

ROMULUS: The Repository of Ontologies for MULTiple USes populated with mediated foundational ontologies

Zubeida Casmod Khan · C. Maria Keet

the date of receipt and acceptance should be inserted later

Abstract A foundational ontology contributes to ontology-driven conceptual data modelling and is used to solve interoperability issues among domain ontologies. Multiple foundational ontologies have been developed in recent years, and most of them are available in several versions. This has re-introduced the interoperability problem, increased the need for a coordinated and structured comparison and elucidation of modelling decisions, and raised the requirement for software infrastructure to address this. We present here a basic step in that direction with the Repository of Ontologies for MULTiple USes, ROMULUS, which is the first online library of machine-processable, modularised, aligned, and logic-based merged foundational ontologies. In addition to the typical features of a model repository, it has a foundational ontology recommender covering features of six foundational ontologies, tailor-made modules for easier reuse, and a catalogue of mappable and non-mappable elements among the BFO, GFO and DOLCE foundational ontologies.

Keywords Foundational Ontology · Top-level Ontology · Upper Ontology · Ontology Repository · Ontology Mediation · Semantic Interoperability · Ontology Metadata · Ontology Module

This work is based upon research supported by the National Research Foundation of South Africa (Project UID: 90041) and the Argentinean Ministry of Science and Technology.

C. Maria Keet
Department of Computer Science, University of Cape Town,
South Africa E-mail: mkeet@cs.uct.ac.za

Zubeida Casmod Khan
Department of Computer Science, University of Cape Town,
South Africa and Council for Scientific and Industrial Research,
Pretoria, South Africa E-mail: zkhan@csir.co.za

1 Introduction

The notion of *ontology-driven information systems* was introduced about 16 years ago [15]. This entails ontology-driven conceptual data modelling [16, 18, 25, 26, 29] that uses principles and solutions from Ontology (philosophy) and ontologies (artifacts in computing) to improve the quality of a conceptual data model and refine its language, which therewith improves the quality of the information system. It concerns both particular aspects of a language or modelling problem in order to devise a solution for conceptual data models—e.g., part-whole relations in conceptual models, aided by a foundational ontology [26]—and, more broadly, it looks at modifying a conceptual data modelling language’s metamodel thanks to a foundational ontology (e.g., [18]). One solution or extension may use the DOLCE foundational ontology [40] for refining UML’s aggregation association [26], another could be informed by the UFO foundational ontology [13, 18, 19, 39], and yet another by GFO [21]. However, it is not clear whether DOLCE, UFO, and GFO are compatible. As a result, modelling improvements may end up to be incompatible if the improvements rest on different philosophical assumptions represented in different foundational ontologies. In addition, besides DOLCE, GFO, and UFO, other foundational ontologies exist, notably SUMO [43], YAMATO [41], and BFO [<http://www.ifomis.org/bfo>]. This potential for incompatibilities for ontologically well-founded conceptual data models is real, and has been recognised in the field of ontologies already, where foundational ontologies serve integration of domain ontologies. This is in particular within the Semantic Web setting, where ontology developers use their preferred foundational ontology that differ in various

aspects and because of that exhibit a semantic interoperability problem for domain ontologies.

A hypothetical solution to such semantic issues has been proposed in 2003 as the “WonderWeb Foundational Ontologies Library” (WFOL), with the aim that one should be able to commit to different but systematically related (modules of) foundational ontologies [40]. Such as library never materialised, however, due to theoretical and implementation limitations. The main theoretical hurdle was to conduct deep content comparisons between the foundational ontologies to create the data for such a library, including the mediation between foundational ontologies. Some theoretical advances have been made in general comparisons [30] and rigorous foundational ontology alignment and mapping [31,32,45,46], i.e., now there is data to fill such a WFOL. The implementation hurdles were primarily due to the absence of a common representation language, and there was scant stable software infrastructure for ontologies. Thanks to technological advances in the meantime, the solvability of the implementation issues is within reach, but it has not yet been realised. The creation of a software-based foundational ontology repository is a necessary first step to both make accessible such theoretical results and to facilitate further investigation, hence, facilitate the management of the issues. This can serve as a one-stop shop for foundational ontologies and therewith foster coordinated, or at least interchangeable, ontology-driven conceptual data modelling with broadly usable results and enable examination of interchangeability of a foundational ontology that is mapped to a domain ontology.

We aim to solve the shortcomings through the creation of the first online library of machine-processable, aligned, merged, and modularised foundational ontologies: the Repository of Ontologies for MULTiple USes ROMULUS. ROMULUS not only has features typical of Open Ontology Repositories [2], such as browsing the ontology, but, moreover, *i*) it incorporates a new web-based version of the extended foundational ontology recommender ONSET [30], *ii*) it contains the included foundational ontologies’ OWLized version and we developed a set of carefully crafted modules thereof to facilitate use and reuse, *iii*) it contains both the logic-based pairwise alignments of DOLCE, BFO, and GFO, as well as a catalogue of individual alignments that are not mappable due to other axioms, *iv*) if one changes one’s mind on the selected foundational ontology, one can automatically ‘swap’ it among DOLCE, BFO, and GFO, and *v*) it uses standardised metadata and refines and extends it considerably with metadata for mediation and modules. ROMULUS is online at <http://www.thezfiles.co.za/ROMULUS/> since December 2012.

In the remainder of the paper, we first describe motivating examples for ontology-driven conceptual data models that avail of foundational ontologies in Section 2.1 and the theory for achieving semantic interoperability in Section 2.2. An overview of the design and features of ROMULUS are presented in Section 3, while modularisation of the foundational ontologies is dealt with in a separate section (Section 4), as it has not been described in previous works about ROMULUS (unlike some of the other features [30–32]). The new, extended, metadata model developed for the ontology library is described afterwards in Section 5. The functionality of ROMULUS is compared with other ontology repositories in Section 6. We discuss ROMULUS and the relevance of the library for ontology-driven conceptual data modelling in Section 7, and conclude in Section 8.

2 Semantic interoperability with foundational ontologies

We first illustrate some issues in ontology-driven conceptual data modelling due to the lack of semantic operability, and subsequently how, in the general case, it has been achieved among the foundational ontologies. This, together with the software infrastructure, can solve the issues described, which we shall return to in Section 7.3.

2.1 Motivating examples for a foundational ontology repository

We provide two motivating examples from the viewpoint of ontology-driven conceptual data modelling, which leads toward the need for a foundational ontology library. The purpose of the examples is not to examine and argue for ‘the best’ way for modelling some aspect of a universe of discourse, but the aim is to illustrate and describe *i*) the consequences of a modelling choice, *ii*) the need for systematically related elements of foundational ontologies, and *iii*) that a foundational ontology library does assist with this.

Example 1 Consider the conceptual data model language element class in UML and entity type in EER or ORM, and its attribute (in UML and EER) or value type (ORM). We wish to link them to their respective ontological version of it in a foundational ontology to make the modelling language more precise, explicate the underlying principles, and foster the development of good quality conceptual models.

A foundational ontology’s counterpart to UML class may be a type of Particular (individual) or Universal

(‘type’). Let us consider several of the foundational ontologies. DOLCE has (categories of) entities subsumed by Particular, such as Endurant (an entity wholly present at a time), Process (entity unfolding in time), Amount of Matter (stuff, like water and gold), and Quality (the ontological version of an attribute) [40]. BFO [<http://ifomis.uni-saarland.de/bfo/>] is an ontology of universals with similar entities; e.g., Independent Continuant (alike dolce:Endurant), Process (alike dolce:Perdurant), and also a Quality. UFO [18] and GFO [21] include both particulars (individuals) *and* universals; they contain, e.g., $\text{gfo:Presential} \sqsubseteq \text{gfo:Particular}$, which is alike dolce:Endurant, and $\text{ufo:Quality} \sqsubseteq \text{ufo:Individual}$.

Which foundational ontology entity should one choose to align class/entity type or attribute/value type to, and does the choice matter? It can, and it does. For instance, take trying to map ‘attribute’: $\text{ufo:Quality} \sqsubseteq \text{ufo:Moment}$ [18], but GFO’s Property, which is similar in idea, is subsumed by gfo:Individual that does not have anything to do with moments. dolce:Quality is also not in agreement with ufo:Quality , for ufo:Moment would have a subsumption alignment with dolce:Perdurant, but $\text{dolce:Perdurant} \sqsubseteq \neg \text{dolce:Quality}$. Put differently, similar ideas, and even the same terms, turn out to be quite distinct ontologically after all. Aligning a conceptual modelling language to a foundational ontology makes explicit the underlying assumptions about those graphical elements: e.g., mapping UML’s Attribute to gfo:Property cf ufo:Quality cf dolce:Quality is essentially disagreeing as to what attributes really are! \diamond

Example 2 A different issue that can arise concerns the coverage of the foundational ontology, i.e., which entities are in its vocabulary and, hence, which can exist, and, following from that, which ones can be modelled in a conceptual data model. Let us take as example GFO’s $\text{Mass_entity} = \text{Amount_of_substrate}$ (like water, gold) and DOLCE’s Amount of Matter for particulars, which convey a similar notion as Guizzardi’s “stuff universal” for which he proposed a stereotype $\ll \text{quantity} \gg$ that is deemed a sortal that, in turn, is a universal in [17] that is associated with UFO. However, UFO does not have anything about amounts of matter in its signature, and nor does BFO. Put differently, and pedantically: stuff does not exist according to UFO and BFO, so we cannot identify and model it. This results in a situation where aligning UML to DOLCE or GFO permits a modeller to create a stereotype for such entities and relate them with, say, subQuantityOf [26,17] to model, e.g., food ingredients, but not when UML would be linked to UFO or BFO. \diamond

These examples may seem confusing, because one does not readily have an overview at hand of each of

the foundational ontologies’ content and structure—i.e., it would be useful if there were one, and we will address this in Section 7.3. For some alignments between the foundational ontology and the conceptual modelling language, it may not matter which foundational ontology is chosen because not all its entities are used in such an alignment. However, one can know this for sure only when there is insight in the systematically assessed mappings between the entities in the foundational ontologies. DOLCE’s and GFO’s notion of amounts of matter, mentioned above, are comparable not only in isolation but also when taking into account the structure of the ontology [31]. We also saw that not all foundational ontologies have the same coverage (Example 2). Instead of the one-off examples described here, what is needed is a systematised way of carrying out such an analysis and, for it to be of computational use, have an easy way to store and access the outcome of such mappings between entities in foundational ontologies.

2.2 Achieving the semantic interoperability

Foundational ontologies contain categories to describe concepts that are common among all domains. As such, it enables semantic interoperability between domain ontologies which are linked to them. However, because there are many foundational ontologies, conceptual modellers, as well as domain ontology developers, end up using their preferred foundational ontology, so that this semantic interoperability issue still exists.

To solve this issue with a library of linked foundational ontologies, theoretical research concerning ontology mediation among foundational ontologies is required. To achieve this, DOLCE, BFO, and GFO were selected, and studied in detail to identify similarities and differences of the structure, organisation and representation of entities among them. There are common aspects among the ontologies, such as each of them containing both 3D (wholly present) and 4D (those unfolding in time) entities, GFO contains both universal and individual entities which aligns with DOLCE’s particular entities and BFO’s universal entities. The organisation and representation of the entities, in general, differ in each foundational ontology and in some cases, entities that appear to be similar fall in incompatible classes of the ontologies.

For interchangeability, it is necessary to perform alignment among the foundational ontologies. Several tools were used for alignment, which resulted in only a few correct alignments due to the philosophical nature of foundational ontologies. The ontologies were then manually aligned resulting in 85 alignments among

DOLCE, BFO, GFO and their variants (GFO-Basic, DOLCE’s Spatial Relation module, etc.).

Ontology mapping uses these alignments to create correspondences among the ontology files to use in practice. There were a total of 43 successful mappings created for the ontologies. An example of an alignment resulting in a successful mapping is that among `dolce:Endurant`, `bfo:IndependentContinuant`, and `GFO:Presential`. Some alignments could not be mapped due to logical inconsistencies among the ontologies. An example of an alignment that could not be mapped is `dolce:temporal-region`, `bfo:Temporal-Region`, and `gfo:Temporal_region` that were incompatible because of their disjointness relations with other entities. A detailed analysis concerning the comparison, and mediation of the foundational ontologies is provided elsewhere [31].

Addressing the need for a systematic investigation into possible alignments and mappings to foster the possibility that the various extensions to conceptual data modelling languages are practically compatible, is only the first step. We also need ways to at least quickly check for this, and, where relevant for the application scenario, switch between foundational ontologies where possible. While this could be carried out and maintained on paper, software-based mapped ontologies where the mappings are at least guaranteed not to lead to any inconsistencies will greatly simplify this process. Such software could then feed into a software-based content negotiation method. Likewise, a catalogue of entities that cannot be mapped serves as an easy online reference of incompatibilities and as points for further investigation by ontologists.

3 Overview of ROMULUS

To meet the generic requirement for creating a foundational ontology repository, we have developed a web-based software system, called the Repository of Ontologies for Multiple USEs (ROMULUS), so that modellers can publicly access and benefit from all the functionalities of the repository. We describe the requirements, design, and features of ROMULUS in this section.

Before the actual design, requirements were formulated specifically for ROMULUS. The functional requirements are briefly described, of which the first three are adapted from the original “WonderWeb Foundational Ontology Library” (WFOL) proposal in [40]. The library must provide a high-level view of the foundational ontologies with only the most general entities common to all implemented foundational ontologies, it must provide a comparison of implemented foundational ontologies, and ontology metadata must be

available [40]. In addition, to serve interoperability and interchangeability, basic ontology mediation must be present, including alignment, mapping, and merging of the foundational ontologies. To facilitate usability, the foundational ontologies in the repository must be modularised, there must be easy and effective online ontology browsing, renderings in human-readable views of each foundational ontology module must be available, and there must be an ontology download facility.

In order to enhance semantic interoperability with foundational ontologies, a Semantic Web language is needed, therefore we use the OWL ontology language which is machine-interpretable and used in Semantic Web applications. While the FOL versions of the ontologies are more expressive and accurate, these versions cannot be practically used in Semantic Web applications. Furthermore, ontology developers more commonly use the OWL versions of the foundational ontologies and not the FOL versions in such applications. We would have liked to use the Distributed Ontology Language (DOL) meta-language [42] to assist with achieving semantic interoperability, but it is still being standardised by the Object Management Group (OMG).

We describe the front-end and back-end features (Section 3.1), the access to ontology mediation data (Section 3.2), the updated ontology recommender (Section 3.3), and the ontology interchangeability tool (Section 3.4) in this section.

3.1 Front-end and back-end features

The modular design of the interface of the foundational ontology library is met through different tabs in the user interface of the repository. WebProtégé [48] is used for an online ontology browsing library, which requires a tomcat server to function. Similar to the ontology browsing library, WebProtégé is also used also to provide easy access to all the mappings and the merged ontologies. Separate HTML pages are available with tables and lists containing the comparison of foundational ontologies for the different categories of criteria¹. Further pages are accessible that contain the user-readable version of the alignments of the ontologies and their metadata. The SWAT Natural Language tool [51] was used to generate the HTML pages of the verbalisation of the axioms in each ontology, and the Protégé-generated description logic axioms of each ontology are available in pdf format. There is a Community page in ROMULUS, that is integrated with the Ontohub repository

¹ There is a page for each of the criteria—ontological commitments, representation languages, software engineering properties, subject domains and applications—within the comparison area of the web interface

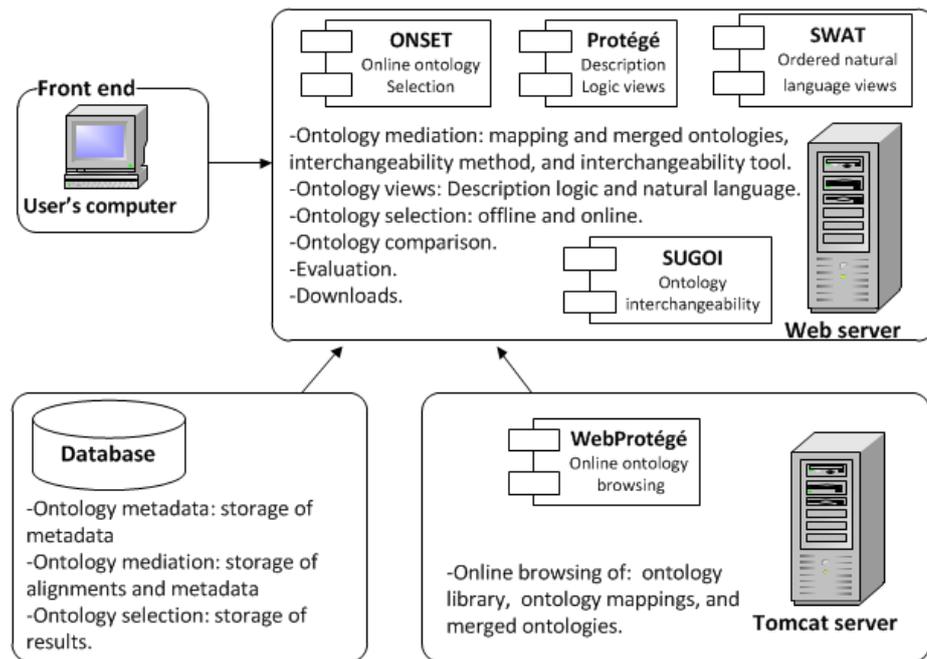


Fig. 1 The interaction of ROMULUS's components.

[37] whereby users are able to upload their own foundational ontology mappings which will then be validated by us and included in the repository, if correctly defined. ROMULUS's alignments and mappings are stored in a database and can be accessed from ROMULUS's foundational ontology interchangeability page for browsing (in HTML format) and querying. Several pages are dedicated to searching for information, such as the mappings and metadata (elaborated on below).

From an implementation viewpoint, the three principal components of ROMULUS are the web server, tomcat server, and a relational database. The PHP-based web server is used to execute the PHP scripts, the tomcat server is used to execute JSP pages, and the MySQL database is used to store all the data. The interaction of the components in ROMULUS is shown in Fig. 1.

3.2 Access to ontology mediation data

ROMULUS contains many resources to enable mediation between foundational ontologies. Its alignments are specifications of correspondences between entities, and are independent of the actual ontology file. Based on these alignments, there are mappings between foundational ontologies. Mappings differ from alignments in that they are the correspondences between entities, and occur in the ontologies themselves such that the ontology remains consistent and has no undesirable deductions. Details of the ontology mediation can be found in

[31,32], and all the alignments, mappings, and descriptions of inconsistencies are contained in the database, which are rendered as HTML tables in the repository.

Fig. 2 shows a screenshot of a part of the metadata for the alignments between BFO and GFO in ROMULUS. Users are also able to search the alignments and mappings; an example is displayed in Fig. 3: the selection is made to examine BFO and GFO (top, left-hand side), and to retrieve only those mediations that have been obtained automatically with LogMap (top-right-hand side), and the results are shown in the figure in the bottom-half.

The mappings and merged ontologies may also be viewed online using WebProtégé. Many alignments cannot be mapped since logical inconsistencies arise between entities that seem to be well-matched on the surface and when considered in isolation. Each logical inconsistency for an alignment that is not mappable is presented together with a human-readable explanation in ROMULUS.

3.3 Online foundational ontology selection

While it is a lofty goal to have a situation that it ought not to matter which foundational ontology is taken for a task, the state of the art is not there yet and will not be in the foreseeable future. To this end, ROMULUS has an integrated ontology recommender tool to aid the process of foundational ontology selection. The ONtology Selection and Explanation Tool (ONSET) [30] as-

Metadata for alignments between BFO and GFO

	Entity URI	Relation	Entity URI	Mappable	Mediation Inconsistency	Mediation Date	Mediation Method	Mediation Tool	Creator
1.	http://www.ifomis.org/bfo/1.1#Entity	Equivalence	http://www.onto-med.de/ontologies/gfo.owl#Entity	Yes	-	2012-11-21	Automatic	YAM++ v1.0, PROMPT v3.0, Optima v3.0, LogMap v2.0, HotMatch v1.0, Hertuda v1.0	Zubeida Khan: zkhan@csir.co.za
2.	http://www.ifomis.org/bfo/1.1/snap#independentContinuant	Equivalence	http://www.onto-med.de/ontologies/gfo.owl#Presential	Yes	-	2012-11-21	Manual	-	Zubeida Khan: zkhan@csir.co.za
3.	http://www.ifomis.org/bfo/1.1/snap#DependentContinuant	Equivalence	http://www.onto-med.de/ontologies/gfo.owl#Dependent	No	GFO's Entity is equivalent to its Item or Set. GFO's Category is a subclass of Item. GFO's Category is disjoint to its Individual. GFO's Dependent is a subclass of Individual. GFO's Entity is equivalent to BFO's Entity. BFO's Entity is equivalent to its	2012-11-21	Manual	LogMap v2.0	Zubeida Khan: zkhan@csir.co.za

Fig. 2 Example of metadata of ontologies and some values for the alignments between BFO and GFO.

Ontologies involved in mediation

Ontology 1 Name

Ontology 1 Acronym

Ontology 2 Name

Ontology 2 Acronym

Entities involved in mediation

Entity 1 Name

Entity 1 URI

Entity 1 type Class Object Property

Entity 2 Name

Entity 2 URI

Entity 2 type Class Object Property

Properties of mediation

Mediation relation Equivalence Subsumption

Mediation mappable Yes No

Mediation method Manual Automatic

Mediation tool used

Person involved in ontology mediation

Person Name

Search results

Advanced search results.

Parameters: Ontology 1 Acronym :bfo, Ontology 2 Acronym :gfo, Mediation method: Automatic, Mediation tool: Logmap

	Ontology 1	Entity 1	Relation	Ontology 2	Entity 2
1.	BFO	http://www.ifomis.org/bfo/1.1#Entity	Equivalence	GFO	http://www.onto-med.de/ontologies/gfo.owl#Entity
2.	BFO	http://www.ifomis.org/bfo/1.1/snap#Function	Equivalence	GFO	http://www.onto-med.de/ontologies/gfo.owl#Function
3.	BFO	http://www.ifomis.org/bfo/1.1/span#Occurrent	Equivalence	GFO	http://www.onto-med.de/ontologies/gfo.owl#Occurrent
4.	BFO	http://www.ifomis.org/bfo/1.1/span#Process	Equivalence	GFO	http://www.onto-med.de/ontologies/gfo.owl#Process
5.	BFO	http://www.ifomis.org/bfo/1.1/snap#SpatialRegion	Equivalence	GFO	http://www.onto-med.de/ontologies/gfo.owl#Spatial_region

Fig. 3 The advanced mediation search and results for alignments between BFO and GFO ontologies that have been generated automatically using the LogMap mediation tool.

sists developers selecting desirable criteria upon which ONSET computes the most appropriate foundational ontology. If the user has requirements matching more than one foundational ontology, conflicting results are displayed, which are those features required by the user but not provided by the selected foundational ontol-

ogy; e.g., when DOLCE scores best overall, but the user wanted a realist ontology (BFO and GFO are realist). A scenario is included in the next example.

Example 3 An example of ontology selection with ONSET is as follows: 1) The requirements of an ontology is for it to have modules of separate 3D and 4D entities,

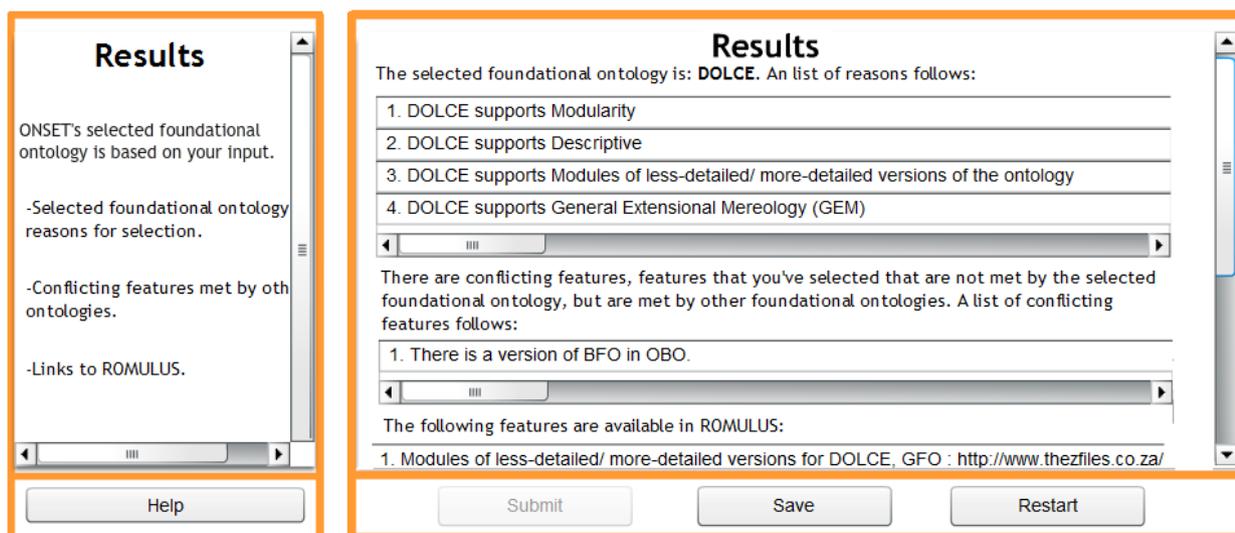


Fig. 4 Output from the new version of ONSET (version 2.0).

descriptive in nature, for it to use the axioms of General Extensional Mereology (GEM) and for it to be representable in OBO language, which each are checkboxes that have to be ticked in the criteria tabs, 2) computation of the results and 3) display of the result and explanations for the selection. Fig. 4 is a screenshot of the new version of ONSET's output for these requirements. In this case, there is a conflicting result, because DOLCE is not available in OBO (but BFO is). ◊

ONSET also generates a list of references to existing projects related to the user's selected domain, where available. Further, if a module was a requirement, then a link to the relevant module in ROMULUS is displayed as well (modules are discussed in Section 4). Overall, several changes have been made to ONSET v2 compared to its first version, of which we highlight four.

First, instead of a stand-alone jar file, there is now a web-based version, although users still can download the offline version of ONSET (albeit without links to ROMULUS's features but simply to perform ontology selection locally). Second, the web-based version has access to ROMULUS's centralised database. This also facilitates the new feature that users can save their ontology selection results, be this locally in a CSV file format or in ROMULUS's database, which can be used for further analysis and investigation with regard to foundational ontology usage and selection. The conceptual data model for that section of the database is displayed in Fig. 5, showing that the database stores both the output explanation of the selection algorithm (Reason-SelectedFO), and any conflicting answers it may have computed (ReasonConflicting).

The ontology selection results stored in ROMULUS's database from March 2014 to July 2015 indicate that

the web-based version of ONSET has been used 74 times. DOLCE was selected as an appropriate foundational ontology 31 times, followed by BFO and GFO at 14 and 9 times respectively. The remaining 20 results were tied between selection combinations of DOLCE, BFO, GFO, SUMO, YAMATO, and GIST. The subject domain criteria had been used during ontology selection 57 times, and the results are shown in Fig. 6.

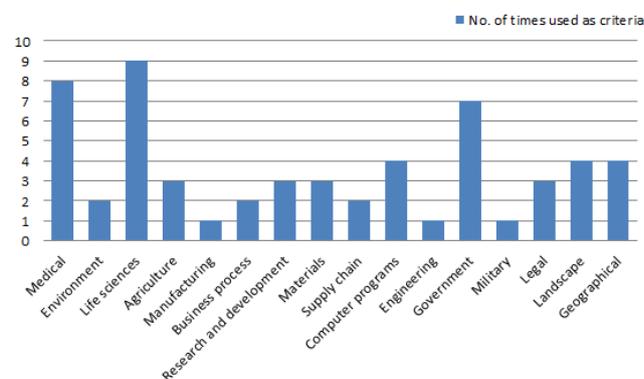


Fig. 6 The frequency of subject domains that were used as criteria in the ontology selection process.

Another, third, difference of the web-based version of ONSET of the separate jar file, is that it provides links to features in ROMULUS, such as its modules and metadata for a particular foundational ontology. Fourth, the YAMATO and GIST foundational ontologies have been added to ONSET v2.0, therewith providing the user with more possible foundational ontology choices (the other ones that were already present in v1.2 are BFO, DOLCE, GFO, and SUMO).

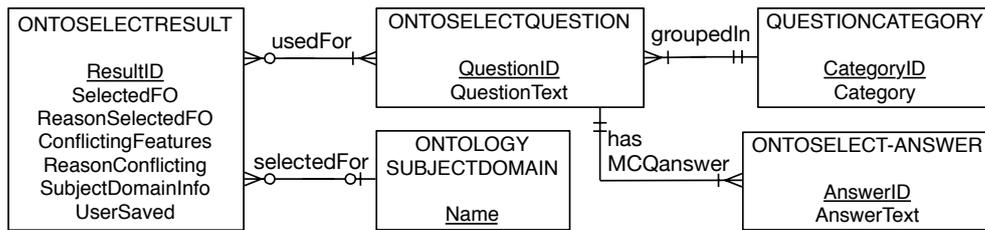


Fig. 5 ER diagram of ONSET's computed data that is saved in the back-end database.

ONSET has been used for ontology development as reported in scientific publications, such as for the data mining [27] and biomedical [4] domain ontologies.

3.4 Changing one's foundational ontology preference

ROMULUS guides the user on foundational ontology interoperability with its step-by-step interchangeability method. To easily perform ontology interchangeability among domain ontologies, the method has also been implemented in a tool, a Software Used to Gain Ontology Interchangeability (SUGOI) [33] which is integrated with ROMULUS.

SUGOI allows a user to input a domain ontology linked to a foundational ontology and automatically interchange it to another foundational ontology. The current version of SUGOI can swap between the DOLCE, BFO, and GFO foundational ontologies, thanks to the available mappings on ROMULUS. It is easily extensible to handle interchangeability with other foundational ontologies, as only new mapping files need to be uploaded to the tool.

There are three versions of SUGOI available currently: the applet integrated into ROMULUS for on-line usage, a desktop online version requiring internet connectivity, and a desktop offline version. Fig. 7 is a screenshot for the applet version of SUGOI. SUGOI generates an interchanged ontology and a log file containing changes that have been made and information about the success of the interchangeability. A scenario for ontology interchangeability using SUGOI is shown in the next example.

Example 4 The OntoDerm ontology [12], a domain ontology for dermatology was created using DOLCE to speed up development and facilitate interoperability using its general categories. If one were to extend the OntoDerm ontology with information about infectious diseases, one could consider the Infectious Disease ontology (IDO) [6]. However, the IDO ontology is linked to BFO foundational ontology while OntoDerm is linked to DOLCE foundational ontology. Thus there are semantic conflicts for these two domain ontologies that

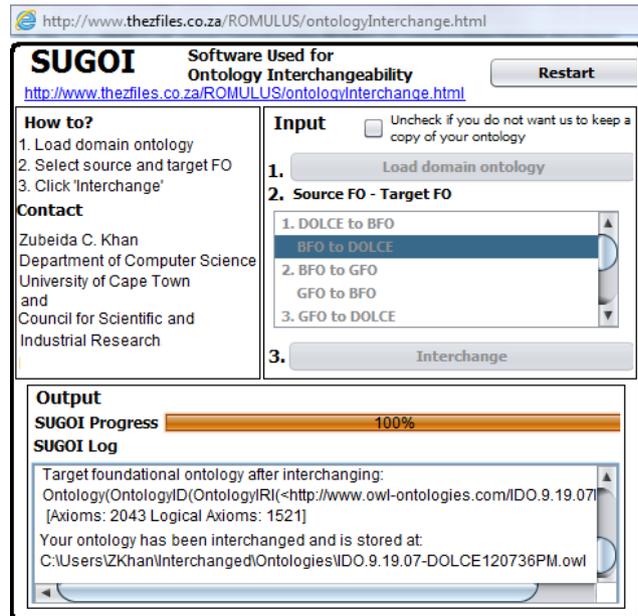


Fig. 7 Output of the applet version of SUGOI.

are linked to different foundational ontologies. To solve this problem, ontology developers could use SUGOI to interchange the IDO ontology from BFO to DOLCE, or the OntoDerm ontology from DOLCE to BFO.

To interchange the IDO ontology from BFO to DOLCE, the steps for the algorithm are as follows:

1. Create a new ontology file, a target ontology (${}^t\mathcal{O}$): `ido-dolce.owl`.
2. Copy the entire target foundational ontology (${}^t\mathcal{O}_f$) to the ${}^t\mathcal{O}$: copy DOLCE into `ido-dolce.owl`.
3. Copy the axioms from the source domain ontology (${}^s\mathcal{O}_d$) to the ${}^t\mathcal{O}$: e.g., consider `ido:Injection` \sqsubseteq `bfo:Object` (*axiom1*) and `ido:Sign` \sqsubseteq `bfo:Role` (*axiom2*) that exist in the IDO source ontology (${}^s\mathcal{O}$). We add these axioms to `ido-dolce.owl` and they are referred to as 'new' axioms.
4. Change the 'new' axioms to reference ${}^t\mathcal{O}_f$ entities: e.g., for *axiom1*, we can use the mapping `bfo:Object` \equiv `dolce:physical-object`, hence we change *axiom1* `ido:Injection` \sqsubseteq `bfo:object` to `ido:Injection` \sqsubseteq `dolce:physical-object`, which is shown in Fig. 8. For *axiom2*, there is no equivalence mapping between `bfo:Role` and

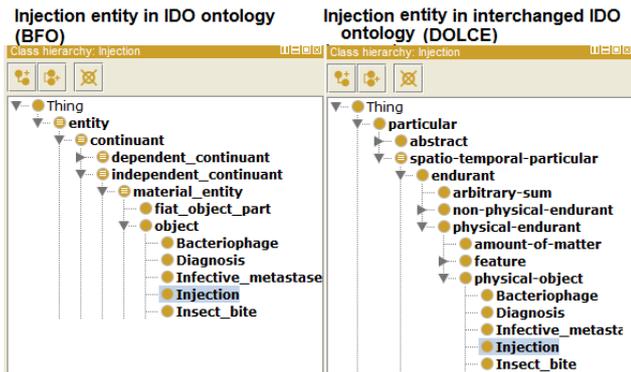


Fig. 8 The position of the `ido:Injection` class in the source and target ontologies.

DOLCE entities, for which we need the next step to resolve.

5. If a mapping does not exist, perform on-the-fly subsumption, if possible: considering *axiom2* again, we note `bfo:Role` \sqsubseteq `bfo:Realizable entity`, but there is still no mapping between this and DOLCE entities. For all the ancestor classes of `bfo:Role`, there is no mappable DOLCE entity, hence finally, *axiom2* (`ido:Sign` \sqsubseteq `bfo:Role`) is contained in the ${}^t\mathcal{O}$ as a subclass of `owl:Thing` outside the scope of DOLCE.
6. Delete entities that exist in the ${}^t\mathcal{O}$ that are from the foundational ontology of the source ontology (${}^s\mathcal{O}_f$) but that do not appear in an axiom with entities from the target domain ontology (${}^t\mathcal{O}_d$), resulting in the final ${}^t\mathcal{O}$, `ido-dolce.owl`. Delete the `bfo:Object` entity from `ido-dolce.owl`. \diamond

Experimental evaluation conducted for SUGOI [33] using 16 domain ontologies indicate that the success of interchangeability ranges from 2% to 82% success, with an average of 36%. Comparing SUGOI to manually mapped domain ontologies, SUGOI found some additional alignments, but also missed a few due to the difference in coverage of foundational ontologies [33].

4 Modularising foundational ontologies

Ontology modularisation deals with creating or altering an ontology to be extracted to serve a specific function. This concerns both the logic and algorithmic aspects [7, 8, 35, 36] and the types of modules [3, 38, 44]. The general idea is to remove or hide unnecessary detail when it is not required. Factors pertaining to modularisation and its techniques are discussed elsewhere (e.g., [11]). Borgo broadly classifies modules as one of the following: modules for a single ontology, modules for several ontologies and modules for everything, where modules are typically created along dimensions of domain coverage,

to isolate branches of a taxonomy, to extract a particular subject domain and/or (sub)theory, to isolate patterns, to assist with scalable automated reasoning, or to reduce the cognitive overload [3].

To classify the modules in ROMULUS, we used Borgo’s classification of branch modules [3]. In addition, we create our own types pertaining to those modules that exist [34]: domain-independent, sub-language, weighted abstraction, and high-level abstraction modules. They are explained in Section 4.1 and used in ROMULUS as described in Section 4.2.

4.1 Module types

The types of modules found in ROMULUS are described here.

Functional modules These modules are those in which the users identify the functional components within an ontology to be separately modularised, with the purpose of selective re-use of an ontology. First, they may cover ‘domain-independent’ orthogonal dimensions, such as the TemporalRelations module that extends DOLCE with temporal relations. Next, one may be interested only in one ‘isolation branch’ or subset of an ontology.

Expressiveness modules These modules are those which are slimmed down by removing some expressiveness power. ‘Sub-language’ modules aid in scalability by removing some features to slim the ontology down to a sub-language, say, OWL 2 EL so that one can use the computationally better behaved ELK reasoner [23].

Abstraction modules Some detail is removed from the ontology to make the ontology conceptually lightweight. One can prune the lower-level entities for ‘high-level abstraction’ (e.g., `gfo-basic`). Next, there is ‘weighted abstraction’, where one can use an abstraction algorithm that uses assigned weights to axioms, where, say, a class with multiple existentially quantified properties are assumed more important than entities that do not have this and are collapsed into the more important ones for that reason (e.g., [5, 24]).

4.2 Modules in ROMULUS

Modularisation ideas have been incorporated in ROMULUS on an experimental basis to facilitate (re) usability of the foundational ontologies. OWL Module extractor [8], Swoop [22], and Protégé [1] have been experimented with to create the modules. Both OWL Module extractor and Swoop use a logic based algorithm to

analyse the axioms only, which resulted in large modules similar to the original ontologies because there are many axioms in DOLCE and GFO in such a way that there are no sparsely connected subsections. For instance, DOLCE’s *endurant* and *perdurant* are related through *participates-in*, making it difficult to separate the hierarchies. Therefore, we had to manually remove some of the axioms relating the two entities. A similar issue existed in the attempt to create a DOLCE module without quality and qualia. Protégé generated smaller modules according to the user’s input, in most cases, but some unnecessary entities were still present after using Protégé and they were manually modularised as well. We created the following modules:

DOLCE modules:

- DOLCE-Endurants, DOLCE-Perdurants, and DOLCE-NoQualityAndQualia; There are axioms relating different types of entities in DOLCE, therefore it was necessary to remove them to create smaller, compact modules; they are of the type *Functional* and of subtype *Isolation branch*.
- DOLCE-EL, DOLCE-QL and DOLCE-RL: corresponding to the OWL 2 profiles; they are of the type *Expressiveness* and of subtype *Sub-language*.

BFO modules:

- BFOContinuants and BFOOccurents: as there are no cross-relationships between entities in BFO, this was easy to generate; they are of the type *Functional* and of subtype *Isolation branch*.
- BFO-EL, BFO-QL, and BFO-RL: corresponding to the OWL 2 profiles; they are of the type *Expressiveness* and of subtype *Sub-language*.

GFO modules:

- GFONoOccurents and GFONoPersistantsAndPre-sentials: same case as with DOLCE-Endurants mentioned above; they are of the type *Functional* and of subtype *Isolation branch*.
- GFOBasicEL, GFOBasicQL, and GFOBasicRL: corresponding to the OWL 2 profiles; they are of the type *Expressiveness* and of subtype *Sub-language*.
- GFOATO (based on the Abstract Top Level layer) and GFOACO (based on the Abstract Core Level): These modules contain the high-level meta-categories of GFO; they are of the type *Abstraction* and of subtype *High-level abstraction*.

In addition, we include the modules *TemporalRelations*, *SpatialRelations*, *FunctionalParticipation* (of the type *Functional*, *Domain-independent*) and *GFO-basic* (of the type *Abstraction*, *Weighted*) in the

foundational ontology mediation, because they have new axioms that are not in their related foundational ontology. Other modules, such as DOLCE-Endurants, do not have any new axioms, so the mappings for those entities still available in the module are taken from their respective full OWL version.

Such ‘slimmed’ modules based on their comprehensive original version of the foundational ontology not only can be used in ontology development projects (see [34] for use cases and further references), but also could be used to help to make interoperable, e.g., user models for knowledge-aware adaptive services [9] and assist with fundamental aspects of interoperability in big data management [10] whilst keeping such systems scalable.

5 Ontology and library metadata

Metadata about the (foundational) ontologies are required to assist ontology developers with understanding the context of the ontology and with reusing an ontology effectively. Metadata values for each ontology—original, modularised, mapped, and merged—have been specified. To do so with an eye on interoperability with other ontology repositories, we have taken into account and used other metadata vocabularies, being the Ontology Metadata Vocabulary (OMV) [20], which is a general OWL-formalised metadata vocabulary, and OM²R metadata model [47], which is aimed at promoting ontology mapping reuse.

The ROMULUS metadata model adopts several classes and attributes from the OMV and the OM²R metadata models, refines some others regarding constraints and entity types, and extends this considerably with its own metadata. Concerning the latter, the additional metadata in ROMULUS that is not present in OMV and OM²R are about modularity—no metadata model includes anything about modularity—and more aspects of the mediation process for an alignment is recorded, such as mediation relation, mediation mappable, inconsistency explanation, mediation creator, mediation set level and mediation set percentage. The refinements involve data about contributors and metrics with respect to the OMV diagram in [20]. We have not used the OWL version of OMV, because there were some ontological issues such as missing object and data property characteristics and *is-a* vs. *instance-of* errors. Among others, transitivity holds for the *isSubDomainOf* object property, but is not declared; data properties such as *URI* and *numAxioms* should have been made functional in OMV, because an ontology (file, version) has at most one URI and one set of axioms; there are many instances declared in OMV that, upon taking a

closer look, are not instances (the instances of `OntologyTask`, such as `AnnotationTask`, can be instantiated and should be declared as subclasses of `OntologyTask`, not members). Also, no deductions are computed upon running the reasoner over OMV, therefore it suffices at this stage to store the metadata in a database and work towards a long-term goal of creating an ontology with more axioms where knowledge can be inferred.

The conceptual model for ROMULUS's metadata (Fig. 9) is explained now for the main entities.

- **Ontology:** This represents an ontology in the repository that is to be annotated with metadata; e.g., BFO.
 - **Language:** This represents the language that the ontology is serialised in; e.g., DOLCE in OWL (called DOLCE-Lite) is represented in OWL DL.
 - **Entity:** This is to store an entity in an ontology's signature, which is used typically only for those entities that are involved in an alignment, mapping, or merging, and can thus be either an OWL class or OWL object property.
 - **Mediation:** This represents the mediation that the entity is involved in, which is identified by a separate identifier for convenience. It has many attributes, some of which have recurring values, which are therefore given their own entity type. The `MediationRelation` describes the type of relation that exists between two entities (to date, only equivalence and subsumption have been used). The attribute `MediationMappable` states whether the mediation of the two entities can be mapped and `MediationInconsistency` describes the inconsistency that may occur if the entities are mapped (hence, the entities cannot be mapped). Each mediation has exactly two entities participating in it, and each participating entity is from a different signature.
 - **MediationSet:** This represents a set of alignments aggregated from the previous mediation. `MediationSetLevel` describes whether the ontology is a mapping ontology or a merged ontology. A mapping ontology contains the mappable relations between two ontologies whereas a merged ontology contains the mappable relations together with the original ontologies. `MediatedPercentage` is a value that measures the amount of the original ontologies that are mappable.
 - **MediationSetType:** This represents the type of mediation that was performed (foundational ontology to domain ontology etc.). The `MediationSetTypeName` attribute indicates the type of ontologies involved in the mediation and `MediationSetTypeDescription` provides further details on the `MediationSet` if there are any; e.g., if a domain ontology has been mediated to a particular module of a foundational ontology or an older version of a foundational ontology has been used in mediation.
 - **Module:** This represents the ontology module, hence, it is a subclass of `Ontology`. The `ModuleType` and `ModuleSubtype` were described in Section 4. `ModuleCoverage` represents the value of the ontology (amount of axioms) that is covered by the module; e.g., 91% of the original DOLCE ontology is covered in the DOLCE-RL module. `ModuleCorrectness` states whether the module is logically correct, i.e., if all the axioms from only the original ontology are found in the modules and nothing new has been added to the module. `ModuleCompleteness` is the reverse of `ModuleCorrectness` and states whether for every axiom in the original ontology the meaning of the axiom is persevered in the module. `ModularisedClassSize` represents the amount of classes of the original ontology that remain in the module. `ModularisedPropertiesSize` represents the amount of the properties of the original ontology that remain in the module. (Note that a possible `ModularisedAxiomSize` attribute is already captured with `ModuleCoverage`.)
 - **Tool:** This entity type serves to store information about the tool that was used for modularisation or mediation of the ontologies. The `ToolMethod` describes the method that the tool implements, such as graph partitioning, and the `ToolAlgorithm` attribute is for the algorithm that the tool applies, such as the greedy graph algorithm.
 - **Method:** This is for recording how the mediation or modularity was performed, which currently can take either manually or automatically.
 - **Metrics:** This represents statistical information about the size and expressivity of the ontology.
 - **Project:** This represents a project that an ontology is applied in. The `ProjectDomain` describes the subject domain of the project, such as in biology or computer science, and the `ProjectUsageApplication` describes the implementation of the project, such as using the ontology in natural language processing or an ontology-driven recommender system.
- ROMULUS stores the metadata for each full and modularised ontology in the back-end database. Storing the metadata makes sense mainly because it is easier to implement queries to search the data, duplication is minimised, and it is possible to convert the database to an ontology or RDF triple store to offer it as linked data, if the need arises. A sample of the queries that can be formulated include, among others:
- Which BFO modules are logically complete?

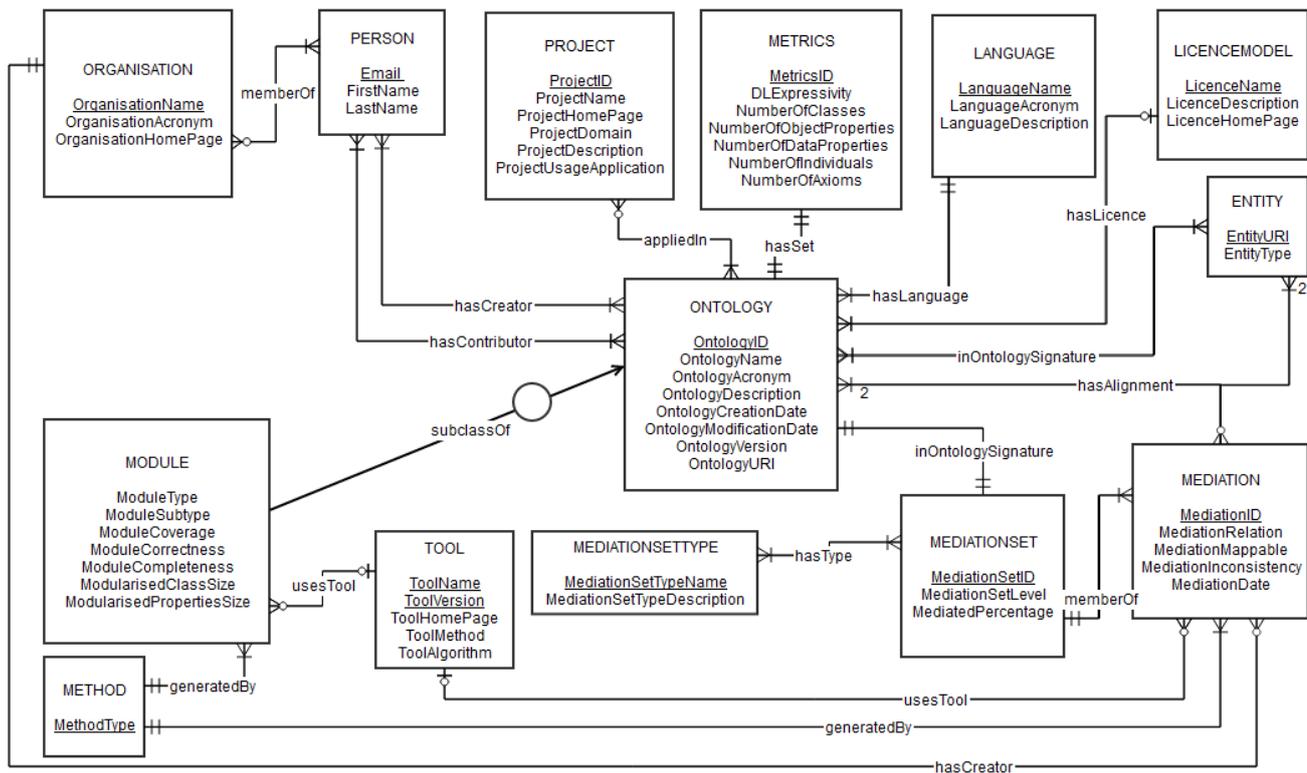


Fig. 9 ER diagram of ROMULUS’s database extending (a subset of) OMV and OM²R; the diagram uses look-across notation.

- Retrieve GFO modules that are <60% of the size of the original ontology.
- Which alignments between DOLCE and GFO have been created manually?
- Which ontology modules have been generated automatically by the Protégé [1] tool?
- Which alignments between DOLCE and GFO are not mappable and what is the explanation for that inconsistency?
- What maps to *dolce:endurant* in other ontologies?
- Are there any subsumption relations for mediation that have been performed automatically?

A screenshot of one such advanced metadata search is displayed in Fig. 10: The search pane (top-half) shows the possible selection criteria, where for the example a module is searched (middle) and the “Module coverage” is set to “less than” “60%” of the original ontology (right-hand side), and the results are displayed in the bottom-half of the figure. Note that the metadata for an ontology is also rendered to HTML pages if the user wants to read it in one overview.

The module metadata together with search query infrastructure can be used in future works to acquire module information in efforts towards characterising modules [11] where there is insufficient criteria to define, assess, and relate different aspects of modularity

(the use cases, modularisation approaches and evaluation criteria). If other ontology repositories employed ROMULUS’s modularity metadata, interested ontologists could acquire results for, say, modules of a particular type, with its respective modularity evaluation properties such as logical properties (Completeness and Correctness) and structural properties (ModuleCoverage), and then create dependencies between them towards devising a foundation for modularity.

6 Related works

We compare ROMULUS to the main other ontology repositories. The feature comparison with OOR [2], BioPortal [50], TONES [49], COLORE [14] and Ontohub [37] is summarised in Table 1.

In terms of repository vision, ROMULUS is a repository of foundational ontologies. Users can upload their own ontologies or data, using the new Community feature page linked to Ontohub, and are also encouraged to download the ontologies and data from the repository. BioPortal, OOR and Ontohub are an open repositories where users are encouraged to upload their ontology projects, contributions, and download resources. BioPortal is a repository of biomedical ontologies. TONES is aimed at being a central location for ontologies that

Ontology details

Ontology Name

Ontology Acronym

Language details

Language Name

Language Acronym

Organisation details

Organisation Name

Organisation Acronym

Developer details

Developer Name

Developer email address

Is the ontology a module?

OR

Is the ontology a mediation ontology?

Mediation details

Mediation Creator:

Mediation Level: Map Merge

Mediation Method: Manual Automatic

Mediation Tool Name:

Mediated Percentage: Less than Equals Greater than % of original ontologies

Module details

Original Ontology Name or Acronym

Module Type: More/less detailed More/less expressive

Reusable component

Module Subtype: Rich/light modules High-level abstraction

OWL profile Domain-independent Branch

Module Correctness: Yes No

Module Completeness: Yes No

Module Coverage: Less than Equals Greater than % of original ontology

Module Class Size: Less than Equals Greater than % of original ontology

Module Property Size: Less than Equals Greater than % of original ontology

Modularity Method: Manual Automatic

Modularity Tool Name:

Search results

Advanced search results.

Parameters: Original ontology of module: gfo, Module coverage: < 60 of original ontology

Ontology ID	Ontology Name	Ontology Acronym	Link to Metadata
17	General Formal Ontology (Abstract-Top-Ontology)	GFO-ATO	http://www.thezfiles.co.za/ROMULUS/Metadata/GFO/gfo-ato.php
18	General Formal Ontology (Abstract-Core-Ontology)	GFO-ACO	http://www.thezfiles.co.za/ROMULUS/Metadata/GFO/gfo-aco.php
19	General Formal Ontology (basic-EL)	GFO-basic-EL	http://www.thezfiles.co.za/ROMULUS/Metadata/GFO/gfo-basic-EL.php
20	General Formal Ontology (basic-QL)	GFO-basic-QL	http://www.thezfiles.co.za/ROMULUS/Metadata/GFO/gfo-basic-QL.php
21	General Formal Ontology (basic-RL)	GFO-basic-RL	http://www.thezfiles.co.za/ROMULUS/Metadata/GFO/gfo-basic-RL.php

Fig. 10 The advanced metadata search and results for GFO modules that are <60% the size of the original GFO ontology.

will be helpful for application developers for testing purposes, but are only allowed to download the ontologies and view some metadata. COLORE aims to be an open repository of Common Logic ontologies to aid in ontology evaluation and integration techniques, and to support the design, evaluation, and application of ontologies in first-order logic.

From the comparison of functionality, ROMULUS provides advanced functionality in most of the criteria used in this evaluation. ROMULUS also provides features that are not available in other repositories, which therefore merited the development of a new repository. These include complete metadata for each ontology that also includes metadata about modularity and ontology mediation beyond the standard metadata vocabularies, carefully analysed alignments and merged ontologies, a foundational ontology interchangeability method and tool, and a foundational ontology selection and explanation tool for guidance to select the most relevant one.

7 Discussion

We discuss first the general design and implementation of the foundational ontology library, reflect on the me-

diation in Section 7.2, and turn to ontology-driven conceptual data modelling in Section 7.3.

7.1 On the realisation of ROMULUS

ROMULUS combines various widely used and recent semantic web technologies to provide a broad set of features and it is the first realisation of the “WFOL” envisioned since 2003. It meets the main WFOL goals [40], which were described in Section 3, and 1) it is a reference point for comparisons between different ontological approaches thanks to the multi-dimensional criteria comparison of foundational ontologies, which are also incorporated in the online and offline versions of the foundational ontology recommender ONSET; 2) it provides a common framework for analysing, harmonising and integrating existing ontologies by availing of its alignments, mapping and merged ontologies, and the metadata for each ontology; 3) it has modularised foundational ontologies to tailor different possible use cases; 4) it has a higher-level ontology, FFO, that contains the common entities of DOLCE, BFO and GFO ontologies, which can serve as a starting point for ontology development; 5) it has many user usability features, such

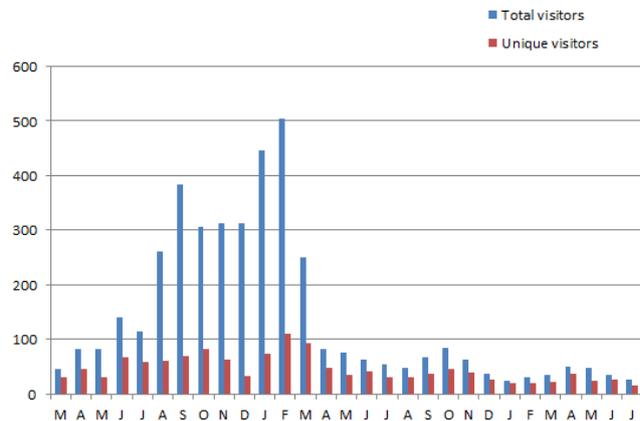
Table 1 Comparison of ROMULUS's features with those of other ontology repositories.

	ROMULUS	BioPortal and OOR	TONES	COLORE	Ontohub
Browse	Uses WebProtégé. Hierarchical ontology view. Displays classes, object properties, individuals and annotations of an ontology.	Hierarchical ontology view. Displays classes and class local neighbourhood. Has advanced visualisation support.	Browsing is currently unavailable.	No support.	Hierarchical ontology view. Displays axioms, classes, object properties, individuals and annotations of an ontology.
Mediation	Alignments and mappings between its ontologies, merged and higher-level foundational ontologies, mapping inconsistencies and a method and tool for foundational ontology interchangeability.	Some mappings between ontologies. Users may specify mappings.	No support.	No support.	Users may specify mappings in DOL format which is rendered as a table.
Metadata	Metadata for ontologies, based on OMV, OM ² R and own extension.	Metadata lists for ontologies based on OMV, but some missing metadata.	Metadata for metrics of each ontology.	Metadata exists in the ontology files.	Metadata lists for ontologies, but currently not as extensive as ROMULUS's.
Ontology selection	Foundational ontology recommender ONSET available online and for download.	An ontology recommender system that allows one to calculate which ontologies are most relevant for a corpus.	No support.	No support.	No support.
Search	Extensive user-friendly fine-grained capabilities to search the alignments, mappings, inconsistencies, and metadata.	Advanced search. Allows a user to search for entity names, ids, synonyms, properties and filter search.	No support.	No support.	Search for ontology name and ontology symbol.
Ontology view	Views in description logics and natural language.	No support.	No support.	Views in common logic.	No support.
Comparison	Ontology comparison in terms of different categories of criteria.	No support.	No support.	No support.	No support.
Ontology access	Users can view and download ontologies and tools. Users can upload ontologies and mappings thanks to the integration with Ontohub.	Users can upload, edit own ontologies and download ontologies.	Users can download ontologies.	Users can view and download ontologies.	Users can upload, edit own ontologies and download ontologies. Users can also create own repositories within Ontohub.

as online browsing of the ontologies and verbalisations. Further, it is rigorous in its logic-based approach, and has been extensively researched.

From an ontology engineering viewpoint, ROMULUS is a major step toward interchangeability of foundational ontologies, because a prerequisite for this is the mapped ontologies so that selection of a foundational ontology has become less of an issue. Meaning negotiation between two domain ontologies that each are linked to a different foundational ontology has now become something within reach, for it can use the mappings in between those two foundational ontologies. Although the technologies that are used to realise ROMULUS might seem fairly straightforward now, they were not until recently, and is it principally the realisation of a comprehensive foundational ontology library with a range of features relevant specifically for foundational ontologies that makes ROMULUS a novelty.

We have been collecting usage statistics for ROMULUS since March 2013, which reveal that the repository has been accessed 4056 times up until July 2015. The number of repository visitors is aggregated by month

**Fig. 11** ROMULUS's visitors from March 2013 to July 2015.

and has increased from 45 visits in March 2013 to 504 visits in February 2014, of which 31 and 111 are unique visitors, respectively. Thereafter from February 2014 to July 2015, it has decreased from 504 to 26 visits respectively. The visitor statistics for this time period is displayed in Fig 11.

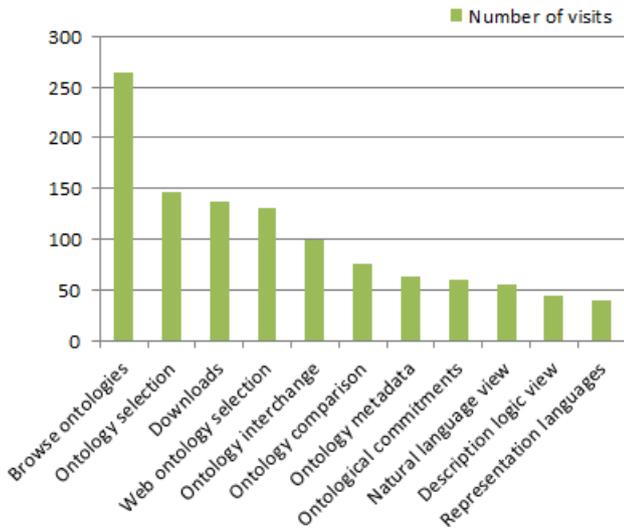


Fig. 12 ROMULUS’s most frequently accessed pages from March 2014 to July 2015.

Detailed usage statistics about which features in ROMULUS are the most popular, or frequently used pages are available from March 2014 to July 2015. The most popular page, with 264 views is the ‘Browse ontology’ page, followed by the ‘Ontology selection’ page, with 146 views. The 10 most frequently visited pages of ROMULUS is shown in Fig 12.

7.2 Notes on the mediation

It is unlikely that ontology developers will commit to using a single unified foundational ontology for ontology development because different foundational ontologies exist that have conflicting philosophies, such as descriptive vs. realist and multiplicative vs. reductionist. To a certain extent, this affects the way that entities are represented. Descriptive ontologies capture concepts based on human common-sense and understanding, whereas realist ontologies capture the world as it is and excludes cognitive aspects such as belief and ‘deprecated’ entities, such as phlogiston. As such, the former allows for abstract entities, while the latter does not. The distinction between possibilism and actualism affects the ontologies in a similar way. Rather than trying to enforce a worldwide ontology, it is achievable to enable interoperability among the existing foundational ontologies by performing ontology mediation. While the differences in philosophical choices affect some processes of ontology mediation, in most cases it is possible to align entities independently of the foundational ontologies’ philosophies. Concerning the current alignments, we have decided to ignore certain aspects of the underlying philosophies of each foundational ontology, be-

cause else it would result in few or no alignments, and from a practical ontology engineering viewpoint, some distinctions do not matter, as they are extra-logical. For instance, an OWL file is agnostic about whether, e.g., an OWL class really refers to a universal in reality or not and some differences in the descriptions of the entities are not reflected in the OWL file due to language feature limitations. For instance, regardless of whether one commits to some ideal of General Extensional Mereology with a connection predicate in one ontology vs. the Kuratowski fragment for GEMT (KGEMT) in the other foundational ontology, the same subset of object properties and object property expressions are represented in OWL [28].

The outcomes of the ontology mediation are based on a combination of ontological analysis and on the formalisations of the foundational ontologies. There were inconsistencies of alignments, including some that one would consider well-established ontologically, such as a mathematical set. It is not our aim to solve such inconsistencies; instead, ROMULUS serves as a systematic foundation for such an investigation, and is a starting point for deeper ontological analyses by philosophers and ontologists. If required by such investigations, additional features could be added to ROMULUS, such as a collaborative wiki or a forum page for each ‘alignment with issues’ where matter can be discussed.

7.3 Revisiting FO interoperability for conceptual data models

Instead of the cumbersome manual analyses that were described in the examples in Section 2.1, we now can conduct a quick look-up in ROMULUS.

Let us consider again amounts of matter from Example 2 in Section 2.1. Ontologists hoping to interchange from DOLCE’s Amount-of-matter to an aligned entity in a different foundational ontology can perform a quick search for ‘matter’ in ROMULUS. Several search results using the “basic search” facility are shown in Fig. 13, searching over the alignments, the mappings, and the inconsistencies. When searching for inconsistencies, the result also contains a brief explanation why mapping the two entities results in an inconsistency. The case of `dolce:amount-of-matter` with `gfo:Amount_of_substrate` is interesting, for, as in Example 2, one could have been pleased to have found manually an entity that seems similar in both DOLCE and GFO and thereby assume they are interchangeable and equally usable to enhance a conceptual data modelling language. However, put into context of the whole ontology, this is not the case; or: apparently, they are not quite the same after all, for the alignment with `gfo:Amount_of`

Basic Search

Use basic search for finding alignments, mappings and inconsistencies.

Alignments
 Mappings
 Inconsistencies

Alignment results for the term matter.

	Ontology 1	Entity 1	Relation	Ontology 2	Entity 2
1	DOLCE-Lite	http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl#amount-of-matter	Equivalence	GFO	http://www.onto-med.de/ontologies/gfo.owl#Amount_of_substrate
2	DOLCE-Lite	http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl#amount-of-matter	Equivalence	GFO-Basic	http://www.onto-med.de/ontologies/gfo.owl#Amount_of_substrate

Mapping results for the term matter.

	Ontology 1	Entity 1	Relation	Ontology 2	Entity 2
1	DOLCE-Lite	http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl#amount-of-matter	Equivalence	GFO-Basic	http://www.onto-med.de/ontologies/gfo.owl#Amount_of_substrate

Inconsistency results for the term matter.

	Ontology 1	Entity 1	Relation	Inconsistency	Ontology 2	Entity 2
1	DOLCE-Lite	http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl#amount-of-matter	Equivalence	DOLCE's amount-of-matter is a subclass of DOLCE's physical-endurant. DOLCE's physical-endurant is equivalent to GFO's Discrete-Substrate	GFO	http://www.onto-med.de/ontologies/gfo.owl#Amount_of_substrate

Fig. 13 Overview of the basic search option, with Screenshot of a basic search in the alignment results on *matter*, which was then repeated for mappings and inconsistencies.

substrate also appears in the inconsistencies. Those problematic axioms are absent from the GFO-basic ontology and there it is mappable. It also shows that it is important to have the ontology name in the results, because if the ontologies and their modules are well-designed, the same entity has the same URI, even though they reside in different ontology files (in casu, in *gfo.owl* and in *gfo-basic.owl*). Whether one should use the mapping is a separate decision.

Recalling Example 1 (Section 2.1) regarding *Quality*, and if one is willing to be slightly lenient on the particular versus universal issue, then the foundational ontologies' version of attributes can be matched among

all three foundational ontologies, using the mappings among *Quality* and *Property*. Observe, though, that some mappings are one of subsumption, not equivalence. The "advanced search" option will let one specify also the mapping relation, and further information is available in the database on the mediation, such as whether it was done manually or automatically, and if the latter, with which tool.

Perhaps a modeller would like to use only those entities that are shared among the included foundational ontologies. This information is available in ROMULUS as well, in the form of a 'foundational foundational' ontology (*FFO.owl*). It contains only 3-dimensional en-

tity, 4-dimensional entity, property, and spatial region, which are the ontological versions of a conceptual data model's class, relationship, attribute and a subset of the possible values for the attributes. For any refinement of a characterisation of the entities in a conceptual data modelling language, one would have to commit to a particular foundational ontology (or none).

ROMULUS contains this, and more, information to aid both the ontological investigations and the ontology-oriented conceptual data modeller, such that one does not have to perform the painstaking analysis of the ontologies themselves anymore, but have this readily available with the pairwise mapped, online, ontologies.

8 Conclusions

We presented a core step in the direction of addressing interoperability issues with the Repository of Ontologies for MULTiple USEs, ROMULUS, software infrastructure. This is the first online library of machine-processable, modularised, aligned, and logic-based merged foundational ontologies. In addition to the typical features of a model repository, ROMULUS has a foundational ontology recommender covering features of six foundational ontologies that is integrated with ROMULUS' features and it has tailor-made modules for easier reuse. Most important for the actual ontology-driven conceptual data modelling, are its features and site content with a catalogue of interesting mappable and non-mappable elements among the BFO, GFO and DOLCE foundational ontologies, and the pairwise machine-processable mapped ontologies.

We are currently working on the preliminary user evaluation of the alignments, which is available in ROMULUS already. We hope to gather submissions for mappings between foundational ontologies from the community page which will be validated and included in the repository. Also, we hope to gather sufficient voluntarily saved ontology selections to analyse them and find patterns in selection criteria.

References

- Protégé. <http://protege.stanford.edu/>. Accessed on 09/03/2014
- Baclawski, K., Schneider, T.: The open ontology repository initiative: Requirements and research challenges. In: Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge, *CEUR-WS*, vol. 514 (2009). 25 October, Washington DC, USA
- Borgo, S.: Goals of modularity: A voice from the foundational viewpoint. In: Proceedings of the Fifth Workshop on Modular Ontologies (WoMO'11), *Frontiers in Artificial Intelligence and Applications*, vol. 230, pp. 1–6. IOS Press (2011). Ljubljana, Slovenia, August
- Camara, G., Desprès, S., Djedidi, R., Lo, M.: Building a schistosomiasis process ontology for an epidemiological monitoring system. In: Innovations in Intelligent Machines-4 - Recent Advances in Knowledge Engineering, *Studies in Computational Intelligence*, vol. 514, pp. 75–99. Springer (2014)
- Campbell, L.J., Halpin, T.A., Proper, H.A.: Conceptual schemas with abstractions: Making flat conceptual schemas more comprehensible. *Data & Knowledge Engineering* **20**(1), 39–85 (1996)
- Cowell, L.G., Smith, B.: Infectious disease ontology. In: V. Sintchenko (ed.) *Infectious Disease Informatics*, pp. 373–395. Springer New York (2010)
- Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: Extracting modules from ontologies. In: Proceedings of the 16th International World Wide Web Conference (WWW'07). ACM (2007). May 8–12, Banff, Alberta, Canada
- Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research* **31**, 273–318 (2008)
- Cuzzocrea, A.: Combining multidimensional user models and knowledge representation and management techniques for making web services knowledge-aware. *Web Intelligence and Agent Systems* **4**(3), 289–312 (2006)
- Cuzzocrea, A., Saccà, D., Ullman, J.D.: Big data: a research agenda. In: Proceedings of the 17th International Database Engineering & Applications Symposium (IDEAS'13), pp. 198–203. ACM Conference Proceedings (2013)
- d'Aquin, M., Schlicht, A., Stuckenschmidt, H., Sabou, M.: Criteria and evaluation for ontology modularization techniques. In: *Modular Ontologies, LNCS*, vol. 5445, pp. 67–89. Springer (2009)
- Eapen, B.R.: ONTOderm—a domain ontology for dermatology. *Dermatology Online Journal* **14**(6) (2008)
- Fiorini, S.R., Abel, M., Scherer, C.M.: An approach for grounding ontologies in raw data using foundational ontology. *Information Systems* **38**(5), 784 – 799 (2013)
- Grüniger, M.: COLORE: Common logic ontology repository. <http://ontolog.cim3.net/file/work/00R-Ontolog-Panel/2009-08-06Ontology-%?Repository-Research-Issues/Colore--MichaelGruninger20090806.pdf>. (2009)
- Guarino, N.: Formal ontology and information systems. In: Proceedings of Formal Ontology in Information Systems (FOIS'98). Amsterdam: IOS Press (1998). Trento, Italy, 6–8 June
- Guarino, N., Guizzardi, G.: In the defense of ontological foundations for conceptual modeling. *Scandinavian Journal of Information Systems* **18**(1), 9p (2006)
- Guizzardi, G.: On the representation of quantities and their parts in conceptual modeling. In: Proceedings of 6th International conference on Formal Ontology in Information Systems (FOIS'10), *Frontiers in Artificial Intelligence and Applications*. IOS Press (2010). Toronto, Canada, May 11–14
- Guizzardi, G., Wagner, G.: Using the unified foundational ontology (UFO) as a foundation for general conceptual modeling languages. In: *Theory and Applications of Ontology: Computer Applications*, pp. 175–196. Springer (2010)

19. Guizzardi, G., Wagner, G., de Almeida Falbo, R., Guizzardi, R.S.S., Almeida, J.P.A.: Towards ontological foundations for the conceptual modeling of events. In: 32nd International Conference on Conceptual Modeling (ER'13), *LNCS*, vol. 8217, pp. 327–341. Springer (2013). 11-13 November, 2013, Hong Kong
20. Hartmann, J., Sure, Y., Haase, P., Palma, R., del Carmen Suárez-Figueroa, M.: OMV - Ontology Metadata Vocabulary. In: *Ontology Patterns for the Semantic Web (OPSW) (2005)*. Galway, Ireland, November
21. Herre, H.: General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In: *Theory and Applications of Ontology: Computer Applications*, chap. 14, pp. 297–345. Springer, Heidelberg (2010)
22. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca Grau, B., Hendler, J.A.: Swoop: A web ontology editing browser. *Journal of Web Semantics* 4(2), 144–153 (2006)
23. Kazakov, Y., Krötzsch, M., Simancik, F.: ELK reasoner: Architecture and evaluation. In: *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE'12)*, *CEUR-WS*, vol. 858 (2012). July 1st, Manchester, UK
24. Keet, C.M.: Enhancing comprehension of ontologies and conceptual models through abstractions. In: 10th Congress of the Italian Association for Artificial Intelligence (AI*IA 2007), *LNAI*, vol. 4733, pp. 814–822. Springer Verlag (2007). Rome, September 10-13, 2007
25. Keet, C.M.: Ontology-driven formal conceptual data modeling for biological data analysis. In: *Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data*, chap. 6, pp. 129–154. Wiley (2013)
26. Keet, C.M., Artale, A.: Representing and reasoning over a taxonomy of part-whole relations. *Applied Ontology – Special issue on Ontological Foundations for Conceptual Modeling* 3(1-2), 91–110 (2008)
27. Keet, C.M., d'Amato, C., Khan, Z., Lawrynowicz, A.: Exploring reasoning with the DMOP ontology. In: 3rd Workshop on Ontology Reasoner Evaluation (ORE'14), *CEUR Workshop Proceedings*, pp. 64–70. CEUR-WS.org (2014). July 1, Vienna, Austria.
28. Keet, C.M., Fernández-Reyes, F.C., Morales-González, A.: Representing mereotopological relations in OWL ontologies with ONTOPARTS. In: *Proceedings of the 9th Extended Semantic Web Conference (ESWC'12)*, *LNCS*, vol. 7295, pp. 240–254. Springer (2012). 29-31 May, Heraklion, Crete, Greece
29. Keet, C.M., Fillottrani, P.R.: Toward an ontology-driven unifying metamodel for UML class diagrams, EER, and ORM2. In: 32nd International Conference on Conceptual Modeling (ER'13), *LNCS*, vol. 8217, pp. 313–326. Springer (2013). 11-13 November, 2013, Hong Kong
30. Khan, Z., Keet, C.M.: ONSET: Automated foundational ontology selection and explanation. In: 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW'12), *LNAI*, vol. 7603, pp. 237–251. Springer (2012). Oct 8-12, Galway, Ireland
31. Khan, Z., Keet, C.M.: Addressing issues in foundational ontology mediation. In: 5th International Conference on Knowledge Engineering and Ontology Development (KEOD'13), pp. 5–16. SCITEPRESS (2013). Vilamoura, Portugal, 19-22 September
32. Khan, Z., Keet, C.M.: Toward semantic interoperability with aligned foundational ontologies in ROMULUS. In: Seventh International Conference on Knowledge Capture (K-CAP'13), p. a27 (poster&demo). ACM (2013). 23-26 June, Banff, Canada
33. Khan, Z., Keet, C.M.: Feasibility of automated foundational ontology interchangeability. In: 19th International Conference on Knowledge Engineering and Knowledge Management (EKAW'14), *LNAI*, vol. 8876, pp. 225–237. Springer (2014). 24-28 November 2014, Linköping, Sweden
34. Khan, Z.C., Keet, C.M.: Toward a framework for ontology modularity. In: *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT'15)*, p. in print. ACM Conference Proceedings (2015). 28-30 September 2015, Stellenbosch, South Africa
35. Konev, B., Lutz, C., Walther, D., Wolter, F.: Semantic modularity and module extraction in description logics. In: *Proceedings of 18th European Conference on Artificial Intelligence (ECAI'08)*, *Frontiers in Artificial Intelligence and Applications*, vol. 178, pp. 55–59. IOS Press (2008). Patras, Greece, July 21-25
36. Konev, B., Lutz, C., Walther, D., Wolter, F.: Formal properties of modularisation. In: *Modular Ontologies*, *LNCS*, vol. 5445, pp. 25–66. Springer (2009)
37. Lange, C., Mossakowski, T., Kutz, O., Galinski, C., Grüninger, M., Vale, D.C.: The distributed ontology language (DOL): Use cases, syntax, and extensibility. *CoRR abs/1208.0293* (2012)
38. Loebe, F.: Requirements for logical modules. In: *Proceedings of the Workshop on Modular Ontologies (WoMO'06)*, *CEUR-WS*, vol. 232 (2006). November 5, Athens, Georgia, USA
39. Martínez Ferrandis, A.M., Pastor López, O., Guizzardi, G.: Applying the principles of an ontology-based approach to a conceptual schema of human genome. In: 32nd International Conference on Conceptual Modeling (ER'13), *LNCS*, vol. 8217, pp. 471–478. Springer (2013). 11-13 November, Hong Kong
40. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: *Ontology library*. Wonder-Web Deliverable D18 (ver. 1.0, 31-12-2003). (2003). [Http://wonderweb.semanticweb.org](http://wonderweb.semanticweb.org)
41. Mizoguchi, R.: YAMATO: yet another more advanced top-level ontology. In: *Proceedings of the Sixth Australasian Ontology Workshop, CRPIT*, pp. 1–16 (2010). December 10, Adelaide, Australia
42. Mossakowski, T., Lange, C., Kutz, O.: Three semantics for the core of the Distributed Ontology Language. In: *Proceedings of the 7th International Conference on Formal Ontology in Information Systems (FOIS'12)*, *Frontiers in Artificial Intelligence and Applications*. IOS Press (2012). 24-27 July, Graz, Austria
43. Niles, I., Pease, A.: Towards a standard upper ontology. In: *Proceedings of the Second International Conference on Formal Ontology in Information Systems (FOIS'01)*. IOS Press (2001). Ogunquit, Maine, Oct. 17-19
44. Parent, C., Spaccaprieta, S.: An overview of modularity. In: *Modular Ontologies*, *LNCS*, vol. 5445, chap. 2, pp. 5–23. Springer (2009)
45. Seyed, A.P.: BFO/DOLCE primitive relation comparison. In: *The 12th Annual Bio-Ontologies Meeting (2009)*. Stockholm, Sweden, 28 June 2009
46. Temal, L., Rosier, A., Dameron, O., Burgun, A.: Mapping BFO and DOLCE. *Studies in Health Technology and Informatics* 160(Pt 2), 1065–9 (2010)
47. Thomas, H., Brennan, R., O'Sullivan, D.: Using the OM2R meta-data model for ontology mapping reuse for the ontology alignment challenge - a case study. In: *Proceedings of the 7th International Workshop on Ontology*

- Matching (OM'12), *CEUR-WS*, vol. 946 (2012). Boston, MA, USA, November 11
48. Tudorache, T., Vendetti, J., Noy, N.F.: Web-Protégé: A lightweight OWL ontology editor for the web. In: Proceedings of the Fifth Workshop on OWL: Experiences and Directions, *CEUR-WS*, vol. 432. CEUR-WS.org (2008). Karlsruhe, Germany, October 26-27
 49. Vale, D.C., et al.: The TONES ontology repository. <http://owl.cs.manchester.ac.uk/repository/browser>. Accessed on 14/13/2014
 50. Whetzel, P.L., Noy, N.F., Shah, N.H., Alexander, P.R., Nyulas, C., Tudorache, T., Musen, M.A.: BioPortal: enhanced functionality via new web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Research* **39**(Web-Server-Issue), 541–545 (2011)
 51. Williams, S., Third, A., Power, R.: Levels of organisation in ontology verbalisation. In: The 13th European Workshop on Natural Language Generation, pp. 158–163. Association for Computational Linguistics (2011). Nancy, France, 28-30 September 2011