

Addressing Issues in Foundational Ontology Mediation

Zubeida Casmod Khan and C. Maria Keet

School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal, and UKZN/CSIR-Meraka Centre for Artificial Intelligence Research, Durban, South Africa
zkhan@csir.co.za, keet@ukzn.ac.za

Keywords: Foundational Ontology; Ontology Mediation; Semantic Interoperability; Ontology Alignment; Ontology Mapping; Ontology Matching; Ontology Merging.

Abstract: An approach in achieving semantic interoperability among heterogeneous systems is to offer infrastructure to assist with linking and integration using a foundational ontology. Due to the creation of multiple foundational ontologies, this also means linking and integrating those ones. In order to achieve this, we have selected the widely used foundational ontologies DOLCE, BFO, and GFO, and their related modules, on which to perform ontology mediation (alignment, mapping, and merging). The foundational ontologies were aligned by identifying correspondences between ontology entities using seven tools, documentation, and our manual alignments, and comparing their effectiveness. Thereafter, based on the alignments, we created correspondences in the ontology files resulting in entity mappings and merged ontologies. However, during the mapping process, it was found that differences in foundational ontologies, such as their hierarchical structure, conflicting axioms due to complement and disjointness, and incompatible domain and range restriction, cause logical inconsistencies in foundational ontology alignments, thereby greatly reducing the number of mappings. We analyse and present these logical inconsistencies with possible solutions to some of them.

1 INTRODUCTION

There has been an exponential growth in ontology development for the Semantic Web, including a move toward modular and networked ontologies. Foundational ontologies are commonly used to facilitate semantic interoperability. However, Semantic Web system developers use their ontologies with a preferred foundational ontology. The semantics and underlying Ontology of each foundational ontology differs, causing a problem in semantic interoperability. Heterogeneous systems on the Semantic Web are restricted to committing to a single foundational ontology in order to promote interoperability. However, no single foundational ontology is used across all systems, therewith preventing interoperability. In order for these applications to share and process information correctly, there is a need for foundational ontology interoperability, so that ontology developers committing to a preferred foundational ontology will achieve seamless linking to other domain ontologies linked to another foundational ontology. Such an infrastructure was envisioned as the “WonderWeb Foundational Ontologies Library” (WFOL) (Masolo et al., 2003), but this infrastructure still does not exist. The main preconditions for a WFOL are content comparisons and ontol-

ogy mediation. Ontology mediation refers to identifying and solving differences between heterogeneous ontologies, in order to allow reuse and interoperability. Its three main processes are alignment, mapping, and merging (de Bruijn et al., 2006). There are only few paper-based alignments of foundational ontologies, being between GFO and DOLCE (Herre, 2010) and between DOLCE and BFO (Seyed, 2009; Temal et al., 2010), which, however, are partial, with older versions of the ontologies, informal, and/or aligned but not mapped. To the best of our knowledge, no systematic comparison of the contents of foundational ontologies has been done, nor full alignments, let alone providing consistent mappings.

We aim to contribute to fill this gap of semantic interoperability by selecting three well-known foundational ontologies, DOLCE (Masolo et al., 2003), BFO (<http://www.ifomis.org/bfo>) with RO (Smith et al., 2005), and GFO (Herre, 2010) with which we perform a rigorous foundational ontology content comparison and mediation to aid in achieving foundational ontology interchangeability. The alignment process is carried out by using the manual alignment as a gold standard and (semi-)automated alignment with seven alignment tools to examine them on their capabilities to align foundational ontologies.

The accuracy and percentage of alignments that were found vary greatly among the tools due to their diverse alignment algorithms, ranging from 18 to 94% and 17 to 31%, respectively. Further alignment issues appear in the transitivity of alignments across the three foundational ontologies due to absence of some entity or conflicting parthood theories, whilst some may be resolved by asserting them as sibling classes. Mapping the aligned entities whilst keeping a consistent ontology reduces the feasible set from 85 alignments to 43 successful mappings due to disjointness and complement axioms elsewhere in the ontology, and due to incompatible domain and range axioms, which in some cases can be solved from a logic viewpoint by asserting subsumption instead. For each mediation process (alignment, mapping and mediation), we present the issues encountered for foundational ontology mediation and aim to solve these issues.

In the remainder of the paper, we provide a literature review in Section 2. A high-level content comparison of the foundational ontologies is described in Section 3, which is followed by an analysis of alignments in Section 4, and of the mappings in Section 5. We discuss the results in Section 6 and conclude in Section 7.

2 LITERATURE REVIEW

There has not been much work done in comparing content of the foundational ontologies. By this, we mean comparing their classes, properties and relations. Some work on comparing the primitive relations of BFO (i.e., the Relation Ontology (RO)) and DOLCE has been done (Seyed, 2009) where it is found that the philosophies behind the foundational ontologies affect the way the relations are modelled. For instance, BFO is based on realist principles and has no abstract entities while GFO is both descriptive and realist in nature and allows abstract entities in an ontology, and BFO's parthood relation `has_part` does not consider abstract entities, while GFO has a parthood relation `abstract.has_part` that considers abstract entities at a higher-level than its `has_part` relation.

Temal et al. (Temal et al., 2010) created a BFO-DOLCE mapping in order to integrate medical information. The classes (universals or categories) are mapped with equivalence and subsumption relations. Based on the older so-called SNAP and SPAN version of BFO, they found that all BFO universals were successfully mapped to DOLCE, but not all DOLCE entities could be mapped to BFO. These alignments were not checked on consistency of the mappings and were done on some First Order Logic version of the ontologies, where the SNAP-BFO has, e.g., Boundary,

that BFO v1.1 in OWL does not have, and DOLCE is claimed to have Collection, which appears neither in the principal documentation (Masolo et al., 2003) nor in the OWLized version of DOLCE. Some of their alignments are useful, however, which we will return to in Section 4.

From a computational viewpoint instead of an Ontology and modelling viewpoint, we consider several aspects of ontology mediation and matching. Ontology mediation (de Bruijn et al., 2006) is divided into three operations: mapping, alignment, and merging. To be precise in the terminology we use throughout the paper, we provide several definitions on ontology matching in this section, which are taken from (Euzenat and Shvaiko, 2007), Chapter 3. First, there is the actual mediation, or matching, process:

Definition 1 (Matching Process). *The matching process can be seen as a function f which, from a pair of ontologies to match o and o' , an input alignment A , a set of parameters p and a set of oracles and resources r , returns an alignment A' between these ontologies: $A' = f(o, o', A, p, r)$.*

To be able to talk about an actual alignment or mapping, the notion of “entity language” has to be introduced, which is used to express precisely those entities that will be matched.

Definition 2 (Entity language). *Given an ontology language L , an entity language Q_L is a function from any ontology $o \subseteq L$ which defines the matchable entities of ontology o .*

Then, a correspondence consists of a relation between two entities in different ontologies, which is uniquely identified and has some confidence value assigned to it.

Definition 3 (Correspondence). *Given two ontologies o and o' with associated entity languages Q_L and $Q_{L'}$, a set of alignment relations θ and a confidence structure over Ξ , a correspondence is a 5-tuple: $\langle id, e, e', r, n \rangle$, such that id is a unique identifier of the given correspondence, $e \subseteq Q_L(o)$ and $e' \subseteq Q_{L'}(o')$, $r \subseteq \theta$, and $n \subseteq \Xi$.*

Ontology alignment, then, is the process of specifying correspondences between entities, by using a particular alignment relation, such as equivalence, subsumption, or a predefined similarity relation.

Definition 4 (Alignment). *Given two ontologies o and o' , an alignment is made up of a set of correspondences between pairs of entities belonging to $Q_L(o)$ and $Q_{L'}(o')$ respectively.*

Ontology mapping deals with creating correspondences between ontologies based on alignments such that the resultant ontology is still consistent and does not have unsatisfiable classes or relations. Euzenat

and Shvaiko do consider this with respect to models of aligned ontologies, which is too lengthy to repeat here, and De Bruijn et al. does not provide a definition of their idea of mapping as a ‘consistent alignment in the context of the whole ontology’ either. Therefore, we capture the gist in the following definition, using Euzenat and Shvaiko’s notational conventions.

Definition 5 (Mapping). *Given two ontologies o and o' , a mapping is made up of a set of correspondences between pairs of entities belonging to $Q_L(o)$ and $Q_L(o')$, respectively, and this mapping is satisfiable and does not lead to an unsatisfiable entity in either o or o' .*

In merging, a new merged ontology is created from the original ontologies.

Definition 6 (Merging). *Given two ontologies o and o' , a merging is the creation of a new ontology o'' containing o and o' and all mappings between entities belonging to $Q_L(o)$ and $Q_L(o')$ such that o'' does not have unsatisfiable entities and is consistent.*

Overviews of approaches, frameworks, and technologies used to perform ontology mapping, alignment and merging are discussed elsewhere (e.g., (de Bruijn et al., 2006)), and more detail about algorithms and issues can be found in (Euzenat and Shvaiko, 2007).

A number of tasks for the problem at hand are based on ontology mediation. Performing foundational ontology mediation with automated tools is a good approach and a starting point because there are many foundational ontology modules, and foundational ontologies are constantly being updated. It is rather time-consuming to explore each foundational ontology, especially when there are differences in hierarchy and structure. By applying several tools to align foundational ontologies, one can determine which tools are better suited for foundational ontologies, and the type of alignments that are misaligned or not discovered by tools. We summarize the alignment tools that are used in the experimental evaluation, of which we note that LogMap, YAM++, HotMatch, Hertuda and Optima have been evaluated with positive results by the Ontology Alignment Evaluation Initiative (OAEI) in terms of their precision, recall and other performance measures.

H-Match (Castano et al., 2003) is an algorithm for matching ontologies at different depth levels, with different accuracies, based on user choices. The algorithm takes into account linguistic and semantic features of ontologies to perform matching and uses one of four matching models: surface, shallow, deep or intensive. The surface model only considers linguistic affinity between entity names to measure similarity.

In shallow, deep and intensive models, context is also considered to determine entity similarity.

PROMPT (Noy and Musen, 2000) is an ontology matching plug-in for Protégé that allows for comparison, mappings, and merging between ontologies. It is a semi-automatic method that invokes algorithms based on a combination of concept-representation structure, the relations between entities and user’s actions. PROMPT offers the user four different algorithms to use for initial comparison: lexical matching, FOAM plugin, lexical matching with synonyms and using UMLS concept identifiers for matching. It is only supported in older versions of Protégé, which makes it unstable.

LogMap (Jiménez-Ruiz and Cuenca Grau, 2011) automatically generates mappings between ontologies using logic-based semantics of the input ontologies. It offers an improvement to other mapping tools in that it addresses scalability and logical inconsistencies. LogMap allows a user to upload ontologies in a number of formats and implements existing reasoners to check the satisfiability of the ontologies.

YAM++ (Ngo and Bellahsene, 2012) aligns entities by information retrieval or machine learning if training data is available. Three matchers are implemented in YAM++: an element level matcher, a structural matcher and a semantic matcher. The element level and structural mapper discover alignments while the semantic matcher revises these alignments to remove inconsistencies and ensure logical mappings.

HotMatch (Dang et al., 2012) is a tool based on a combination of many matching algorithms. The two types of algorithms are element level and structural matching. However, there is more than one of each implemented. There are also filters in HotMatch, used to remove duplicate mappings found by the matchers. Upon input of a source and target ontology, HotMatch deploys its matchers and filters sequentially resulting in mappings between the two.

Hertuda (Hertling, 2012) is an entity matcher that applies element level matching with a string comparison. The alignments generated by Hertuda are only satisfiable in OWL Lite/DL. As a result, object properties in the ontologies are handled separately. This may cause some difficulties in aligning object properties in the foundational ontologies because their domains and ranges affect the alignments.

Optima (Kolli and Doshi, 2008) is a fully automatic tool which iteratively improves alignments. It is aimed at aligning large ontologies but may also be used for smaller ontologies. Its similarity measure is based on both syntactic and semantic similarity.

3 FOUNDATIONAL ONTOLOGY CONTENT COMPARISON

In this section, we provide an informal content comparison between the foundational ontology pairs by identifying differences and similarities between them. A content comparison is beneficial in that it forms the basis for performing ontology mediation operations. A content comparison does not include abstract comparisons such as those based on philosophical choices, ontological alignments and software engineering properties, which has been addressed elsewhere (Khan and Keet, 2012), but rather a high-level comparison of the structure, organisation, and entities of the foundational ontologies.

DOLCE, BFO and GFO contain both 3D and 4D entities. Both BFO and GFO name these entities *Continuant* and *Occurrent* while DOLCE names them *endurant* and *perdurant*. Some syntactic variants exist between DOLCE, BFO, and GFO, e.g., DOLCE's *space-region* vs. BFO's *SpatialRegion* vs. GFO's *Spatial_region*. In DOLCE, BFO, and GFO, classes that share the same name and idea are *process*, *function* and *role*.

DOLCE entities are of type *particular*, BFO's entities are *Universals* while GFO contains a combination of the two, both *Individual* and *Universal* entities. DOLCE and BFO have similar structures at a high-level only in that both have separate branches of 3D and 4D entities. GFO's high-level structure is different as it offers a distinction between *Category* and *Individual* entities. DOLCE's *endurant* and *perdurant* branches are linked by *participation* relations; BFO's and GFO's 3D and 4D entity branches are completely independent of each other.

DOLCE and GFO have advanced support for representing entity properties (e.g., *colour*) and their values (e.g., *blue*) while BFO has limited support for this. To describe these entities, DOLCE uses the terms *quality*, *quale* and *quality-space*, BFO uses the term *Quality* while GFO names these entities *Property*, *Property_value* and *Value_space*. DOLCE's *quality* branch is disjoint to its *endurant* and *perdurant* branches. In BFO, on the other hand, *Quality* is subsumed by *Continuant* branch, while GFO has *Property* subsumed by the higher-level *Individual*. BFO's temporal entities, including temporal regions, intervals, and instants, are subsumed by *Occurrent*, while DOLCE's temporal entities are split up into three parts, being temporal regions that are subsumed by abstract entities, temporal qualities that are subsumed by *quality* entities, and subclasses of *perdurants*. All of GFO's temporal entities are subsumed by its *Space-time* entity, which is disjoint to its *Concrete* and *Abstract* entities. Most of DOLCE's spatial

entities are subsumed by its abstract entity with the exception of *spatial-location-q* which is subsumed by *quality*, while GFO's spatial entities are subsumed by its *space-time* entity. BFO's *SpatialRegion* entity is subsumed by *Continuant*. DOLCE and GFO have abstract entities while BFO does not.

DOLCE and GFO contain relational properties. BFO does not have relational properties included in the ontology, but rather as a separate ontology, the *Relational Ontology (RO)* (Smith et al., 2005). BFO 2.0 is currently being developed, where BFO is integrated with RO. DOLCE's relational properties are all based on either of its six primitive relations: *parthood*, *temporary parthood*, *constitution*, *participation*, *quality*, and *quale*. For mereology, DOLCE adopts the axioms of *General Extensional Mereology (GEM)*. BFO core is a comprehensive mereology represented in first-order logic and contains collections, sums and universal axioms. GFO's mereology contains the following axioms: *antisymmetry*, *transitivity*, *set inclusion*, *proper parthood*, and other GFO-specific axioms based on these.

Thus, the organisation of entities within the three ontologies differ. In some cases, entities that seem similar fall in contradicting or disjoint classes. These differences in structure and organisation may cause inconsistencies when performing mapping, as we shall see later in detail.

4 ALIGNMENT

For foundational ontology alignment, i.e., aligning on an entity-by-entity basis, certain aspects of the underlying philosophies of each foundational ontology have been ignored, because else it would result in few or no alignments and for practical usage of their OWL files, they are less pressing issues (e.g., DOLCE is descriptive and contains particulars, while BFO is realist and contains universals). We align classes and object properties with equivalence relations first, and use subsumption relations afterward to resolve some mapping inconsistencies.

We create alignments for 20 pairs of ontologies. These ontologies include DOLCE-Lite, BFO, GFO, *FunctionalParticipation*, *SpatialRelations*, and *TemporalRelations* (which are more-detailed modules of DOLCE), BFORO and GFO-Basic. BFORO refers to the merged ontology of BFO with the RO, and GFO-Basic is a less-detailed module of GFO. We perform ontology alignment by using existing tools, documentation and manually using the content comparison, entity axioms and annotations. Further, for each resource (tool, documentation or manual alignment), we measure its *accuracy* by firstly examining each of

its output alignments to determine whether or not the equivalence relation is correct. Thereafter we define accuracy as the number of ‘correct’ alignments over the total alignments given by the resource (Eq. 1), where ‘correct’ denotes the alignment is also in the set of alignments found manually, i.e., what is typically considered as the ‘gold standard’. We define the *found* measure of the resources as the number of correct alignments over the total possible correct alignments, after manual intervention (Eq. 2).

$$Accuracy = \frac{|correct\ alignments|}{|total\ alignments_{resource}|} \times 100 \quad (1)$$

$$Found = \frac{|correct\ alignments|}{|total\ alignments_{gold}|} \times 100 \quad (2)$$

4.1 Alignment results

We describe the results of the manual alignments first, and then the results obtained with the matching tools.

4.1.1 Manual alignments

The yield of the manual alignments between the main foundational ontologies (DOLCE-Lite, BFO and GFO) resulted in 17 alignments for DOLCE-Lite \leftrightarrow BFO, 23 alignments for BFO \leftrightarrow GFO and 35 alignments for GFO \leftrightarrow DOLCE-Lite, hence, 75 in total; the complete set of alignments can be accessed at <http://www.thezfiles.co.za/ROMULUS/>. Of these 75 alignments, 37 are for alignments between object properties and are displayed in Tables 1, 2, and 3. The mapped class alignments are available and discussed elsewhere (Khan and Keet, 2013) and online accessible, and the difference between class alignments and mappings for GFO \leftrightarrow DOLCE-Lite are displayed in Table 4. When we consider entity alignments including the related modules of the foundational ontologies (e.g., GFO-Basic), there is a total of 85 alignments. Naturally, there are many more than 85 alignments if we consider identical alignments that occur among the same entities in related modules; e.g., DOLCE-Lite:particular \leftrightarrow GFO:Individual and FunctionalParticipation:particular \leftrightarrow GFO:Individual. There are 14 alignments common between these three ontologies, which is displayed in Table 5.

The manual alignments were aided by the GFO documentation (Herre, 2010) and checked against the alignments proposed by (Temal et al., 2010; Seyed, 2009). The GFO documentation (Herre, 2010) contains a list of similarities between GFO and DOLCE which helped with the alignment process. Some of the alignments could not be used, however, due to changes in the two foundational ontologies. We were

Table 1: Equivalence alignments for relational properties between DOLCE-Lite and BFO; the alignments numbered in bold font can also be mapped.

	DOLCE-Lite	BFORO
1.	generic-location	located_in
2.	generic-location-of	location_of
3.	part	has_part
4.	part-of	part_of
5.	proper-part	has_proper_part
6.	proper-part-of	proper_part_of
7.	participant	has_participant
8.	participant-in	participates_in

Table 2: Equivalence alignments for relational properties between DOLCE-Lite and GFO; the alignments numbered in bold font can also be mapped.

	DOLCE-Lite	GFO
1.	generic-constituent	has_constituent_part
2.	generic-constituent-of	constituent_part_of
3.	generically-dependant-on	depends_on
4.	generic-dependant	necessary_for
5.	has-quale	has_value
6.	quale-of	value_of
7.	boundary	has_boundary
8.	boundary-of	boundary_of
9.	q-present-at	exists_at
10.	temporary-participant-in	agent_in
11.	temporary-participant	has_agent
12.	generic-location	occupies
13.	generic-location-of	occupied_by
14.	part	abstract_has_part
15.	part-of	abstract_part_of
16.	proper-part	has_proper_part
17.	proper-part-of	proper_part_of
18.	participant	has_participant
19.	participant-in	participates_in

Table 3: Equivalence alignments for relational properties in BFO and GFO; the alignments in bold are also mapped.

	BFORO	GFO
1.	has_part	has_part
2.	part_of	part_of
3.	has_proper-part	has_proper_part
4.	proper_part_of	proper_part_of
5.	has_participant	has_participant
6.	participant_in	participates
7.	located_in	occupies
8.	location_of	occupied_by
9.	has_agent	has_agent
10.	agent_in	agent_in

able to use 42% of the alignments from the documentation. We discuss four equivalence alignments from (Temal et al., 2010). We changed the alignment `bfo:ProcessualEntities` \leftrightarrow `dolce:perdurant` to `bfo:Occurrent` \leftrightarrow `dolce:perdurant`, because by definition occurrents and perdurants both represent entities that have temporal parts and unfold in time. Temal et al.’s alignment of `bfo:Quality` with `dolce:physical-quality` is more precise than ours, because, as mentioned above, we chose to ignore the some philosophies (the realist debate) with the hope of achieving a higher number of alignments. That is, our mapping has `bfo:Quality` \leftrightarrow `dolce:quality`, thereby ignoring the fact that BFO does not consider abstract entities. We agree with `bfo:SpatialRegion` \leftrightarrow `dolce:space-region` and `bfo:TemporalRegion` \leftrightarrow `dolce:temporal-region`, and use this equivalence, too. Seyed (Seyed, 2009) examined only three relations—dependency, quality, and constitution—and found that they are different in DOLCE and BFO. The basic numbers of the alignments are included in Table 6.

Table 4: Equivalence alignments for classes between DOLCE-Lite and GFO ontologies; the alignments numbered in bold font can also be mapped.

	DOLCE-Lite	GFO
1.	particular	Individual
2.	endurant	Presential
3.	physical-endurant	Discrete_presential
4.	physical-object	Material_object
5.	amount-of-matter	Amount_of_substrate
6.	perdurant	Occurrent
7.	process	Process
8.	state	State
9.	abstract	Abstract
10.	set	Set
11.	quality	Property
12.	quale	Property_value
13.	quality-space	Value_space
14.	time-interval	Chronoid
15.	space-region	Spatial_Region
16.	temporal-region	Temporal_Region

4.1.2 Automated alignments

Table 7 lists the numbers of alignments found by the selected tools. We describe some further data in the remainder of this section.

H-Match generated many alignments, but most of the output was not accurate. Many entity pairs that were matched using H-Match were found to be incorrectly aligned; e.g., `DOLCE-Lite:quale` \leftrightarrow `bfo:Role`. This resulted in us being able to use only 18% of these alignments, with the rest being false positives.

Table 5: Common alignments between DOLCE-Lite, BFO and GFO.

	DOLCE-Lite	BFORO	GFO
Class			
1.	endurant	Independent Continuant	Presential
2.	physical-object	Object	Material_object
3.	perdurant	Occurrent	Occurrent
4.	process	Process	Process
5.	quality	Quality	Property
6.	space-region	SpatialRegion	Spatial_region
7.	temporal-region	Temporal-Region	Temporal_region
Relational property			
1.	proper-part	has_proper_part	has_proper_part
2.	proper-part-of	proper_part_of	proper_part_of
3.	participant	has_participant	has_participant
4.	participant-in	participates_in	participates_in
5.	generic-location	located_in	occupies
6.	generic-location-of	location_of	occupied_by

Table 6: Comparison of manually performed alignment accuracies of the GFO documentation (Herre, 2010), related works, and ours, and aggregates for mappings.

	Seyed	Herre	Temal et al.	Ours
<i>Class alignments</i>				
DOLCE-Lite \leftrightarrow BFO	-	-	2/7	9/9
BFO \leftrightarrow GFO	-	-	-	13/13
GFO \leftrightarrow DOLCE-Lite	-	13/31	-	16/16
<i>Object property alignments</i>				
DOLCE-Lite \leftrightarrow BFO	0	-	-	8/8
BFO \leftrightarrow GFO	-	-	-	10/10
GFO \leftrightarrow DOLCE-Lite	-	0	-	19/19
Overall alignments				
Total	0/0	13/31	2/7	75/75
Accuracy	0%	42%	29%	100%
Found	0%	37%	12%	100%
Overall mappings				
Total	0/0	8/31	1/7	40/40
Accuracy	0%	26%	14%	100%
Found	0%	61%	9%	100%

PROMPT was generally unstable resulting in force closure of the application. It generated suggestions of which we were able to use 56%, with the rest being false positives; e.g., `bfo:Site` \leftrightarrow `gfo:Situoid`.

While LogMap provided few alignments between the foundational ontologies (less than ten in all cases), most alignments were accurate. The one false positive in LogMap was the alignment of `bfo:IndependentContinuant` \leftrightarrow `gfo:Independent`.

Table 7: Comparison of alignment accuracies of the matching tools and aggregates for mappings.

	H-Match	PROMPT	LogMap	YAM++	Hot Match	Hertuda	Optima
<i>Class alignments</i>							
DOLCE-Lite↔BFO	4/16	3/8	2/2	4/4	3/3	3/3	4/12
BFO↔GFO	5/31	7/12	7/8	6/7	7/7	7/7	8/14
GFO↔DOLCE-Lite	4/25	4/8	3/3	8/11	5/5	5/5	5/16
<i>Object property alignments</i>							
DOLCE-Lite↔BFO	0	0	0	0	0	0	0/1
BFO↔GFO	0	0	4/4	0	0	0	1/3
GFO↔DOLCE-Lite	0	4/4	0	5/14	5/7	6/8	2/23
Overall alignments							
Total	13/72	18/32	16/17	23/36	20/22	21/23	20/69
Accuracy	18%	56%	94%	64%	91%	91%	29%
Found	17%	24%	21%	31%	27%	28%	27%
Overall mappings							
Total	10/72	11/32	16/17	15/36	11/22	12/23	13/69
Accuracy	14%	34%	94%	42%	50%	52%	19%
Found	25%	28%	40%	38%	28%	30%	33%

YAM++ generated many alignments. However, while most of the alignments for DOLCE ↔ BFO and BFO ↔ GFO were accurate, only about half were accurate for GFO ↔ DOLCE. Overall we were able to use almost 64% of its alignments. Like LogMap, YAM++ also incorrectly aligned `bfo:IndependentContinuant` ↔ `gfo:Independent`. Some of YAM++’s other false positive alignments include `dolce:generic-constituent` ↔ `gfo:has_sequence_constituent`, `dolce:quality-space` ↔ `gfo:Space` and `dolce:temporary-proper-part` ↔ `gfo:has_constituent_part`.

HotMatch generated a fair amount of alignments between the ontologies. Overall, we were able to use 91% of HotMatch’s alignments, with just 2 alignments out of all 22 being false positives. Hertuda’s output was surprisingly similar to HotMatch’s output, with just one more alignment than HotMatch. We were able to use 91% of Hertuda’s alignments, with just 2 alignments out of all 23 being false positives. Common false positives in YAM++, Hertuda and HotMatch were the alignments between `dolce:part` ↔ `gfo:has_part` and `dolce:part-of` ↔ `gfo:part_of`, which is discussed in Section 4.2.1. Optima generated many alignments for each pair. However, there were many false positives, consequently we were able to use 29% of its alignments overall. This may be because Optima is aimed at aligning large ontologies and the foundational ontologies in question are of reasonable size. Optima incorrectly aligned `gfo:Continuous` ↔ `bfo:Continuant`, `dolce:Region` ↔ `bfo:SpatialRegion` and `dolce:dependent-place` ↔ `bfo:Dependent`.

4.2 Alignment Issues

We have encountered two types of issues in alignment: transitivity, where there was no ‘full circle’ alignment between some entities of the three ontolo-

gies, and approximate alignments, where there is no clear relationship to describe the match.

4.2.1 Transitivity

Transitivity in entity alignments works as follows: if the equivalence relation holds between entities from the first and second ontology and it holds between entities from the second and third ontology; it necessarily holds between entities from the first and third ontology. Applying transitivity to entity alignments assists in detecting errors. For instance, if one were to align `dolce:endurant` ↔ `gfo:Persistent`, and `gfo:Persistent` ↔ `bfo:Continuant`, then by transitivity this means that `dolce:endurant` is equivalent to `bfo:Continuant`, which is incorrect. In most cases, the foundational ontology alignments are transitive. There were two types of exceptions, being the absence of an entity and what can be termed consequences of conflicting philosophies.

Absence of an entity. An alignment cannot be a candidate for transitivity if there is an equivalence between only two out of the three ontologies. From the three main ontology alignments, the following ones were not transitive due to the absence of an entity:

- Absence of a DOLCE entity (7 cases): `bfo:Entity` ↔ `gfo:Entity`, `bfo:DependentContinuant` ↔ `gfo:Dependent`, `bfo:ObjectBoundary` ↔ `gfo:Material_boundary`, `bfo:Function` ↔ `gfo:Function`, `bfo:Role` ↔ `gfo:Role`, `bfo:has_agent` ↔ `gfo:has_agent`, `bfo:agent_in` ↔ `gfo:agent_in`.
- Absence of a GFO entity (1 case): `dolce:spatio-temporal-region` ↔ `bfo:SpatioTemporalRegion`
- Absence of a BFO entity (17 cases): `gfo:Individual` ↔ `dolce:particular`, `gfo:Amount_of_substrate` ↔ `dolce:amount-of-matter`, `gfo:State` ↔

dolce:state, gfo:Abstract ↔ dolce:abstract,
 gfo:Set ↔ dolce:set, gfo:Property_value ↔
 dolce:quale, gfo:Value_space ↔ dolce:quality-
 space, gfo:Chronoid ↔ dolce:time_interval,
 gfo:has_constituant_part ↔ dolce:generic-
 constituent, gfo:constituant_part_of ↔
 dolce:generic-constituant-of, gfo:necessary_for
 ↔ dolce:generic-dependent, gfo:depends_on ↔
 dolce:generically-dependent-on, gfo:has_value
 ↔ dolce:has-quale, gfo:value_of ↔ dolce:quale-
 of, gfo:has_boundary ↔ dolce:boundary,
 gfo:boundary_of ↔ dolce:boundary-of,
 gfo:exists_at ↔ dolce:q-present-at

From this type of transitivity issue, we see that for the three main ontology alignments, in most cases BFO entities are absent. There are a few cases of absent DOLCE entities and one case of an absent GFO entity.

Conflicting philosophies. The philosophies of foundational ontologies affect their entities to a certain extent. In some cases, two entities that are aligned to each other may not be aligned to the same entity of a third ontology.

- *dolce:physical-endurant* ↔ *bfo:MaterialEntity*, *dolce:physical-endurant* ↔ *gfo:Discrete-presential* and *bfo:MaterialEntity* ↔ *gfo:Material-persistent*. Let us align *bfo:MaterialEntity* ↔ *dolce:physical-endurant*, ignore their underlying philosophies (i.e., that BFO is an ontology of universals and DOLCE of particulars). However, in GFO, there are two entities for representing this type of entity, based on distinct philosophical notions: *gfo:Discrete-presential*, being subsumed *gfo:Individual*, is suited for *dolce:physical-endurant* while *gfo:Material-persistent*, being subsumed by *gfo:Universal*, is suited for *bfo:MaterialEntity*.
- *dolce:part* ↔ *bfo:has-part*, *dolce:part* ↔ *gfo:has-abstract-part* and *bfo:has-part* ↔ *gfo:has-part* (idem for their inverses). In DOLCE, both the domain and range of *part* is particular. In BFORO, there is no domain and range for *has-part*. In GFO, both the domain and range of *abstract.has-part* is *Item*, while both the domain and range for *has-part* is *Concrete*. The former relational property may be better suited for DOLCE because it is a descriptive ontology and contains abstract entities. The latter is better suited for BFORO as it is a realist ontology, representing the world as is, thereby containing concrete entities only.

The ontology matching tools discussed in Section 4.1.2 misaligned *dolce:part* ↔ *gfo:has-part* and their inverses. This is because object property inconsistencies are not fully recognised by reasoners (Keet,

2012), hence their conflicting domains and ranges did not affect the satisfiability of the ontology.

4.2.2 Approximate Alignments

There are a number of approximate alignments between foundational ontology entities. By this we mean that they are not equivalent to each other or subsumed by one another, but share some common characteristics. By identifying these relations between these entities, foundational ontology developers could possibly relate them as sibling classes by grouping them both under a common superclass. We mention three of them.

- *dolce:arbitrary-sum*, *bfo:ObjectAggregate* and *gfo:Configuration*: All three of these entities describe a collection of something. *dolce:arbitrary-sum*, however, has no unity criterion e.g., a pencil and laundry basket are together a *dolce:arbitrary-sum*, and it can contain both *dolce:physical-endurant* and *dolce:non-physical-endurant* entities. *dolce:physical-endurant* is not restricted just to instances of *dolce:physical-object* but can possibly include *dolce:feature* and *dolce:amount-of-matter*. *bfo:ObjectAggregate*, on the other hand, has overall unity and can be considered as a whole. It is restricted to *bfo:Object* only, and in the case of BFO, all objects are physical. *gfo:Configuration* is simply a collection of *gfo:Presential* facts. *gfo:Presentials* are not restricted to whole physical objects and can include other *gfo:Presential* entities. For this reason, it cannot equate to *bfo:ObjectAggregate*. Furthermore, it holds a restriction that it must contain at least one material entity. *dolce:arbitrary-sum* could contain physical, non-physical or both entities, with no restrictions.
- *dolce:state* and *bfo:SpatioTemporalInstant*: DOLCE describes *dolce:state* by using an example of a rock erosion describing state as a time interval of the erosion is collapsed into a time point. Similarly BFO defines *bfo:SpatioTemporalInstant* as a “connected spatiotemporal region at a specific moment”. The difference between the two lies in the fact that *dolce:state* is homeomeric while *bfo:SpatioTemporalInstant* is not.
- *dolce:relevant-part* and *bfo:FiatObjectPart*: DOLCE describes *dolce:relevant-part* as a feature that is a relevant part of their host; e.g., the edge of a cube. BFO defines *bfo:FiatObjectPart* as a material entity that is part of an object but not demarcated by physical discontinuities; e.g., the lower portion of the leg. In this sense they are both part objects that are physical entities. However, it is unclear whether *dolce:relevant-part*

is demarcated by physical discontinuities or not and whether BFO's fiat object parts are 'relevant' somehow. This requires further investigation.

5 MAPPING AND MERGING

Ontology Mapping uses the alignments from the alignment process to create correspondences between entities in the ontologies. The output from the alignment process is broader, while the output from the mapping process is narrower as inconsistencies affect the mapping process. Merging is performed by creating a new ontology of the source ontologies with their mappings between each other. Ontology mapping and merging was performed by relating classes and object properties in Protégé. Entities were mapped in the order of their level in the hierarchy, from higher to lower level, because foundational ontologies by definition are general high-level ontologies. Therefore, in mapping, preference must be first given to high-level entities to have agreement among general entities and avoid inconsistencies at that level.

Alignments that cannot be mapped due to logical inconsistencies result in unsuccessful mappings. The inconsistencies were identified by using the following method. For each candidate class mapping:

1. Assert the equivalence for the found alignment.
2. Run the automated reasoner.
3. Check if there are any unsatisfiable classes.
4. If there are unsatisfiable classes, use the reasoner explanation feature to generate an explanation.
5. Analyse explanations.
6. Remove inconsistent mapping, if applicable.

For each candidate object property mapping, since object property inconsistencies and flaws are not properly recognised by reasoners (Keet, 2012), we identified inconsistencies by checking if an object property pair's domain and range restrictions are satisfiable by using the above method.

The numbers in bold face in Table 5 represent the alignments that resulted in successful mappings between the common entities of the three main ontologies. From the 14 alignments in Table 5, six successful mappings exist. Recall from the previous section on alignment, there was a total of 85 distinct alignments between all foundational ontologies and related modules, and 75 alignments between the main foundational ontologies. Performing the method to identify inconsistencies in alignments resulted in 42 distinct logical inconsistencies of which 35 alone were from the main ontologies. From all the distinct equivalence alignments, only half were satisfiable and resulted in successful mappings. Comparing these mappings to the alignments found by the tools, LogMap

doubled its percentage found to 40% and performed best compared to the six others evaluated (see Table 7, bottom three rows).

To solve inconsistencies in the mapping attempts, we analysed each alignment on the logical explanation for the inconsistency and the description of the entity provided by the foundational ontology developers, and checked whether it was possible to change the alignment from equivalence to subsumption. However, there are still many unsolvable inconsistencies, mainly due to hierarchical and structural differences in the foundational ontologies. Due to space limitations, we describe only a representative selection of the logical inconsistencies and (logically satisfiable) possible solutions; the full list of inconsistencies is available at <http://www.thezfiles.co.za/ROMULUS/>.

Inconsistencies due to disjoint classes. For this type of inconsistency, the entities to be aligned are disjoint to each other, either directly, through higher-level equivalence relations or through their subclasses or superclasses. If entities are disjoint, they cannot overlap, hence cannot be equivalent.

- *dolce:temporal-region - gfo:TemporalRegion - bfo:TemporalRegion:* The issue with incompatible temporal regions between BFO, GFO, and DOLCE is depicted in Fig. 1 and is a result of the OWL DisjointClasses class axiom between *gfo:Concrete*, *gfo:Space_Time* and *gfo:Abstract*, and between *dolce:Abstract* and *dolce:Perdurant*, or, from the other viewpoint: because BFO made *TemporalRegion* an Occurrent, DOLCE made it Abstract, and GFO neither. This does not seem to be resolvable.
- *bfo:Role - gfo:Role:* *gfo:Processual_role* is a subclass of *gfo:Role* and *gfo:Process*. *gfo:Process* is a subclass of *gfo:Occurrent*. *gfo:Occurrent* is equivalent to *bfo:Occurrent*. *bfo:Role* is a subclass of *bfo:Continuant*. *bfo:Continuant* is disjoint to *bfo:Occurrent*. In this equivalence relation, both *gfo:Role* and *gfo:Occurrent* are superclasses of *gfo:Processual_role*, and *bfo:Role* is a subclass of *bfo:Continuant*; *gfo:Occurrent* and *bfo:Continuant* are disjoint, hence the two classes cannot be equivalent. Solution: Logically, *bfo:Role* cannot be equivalent to *gfo:Role*. However, *bfo:Role* can be subsumed by *gfo:Role*. Therefore the relation can be changed to *gfo:Role* subsumes *bfo:Role*.
- *gfo:necessary_for - dolce:generic-dependent:* If we were to equate these object properties, we would have to assume that their domains and ranges are equivalent, which is not the case; the situation is depicted in Fig. 2. Solution:

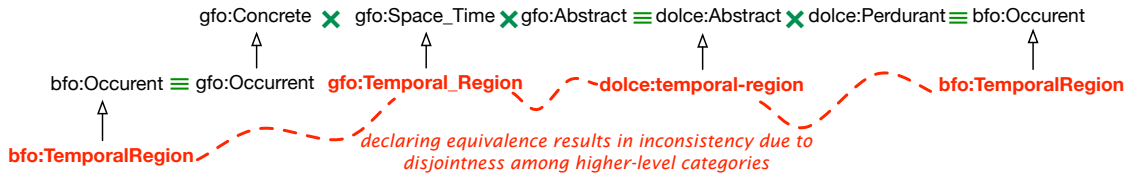


Figure 1: Graphical depiction of why the aligned `gfo:Temporal_Region`, `bfo:TemporalRegion`, and `dolce:temporal-region` cannot be mapped in any way without causing an inconsistency; \equiv : aligned entities, \times : disjoint entities.

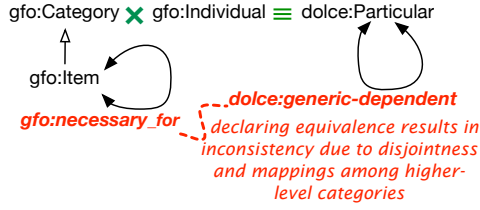


Figure 2: Visualisation of the root cause of the non-mappable `gfo:necessary_for` and `DOLCE-Lite:generic-dependent`; \times : disjointness, \equiv : equivalence mapping.

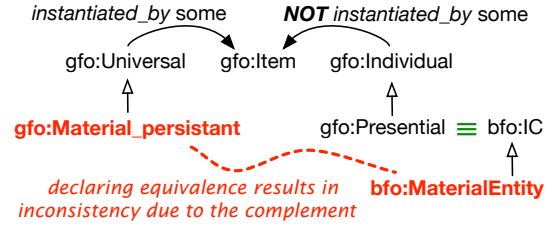


Figure 3: Visualisation of the root cause of the non-mappable `bfo:MaterialEntity` and `gfo:Material_persistent`; \equiv : equivalence mapping, IC = IndependentContinuant.

Logically, `gfo:necessary_for` cannot be equivalent to `dolce:generic-dependent`, because equating their domains and ranges causes inconsistencies. However, `dolce:generic-dependent`'s domain and range, `dolce:particular` can logically be subsumed by `gfo:necessary_for`'s domain and range, `gfo:item`. Therefore the relation can be changed to `gfo:necessary_for` subsumes `bfo:generic-dependent`.

- `dolce:generic-location` - `bfo:located_in`: This issue is due to disjointness among domain/range. `dolce:generic-location`'s range is `dolce:particular` and `bfo:located_in`'s range is `bfo:Continuant`. `bfo:Continuant` is disjoint to `bfo:Occurrent` and `bfo:Occurrent` \equiv `dolce:perdurant`. In DOLCE, `perdurant` \sqsubseteq `\exists has-Quality.temporal-location-q` and the domain of `dolce:has-Quality` is `dolce:particular` (the superclass of `dolce:perdurant`). Thus, `bfo:Continuant` is disjoint to `\exists has-Quality.temporal-location-q`, and cannot be equivalent, causing the range restrictions to be unsatisfiable in the alignment. Therefore `dolce:generic-location` cannot map to `bfo:located_in`.

Other unresolvable cases are, among others, `dolce:set` - `gfo:Set` and `dolce:quality-space` - `gfo:Value space`.

Inconsistencies due to complement classes. For this type of inconsistency, the entities to be aligned were found to be complements of each other, either directly, through higher-level equivalences or through subsumption. We describe here one such case.

- `bfo:MaterialEntity` - `gfo:Material_persistent`, which is visualised in Fig. 3. The crucial aspect in GFO is the class axiom `Universal` \sqsubseteq `\exists instantiated_by.Item`, and the complement for individuals. Concerning mappings, `bfo:IndependentContinuant` \equiv `gfo:Presential`. However, GFO has `Presential` \sqsubseteq `Individual` and `Individual` \sqsubseteq `\neg \exists instantiated_by.Item`. Thus, `gfo:Material_persistent` is a subclass of `gfo:instantiated_by some gfo:item` while `bfo:MaterialEntity` is a subclass of the *complement* of that class, hence `bfo:MaterialEntity` cannot be equivalent to `gfo:Material_persistent`. **Solution:** The alignment can be changed into `bfo:MaterialEntity` - `gfo:Discrete_presential`, which avoids the complement issue but it is not free of argument (recall the “conflicting philosophies” item in Section 4.2.1).

6 DISCUSSION

Given the size of the ontologies and our high tolerance by ignoring underlying philosophies, the amount of alignments, and, even more so, the amount of mappings is less than one may have expected; or: once investigated in detail, the foundational ontologies are, at present, not particularly interchangeable even at the logical level. Only six pairwise mappings exist, i.e., they being, essentially, equivalent throughout all three examined foundational ontologies.

Concerning feasibility to carry out automated alignments, in most cases, the tools evaluated with

the OAEI performed better than the others, with the exception of Optima. LogMap had the highest accuracy, because it also considers the logic-based semantics of the ontologies and uses automated reasoning services throughout the process, therewith eliminating those false positives that would have led to a logical inconsistency. However, LogMap generated very few alignments compared to other accurate tools (YAM++, Hertuda and HotMatch), indicating that the additional heuristics implemented are too strict at least for foundational ontology alignment.

Most false positive alignments generated by the tools, such as `bfo:IndependentContinuant` ↔ `gfo:Independent`, indicate that the algorithms implement syntactic matching, which, based on the results we obtained, is not sufficient or suitable for foundational ontology matching because many entities have a common syntax e.g., `dolce:quality-space` ↔ `gfo:Space` both have the string ‘space’ in common but are entirely different entities; Table 8 includes a selection of such false positives that are caused by syntactic matching in the tools when aligning the three foundational ontologies. The tools failed to recognise simple alignments such as `dolce:perdurant` ↔ `gfo:Occurrent`, `bfo:Quality` ↔ `gfo:Property`. In this sense, semantic matching is not considered, or if it is, it fails to recognise synonyms of the philosophical scope on which foundational ontologies are built upon. Structural matching is not an effective method either, due to the fact that the hierarchies and structures of the foundational ontologies differ greatly which causes the root distances of mappable entities to differ. For aligning foundational ontologies, it will be useful if existing semantic matchers would include something alike a ‘philosophy WordNet’ that specialises in philosophical terms, synonyms, and definitions used in foundational ontologies.

The results of the tool analysis is a good indication of which tools to experiment with for foundational ontology alignment in general. However, they found less than a third of the actual alignments at this stage, and therefore it is still vital to perform manual alignment for foundational ontologies. The tools also did not generate subsumption relations for any of the alignments, but this could perhaps be an extension to the basic idea of LogMap by means of another call to the reasoner. One could investigate whether Optima is useful to identify accurate alignments among the larger foundational ontologies SUMO (Niles and Pease, 2001) and YAMATO (Mizoguchi, 2010).

On a positive note, the systematised list of issues now can be taken up by ontologists. While some of the inconsistencies found are quite elaborate, others should be easier to resolve both ontologically (philosophically) and where in the ontology the entity is

Table 8: False positives caused by syntactic matching generated by the alignment tools; the terms in italics represent the strings that are common between aligned entities.

DOLCE-Lite	BFO
<i>physical-region</i>	ConnectedSpatio TemporalRegion
<i>non-physical-object</i>	<i>Object</i>
<i>region</i>	SpatioTemporalRegion
BFO	GFO
<i>IndependentContinuant</i>	<i>Independent</i>
<i>Site</i>	<i>Situoid</i>
<i>Continuant</i>	<i>Continuous</i>
GFO	DOLCE-Lite
<i>has_sequence_</i> <i>constituent</i>	<i>generic-constituent</i>
<i>has-part</i>	<i>part</i>
<i>Space</i>	<i>quality-space</i>

positioned; e.g., the notion of a mathematical Set is fairly well investigated already, and likewise the different theories of parthood. As such, the results presented here provide a solid foundation for ample ontological investigations. From an engineering viewpoint and in case of urgent need for interoperability, one could take a quite different strategy: OWL 2 EL does not have negation, and therefore it should be possible to assert more mappings between the OWL 2 EL modules of the foundational ontologies. Whether that is the best strategy is a different matter, and it does not take away the substantial list for which there was no transitivity due to ‘missing’ entities. In any case, we now know that some mappings are possible, hence, also some foundational ontology interoperability.

7 CONCLUSION

The foundational ontologies DOLCE, BFO, and GFO were pairwise aligned and mapped. They were aligned manually, which served as the ‘gold standard’, and with the aid of seven alignment tools. The accuracy and percentage of alignment found were compared, where LogMap had the highest accuracy with 94% and HotMatch and Hertuda as close second, and YAM++ found the most correct alignments (31% of the total manual alignments among the three main ontologies (75)). The evaluation of the tools indicated that the algorithms currently implemented by the tools are not well-suited for foundational ontology mediation. Semantic matching in the tools need to be improved to include philosophical synonyms which are used in foundational ontologies. Declaring the correspondences in all ontology files based on its 85

alignments resulted in only 43 mappings, with the remaining 42 causing logical inconsistencies. The inconsistencies are due primarily to differences in their respective hierarchical structure with conflicting axioms, such as complement and disjointness, and incompatible domain and range restriction. On closer inspection, some inconsistencies may be resolved using subsumption or making them sibling classes.

Future research includes mapping other foundational ontologies, adding subsumption mappings, and evaluating the current alignments with the foundational ontology developers. We also aim to implement a facility for community input on the alignments and mappings, which could be facilitated via the foundational ontology library that is available online at <http://www.thezfiles.co.za/ROMULUS/>.

REFERENCES

- Castano, S., Ferrara, A., and Montanelli, S. (2003). H-MATCH: an algorithm for dynamically matching ontologies in peer-based systems. In *The first International Workshop on Semantic Web and Databases (SWDB'03)*, pages 231–250, Humboldt-Universitt, Berlin, Germany.
- Dang, T. T., Gabriel, A., Hertling, S., Roskosch, P., Wlotzka, M., Zilke, J. R., Janssen, F., and Paulheim, H. (2012). HotMatch results for OEAI 2012. In *Seventh International Workshop on Ontology Matching (OM'12)*, volume 946 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- de Bruijn, J., Ehrig, M., Feier, C., Martins-Recuerda, F., Scharffe, F., and Weiten, M. (2006). Ontology mediation, merging, and aligning. In *Semantic Web Technologies*, pages 1–20. Wiley Online Library.
- Euzenat, J. and Shvaiko, P. (2007). *Ontology Matching*. Springer.
- Herre, H. (2010). General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In *Theory and Applications of Ontology: Computer Applications*, chapter 14, pages 297–345. Springer.
- Hertling, S. (2012). Hertuda results for OEAI 2012. In *Seventh International Workshop on Ontology Matching (OM'12)*, volume 946 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Jiménez-Ruiz, E. and Cuenca Grau, B. (2011). LogMap: Logic-based and scalable ontology matching. In *10th International Semantic Web Conference (ISWC'11)*, volume 7031 of *Lecture Notes in Computer Science*, pages 273–288. Springer.
- Keet, C. M. (2012). Detecting and revising flaws in OWL object property expressions. In ten Teije et al., A., editor, *18th International Conference on Knowledge Engineering and Knowledge Management (EKAW'12)*, volume 7603 of *Lecture Notes in Artificial Intelligence*, pages 252–266. Springer. Oct 8-12, Galway, Ireland.
- Khan, Z. and Keet, C. M. (2012). ONSET: Automated foundational ontology selection and explanation. In ten Teije et al., A., editor, *18th International Conference on Knowledge Engineering and Knowledge Management (EKAW'12)*, volume 7603 of *Lecture Notes in Artificial Intelligence*, pages 237–251. Springer. Oct 8-12, Galway, Ireland.
- Khan, Z. and Keet, C. M. (2013). Toward semantic interoperability with aligned foundational ontologies in ROMULUS. In *Seventh International Conference on Knowledge Capture (K-CAP'13)*. ACM proceedings. 23-26 June, Banff, Canada. (poster/demo, in press).
- Kolli, R. and Doshi, P. (2008). OPTIMA: tool for ontology alignment with application to semantic reconciliation of sensor metadata for publication in SensorMap. In *IEEE International Conference on Semantic Computing (ICSC'08)*, pages 484–485. IEEE. 4-7 Aug. 2008, Santa Clara, CA, USA.
- Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). Ontology library. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003). <http://wonderweb.semanticweb.org>.
- Mizoguchi, R. (2010). YAMATO: yet another more advanced top-level ontology. In *Proceedings of the Sixth Australasian Ontology Workshop*, Conferences in Research and Practice in Information, pages 1–16. Sydney : ACS.
- Ngo, D. and Bellahsene, Z. (2012). YAM++ : A multi-strategy based approach for ontology matching task. In ten Teije et al., A., editor, *18th International Conference on Knowledge Engineering and Knowledge Management (EKAW'12)*, volume 7603 of *Lecture Notes in Artificial Intelligence*, pages 421–425. Springer. Oct 8-12, Galway, Ireland. Demo.
- Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Second International Conference on Formal Ontology in Information Systems (FOIS'01)*. IOS Press. Ogunquit, Maine, October 17-19, 2001.
- Noy, N. F. and Musen, M. A. (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment. In *Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, pages 450–455. AAAI Press.
- Seyed, A. P. (2009). BFO/DOLCE primitive relation comparison. In *The 12th Annual Bio-Ontologies Meeting Colocated with Intelligent Systems for Molecular Biology (ISMB'09)*. Stockholm, Sweden, 28 June 2009.
- Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A., and Rosse, C. (2005). Relations in biomedical ontologies. *Genome Biology*, 6(5):46.
- Temal, L., Rosier, A., Dameron, O., and Burgun, A. (2010). Mapping BFO and DOLCE. *Studies in Health Technology and Informatics*, 160(Pt 2):1065–9.