

Using Philosophy to Improve the Coherence and Interoperability of Applications Ontologies: A Field Report on the Collaboration of IFOMIS and L&C

Jonathan Simon, Matthew Fielding, and Barry Smith, PhD.

Institute for Formal Ontology and Medical Information Science, University of Leipzig,
Härtelstrasse 16, 04107 Leipzig, Germany
jonathan.simon@ifomis.uni-leipzig.de

Abstract. The collaboration of Language and Computing (L&C) and the Institute for Formal Ontology and Medical Information Science (IFOMIS) is guided by the hypothesis that quality constraints on ontologies for software application purposes closely parallel the constraints salient to the design of sound philosophical theories. The extent of this parallel has been poorly appreciated in the informatics community, and it turns out that importing the benefits of philosophical insight and methodology into application domains yields a variety of improvements. L&C's LinKBase® is one of the world's largest medical domain ontologies. Its current primary use pertains to natural language processing applications, but it also supports intelligent navigation through a range of structured medical and bioinformatics information resources, such as SNOMED-CT, Swiss-Prot, and the Gene Ontology (GO). In this report we discuss how and why philosophical methods improve both the internal coherence of LinKBase®, and its capacity to serve as a translation hub, improving the interoperability of the ontologies through which it navigates.

1 Introduction

We may understand an application ontology as a system of representations of elements of reality, structuring data according to some hierarchy of classes for the purpose of managing and manipulating that data, and supporting interoperability of various resources in automatic fashion. We may understand a philosophical ontology as a system of representations of elements of reality structured according to some hierarchy for the purposes of better understanding and relating those elements of reality to one another. These two forms of ontology can in principle support each other. The principal distinction is the demand, crucial in philosophical circles, that an ontology be maximally comprehensive. The philosopher strives for logical rigour, which means that she is not free to ignore irrelevant or rare counterexamples to her general schema. Such a demand is not present in many application ontologies, where the goal-driven context tends to encourage a view of such perfectionism as excessive and costly. Rather ad hoc algorithms are used which are designed to protect the system against counterexamples under given externally determined local conditions. More and more, however, researchers are coming to realize that this quick-fix methodology has not

fulfilled its promise of bringing about interoperability between information resources, and that it has indeed hindered adaptability of preexisting systems to handle new applications, and to support new software.

As researchers from various fields – in the medical domain for example in fields such as natural language processing, clinical trials management, genetics research, anatomy representation and visualization – struggle with the same set of issues, they find themselves unknowingly appealing to the very same principles and methodologies that have driven philosophical research for thousands of years. The more global and flexible an application ontology strives to be, the more general are the data sets it must be prepared to manage, and the more it becomes possible to establish the beginning of an isomorphism between the data sets relevant to the domain in question, and the elements of reality represented by a maximally comprehensive philosophical ontology of that domain. Where the application ontologist evolves and tests his system in response to the cases actually presented by new data, the philosophical ontologist can evolve and test her system according to the methodology of the *Gedankenexperiment*, the practice of imagining possible scenarios which testify to the inadequacy of an existing representation. The thought experiments of ontologically-minded philosophers such as Aristotle, Brentano, Husserl, and Ingarden have in fact proven to be astonishingly prescient in anticipating the problems faced by application ontologies when new types of data need to be dealt with, and the responses these philosophers have suggested sometimes parallel the optimal revisions available to application ontologists in the relevant cases. [1],[2]

The hypothesis which drives the collaboration between the commercial enterprise Language and Computing (L&C) and the academic research group IFOMIS, the Institute for Formal Ontology and Medical Information Science, is that such parallels should be pushed as a matter of principle, and that the construction of an application ontology based on philosophical principles can yield considerable practical benefits. The procedure, broadly speaking, has been the callibration of LinKBase®, L&C's application ontology, in conformity with BFO, the philosophical ontology developed by IFOMIS. This callibration has already yielded positive benefits along two dimensions: 1) improving LinKBase®'s capacity to model certain types of data for the purposes of L&C's software applications, and 2) improving LinKBase®'s capacity to serve as a translation hub for ontologies like GO and SNOMED-CT by enabling the development of the mapping software MaDBoKs. In what follows we discuss these improvements, emphasizing the philosophical nature of the innovations which enable them, and drawing conclusions for the value of philosophical methodology in the advance of information systems. [3]

2 Methods

2.1 LinKBase® and BFO

LinKBase® is a biomedical domain ontology that has been designed to integrate terminologies and databases with applications designed for natural language processing and information retrieval. The ontology contains 543 different relations (linktypes),

divided into different groups, including spatial, temporal and process-related link types. LinKBase® currently contains over 2,000,000 medical concepts organized in a graph with over 5,300,000 link type instantiations. Both concepts and links are language independent, but they are cross-referenced to about 3,000,000 terms in various languages. LinKBase® provides a central hub with fixed structured definitions into which external medical terminologies and databases may be embedded. This task turns out to be a complex endeavor, not least because the different terminologies or databases that are to be integrated are often internally and mutually inconsistent. Yet, as all these terminologies must essentially speak about the same reality, there is a common thread that runs through them and the LinKBase® methodology is based on the idea that it is possible to integrate them precisely by reference to those basic categorical distinctions that are common to them all.

Basic Formal Ontology (BFO) is a philosophically inspired top-level ontology which provides a coherent, unified understanding of these basic distinctions and which is currently being implemented as a top-level open source backbone ontology for LinKBase®. [4] BFO incorporates theories of continuants and occurrents, mereology, mereotopology, universals and particulars, biological classes (natural kinds) and instantiations, and of granular partitions, as well as respecting the more general demands on good ontology recognized by the philosophical community. [5] BFO is thus ideal as a framework for mapping external ontologies, terminologies, and databases onto LinKBase® in a way that is designed to provide for successful integration, and as a useful guide for the future algorithm development that will allow for cross-ontology navigation. The core of BFO is expressed as a simple is-a tree structure, with which is associated a more comprehensive first-order formalism, also available in a KIF representation. [6] In its logical manifestation, the richness of the BFO theory is exploited to guide changes and adaptations of the LinKBase® system. BFO is the result of collaboration among philosophers, linguists, computer scientists and physicians, and is currently being extended to a top-level formal ontology of biomedical categories such as function, site, system, anatomic structure, and so on.

2.2 First-Order Standardization

As ontologies and terminologies expand and are integrated together, it is natural that consistency will become increasingly difficult to maintain. One cause of this difficulty lies in the many ambiguities and inconsistencies that result from the lack of a standard unified framework for understanding those basic relations that structure our reality. The BFO formal ontology provides application ontologies with a set of standardized, first-order definitions for these ontological elements, definitions which can be exploited by reasoning applications, including applications designed for natural language understanding. By disambiguating the ontological structures underlying those informal definitions currently used, which characteristically fall below acceptable standards of formal precision, these formalizations can aid in the passage of domain knowledge between users and software agents, and thus improve coherence and adaptability in and between ontologies. [7]

The resultant standardization reflects an implementation of philosophical rigor along two dimensions. First, it establishes internal consistency on the basis of precise

analyses of the concepts involved. Ontologies such as LinKBase® (as well as SNOMED and GO) are viewed as object languages with a certain “surface structure.” They consist of systems of concepts joined together in binary relations such as is-a and part-of. For the most part however, these relations and concepts are given only in natural language and in a form that leads to various characteristic ambiguities. Thus, the project of defining a unique deep structure to which every such concept, relation, and axiom can be mapped requires sound conceptual analysis. The standardization effort gives us a methodology with which to identify and repair internal inconsistencies and ambiguities in LinKBase® and other ontologies.

The second dimension of rigor requires the use of the standard first-order logical language in which the concepts of BFO are defined and axiomatized. In this way the rigor of the BFO classification system is imported into an ontology from the outside. This importation is meta-ontological, in the sense that changes are not made directly within the external ontology itself; rather, their place in the BFO re-articulated domain ontology, in this case LinKBase®, is marked via an external mapping algorithm in a way that provides the degree of consistency required to navigate between different third-party ontologies.

The standardization on concepts, relations and axioms of LinKBase® runs as follows:

- 1) For every concept C , the definition consists in a mapping to a pair: <the class named by C , the extension of the class named by C >
- 2) For every relation $R(X,Y)$, the definition consists in a mapping to a logical formula of the following form: For all x such that x is in the extension of the class named by ‘ X ’, there is a y such that y is an element in the extension of the class named by ‘ Y ’, and $R^*(x,y)$ (where R^* is a relation in the formal language of BFO, for example part-of, defined as a relation between individuals, including those individuals which are instances of the classes with which we began).

Axioms, which are essentially instantiated relations, are defined by a mapping similar to the definition of relation presented above, differing only in that the variables are replaced by specific concepts within the ontology.

In the remainder of this essay we seek to accomplish two goals. We first examine ways in which the philosophical insights afforded by this standardization have allowed us to understand and resolve modelling errors within the LinKBase® ontology. We then discuss the way in which the BFO standardization has assisted in the effort of ontology integration in the biomedical domain.

3 Results

3.1 Resolving Ambiguities and Modelling Conflicts in LinKBase®

3.11 Objects and Processes in LinKBase®

In philosophical circles it is well understood that the universe accessible to our everyday cognition contains two types of entities that relate differently to time. There are on the one hand objects, such as tables, chairs, countries, and people. These entities are said to *endure* through time, which means that they do not have temporal parts but

are rather wholly present at every moment in which they exist. On the other hand are processes like brain surgeries, heart attacks, lives. These are said to *perdure* through time, which means that they do have temporal parts, such as the first half of the surgery, the last phase of the heart attack, one's childhood. This distinction is not adequately made in existing application ontologies and taxonomies. In particular when the ultimate tribunal for those ontologies are natural language practices, it becomes very important to identify the ambiguity in terms like 'injury', 'dilation', and 'dislocation'. For each of these terms in fact corresponds to two distinct concepts. We speak both of an injury as a perdurant ('when did that injury occur?') and as an endurant ('that injury looks terrible'). Likewise with kinds of injuries, like dislocations: 'The dislocation of his shoulder occurred yesterday' vs. 'The doctor reduced the dislocation.' Indeed, in the medical domain it is commonplace for a sort of process and the state resulting from that process to share a name.

'Dilation' may stand for the process of dilation, i.e. of becoming broader: 'Once in place, a small balloon tip is inflated for a few seconds to *dilate* the artery.' Or, it may stand for the dilated, broadened structure: '*Dilation* of the posterior mitral ring was corrected.'

Here the philosophical distinction between endurants and perdurants allows us to maintain the separation of concepts which would otherwise be, and in standard medical terminologies often are, conflated. By implementing this distinction into the LinKBase® top level, we have been able to recognize these instances of homonymy when they appear. We thereby avoid a range of modeling errors that emerge in standard systems. [8],[9]

3.12 Absences in LinKBase®

It is a tenet of contemporary philosophy that absences are not entities in their own right, but rather, precisely, the absences of entities. Yet medical ontologies must represent natural medical language concepts like 'absence of bacteriuria (bacteria in the urine)', and 'sputum without blood'. Further, though less common, medical texts may feature reference to absences without a specified location of absence, because the location is determined by context.

The straightforward approach, and the approach that LinKBase® formerly used, violated the philosophical tenet mentioned above, and construed absences as special kinds of entities, called 'processes of absence'. With this approach, it was necessary to provide further specification of the processes in question. What kind of process is an absence? What is its duration? Who are its participants? How do we know when two descriptions of absences actually refer to the same entity?

Processes are perdurants, entities located in spacetime. They thus have boundaries, volumes, and locations ('the surgery took place in the operating room'). An adequate inference engine will know various things about such bounded objects: it will know, for example, that if the boundary of object x is different from the boundary of object y, then x cannot be the same object as y, and so on. In a natural language data extraction application, information about the boundary of an absence might be specified via a description like 'an absence *in the liver*.'

Philosophical scrutiny (one of whose functions is to test the adaptability of an ontology framework by demanding responses to creative counterexamples) tells us that the treatment of absences as processes is unstable. A reasoning engine attempting to

handle and infer information about absences so construed runs the risk of deriving contradictions. This possibility arises when we need to establish whether differently described absences are identical. ‘The book was absent from my apartment’ and ‘The book was absent from my bedroom’ seem to refer to the same absence. However, as soon as we instruct our inference engine to consider the two absences here described as identical, we will encounter inconsistency. For the system will record both that the absence has as boundary: *my apartment*, and that it has as boundary: *my room*. But this is a contradiction, since of course $x = y$ implies $\text{boundary_of}(x) = \text{boundary_of}(y)$. How, then, should absences be treated in a more philosophically adequate framework?

Another tenet of philosophy is: distinguish the particular from the universal. When we say ‘There is an absence of bacteria in the patient’s urine’ we clearly are not saying *of* the bacteria in the urine, that *it* is not there. Rather, we are saying *of the universal*: bacteria, that *it* has no *instances* in the patient’s urine. Following this intuition, LinKBase®’s current modeling eliminates concepts of absence themselves. Rather, relations of absence (like: the absence of bacteria in the urine) are construed as relations between the relevant bacteria concept, and the urine concept, but here it is the universal bacteria that is involved: ‘If x is the bacteria universal, and y is an instance of urine, then x has no instance located in y .’ This technique allows us to make inferences very naturally that would be artificial and error prone on the basis of the absences-as-entities model. We no longer need to answer the question whether the absence of the book from my apartment is the same absence as that of the book from my room. Rather we may naturally infer that there is an absence of the book from my room, given that there is an absence of the book from my apartment. This will follow from our general knowledge of location and parthood.

Along with improving our reasoning power, this solution improves our representation structure, rendering applications involving absences more elegant and simple. The old representation of absences as processes blocked us from directly linking two entities where one entity is “absent in” the other entity. It forced, rather, the creation of a third concept: the process of “absence of entity” which related the two.

By representing absence in terms of universals and non-instantiation we avoid the need to create this third concept, and reduce the distance between the related concepts to one relation instead of two. (E.g. the concept “sputum without blood” can be represented with a direct link to the concept “blood”, which will be interpreted formally as: ‘The blood universal has no instance located in (the patient’s) sputum’.) The distance between concepts, and between links on parent-child trees is relevant to many LinK-Base® applications. [10]

3.2 How Philosophy Engenders Interoperability: GO and MaDBoKs

3.21 Objects and Processes within the Gene Ontology

The Gene Ontology (GO) is divided into three disjoint hierarchies: the *cellular component*, *biological processes*, and *molecular function* ontologies. [11] The first, equivalent to anatomy in the medical domain, is an ontology of endurants. It allows users to access the physical structure with which a gene or gene product is associated. A biological process, on the other hand, is defined in GO as ‘a phenomenon marked

by changes that lead to a particular result, mediated by one or more gene products.’ This ontology is therefore a hierarchy of occurrents.

There are however some confusions over the role and nature of GO’s molecular function hierarchy. While GO defines molecular function as ‘the action characteristic of a gene product,’ what biologists characteristically assert about functions makes it clear that functions do not occur, but rather endure; the function of a gene or gene product exists identically for as long as its bearer exists and it is present at all times, even if that function is never realized. Even mutant genes retain their function. Thus for example, “signal transducer activity” remains the function of the EPO_HUMAN protein even when the latter is incapable of performing the signal transduction process.

Molecular functions and biological processes are obviously closely related. The function “signal transducer activity” certainly *involves* performing “signal transduction” in some sense; yet in GO this relationship is undefined. The authors of GO have attempted to clarify the matter by stating, ‘a biological process is accomplished via one or more ordered assemblies of molecular functions,’ in order to suggest that the relation is one of agency. Here, functions *initiate* biological processes, but this would suggest that the one stands to the other in a relation of parthood, which GO on the other hand explicitly rules out. For GO’s authors insist, correctly in our view, that parthood only holds between entities of the same hierarchy. So long as the associated relations continue to conflate the distinct categories of function and process within the ontology, however, architectural flaws in GO will continue to constrain the sorts of reasoning systems which can support. [12]

3.22 MaDBoKS: Philosophically Inspired Ontology Integration

The Mapping Databases onto Knowledge Systems tool (or MaDBoKS) is an extension of the LinkFactory® ontology management system that administers and generates mappings from external databases such as GO or Swiss-Prot onto LinKBase®. This mapping mediates the data contained in the external database in a manner that expands the hub ontology, leaving the structure of the foreign ontology untouched. The MaDBoKS system is designed in such a way that all implicit and explicit relationships between data from the different databases are mapped to the hub ontology. Administration of the mapping mediates the data contained in the different databases in such a way that it is associated with ontological information and the ontology is thereby virtually expanded with the data and relations from the external sources. In this manner we are able to navigate across problematic definitions and relations within an external database using the BFO standardization as translation mechanism.

We now discuss how this works in the case of GO. We first carefully investigated the top-layer categories of the three GO sub-domains that act as our gateway between the LinKBase® concepts and the remaining terms in GO. We identified the more general concepts of GO in LinKBase® and created new concepts in those cases where suitable equivalents were not already recognized. In this way we were able to relate GO’s molecular function hierarchy to the two other GO hierarchies by integrating all three simultaneously into BFO.

In the case of the EPO_HUMAN protein example mentioned earlier, we established that by mirroring BFO defined structures, LinKBase® is able to appropriate this example and model the associated relations with an improved degree of clarity.

The connection between a protein and its function is captured in LinKBase® by a “has-function” relation, and the connection between a function and its corresponding processes is captured by the LinKBase® “realization” relation. The former reflects the relation between a substance and its function, and the latter that between a function and its expression or actualization. Clearly, this latter relation is skew to the whole/part relation, which is properly left exclusive to each hierarchy.

In this manner not only is GO consistently mapped to LinKBase®, but the expressiveness of GO itself has been expanded without any major alterations required in its core structure. [13]

4 Concluding Remarks

It is a tangled web we weave when we seek to create application ontologies without a basis in philosophically sound formal theories. The BFO formalism structuring LinKBase® yields clean data, improves the efficiency of LinKBase®’s own software applications, and supports the integration (and thereby the untangling) of data from different external data sources in a transparent way. It captures the intended semantics of the database terms, and filters out erroneous synonyms and other errors.

Support from the Wolfgang Paul Program of the Alexander von Humboldt Foundation is gratefully acknowledged.

References

1. Flett A, Dos Santos M, Ceusters W.: Some Ontology Engineering Procedures and their Supporting Technologies. EKAW2002 (2003)
2. Smith B., Rosse C.: The Role of Foundational Relations in the Alignment of Biomedical Ontologies. <http://ontology.buffalo.edu/medo/isa.doc>
3. Montayne F, Flanagan J.: Formal Ontology: The Foundation for Natural Language Processing <http://www.landcglobal.com> (2003)
4. Smith B.: Basic Formal Ontology. <http://ontology.buffalo.edu/bfo> (2002)
5. Smith, B.: Mereotopology: A Theory of Parts and Boundaries. *Data & Knowledge Engineering* (1996) 20:287-303
6. Anderson, W. and Grenon, P.: KIF Axiomatization of BFO. IFOMIS Report 01/04, <http://www.ifomis.uni-leipzig.de> (2004)
7. Smith B., Köhler J., Kumar A.: On the Application of Formal Principles to Life Science Data: A Case Study in the Gene Ontology. http://ontology.buffalo.edu/medo/Database_Integration.pdf
8. Ceusters W., Smith B., Kumar A., Dhaen C.: Mistakes in Medical Ontologies: Where Do They Come From and How Can They Be Detected? In Pisanelli D. ed., *Ontologies in Medicine: Proceedings of the Workshop on Medical Ontologies, Rome, October 2003* Amsterdam, IOS Press, forthcoming
9. Kumar, A., Schulze-Kremer, S.: Revising the UMLS Semantic Network. *Proceedings of MedInfo, September 7-11 2004*. forthcoming
10. Van Geyt L, Martens P, Terzic B, Flanagan J.: Get More Out of Your Unstructured Medical Documents. <http://www.landcglobal.com> (2002)
11. GO (Gene Ontology General Documentation) <http://www.geneontology.org/doc/GO.doc.html>
12. Smith B., Williams J., Schulze-Kremer, S.: The Ontology of the Gene Ontology. Forthcoming in the *Proceedings of AMIA 2003*
13. Vershelde J.L., Dos Santos M, Deray T, Smith B, Ceusters W.: Ontology-Assisted Database Integration to Support Natural Language Processing and Biomedical Data-mining. *Journal of Integrative Bioinformatics*, forthcoming