Philosophical Scrutiny for Run-Time Support of Application Ontology Development

Mariana Casella dos Santos[1], Christoffel Dhaen[1], Matthew Fielding[2], Barry Smith, PhD[2, 3] and Werner Ceusters[1]
1. *Language and Computing NV*
Maaltecenter Blok A Derbystraat 79, 9051 St. Denijs Westrem, Belgium
2. *Institute for Formal Ontology and Medical Information Science, University of Leipzig*
Haertelstrasse 16, 04107 Leipzig, Germany
3. Department of Philosophy, University of Buffalo, N4 14260, USA

Abstract. The development and maintenance of domain-specific application ontologies require knowledge input from domain experts who are usually without any formal ontology or AI background. When dealing with large-scale ontologies, for example of the kind with which we are currently familiar in the biomedical spheres, quality assurance becomes important in minimizing modelling mistakes and the application errors which they bring in their wake. In this paper we describe how the upper-level framework BFO (for: Basic Formal Ontology), developed by the Institute for Formal Ontology and Medical Information Science, is being used to provide automatic error detection and run-time modelling support to the development of LinKBase®, a large-scale medical domain ontology developed by Language and Computing NV to serve a range of natural language processing applications.

1 Introduction

Domain-specific application ontologies [1] can be understood as knowledge bases whose elements represent entities of different sorts within particular domains of reality. These elements are hierarchically organized and connected by a network of relationships such as *IS-A* and *Part-of*. Such application ontologies have the goal of storing and structuring data in a way that supports the functionality of software applications and allows for information exchange. An ontology designed to support Natural Language Processing (NLP) applications must consider not only the aspect of language (i.e. how language is construed and used to represent reality), but also the aspect of reality itself (i.e. what language is trying to represent) [2]. These two aspects are often confused by those developing application ontologies, leading to problems when the same expressions can be used to represent different aspects of reality, or when the same reality can be symbolized by different expressions [3, 4].

Application ontologies that serve the purposes of data integration must in addition be in a position to represent the intrinsic semantics of elements present in several distinct external data sources, including other application ontologies [5]. Even where these data sources represent the same particulars in a given domain of reality, their view and understanding of these particulars may be different, and this results in several different and potentially incompatible types of representation, no one of which is true to the reality itself. The chosen form of representation often reflects the needs of the specific applications which the ontology has been designed to support, or they reflect the particular line of thought of particular domain-experts. Distinctive applications call for distinctive types of data structure, and different experts accentuate different aspects of the same reality. Thus a clinician would look at the entity “fever” from the perspective of symptoms and underlying diseases, while a physiologist would consider mainly the body processes and substances involved. In order to integrate such diverse data sources, a way must be found to build an ontology framework that is true to reality and is as far as possible independent from application-dependent forms of representation.

The development of domain-specific ontologies requires the collection and representation of domain-specific data, and this in turn requires input from domain experts. The professional expertise of the latter (a doctor or nurse in the medical domain, of an attorney in the domain of law) characteristically does not include within its scope the skills required for good knowledge representation or for ontology development. The formal and application ontology building expertise of domain modellers is also often limited, which complicates still further the process of development of large-scale domain-specific ontologies. Modellers require extensive training before they are able to edit an ontology in such a way as to maintain its quality. The avoidance of fundamental errors of a sort, which can bring undesired consequences that propagate throughout the system, remains an extremely demanding task [6].

The hypothesis which drives the collaboration between the commercial enterprise Language and Computing (L&C) and the academic research group IFOMIS, the Institute of Formal Ontology and Medical Information Sciences, is that philosophical principles can aid application ontologies in the development of the perspective-independent representation of reality of the sort that is required for sound NLP and data integration applications [7]. Taking philosophical principles as our starting point, we have designed a number of in-built algorithms to assist domain modellers during the process of ontology editing. The algorithms operate by issuing alerts when these principles are contravened. We thus constrain the modelling space in a way that does not require a deep understanding of formal ontology on the part of the domain modeller himself. This paper describes part of the process of integrating BFO, the philosophical upper-level ontology developed by IFOMIS, into LinKBase®, L&C’s medical domain ontology. It also provides examples of how the rigor of BFO has assisted in the maintenance of accuracy and optimization of the modelling process of LinKBase®.

2 Materials and Methods

2.1 LinKBase® and BFO

LinKBase® is a medical domain ontology developed and maintained with the purpose of serving NLP and data integration applications. It consists of approximately 2,100,000 medical concepts hierarchically structured and horizontally connected by 614 operational, formal and linguistic relations (linktypes) on over 5,600,000 linktype instantiations. The ontology is language independent and is related to a lexicon of about 3,600,000 terms in several languages. For purposes of data integration it contains a field labelled “Meta”, in which external ontologies and terminologies are stored, and from which they are mapped towards the central domain ontology. LinKBase® uses binary relations to represent and define the semantics of medical concepts and to provide a central reference point to the elements in several ontologies, terminologies and databases in such a way as to allow cross-mapping between them. Its relationships are designed both to represent the perspective of reality and also to enable calibration with the perspectives conveyed by different languages.

Basic Formal Ontology (BFO) is a philosophically inspired top-level ontology which is designed to provide a coherent, unified understanding of basic categorical distinctions and which is currently being implemented as a top-level open source backbone ontology for

LinKBase®. BFO incorporates theories of endurants and perdurants [8], mereology, topology, universals and particulars, space and time. It also contains modules for dealing with biological classes (natural kinds) and their instantiations [9], and also with granular partitions [10], as well as respecting the more general demands for a sound ontology recognized by the wider philosophical community [11]. We have attempted to demonstrate empirically that BFO is situated to the task of providing a framework for mapping external application ontologies, terminologies, and databases onto a system like LinKBase® in a manner that can facilitate successful integration [12].

The core BFO ontology is expressed as a simple ***IS-A*** tree structure, with which is associated a more comprehensive first-order formalization, also available in a KIF representation in the Wonderweb Library of Foundational Ontologies [13]. In its logical form, the expressiveness of the BFO theory may be exploited to inform models for information integration as well as to help in regimenting the core structure of LinKBase® itself. BFO is the result of a collaboration between philosophers, linguists, informaticians, and physicians, and is currently being extended to a top-level formal ontology of biomedical categories such as function, site, system, and anatomical structure [20].

2.2 The LinkFactory® Ontology Management System

LinkFactory® is a platform-independent ontology management system built with the goal of enabling the development of large and complex language-independent ontologies, and of connecting these with other ontologies and terminologies in a single information network [14]. LinkFactory® stores the data in a relational database. Access to the database is via a set of ontology-related API functions such as “get-children”, “find-path”, “join concepts”, “get terms for concept”, etc. LinkFactory® also contains a server-side component that allows developers to use a standardized API to program applications on top of the database without requiring intimate knowledge of the internal structure of the database. This component is stable, database-independent, and capable of dealing with multiple concurrent users. In order to comply with all the features mentioned above Java chosen as the underlying programming language. The LinkFactory® system consists of two major components, the LinkFactory® Server, and the LinkFactory® Workbench (client-side component). The LinkFactory® Workbench allows the user to browse and model the ontology data. The workbench is a dynamic framework for the LinkFactory® beans. Each bean has its own specific functionality and yields its own specific view on the underlying ontology, and the separate beans can be linked and made to interact with each other by displaying corresponding information on the selected ontological elements. Users can compose their own scenarios of bean configuration according to their modelling focus.

2.3 Ontology Verification Mechanisms

2.3.1 Domain-Range Restrictions

Usually (for example in [15]) ontology management systems do not enforce domain and range restrictions at the modelling stage, but rather leave such restrictions do be dealt with by a reasoning engine where they can be used to deduce the most appropriate classes that an individual (i.e. an instance of a given concept) should instantiate.

Domain and range restrictions are applied on linktypes. The domain-restriction on a linktype limits the individuals which can serve as first term of the corresponding relation to those which fall under the concept specified in the domain-restriction. If a linktype relates a conceptual instance to another conceptual instance, and the linktype has a domain-restriction, then the instance must belong to (i.e. be subsumed by) the concept specified as the domain of the linktype. The range-restriction on a linktype similarly limits the instances of the target-concept to which the given linktype can be applied (i.e. it limits what can serve as the second term of the relation). If a linktype relates a source instance to a target instance, and the linktype has a given concept as its range, then the target instance must instantiate the concept specified in the range-restriction of this linktype.

When taking the hierarchy of linktypes and concepts into account, a global or ontology-wide domain/range restriction on a linktype affects not only the linktype and concepts related therein, but also the subtypes of this linktype and all their corresponding concepts. If there exists a global domain-range restriction on a linktype L with domain C1 and range C2, and L_i is a subtype of L, then every restriction using $L_\mathbf{i}$ on the concept-level must at least fulfil the global restriction. Thus a local restriction such as “all instances of Cx have at least one Li relation to some instance of Cy” can be satisfied only if Cx is a subclass of C1 and Cy is a subclass of C2.

Because the system enforces these restrictions during the modelling-process, the knowledge engineer is warned immediately should either domain or range restrictions be contravened. LinkFactory automatically performs a real-time check whenever the knowledge engineer creates new criteria (i.e. new local restrictions), and issues a warning whenever a restriction is violated. This check covers all creations or modifications of criteria using the non-hierarchical linktypes (i.e. linktypes other than *IS-A*).

When changes are made to the hierarchy (either to its concepts or to its linktypes), this can affect the restrictions previously imposed. Such effects are not checked in real-time; rather, an ontology-wide check can be performed, preferably at regular intervals, to re-check for any violations. While the real-time check triggers any applicable domain-range restrictions that need to be enforced, the ontology-wide check employs the reverse approach, which means that it loads the necessary data from the ontology and checks whether the domain-range restrictions can be applied successfully to the data. The checking methodology has been optimized for large knowledge bases and is able to verify within minutes the satisfaction of global constraints on an ontology with over 500,000 conceptual entities and 400 different types of relationships. Relations between concepts that are not compatible with given global domain-range restrictions are returned to the knowledge engineer for further verification.

2.3.2 Disjoint-restrictions

Disjointness restrictions can be used not only to compute information about instances, but also to aid the construction of the conceptual model that underlies the knowledge base, since a concept that is a subclass of two disjoint concepts could of course never have any instances. Excluding such subclasses can thus help the knowledge engineer to construct a valid model of reality. LinkFactory enforces disjointness by ensuring that, when changes are made to the hierarchy, no concept can be a subclass of two disjoint concepts, and no two concepts can be made disjoint where concepts which they subsume in common and are thus violating this restriction already exist. These checks, too, are performed real-time whenever the modeller tries to enter corresponding changes.

2.4 Method for Integration of LinKBase® with BFO

The medical concepts in LinKBase® are structured under a higher-level domain independent ontology whose concepts represent general categories such as process, object and property, instantiated also in the medical world. Part of our integration effort was to map these higher-level concepts of LinKBase® to the two constituent ontologies, SNAP and SPAN, of BFO [8], representing continuants and occurents respectively. In LinkFactory®, a disjointness constraint (described at *2.3.2*) was imposed upon the SNAP and SPAN categories in BFO, in such a way that is ruled out that any mapped LinKBase® concept should be subsumed simultaneously by different disjointed BFO entities.

The LinKBase® linktypes are organized in a complex hierarchical structure reflecting both the demands of formal ontological realism, and also the need to do justice to aspects of linguistic representation [16]. Properties such as satisfaction of constraint or inference algorithms are inherited through the hierarchy. The integration method maps the linktypes to BFO relations, such as dependence, inherence and also part-whole and other mereotopological relations [17]. As described in *2.3.1* LinkFactory® has embedded within it a check, for each linktype instantiation, that verifies for each linktype compatibility of source (domain) and target (range) concepts according to pre-defined domain-range values. The BFO relations were themselves introduced into LinkFactory® as linktypes, and were given domain-range attributes with values corresponding to the SNAP or SPAN entities of BFO between which they apply. This enables the constraint set at the level of the formal relations of BFO to be inherited by the mapped LinKBase® linktypes and thereby allows the system to detect and prevent linktype instantiations which are inconsistent.

This paper discloses two examples of disjointness between BFO entities, and two examples of domain-range constraints on BFO formal relations, giving examples as to how they are used for run-time modelling support and error prevention. We discuss frequent erroneous ontological assumptions encountered in the modelling of medical domain experts.

3 Results

3.1 Endurant vs. Perdurant Disjointness

At its most basic level the BFO framework draws a distinction between two sorts of entities: endurants (SNAP) and perdurants (SPAN). These two sorts of entities relate differently to time. Endurants are those entities, which, as the name implies, endure through time; they are wholly present at each moment of their existence. Examples of endurants are entities such as tables and chairs, people, operating rooms, cells, and chromosomes. All of these kinds of entities, and all of their parts, maintain their full identity from one moment to the next. Perdurants, on the other hand, are those sorts of entities that are never fully present at any one given moment in time, but instead unfold themselves in successive phases or temporal parts. Entities that perdure are processes or events such as: a morning run, a surgery session, a case of cellularization. Where your arm is a part of you and your hand is a part of your arm, your youth is a part of the process which is your life and your first day at school is a part of your youth. It is important to note that parthood never crosses the SNAP-SPAN boundary; parts of endurants are always endurants; and parts of perdurants themselves always perdure.

The BFO categories of *Endurant* and *Perdurant* were introduced into LinKBase® and made disjoint according to the algorithm described at *2.3.2*. As such, LinKBase® domain concepts cannot be simultaneously children of both of these categories. The LinKBase® domain concepts “process”, and “region of time” were mapped to *Perdurant*. The LinKBase® domain concepts “structure”, and the BFO categories *Dependent Endurant* and *Independent Entity* were mapped to *Endurant*, so that the latter subsumes also all LinKBase® domain concepts mapped to *Dependent Endurant* and *Independent Entity* (section *3.2* below).

Problematic modelling that was brought to light as a result of these changes include for example with the modelling of the concept “furuncle of hand”. Furuncles are boils filled with pus and caused by an infection process. Some external ontologies or terminologies represent “furuncle” from the perspective of the material lesion itself, others from the perspective of the infection that caused it, implying a misuse of the *IS-A* relation. If the modeller, during the process of integration of elements coming from the corresponding external terminologies, attempts to maintain their incompatible hierarchies by entering both these: “*furuncle of hand IS-A furuncle of upper extremity*” and “*furuncle of hand IS-A infection of skin*”, then he will be prevented by the disjointness mechanism. “Furuncle of upper extremity” is modelled as a “lesion”, consequently as a “material entity” and as an *Endurant*, while “infection of skin” is modelled as a “process” and therefore as a *Perdurant*. Since the algorithm does not allow for a concept to inherit from both *Endurant* and *Perdurant*, the introduction of the corresponding error is excluded. This helps the modeller to reflect on the nature of the relationships he is using and to realize that the relation of furuncle to infection is one of causation rather than of subsumption.

3.2 Dependent vs. Independent Entities Disjointness

Some entities, which we call independent, have the ability to exist without the ontological support of other entities; these are entities such as people, tables, or molecules, and they belong always to the SNAP ontology. Other entities, which are every bit as real, require for their existence the existence of the first sort of entities. A morning run needs a runner; a viral infection is dependent on the virus and on the organism infected. All perduring (SPAN) entities require at least one independent entity in which to inhere; in other words, there is no process without a substance to bear it. But there are dependent entities also within the category of endurant entities, for example the function of an organ, which depends on the existence of that organ, or the temperature of the body, which depends on the body.

When introduced into LinkFactory® the BFO formal categories *Dependent Endurant* and *Independent Endurant* of BFO were made disjoint according to the algorithm described at *2.4.2*, thus preventing child concepts of all the concepts directly mapped to these BFO entities from inheriting simultaneously from both of them. The LinKBase® domain concept “property”, which represents qualities and powers, was mapped to *Dependent Endurant*. The LinKBase® domain concepts “conscious thing”, “entity of body” (representing material parts of the body, like the arm or the heart, or spatial, like the abdominal cavity), “material entity” (representing material objects), and “substance” (representing chemical substances such as glucose or protein) were mapped to *Independent Entity*. Some concepts, for example “dendritic” (see bellow) were found to violate the disjointness constraint, and were corrected upon implementation of the check.

Amongst philosophers, the principle of subsumption and the *IS-A* relation are very well understood. But what might seem obvious to some can be severely misunderstood by others, and there are a number of very common confusions manifested by domain-experts is in the assignment of the *IS-A* relation. Other formal relations, such as parthood, inherence or participation are often introduced as if they were *IS-A* relations, in a way which on application of subsumption algorithms and the use of inheritance brings severe errors in its wake. Domain modellers usually understand the real semantics of the entities they are trying to represent; what remains cloudy is the distinction between the formal relations themselves and the knowledge of when to apply them, a proper grasp of which, we believe, requires some minimal philosophical background. The disjointness mechanism based on BFO's categories helps us guide the modeller in this process. Thus the word "dendritic" symbolizes quality of having a shape that is similar to a tree with branches. The domain modeller might want to represent "dendritic" with two simultaneous *IS-A* relations: "*IS-A branch*" and "*IS-A shape state*". The system will not allow this, because a "branch" is a "material entity" and therefore an *Independent Entity*, while "shape state" is a "property" and consequently a *Dependent Endurant.*

3.3 Ontological Relations

At the base of our theory lies the notion of ontological dependence [18]. By this, we understand those relationships that express existential necessity, as a smile is existentially dependent on the one who smiles; a surgery is existentially dependent on a patient as well as on one who performs the surgery. Thus if a smile exists, then we can infer from this that there exists one who is smiling. For our purposes here, we deal with that type of ontological dependence that is one-sided. This relationship holds between categorically similar entities (such as dependent and independent endurants) as well as categorically diverse entities (such as endurants and perdurants) and implies that, while one entity requires the existence of the other for its own existence, this does not hold in the reverse direction. Here we see that, while a surgery requires the existence of a doctor to perform some given surgical procedure, the reverse is certainly not true: a surgeon remains a surgeon even when not performing a surgery.

3.3.1 Involvement Axioms and the Domain-Range Constraint

The relation that we have termed *Involvement* is the most general form of the relation that holds specifically between an enduring substance and perduring process. The basic constraints imposed upon this relation require that one of its bearers be a perdurant and the other an endurant, and that a one-sided relation of dependence also holds between them. This formal constraint was applied, through the algorithmic Domain-Range constraint described in *2.4.1* above, to the formal relation of *Involvement* when the latter was introduced in LinkFactory®. As such all relations that make use of the linktypes mapped to *Involvement* must comply with this constraint in order to be introduced into the LinKBase® domain ontology. A total of forty-nine relations were mapped to *Involvement*, and they represent mainly linguistic variations of this formal relation (i.e. the various different perspectives of the relation between endurants and perdurants captured by language). The linguistic relations are assigned both in the direction from an endurant towards its dependent perdurant (*Involvement*), as also in the direction from the dependent perdurant towards the endurant it depends upon (*Reverse Involvement*). Examples of these linguistic relations are *Has-actee/Is-actee-of*, which relates processes and substances participating passively, from a linguistic perspective, in these processes; *Has-agent/Is-agent-of*, which relates processes and substances participating actively and with agency, from a linguistic perspective, in these processes; *Has-theme/Is-theme-of*, which relates processes of motion and substances participating as the element being moved in these processes.

Ensuring conformity to this Domain-Range constraint for the *Involvement* relation might seem to be a simple task from the philosophical perspective; but at the level of domain modelling the notions are easily confused. A common misunderstanding when building ontologies for NLP purposes is to confuse the essence of an entity in reality with the perspective language takes to describe it. Take for example the entity represented by the term “visual floaters”. In reality this entity is a sensorial process in which a person has the sensation of having spots floating before his eye. The process of “floating” and the substance “spot” are here however simply metaphors of language and do correspond to anything in reality. During the modelling of this entity, which was already classified as a sensorial process and therefore as a perdurant, the Domain-Range restriction ensured that the modeller cannot introduce the relation “*visual floaters - is-theme-of - motion process*” because the is-theme-of linktype requires an endurant as its source. This erroneous association could cause damaging results in applications by allowing, “visual floater” to be understood as a material “spot” and therefore as a lesion of the eye.

In medical terminology it is very common that one and the same term or description is used to symbolize both a process, either intrinsic to the body or performed by a physician, and the structural change in the body that results therefore (e.g. the term “implant” represents both the surgical procedure of implanting as well as the material structure implanted; the term “bypass” represents both the surgical procedure of creating a bypass as well as the structural deviation created thereby). In LinKBase® both the process and the consequent endurant are represented, but modellers find difficulty in understanding the distinction and consequently often relate the concepts erroneously.

The implemented BFO constraint has helped also in this respect, for example in the modelling of the concept “removal of orthopedic implant”. When the modeller searches for the term “implant” he finds concepts representing both the “implanted structure” as well as “implantation procedure”. If the modeller is not aware of the distinction and reasons only based on the terms, then he might try to add the link “*removal of orthopedic implant - has-theme - implantation procedure*”, but he is prevented from doing this by the constraint requiring an endurant as target concept, and by the fact that “implantation procedure” is already classified as a perdurant. Again we prevent hereby errors which would otherwise arise when the system is used by NLP applications, for example on inferences of the kind “the surgical procedure has been removed”.

3.3.2 Inherence Axioms and Domain-Range Constraint

Inherence is the relation that holds in BFO between dependent and independent endurants, for example between properties, powers and functions and the substances upon which they depend. The basic constraint of this relation requires that one of the entities involved be a dependent and the other an independent entity, while both are endurants and the former is one-sidedly dependent on the latter. This formal constraint was applied via the algorithmic Domain-Range constraint described on *2.4.1* to the formal relation of *Inherence* when introduced in LinkFactory®. As such all relations that make use of the linktypes mapped to *Inherence* must comply with the constraint in order to be introduced into the LinKBase® domain ontology.

A total of four relations were mapped to *Inherence*, and they represent the inherence of qualities and powers both as determinables and determinates (e.g. *Has-inherent-state/Is-inherent-state-of*, which relates a determinable to the independent entity upon which it inheres). In LinKBase® these relations are assigned both in the direction from a dependent to an independent endurant (*Inherence*), as well as in the direction from the independent entity to the dependent endurant depending upon it (*Reverse Inherence*).

A problematic issue frequently encountered in domain-specific ontology modelling is the proper identification of dependent entities (as properties, qualities, functions and so on). Especially when the requirements of natural language understanding are involved, it is common to see properties wherever terms play the syntactic role of an adjective. But while words like “increased”, “oriented” or “bilateral” can be syntactically construed as adjectives, which means that they characterize another entity in a sentence they represent not a property but rather a participation in a given process or a projection on a spatial region. The Domain-Range constraint can help us avoid this problem of representation, as in the modelling of a concept like “undiagnosed bleeding”. Even though the concept “undiagnosed” has already erroneously been classified as a property, when a modeller tries to enter the relationship “ *undiagnosed bleeding – has-inherent- state - undiagnosed* ” the system does not allow the insertion because “undiagnosed bleeding” *IS-A* “bleeding”, and consequently a process and a perdurant, and the *Inherence* relation can relate only endurants.

Another confusion often made is between location relations (i.e. the overlap of a substance or process with a specific spatial region) and properties. Certain linguistic constructions also allow terms such as “inferior”, “apical” or “bilateral” to be mistakenly represented as properties. In modelling the concept “anisocoria” (i.e. the condition of having pupils of unequal size), for example, if the modeller tries to enter the relationship “*anisocoria – has-inherent- state - bilateral location*”, the system does not allows it, since “bilateral location” is not a dependent entity but a spatial region and is therefore mapped to an independent entity because the constraint on *Reverse Inherence* requires a dependent entity as target.

Not only ontological mistakes are thereby prevented at run-time, but also the methodology offers the modellers a basis for reflecting on the true ontological aspects of the entities they are trying to represent. The methodology also provides assistance in finding underspecifications that arise when in the course of time, descriptions already present in the ontology need to be refined as more detailed relationships are required [6, 19].

4 Conclusion

In applying the BFO framework to the LinKBase® medical domain ontology we have seen how the adoption of philosophical rigor provides for clarity of representation within application ontologies, allowing these ontologies to grasp more nearly the true perspective upon reality required both for natural language understanding and for the integration of different data sources. The BFO formalism embedded into the system in the form of efficient ontology verification checks provides modelling support and quality control mechanisms in such a way as to facilitate the development of application domain ontologies like LinKBase®. Building and maintaining large-scale domain ontologies according to a formal realist representation still involves a great amount of laborious effort, but the results are of value in providing a sound basis for applications.

The erroneous ontological assumptions encountered in the modelling of medical domain experts here discussed are characteristic of much work in the biomedical informatics field. Hence we believe our experiment has significance far beyond the single case of LinKBase® and its applications.

Acknowledgements

The present paper was written under the auspices of the Language and Computing Research Division along with the Wolfgang Paul Program of the Alexander von Humboldt Foundation, the Network of Excellence in Semantic Interoperability and Data Mining in Biomedicine of the European Union, and the project Forms of Life sponsored by the Volkswagen Foundation.

References

[1] Nicola Guarino. Understanding, Building and Using Ontologies: A Commentary to “Using Explicit Ontologies in KBS Development”. *International Journal of Human and Computer Studies*, 46:293--310, 1997.

[2] Ceusters W, Smith B, Van Mol M. *Using ontology in query answering systems: scenarios, requirements and challenges*. In Bernardi R, Moortgat M (eds) Proceedings of the 2nd CoLogNET-ElsNET Symposium, pages 5-15, Amsterdam, 2003.

[3] Ceusters W, Buekens F, De Moor G, Waagmeester A. *The distinction between linguistic and conceptual semantics in medical terminology and its implications for NLP-based knowledge acquisition.* Meth Inform Med 1998; 37:327-33.

[4] Zajac, R. *“Engineering Ontologies vs. Linguistic Ontologies for Web Applications”*, Proceedings of MAA Conference, Tokyo, 2001.

[5] Ceusters W, B Smith, JM Fielding, “*LinkSuite™: Formally Robust Ontology-Based Data and Information Integration*, in Erhard Rahm (Ed.): Data Integration in the Life Sciences, First International Workshop, DILS 2004, Leipzig, Germany, March 25-26, 2004, Proceedings. Lecture Notes in Computer Science 2994 Springer 2004, ISBN 3-540-21300-7, 124-139 (electronic version)

[6] Alan Flett, Mariana Casella dos Santos, Werner Ceusters. *Some Ontology Engineering Processes and their Supporting Technologies.* Siguença, Spain, October 2002. EKAW 2002

[7] Smith B, Ceusters W. Towards Industrial-Strength Philosophy; How Analytical Ontology Can Help Medical Informatics. *Interdisciplinary Science Reviews*, 2003, vol 28, no 2, 106-111.

[8] Grenon, Pierre and Smith, Barry. SNAP and SPAN: Prolegomena to Geodynamic Ontology. In : Hornsby and Worboys, *Spatial Cognition and Computation*, forthcoming.

[9] Smith, Barry. *The Logic of Biological Classification and the Foundations of Biomedical Ontology*. In: Westerstahl (ed.), Invited Papers from the 10th International Conference in Logic Methodology and the Philosophy of Science, Oviedo, Spain, 2003, forthcoming.

[10] Bittner, Thomas and Smith, Barry. A Theory of Granular Partitions. In: Duckham, Goodchild, and Worboys (eds.), _*Foundations of Geographic Information Science*, pages 117-151.London: Taylor & Francis Books, 2003.

[11] Smith, Barry. Basic Tools of Formal Ontology. In: Nicola Guarino (ed.), *Formal Ontology in Information Systems*, pages 19-28. Washington, DC: IOS Press (Frontiers in Artificial Intelligence and Applications), 1998.

[12] Fielding, James; Simon, Jonathan; Ceusters, Werner; Smith, Barry. *Ontological Theory for Ontological Engineering*. Proceedings of the Ninth International Conference in the Principles of Knowledge Representation and Reasoning, Whistler, Canada, 2004, forthcoming.

[13] WonderWeb Library of Foundational Ontology (http://wonderweb.semanticweb.org).

[14] Werner Ceusters, Peter Martens, Christoffel Dhaen and Boris Terzic. *Linkfactory: an advanced formal ontology management system*. Victoria, Canada October 2001. K-CAP 2001.

[15] Owl Web Ontology Language (http://www.w3.org/TR/2004/REC-owl-semantics/20040210/semantics-all.html)

[16] J. A. Bateman, R. Henschel, and F. Rinaldi. *The generalized upper model 2.0*. Technical report, GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany, 1995.

[17] Smith, Barry. *Mereotopology: A Theory of Parts and Boundaries*. Data mining Engineering, 20: 287-303, 1996.

[18] Barry Smith, and Pierre Grenon. *A Cornucopia of Formal Relations*. Dialectica, forthcoming.

[19] Werner Ceusters, Christoffel Dhaen, Barry Smith: *Context-specific quality assurance in very large and rapidly changing ontologies*, forthcoming.

[20] Basic Formal Ontology (http://ontology.buffalo.edu/bfo/BFO.htm)