O}}$ is.*

Proof. Note that the diagrams for obtaining $\widetilde{\mathcal{O}}$ resp. $\mathcal{O}$ can be turned into connected diagrams by adding the inclusions of the empty signature into all involved theories (the colimit does not change by this addition). The rationale is that this ensures in $\mathcal{OWL}$ that all compatible model families are built over the same universe of individuals. More formally, by Prop. 4.4.15 of [11], in any semi-exact logic with an initial signature, all finite non-empty connected diagrams enjoy the amalgamation property. With this, the proof is a straightforward generalisation of the proof of Thm. 1. ⊣

[13] Consider Fig. 4, but replace Σ by $\Sigma_{i,j}$, σ_1 by $\sigma_{i,j}$, σ_2 by $\sigma_{j,i}$, O_1, Σ_1 by O_i, Σ_i, and O_2, Σ_2 by O_j, Σ_j.

Note that Thm. 1 does not hold for consequence-theoretic conservativity. Consider the following example, adapted from [7]. In $\mathcal{ALCO}$, let O_1 be

$$\begin{aligned} &\mathsf{IntroTCS} && \sqsubseteq \exists \mathsf{has\_subject}.\mathsf{AutomataTheory} \\ &\mathsf{IntroTCS} && \sqsubseteq \exists \mathsf{has\_subject}.\mathsf{ComplexityTheory} \\ &\mathsf{AutomataTheory} \sqcap \mathsf{ComplexityTheory} && \sqsubseteq \bot \end{aligned}$$

and O_2 be

$$\begin{aligned} \mathsf{IntroTCS} &\sqsubseteq \forall \mathsf{has\_subject}.\{\mathsf{moore\_automata}\} \\ \mathsf{IntroTCS} &\sqsubseteq \exists \mathsf{has\_subject}.\{\mathsf{moore\_automata}\} \end{aligned}$$

Let Σ be $\{\mathsf{IntroTCS}, \mathsf{has\_subject}\}$. Then assuming a consequence-theoretically conservative (minimal) module generator, $\Pi(O_1, \Sigma) = \Pi(O_2, \Sigma) = \widetilde{\mathcal{O}}$ is

$$\mathsf{IntroTCS} \sqsubseteq \exists \mathsf{has\_subject}.\top$$

But this is consistent, while $O_1 \cup O_2$ is not (assuming IntroTCS is also instantiated).

3 Worked Out Example

The following worked out example serves two purposes: it should illustrate the last theorem on combination of multiple modules, and it should demonstrate its practical impact.

Our ontology repository ORATE[14] contains among others three ontologies, namely: `SpatialRelations`, `ExtendedDnS`, and `FunctionalParticipation`. Let us denote them by O_1, O_2, and O_3 resp., to be notationally conform with Theorem 2 above. We combine them in different ways and show how consistency issues of the combined ontologies can already be answered in the combination of modules. Automated matching resulted in the concept identifications listed in the two tables below.

`SpatialRelations`	endurant	participant	place-of
`ExtendedDnS`	physical-endurant	participant-place-of	situation-place-of

Table 1. Matching `SpatialRelations` against `ExtendedDnS`

Our first matching thus induces an interface $\langle \Sigma_{12}, \theta_{12}, \theta_{21} \rangle$ with

$$\Sigma_{12} = \{\mathsf{endurant}, \mathsf{participant}, \mathsf{place\text{-}of}\},$$

$\theta_{12} = id$, and $\theta_{21} = \{\text{endurant} \mapsto \text{physical-endurant}, \text{participant} \mapsto \text{paticipant-place-of}, \text{place-of} \mapsto \text{situation-place-of}\}$.

The second an interface $\langle \Sigma_{23}, \theta_{23}, \theta_{32} \rangle$ with $\Sigma_{23} = \{\text{endurant}, \text{physical-object}, \text{region}\}$, $\theta_{23} = \{\text{endurant} \mapsto \text{non-physical-endurant}, \text{physical-object} \mapsto \text{agentive-physical-object}, \text{region} \mapsto \text{space-region}\}$, and $\theta_{32} = id$.

From Σ_{12} and Σ_{23}, the Σ_i's can be determined: $\Sigma_1 = \theta_{12}\Sigma_{12} = \{\text{physical-endurant}\}$, $\Sigma_2 = \theta_{12}\Sigma_{12} \cup \theta_{23}\Sigma_{23} = \{\text{endurant}\}$, and $\Sigma_2 = \theta_{23}\Sigma_{32} = \{\text{non-physical-endurant}\}$.

[14] `http://ontologies.informatik.uni-bremen.de`

SpatialRelations	endurant	physical-object	region
FunctionalParticipation	non-physical-endurant	agentive-physical-object	space-region

Table 2. Matching SpatialRelations against FunctionalParticipation

The signatures Σ_i, for $i = 1, 2, 3$, together with corresponding ontologies O_i are sent to the module generator that gives us for each ontology the corresponding module $M_i := \Pi(O_i, \Sigma_i)$. Finally, from the signatures Σ_i and the signature morphisms θ_{ij}, the colimit (i.e. the alignment) $\tilde{M}$ of the modules M_1, M_2, and M_3 is obtained. Similarly, the aligned ontology $\tilde{M}$ can be determined from the three ontologies O_1, O_2, and O_3. A practical result of this alignment can be a check of the merged module $\tilde{M}$ for consistency. As we would expect, the merge of the concept "physical-endurant", "endurant", and "non-physical-endurant" into a single concept makes it unsatisfiable—this result can be automatically verified by a prover. Since we know that $\tilde{O}$ must be a conservative extension of $\tilde{M}$, it is in fact not necessary to build $\tilde{O}$ and check its consistency: this information can already be inferred from $\tilde{M}$. To repair the detected inconsistency of $\tilde{M}$, we can abandon either M_1 or M_3 from the alignment process. In both cases, M_1 aligned with M_2 only, or M_3 aligned with M_2 only, the resulting merged module turns out to be consistent, and we know by conservativity that the corresponding ontologies would be consistent, too.

4 An Interoperability Workflow and Prototypical Implementation

4.1 The Component Tools

We implemented a workflow for aligning arbitrary matching configurations taking advantage of several third-party tools. We here briefly introduce these tools and describe in the subsequent subsection their interoperation. Our ontologies to be matched and aligned are taken from our ontology repository ORATE which is being developed and maintained within the EU-project OASIS[15]. The software is based on BioPortal [10]. As matching system we use Falcon [3] which matches $\mathcal{OWL}$ ontologies by means of linguistic and structural analysis. Falcon can be comfortably used in a batch mode and thus makes it suitable for a pipe workflow. For module extraction as well as consistency checks we use Pellet [12] which in particular makes use of the $\mathcal{OWL}$-API [16]. Finally, we use Hets [17] for the computation of colimits (i.e. alignments). Hets is a parsing, static analysis and proof management tool incorporating various provers and different specification languages, thus providing a tool for heterogeneous specifications.

4.2 Workflow Description

Our workflow of multiple ontology alignment consists of two phases: 1) the preprocessing of the whole repository (ORATE) to a complete list of pairwise **matching records**

[15] See http://www.oasis-project.eu
[16] See http://owlapi.sourceforge.net
[17] See www.informatik.uni-bremen.de/cofi/hets

and 2) the alignment of a connected subgraph of the **hyperontology graph**. It is up to user what part of the hyperontology graph the user selects as connected subgraph. The rationale behind this user interaction is to dismiss false matchings produced by the matching system. Different matching configuration consequently lead to different alignment outcomes.

```
preprocess_repository = {
  foreach (o1,o2) in repository
   match_ontologies o1 o2
  end
}

match_ontologies o1 o2 = {
  mapping = falcon o1 o2
  cs1 = extract_concepts_from_o1 mapping
  cs2 = extract_concepts_from_o2 mapping
  s12 = build_interface_signature cs1 cs2
  sm1 = build_interface_signature_morphism s12 cs1
  sm2 = build_interface_signature_morphism s12 cs2
  matching_record = (o1,o2,s12,sm1,sm2)
  if (not_empty mapping) store_in_hyperontology_graph matching_record
}

compute_module_graph hyperontology_subgraph = {
  foreach node in hyperontology_subgraph
    in_edges = edges incident with node
    sig = compute_interface_signature node in_edges
    module = compute_module node sig
  end
  replace nodes in hyperontology_subgraph by modules
}

align_modules hyperontology_subgraph = {
  module_graph = compute_module_graph hyperontology_subgraph
  interfaces = {compute_interface edge | edge in module_graph}
  views = {compute_views edge | edge in module_graph}
  spec = write_alignment_spec modules interfaces views
  alignment = compute_alignment spec % hets
}
```

Fig. 5. Pseudo code of the workflow for multiple ontology alignment

Fig. 5 shows the whole workflow in pseudo code. We are going to explain it now line by line. Procedure `preprocess_repository` takes each pair of ontologies `(o1,o2)` and applies the procedure `match_ontologies` to it, i.e., it matches pairwise all ontologies from the repository. The output of `match_ontologies` is a `matching_record`: the matching system FALCON computes the `mapping` between the two ontologies `o1` and `o2`. From the `mapping` the concepts `cs1` (`cs2`) belonging to ontology `o1` (`o2`) are extracted. Based on the concepts `cs1` (`cs2`), the interface signature `s12` is built and the corresponding signature morphisms `sm1` and `sm2` to the ontologies `o1` and `o2`. The two ontologies, the interface signature, and the two signature morphisms form the `matching_record`. This record can be viewed as a link between two ontologies. We call this network whose nodes are

ontologies and whose edges are the matching records *hyperontology graph*. Although all ontologies are matched pairwise, the graph is not complete, i.e., some pairs of ontologies (in practice even the majority) are not linked, namely when the matching system cannot find any mappings.

Once the hyperontology graph of the ontology repository is computed we can choose an arbitrary subgraph of it to compute the colimit of modules implicitly given in this subgraph. For that the procedure `compute_module_graph` takes the hyperontology subgraph and basically replaces its ontologies by modules extracted from them. More precisely, for each ontology (=`node`) it collects all interfaces (=`in_edges`) connected to this node and computes (according to Equation 1 in Thm. 2) a signature (`sig`) that is finally used to compute the `module`. Practically, this last step is delegated to Pellet ($\mathcal{OWL}$-API).

Aligning modules implicitly given in a hyperontology subgraph (cf. `align_modules` procedure) comprises the following steps: we first transform the graph with the just mentioned procedure `compute_module_graph` to the `module_graph`. From the module graph we extract all signatures (=`interfaces`) and signature morphisms (`views`) to the modules. From the modules, the interfaces, and the views we compose a specification document (`spec`) that can be understood by Hets. Finally, Hets computes the colimit (`alignment`) of this `spec`.

5 Related Work and an Outlook

Matching and revision in networks of ontologies seems to be a rather unexplored topic. Although only studying pairs of ontologies, the most closely related work in spirit appears to be [4], where a semi-automatic procedure is presented for the integration of ontologies that involves revision of mappings. This approach is implemented as the Protégé 4 plugin ContentMap. Here, a selected ontology matcher is used to compute mappings between the signatures of two ontologies chosen for integration, the mappings are explicitly internalised as axioms in an $\mathcal{OWL}$-2 ontology, and the result of the integration is then taken to be the (disjoint) union of the original ontologies together with the mapping axioms. The integration is assumed to be successful if a user does not identify unintended logical consequences. This decision is guided by justifications (explanations for entailment) that can automatically be computed, e.g., by ContentMap and the confidence values created by the matcher for the mapping axioms. To eliminate the unintended consequences, a repair plan is created describing which axioms from the original ontologies or the mapping should be removed. It should be noted that such a plan does not always exist and that a desired integration may require the iteration of these steps.

The main differences to our approach are a) we support heterogeneity of ontology languages, b) we do not internalise the mappings but make explicit the structure of the alignment graph, c) we generically support arbitrary matching configurations and use category-theoretic techniques to compute the merged ontologies without duplicating matched signature items. Apart from these differences, the necessity for debugging and revising matchings also applies to our approach, and the proposed techniques can be made fruitful also in this setting.

We could here only scratch the surface of the area of problems related to matching in networks of ontologies. We have laid out the necessary engineering infrastructure to combine matching, structuring, and reasoning tools, and obtained some theoretical results concerning the reduction of consistency checks to merged modules.

However, a lot of open questions remain. For instance, internalising mappings can be extended to internalising the confidence values of mappings building on similarity-based $\mathcal{E}$-connections [2]. Moreover, a full statistical analysis of major ontologies and repositories remains to be done to understand the impact of iterated matching on consistency.

Acknowledgements

Work on this paper has been supported by the DFG-funded collaborative research centre SFB/TR 8 'Spatial Cognition', the EU-funded OASIS project, and by the German Federal Ministry of Education and Research (Project 01 IW 07002 FormalSafe). Dirk Walther is supported by a 'Juan de la Cierva' postdoctoral fellowship.

References

1. Ghilardi, S., Lutz, C., and Wolter, F. Did I Damage My Ontology? A Case for Conservative Extensions in Description Logics. In *Proceedings of KR-06* (2006), pp. 187–197.
2. Hois, J., and Kutz, O. Counterparts in Language and Space—Similarity and $\mathcal{S}$-Connection. In *Proc. of FOIS 2008* (2008), C. Eschenbach and M. Grüninger, Eds., IOS Press, pp. 266–279.
3. Hu, W., and Qu, Y. Falcon-AO: A practical ontology matching system. In *Proc. of WWW-07* (2008), pp. 237–239.
4. Jimenez Ruiz, E., Cuenca Grau, B., Horrocks, I., and Berlanga, R. Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences. In *Proc. of the 6th European Semantic Web Conference (ESWC 2009)* (2009), LNCS, Springer.
5. Kutz, O., and Mossakowski, T. Conservativity in Structured Ontologies. In *18th European Conf. on Artificial Intelligence (ECAI-08)* (Patras, Greece, 2008), IOS Press.
6. Kutz, O., Mossakowski, T., and Lücke, D. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis 4*, 2 (2010). Special issue on 'Is Logic Universal?'.
7. Lutz, C., Walther, D., and Wolter, F. Conservative Extensions in Expressive Description Logics. In *Proc. of IJCAI-07* (2007), M. Veloso, Ed., AAAI Press, pp. 453–458.
8. Lutz, C., and Wolter, F. Conservative Extensions in the Lightweight Description Logic $\mathcal{EL}$. In *Proc. of CADE-07* (2007), Springer, pp. 84–99.
9. Normann, I., and Kutz, O. Ontology Reuse and Exploration via Interactive Graph Manipulation. In *Proc. of the ISWC Workshop on Ontology Repositories for the Web (SERES-2010)* (ISWC-2010, November 7, 2010, Shanghai, China), CEUR-WS.
10. Noy, N. F., Shah, N. H., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Montegut, M. J., Rubin, D. L., Youn, C., and Musen, M. A. Bioportal: A web repository for biomedical ontologies and data resources [demonstration], 2007.
11. Sannella, D., and Tarlecki, A. *Foundations of Algebraic Specifications and Formal Program Development*. Springer Verlag. to appear.
12. Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., and Katz, Y. Pellet: A practical OWL DL reasoner. *Journal of Web Semantics 5*, 2 (2007), 51–53.
13. Zimmermann, A., Krötzsch, M., Euzenat, J., and Hitzler, P. Formalizing Ontology Alignment and its Operations with Category Theory. In *Proc. of FOIS-06* (2006), pp. 277–288.