EXPERIENCE IN REASONING WITH THE FOUNDATIONAL MODEL OF ANATOMY IN OWL DL

SONGMAO ZHANG [1], OLIVIER BODENREIDER [2], CHRISTINE GOLBREICH [3]

[1] *Institute of Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing, China {smzhang@math.ac.cn}*

[2] *U.S. National Library of Medicine, National Institutes of Health, Bethesda, MD, USA {olivier@nlm.nih.gov}*

[3] *LIM, University Rennes 1, Rennes, France {christine.golbreich@univ-rennes1.fr}*

The objective of this study is to compare description logics (DLs) and frames for representing large-scale biomedical ontologies and reasoning with them. The ontology under investigation is the Foundational Model of Anatomy (FMA). We converted it from its frame-based representation in Protégé into OWL DL. The OWL reasoner Racer helped identify unsatisfiable classes in the FMA. Support for consistency checking is clearly an advantage of using DLs rather than frames. The interest of reclassification was limited, due to the difficulty of defining necessary and sufficient conditions for anatomical entities. The sheer size and complexity of the FMA was also an issue.

1 Introduction

As virtually all other biomedical ontologies relate to them, reference ontologies for core domains such as anatomical entities and small molecules form the backbone of the Semantic Web for Life Sciences. One such ontology is the Foundational Model of Anatomy (FMA). However, existing reference ontologies sometimes need to be adapted to Semantic Web technologies before they can actually contribute to the Semantic Web. Converting ontologies into the formalisms supported by the Semantic Web can also benefit these ontologies as such formalisms enable consistency checking and reasoning support. This study explores the benefits of converting the FMA, a large reference ontology, to the Web Ontology Language OWL.

Biomedical terminologies and ontologies are increasingly taking advantage of Description Logic (DL)-based formalisms in representing knowledge. GALEN [1] and SNOMED Clinical Terms® (SNOMED CT)[2] were both developed in a native DL formalism. Other terminologies have been converted into DL formalism, including the UMLS® Metathesaurus® [1-3] and Semantic Network [4], the Medical Subject Headings (MeSH) [5], the Gene Ontology™ [6] and the National Cancer Institute Thesaurus [7].

[1] http://www.opengalen.org/

[2] http://www.snomed.org/snomedct_txt.html

However, many ontologies developed in the frame paradigm – often with the ontology editor Protégé (e.g., the Foundational Model of Anatomy) – cannot benefit from the reasoning support provided by description logics and cannot directly contribute to the Semantic Web.

While developed out of frame-based structures, description logics provide more precise specification of domain knowledge and enable powerful reasoning support. The most popular description logic formalism is currently the Web Ontology Language (OWL) [8, 9]. Serving as the logical basis for the Semantic Web, OWL is used to formalize a domain, assert properties about individuals and reason about classes and individuals. OWL comes in three flavors (OWL Lite, OWL DL and OWL Full), corresponding to different levels of expressivity (i.e., what knowledge can be represented with the language) and decidability (i.e., whether reasoning support is assured). OWL Full is maximally expressive but undecidable; in contrast, OWL Lite is efficient but has limited expressivity. Based on description rather than predicate logic, OWL DL offers a trade-off between expressivity and decidability. All versions of OWL use the Semantic Web technology RDF (Resource Description Framework) for their syntax.

In previous work, we proposed a method for converting the Foundational Model of Anatomy (FMA) from its original frame-based representation to OWL DL [10, 11]. In addition to the conversion process, this study focuses on the reasoning support enabled by OWL DL for the FMA. Dameron et al. have explored the conversion of the FMA to OWL Full rather than OWL DL [12]. Their goal is to stay as close as possible to the Protégé representation constructs, which is not possible with OWL DL (e.g., representing metaclasses). Beck et al. also transformed the FMA into a description logic-based representation (but not OWL), with special emphasis on the representation of partitive relations ("Structure-Entirety-Part triplets") [13].

This study is composed of two parts: conversion and reasoning. Section 3 presents the conversion rules we established to automatically convert the FMA from its frame-based representation in Protégé into OWL DL, followed by results (section 4) and optimization issues (section 5). A reasoner (Racer) is used to reason over the OWL version of the FMA converted from Protégé. After a brief overview (section 6), we study satisfiability (section 7) and reclassification (section 8). The benefits and limitations of using description logic to model the FMA are discussed in section 9. Examples of FMA classes in OWL DL could not be included in this manuscript, but are available as supplementary material at: `mor.nlm.nih.gov/pubs/supp/2006-psb-sz/`. They are referenced by "Supp x" markers.

2 The Foundational Model of Anatomy

The Foundational Model of Anatomy[3] (FMA) is an evolving ontology that has been under development at the University of Washington since 1994 [14, 15]. The FMA is implemented in Protégé[4], a frame-based ontology editing and knowledge acquisition environment developed at Stanford University [16]. The objective of the FMA is to conceptualize the physical objects and spaces that constitute the human body. 70,169 classes cover the entire range of macroscopic, microscopic, and subcellular canonical anatomy. Additionally, 187 slots are specified and used. Seven of them correspond to partitive relationships (*e.g.*, *CONSTITUTIONAL_PART_OF* and *2D_PAR_ OF*). In canonical anatomy, all partitive relationships have inverses (*e.g.*, *CONSTITUTIONAL_PART* and *2D_PART*, respectively). 80 slots represent associative relationships between classes, of which 42 have inverses (*e.g.*, *BRANCH* / *BRANCH_OF* and *CONTAINS* / *CONTAINED_IN*); *CONTINUOUS_WITH* is its own inverse; 37 slots do not have inverses (*e.g.*, *FASCICULAR_ARCHITECTURE* and *HAS_WALL*). In addition to slots linking classes, there are 61 slots in FMA describing atomic properties of classes (*e.g.*, the slot *HAS_MASS* accepts a Boolean value: TRUE or FALSE). Finally, 32 slots in the FMA link classes to instances[5] (e.g., *LOCATION* and *PREFERRED_NAME*).

In order to reduce the number of classes under investigation while keeping most of the complexity of the FMA, we ignored the classes differing from their parents solely by laterality (e.g., *Left ligament of wrist* vs. *Ligament of wrist*). The remaining subset comprises 39,337 classes. A CLIPS representation of the FMA was generated in Protégé, provided by the FMA developers. The features in the CLIPS representation of FMA are generally the same as in the Protégé environment. However, slots typed as Boolean in the Protégé environment are represented as type SYMBOL in CLIPS (with allowed-values of TRUE and FALSE). The version of the FMA used in this study is dated of July 2004.

3 Conversion rules

Ontologies developed in Protégé[6] are composed of classes and instances, the classes being organized in a taxonomy. Slots and facets are another important component of frame-based systems: slots specify relationships between classes and describe class properties; facets express constraints on slots. OWL ontologies contain classes, proper-

[3] http://fma.biostr.washington.edu/

[4] http://protege.stanford.edu/

[5] Instances in FMA correspond to special types of slot values, not to the realization of anatomical concepts as it is generally understood. (See Supp 4 and Supp 9 for examples)

[6] Throughout this paper, Protégé refers to the "core Protégé", i.e., the frame-based editor, ignoring its popular OWL plugin.

ties and individuals. Classes are specified by necessary conditions and/or defined by necessary and sufficient conditions.

We designed conversion rules and implemented them in order to convert the FMA into OWL DL automatically. In practice, our tools convert the original CLIPS file into an OWL file. The conversion can be summarized as follows. Classes in Protégé become classes in OWL DL[7]. Slots in Protégé become properties in OWL DL (including annotation properties). Finally, necessary and sufficient conditions are defined for OWL DL classes.

3.1 *Converting slots of the FMA in Protégé into properties in OWL DL*

All slots used in the FMA are represented in a top-level slot class. Each of these slots is converted into a property in OWL DL. Slots have a type specification (*e.g.*, INTEGER and SYMBOL) and constraints about the allowable values (*i.e.*, in allowed-parents / allowed-classes / allowed-values), which are used to delimit the type and range of property in OWL DL, as shown in Table 1. Additionally, the number of values allowed in a slot (single-slot or multi-slot specification) corresponds to the cardinality (at most one or multiple) of the corresponding property. Slots with single-slot specification are converted into functional properties in OWL DL. Finally, slots having inverses (inverse-slot specification) are converted as to stand in a owl:inverseOf relation in OWL DL; when a slot is its own inverse, the corresponding property becomes symmetric in OWL DL.

Slot of the FMA in CLIPS	Property in OWL DL
Typed INTEGER, FLOAT or STRING	owl:DatatypeProperty with range being XML Schema datatypes integer, float and string, respectively
Typed SYMBOL with allowed-values TRUE and FALSE	owl:DatatypeProperty with range being XML Schema datatype Boolean
Typed SYMBOL with allowed-values that are neither TRUE nor FALSE	owl:ObjectProperty with range being an enumerated class of all individuals in allowed-values
Typed SYMBOL with allowed-parents	owl:ObjectProperty with range being owl:unionOf all classes in allowed-parents
Typed INSTANCE with allowed-classes	owl:ObjectProperty with range being owl:unionOf all classes in allowed-classes

Table 1 – Rules for converting slots of the FMA into properties in OWL DL

In addition to the overall top-level definition, slots can be introduced in class descriptions in CLIPS, representing that the class is allowed to have the slot. We use such specification to delimit the domain of property in OWL DL. If slot *S* is introduced in

[7] OWL classes are either named or unnamed. Throughout this paper, unless we explicitly specify "unnamed", "class" refers to named classes.

class *X*, then *X* becomes an element of the domain of the property *S*. As one slot can be introduced into multiple classes, the domain of *S* is the union of all these classes. (see Supp 1- Supp 4 for examples)

In order to convert slots of type SYMBOL with allowed values other than TRUE or FALSE into properties having an enumerated class as their range, one individual has to be generated in OWL DL for each of the allowed values of these slots (see Supp 5).

3.2 *Converting classes of the FMA in Protégé into classes in OWL DL*

Every class of the FMA is represented both as a metaclass and an instance of another metaclass in Protégé, "as a technical solution for enabling the selective inheritance of attributes" [16]. The *metaclass definition* of a class, inherited by its subclasses, specifies its name, its direct superclass(es), and the slots introduced in this class. Therefore, allowable slots for a class include the slots introduced in this class and those inherited from its superclasses. In contrast, the *instance definition* of the class, not inheritable, specifies the metaclass of which this class is an instance (*i.e.*, metaclass instantiation), and all the values for the slots in this class. When converted into a class in OWL DL, the metaclass and instance definitions of the class in Protégé are merged, as shown in Table 2. (See Supp 6-Supp 8 for examples).

Class of the FMA in CLIPS		Class in OWL DL
Metaclass definition	Every taxonomic relation to direct superclass	rdfs:subClassOf axiom to a named class representing the direct superclass
	Every slot introduced with allowed-parents (or allowed-classes)	property restriction with owl:allValuesFrom constraint on owl:unionOf all classes in allowed-parents
	Every slot introduced with allowed-values and a concrete value specification	property restriction with owl:hasValue constraint on the value
Instance definition	Metaclass instantiation	rdfs:subClassOf axiom to a named class representing the metaclass that this class is an instance of
	Every slot value where the slot is converted into a datatype property	property restriction with owl:hasValue constraint on the value
	Every slot value where the slot is converted into an object property ranging over an enumerated class	rdfs:subClassOf axiom to the property restriction with owl:hasValue constraint on the value
	Every slot value where the slot is converted into an object property ranging over a named class or disjunction of named classes	property restriction with owl:someValuesFrom constraint on the value

Table 2 – Rules for converting classes of the FMA into classes in OWL DL

Attributed slots are used to represent the properties of relations. For example, because the partitive relation between *Wall of esophagus* and *Esophagus* is not shared with other anatomical structure, *unshared* is an attribute of this *ATTRIBUTED_PART* slot. Attributed slots and their values are converted into subClassOf axioms to the property restrictions with owl:someValuesFrom constraints on the nested classes generated for the values (see Supp 9 for an example).

3.3 *Defining classes in OWL DL by necessary and sufficient conditions*

In modeling classes, OWL distinguishes between two types of conditions: *necessary and sufficient conditions* (owl:equivalenceClass) which define classes *and necessary conditions* (owl:subClassOf). Slot values generally correspond to necessary conditions. However, there is no correspondence in Protégé for necessary and sufficient conditions in OWL. One trivial solution consists of simply describing the classes with necessary conditions rather than defining them with necessary and sufficient conditions. In this case, only limited reasoning support can be expected, as reasoners such as Racer rely in part on defined classes. Alternatively, we had to select – somewhat arbitrarily – the properties that would define FMA classes. Intuitively *and as a first approximation*, we considered anatomical structures to be "the sum of their parts" and selected one of the mereological views, the slot *CONSTITUTIONAL PART* – with all its values – as the source for necessary and sufficient conditions for classes in OWL.[8] Other slots and combination thereof could also be selected, leading to different reasoning results in OWL. (See Supp 10 for an example).

In addition to necessary and sufficient conditions, defined classes can also have necessary conditions, called *global axioms* in this case (coming from slots other than those selected for necessary and sufficient conditions). However, global axioms are known to dramatically increase the reasoning complexity in Racer and were therefore purposely removed from defined classes.

3.4 *Designating annotation properties in OWL DL*

Similarly to the necessary and sufficient conditions of classes, annotation properties in OWL have no direct correspondence in Protégé. Slots for identifiers and names of anatomical structures (e.g., *UWDAID*, *PREFERRED_NAME* and *SYNONYMS*) typically become annotation properties in OWL DL. Such slots must be identified manually. Their values are converted into data literals in OWL DL. (See Supp 11 and Supp 12 for examples).

[8] A class is defined to be equivalent to a conjunction of its direct superclasses, the metaclass of which this class is an instance, someValuesFrom restriction on S over U_1, …, and someValuesFrom restriction on S over U_n

4 Results of the conversion

After the conversion of the 39,337 classes and 187 slots from FMA in Protégé (ignoring laterality distinctions), FMAinOWL contains 39,337 classes, 187 properties and 85 individuals. Among the properties, 20 correspond to annotations (including 3 from attributed slots), 19 to datatypes and 148 to object properties (including 29 from attributed slots). 115,203 subClassOf axioms are generated, including 39,331 from taxonomy and 3,406 from metaclass instantiation. Additionally, 2,310 nested classes are generated for the values of attributed slots, and 9,092 subClassOf axioms are contained in these nested classes. 559 classes are defined through equivalentClass axioms after using slot *CONSTITUTIONAL_PART* as source of necessary and sufficient conditions. With these defined classes, the total number of subClassOf axioms in FMAinOWL has decreased to 107,238, including 38,772 from taxonomy, and 3,378 from metaclass instantiation.

5 Optimizing the conversion

Optimization techniques have been explored to downsize FMAinOWL, for the purpose of enabling the OWL reasoning and to make it more efficient. Unlike removing global axioms as presented earlier, the optimization does not change the logical definitions or reasoning results of FMAinOWL.

Optimizing domains. As stated earlier, the domain of a property in OWL DL is the disjunction of all classes where the corresponding slot is introduced. Some properties contain a large number of classes in their domains (*e.g.*, 1,618 for location), leading to inefficient OWL reasoning. Classes that are descendants of other classes in the domain can be removed from the domain without changing the definition or application of the property. The optimization results in downsizing the domain of 40 of the 187 properties. For example, only 2 classes remain in the domain of location after optimization.

Optimizing subClassOf axioms. As stated earlier, classes receive subClassOf axioms to named classes in OWL DL from two sources: taxonomic relations and metaclass instantiation. For example, class *X* is represented as "X is-a Y" in metaclass definition and "[X] of Z" in instance definition. Optimization techniques prevent the generation of 28 reflexive subClassOf axioms (from "[X] of X"), 24,307 duplicate subClassOf axioms (from "X is-a Y" and "[X] of Y") and 11,430 transitively redundant axioms (from "X is-a Y" and "[X] of Z" where Y is a descendant of Z). Overall, only 9% of 39,337 classes end up having subClassOf axioms from both taxonomy and metaclass instantiation.

6 Reasoning over FMAinOWL with Racer

Besides making it available in a popular formalism, the principal motivation for converting the FMA into OWL is to benefit from reasoning support. Because it is mapped to description logic, OWL DL makes use of existing reasoners such as Racer [17]. Reasoning support allows users to check the consistency of the onlology and the hierarchical organization of the classes (classification). Unlike consistency checking, classification requires classes to be defined with necessary and sufficient conditions, not only described with necessary conditions.

The sheer size and complexity of FMAinOWL, even after limiting the number of classes and optimizing the conversion, caused Racer to fail to reason over the whole file. Extracting a subset (e.g., for the cardiovascular system) would alleviate this problem but is likely to hide issues specific to other subsets. Instead, we elected to reason over the whole domain. As suggested by the developers of Racer, we tested only a limited number of properties at a given time (*e.g.*, no properties, only Boolean typed properties, two inverse object properties). In practice, we generated smaller versions of the FMAinOWL file, containing all classes but limited to the properties to be tested. Version 1.7 of Racer was used in this study on a Microsoft Windows platform.

7 Checking consistency: class satisfiability

We checked the consistency of the ontology based on Boolean properties and on the domain and range of properties. Importantly, not only do the descendants of unsatisfiable classes become unsatisfiable themselves, but this is also the case of all classes which have an unsatisfiable class as value for some property.

7.1 *Consistency based on Boolean datatype properties*

In the FMA in Protégé, Boolean slots are used to record differentiae between high-level anatomical categories. For example, material physical anatomical entities have mass (hasValue (has_mass true)), while non-material physical anatomical entities do not (hasValue (has_mass false)). Classes specified as descendants of both Material_physical_anatomical_entity and Non-material_physical_anatomical_entity were identified as unsatisfiable by Racer.

113 such classes were identified by Racer in FMAinOWL. Inconsistencies were traced back to inconsistent descriptions in the FMA (39 cases) and to the conversion process (74 cases). Examples of **inconsistent descriptions in the FMA** include the class Zone_of_cell. This class inherits hasValue (has_mass true) from its ancestor Material_physical_anatomical_entity and has value false in its own slot has_mass. Note that because Zone_of_cell is unsatisfiable, all its descendants also become unsatisfiable. During the **conversion process**, we showed that both taxonomic relations and metaclass instantiation are converted into subClassOf axioms (section 3.2). Merging the two definitions may result in conflicting values for a given Boolean property. For example, the class Compartment_subdivision is a descendant of Material_physical_anatomical_entity and an instance of the metaclass Anatomical_space, itself a descendant of Non-material_physical_anatomical_entity. Again, this

class inherits both true and false for the property has_mass and is therefore unsatisfiable (as are, in turn, its descendants).

7.2 *Consistency based on the domain and range of object properties*

Racer checks the consistency between the domain and range defined for a given object property *P* (see 3.1) and the restriction(s) involving this property in the definition of a class *C*. Consistency checking based on domain and range in OWL is different from type checking in programming languages. Here, consistency implies that the intersection between the domain (or range) of *P* and the value of *P* in *C* is not empty. The class *C* is declared unsatisfiable by Racer if this condition is not met. For example, the property *D2D_PART* has range Non-Material_physical_anatomical_entity. In the class Surface_of_wrist, the property *D2D_PART* has value Anatomic_snuff_box, a descendant of Material_physical_-anatomical_entity. The value of *D2D_PART* in Surface_of_wrist is disjoint from the range of *D2D_PART*. The class Surface_of_wrist is thus identified as unsatisfiable by Racer. Overall, this error is the only one revealed by this type of consistency checking in the FMA.

8 Reclassification

The whole taxonomy of the FMA is built manually by the domain experts under FMA-specific modeling principles [15]. In contrast, Racer automatically recreates the class hierarchies based on the definition of the classes. Discrepancies between the original taxonomy and Racer's hierarchy, i.e., reclassified classes, typically correspond to inconsistent descriptions in the FMA or issues in the conversion process. As for unsatisfiability, reclassification may have far-reaching effects due to propagation.

8.1 *Reasoning on necessary and sufficient conditions*

Based on necessary and sufficient conditions (i.e., the property *CONSTITUTIONAL_PART* in this experiment), 286 classes were reclassified by Racer, bringing to light the following issues: sibling classes having the same constitutional parts become equivalent; a class and its direct superclass having the same constitutional parts become equivalent; a class and its direct superclass become equivalent when the class and one of its indirect superclasses have the same constitutional parts; and a class becomes a subclass of its sibling. An analysis of some of the classes reclassified confirms that the property *CONSTITUTIONAL_PART* – as currently defined in the FMA – is not a reliable source of necessary and sufficient conditions. For example, the class Atrioventricular_valve and its two direct subclasses Mitral_valve and Tricuspid_valve are all identified as equivalent because Mitral_valve and Tricuspid_valve have the same constitutional parts as their indirect superclass Cardiac_valve.

8.2 *Reasoning on transitive properties*

Partitive relationships among (canonical) anatomical entities are generally transitive. Unlike in Protégé, the transitivity of properties is supported in OWL DL. The property *CONSTITUTIONAL_PART*, for example, was defined as transitive, which helped identify additional issues in the class definitions in FMAinOWL, most of which being related to selecting *CONSTITUTIONAL_PART* as the source for necessary and sufficient conditions. One such issue can be summarized as follows. The constitutional parts of Prostate include, by transitivity, cells such as Luminal_cell_of_prostatic_acinus, which have the same values for *CONSTITUTIONAL_PART* as Cell. This causes Prostate to be reclassified – along with 137 other classes – as a direct subclass of Cell. Our point here is to not to argue whether prostate is a kind of cell or not, but rather to emphasize the power of reasoning in identifying modeling or conversion insufficiencies. Of course, adding constraints to the definition of Cell would prevent such infelicitous reclassification [11].

9 Discussion

Reasoning support as a quality assurance tool. Large ontologies are notoriously difficult to develop and maintain in a consistent state, especially when they are developed with little or no support for consistency checking. Frame-based ontology environments such as Protégé do accommodate plugins allowing users to perform consistency checks, but offer little built-in support for consistency checking. “DL-izing” the FMA makes it amenable to reasoning support and can therefore be used for quality assurance purposes. In our experience with the FMA, consistency checking helped detect modeling errors otherwise difficult to identify (e.g., low-level classes inheriting from two disjoint high-level classes). The benefit in terms of reclassification is more subtle, due to the difficulty of defining necessary and sufficient conditions for anatomical entities.

More work is needed to determine the place of DL-based techniques in the validation and verification of ontologies. While our experience seems consistent with recent work on ontology “debugging” [18], such techniques are certainly complementary to visual (e.g., Jambalaya plugin) or other validation approaches (e.g., [19, 20]).

Size matters. The FMA is one of the largest ontologies developed with Protégé so far, and probably the largest to be converted from Protégé to OWL DL. In comparison to the 70,169 classes and 187 properties of the FMA, the NCI Thesaurus contains “only” about 34,000 classes, 100 properties, and 9,000 conditions of classes [21]. Moreover, no necessary and sufficient conditions are defined, nor are any owl:hasValue or owl:allValuesFrom restrictions specified in the NCI Thesaurus. The sheer size and complexity of the FMA represented an issue not for the conversion, but for the reasoning. In fact, Racer could not digest the entire FMA, even after removing 43% of its classes and optimizing the representation. In order to enable consistency checking, properties had to be tested individually or in small groups rather than all together. Reasoning over large ontologies remains technically challenging.

Necessary and sufficient conditions. The biggest challenge in this experiment is certainly to define the classes in OWL DL automatically by selecting the appropriate necessary and sufficient conditions. Other attempts to convert existing ontologies or terminologies into OWL generally did not address necessary and sufficient conditions (e.g., [21]) or deferred this issue to the applications using these ontologies (e.g., [12]). We attempted to define anatomical entities by combining the following properties into necessary and sufficient conditions: taxonomic relations, metaclass instantiation and constitutional parts. This simple method can be automatically implemented as part of the conversion process, but is insufficient in many respects. Defining anatomical entities solely on the basis of their constitutional parts is not correct, in part because no such constitutional parts are defined for most FMA classes. The absence of precisely defined classes was a serious limitation for reasoning support, especially reclassification. A closer collaboration with the authors of the FMA should lead to better class definitions for anatomical entities. Analogously, our conversion strategy generally consisted in preserving most of the features of the frame representation. However, a better understanding of the original modeling choices in Protégé for the FMA would certainly result in a more accurate representation in OWL. Alternative representations should be tested and evaluated.

Acknowledgments

This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM).This work was done while Songmao Zhang and Christine Golbreich were visiting scholars at the Lister Hill National Center for Biomedical Communications, NLM, NIH, Thanks for their support to the developers of Racer: Volker Haarslev and Ralf Möller. Our thanks also go to Cornelius Rosse, José Mejino and Todd Detwiler for the FMA.

References

1. Cornet R, Abu-Hanna A. Usability of expressive description logics--a case study in UMLS. Proc AMIA Symp 2002:180-4
2. Hahn U, Schulz S. Towards a broad-coverage biomedical ontology based on description logics. Pac Symp Biocomput 2003:577-88
3. Pisanelli DM, Gangemi A, Steve G. An ontological analysis of the UMLS Methathesaurus. Proc AMIA Symp 1998:810-4
4. Kashyap V, Borgida A. Representing the UMLS Semantic Network using OWL: (Or “What's in a Semantic Web link?”). In: Fensel D, Sycara K, Mylopoulos J, editors. The SemanticWeb - ISWC 2003. Heidelberg: Springer-Verlag; 2003. p. 1-16

5. Soualmia L, Golbreich C, Darmoni S. Representing the MeSH in OWL: Towards a semi-automatic migration. Proceedings of the KR 2004 Workshop on Formal Biomedical Knowledge Representation 2004:81-87 http://CEUR-WS.org/Vol-102/soualmia.pdf.
6. Wroe CJ, Stevens R, Goble CA, Ashburner M. A methodology to migrate the Gene Ontology to a description logic environment using DAML+OIL. Pac Symp Biocomput 2003:624-35
7. Golbeck J, Fragoso G, Hartel F, Hendler J, Oberthaler J, Parsia B. The National Cancer Institute's thesaurus and ontology. Journal of Web Semantics 2003;1(1) http://www.websemanticsjournal.org/ps/pub/2004-6.
8. Antoniou G, van Harmelen F. Web Ontology Language: OWL. In: Staab S, Studer R, editors. Handbook on Ontologies: Springer-Verlag; 2004. p. 67-92
9. Smith MK, Welty C, McGuinness DL. OWL Web Ontology Language Guide, W3C Recommendation, 10 February 2004; 2004 http://www.w3.org/TR/2004/REC-owl-guide-20040210/.
10. Golbreich C, Zhang S, Bodenreider O. Migrating the FMA from Protégé to OWL. Proceedings of the Eighth International Protégé Conference 2005:108-111
11. Golbreich C, Zhang S, Bodenreider O. The Foundational Model of Anatomy in OWL: Experience and perspectives. Proceedings of the workshop "OWL Experiences and Directions", November 11-12, 2005, Galway, Ireland 2005:(electronic proceedings: http://CEUR-WS.org)
12. Dameron O, Rubin D, Musen A. Challenges in converting frame-based ontology into OWL: the Foundational Model of Anatomy case-study. Proc AMIA Symp 2005:(in press)
13. Beck R, Schulz S. Logic-based remodeling of the Digital Anatomist Foundational Model. AMIA Annu Symp Proc 2003:71-5
14. Rosse C, Mejino JL, Modayur BR, Jakobovits R, Hinshaw KP, Brinkley JF. Motivation and organizational principles for anatomical knowledge representation: the digital anatomist symbolic knowledge base. J Am Med Inform Assoc 1998;5(1):17-40
15. Rosse C, Mejino JL, Jr. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J Biomed Inform 2003;36(6):478-500
16. Noy NF, Musen MA, Mejino JLV, Rosse C. Pushing the envelope: challenges in a frame-based representation of human anatomy. Data & Knowledge Engineering 2004;48(3):335-359
17. Haarslev V, Möller R. Racer: An OWL reasoning agent for the Semantic Web. Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems 2003:91-95
18. Schlobach S. Debugging and semantic clarification by pinpointing. In: The Semantic Web: Research and Applications. Proceedings of the Second European Semantic Web Conference: Springer-Verlag; 2005. p. 226-240
19. Noy NF, Musen MA. PROMPT: algorithm and tool for automated ontology merging and alignment. Proc of AAAI 2000:450-455
20. Zhang S, Bodenreider O. Law and order: Assessing and enforcing compliance with ontological modeling principles. Computers in Biology and Medicine 2005:(in press)
21. de Coronado S, Haber MW, Sioutos N, Tuttle MS, Wright LW. NCI Thesaurus: using science-based terminology to integrate cancer research results. Medinfo 2004;11(Pt 1):33-7

Supplementary material

Examples of FMA classes in OWL DL

Available online at:
`http://mor.nlm.nih.gov/pubs/supp/2006-psb-sz/`.

Converting slots of the FMA in Protégé into properties in OWL DL

The single-slot *HAS_MASS* is specified as type SYMBOL with allowed-values TRUE and FALSE. Moreover, this slot is introduced in two classes, *Material_physical_anatomical_entity* and *Non-material_physical_anatomical_entity*. The conversion is shown in Supp 1.

has_mass in CLIPS	has_mass in OWL DL
(defclass CLIPS_TOP_LEVEL_SLOT_CLASS	<owl:DatatypeProperty rdf:ID="has_mass">
(single-slot has_mass	<rdfs:domain>
(type SYMBOL)	<owl:Class>
(allowed-values FALSE TRUE)	<owl:unionOf rdf:parseType="Collection">
(cardinality 0 1))	<owl:Class rdf:about="#Material_physical_anatomical_entity" />
...)	<owl:Class rdf:about="#Non-material_physical_anatomical_entity" />
	</owl:unionOf>
(defclass Material_Physical_anatomical_entity	</owl:Class>
(single-slot has_mass...)...)	</rdfs:domain>
	<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean" />
(defclass Non-material_Physical_anatomical_entity	<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
(single-slot has_mass ...)...)	</owl:DatatypeProperty>

Supp 1 – Converting slot *HAS_MASS* of the FMA into property in OWL D

The single-slot *DIMENSION* is specified as type SYMBOL with allowed-values 0-dimension, 1-dimension, 2-dimension and 3-dimension. The conversion is shown in Supp 2.

Dimension in CLIPS	dimension in OWL DL
(defclass CLIPS_TOP_LEVEL_SLOT_CLASS	<owl:ObjectProperty rdf:ID="dimension">
(single-slot dimension	<rdfs:domain rdf:resource="#Physical_anatomical_entity" />
(type SYMBOL)	<rdfs:range>
(allowed-values 3-dimension 2-dimension	<owl:Class>
1-dimension 0-dimension)	<owl:oneOf rdf:parseType="Collection">
(cardinality 0 1))	<owl:Thing rdf:about="#individual_1-dimension" />
...)	<owl:Thing rdf:about="#individual_0-dimension" />
	<owl:Thing rdf:about="#individual_2-dimension" />
(defclass Physical_anatomical_entity	<owl:Thing rdf:about="#individual_3-dimension" />
(single-slot dimension...)...)	</owl:oneOf>
	</owl:Class>
	</rdfs:range>
	<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
	</owl:ObjectProperty>

Supp 2 – Converting slot *DIMENSION* of the FMA into property in OWL DL

The multi-slot *CONSTITUTIONAL_PART_OF* is specified as type SYMBOL with allowed-parents *Physical_anatomical_entity*. Moreover, this slot has inverse-slot *CONSTITUTIONAL_PART*, and is introduced in classes *Anatomical_space*, *Body_substance* and *Anatomical_structure*. The conversion is shown in Supp 3.

constitutional_part_of in CLIPS	constitutional_part_of in OWL DL
(defclass CLIPS_TOP_LEVEL_SLOT_CLASS	<owl:ObjectProperty rdf:ID="constitutional_part_of">
(multislot constitutional_part_of	<owl:inverseOf rdf:resource="#constitutional_part" />
(type SYMBOL)	<rdfs:domain>
(allowed-parents Physical_anatomical_entity)	<owl:Class>
(inverse-slot constitutional_part))	<owl:unionOf rdf:parseType="Collection">
…)	<owl:Class rdf:about="#Anatomical_space" />
	<owl:Class rdf:about="#Body_substance" />
(defclass Anatomical_space	<owl:Class rdf:about="#Anatomical_structure" />
(multi-slot constitutional_part_of…)…)	</owl:unionOf>
(defclass Body_substance	</owl:Class>
(multi-slot constitutional_part_of…)…)	</rdfs:domain>
(defclass Anatomical_structure	<rdfs:range rdf:resource="#Physical_anatomical_entity" />
(multi-slot constitutional_part_of…)…)	</owl:ObjectProperty>

Supp 3 – Converting slot *CONSTITUTIONAL_PART_OF* of the FMA into property in OWL DL

Slots typed INSTANCE are the attributed slots linking classes to instances in Protégé. As shown in Supp 4, *ATTRIBUTED_PART* is an attributed slot whose allowed classes are instances of class *Part_of_relationship_value.*

Attributed_part in CLIPS	attributed_part in OWL DL
(defclass CLIPS_TOP_LEVEL_SLOT_CLASS	<owl:ObjectProperty rdf:ID="attributed_part">
(multislot attributed_part	<rdfs:domain rdf:resource="#Anatomical_structure" />
(type INSTANCE)	<rdfs:range rdf:resource="#Part_of_relationship_value" />
(allowed-classes Part_of_relationship_value))	</owl:ObjectProperty>
…)	
(defclass Anatomical_structure	
(multi-slot attributed_part…)…)	

Supp 4 – Converting slot *ATTRIBUTED_PART* of the FMA into property in OWL DL

Generating individuals in OWL DL

Based on the slot *DIMENSION* presented earlier, one individual[9] is generated under owl:Thing for each allowed value of this slot, as shown in Supp 5.

```
<owl:Thing rdf:ID="individual_0-dimension" />
<owl:Thing rdf:ID="individual_1-dimension" />
<owl:Thing rdf:ID="individual_2-dimension" />
<owl:Thing rdf:ID="individual_3-dimension" />
```

Supp 5 – Generating individuals in OWL DL

[9] Individuals are prefixed by "individual_", because some allowed-values of slots share names with classes in the FMA in Protégé, such as *Inferior* and *Liquid*.

Converting classes of the FMA in Protégé into classes in OWL DL

As shown in Supp 6, the metaclass and instance definitions of class *Integument_of_abdomen*, are merged into OWL DL.

Integument_of_abdomen in CLIPS	Integument_of_abdomen in OWL DL

Metaclass definition:
 	
(defclass Integument_of_abdomen
    (is-a integument_of_body_part_subdivision))
	<owl:Class rdf:ID="Integument_of_abdomen">
	   <rdfs:subClassOf rdf:resource="#Integument_of_body_part_subdivision" />
	   <rdfs:subClassOf rdf:resource="#Anatomical_structure" />
	   <rdfs:subClassOf>
	      <owl:Restriction>
	         <owl:onProperty rdf:resource="#dimension" />
	         <owl:hasValue rdf:resource="#individual_d3-dimension" />
	      </owl:Restriction>
	   </rdfs:subClassOf>
	   <rdfs:subClassOf>
	      <owl:Restriction>
	         <owl:onProperty rdf:resource="#constitutional_part_of" />
	         <owl:someValuesFrom rdf:resource="#Abdominal_wall" />
	      </owl:Restriction>
	   </rdfs:subClassOf>
	   … …
	</owl:Class>

Instance definition:
 	
([Integument_of_abdomen]
    of Anatomical_structure
    (dimension 3-dimension)
    (constitutional_part_of Abdominal_wall)
    …
)	

Supp 6 – Converting class *Integument_of_abdomen* of the FMA into class in OWL DL

Few classes have two direct superclasses. Such classes (e.g., Physical_anatomical_entity is-a Anatomical_entity_template and Physical_anatomical_entity is-a Anatomical_entity) in CLIPS are converted into two subClassOf axioms as shown in Supp 7.

```
<owl:Class rdf:ID="Physical_anatomical_entity">
  <rdfs:subClassOf rdf:resource="#Anatomical_entity_template" />
  <rdfs:subClassOf rdf:resource="#Anatomical_entity" />
  … …
</owl:Class>
```

Supp 7 – Converting classes with two direct superclasses into OWL DL

The class *Body_substance* has slots *CONTAINED_IN* (with allowed-parents) and *HAS_INHERENT_3-D_SHAPE* (with allowed-values and a concrete value specification) introduced in its metaclass definition, the conversion shown in Supp 8.

Body_substance in CLIPS	Body_substance in OWL DL

Metaclass definition:
 	
(defclass Body_substance
   …
   (multi-slot contained_in
     (type SYMBOL)
     (allowed-parents Anatomical_space))
   (single-slot has_inherent_3-D_shape
     (type SYMBOL)
     (allowed-values FALSE TRUE)
     (value FALSE))
   …
)
	<owl:Class rdf:ID="Body_substance">
	   <rdfs:subClassOf>
	      <owl:Restriction>
	            <owl:onProperty rdf:resource="#contained_in" />
	            <owl:allValuesFrom rdf:resource="#Anatomical_space" />
	      </owl:Restriction>
	   </rdfs:subClassOf>
	   <rdfs:subClassOf>
	       <owl:Restriction>
	            <owl:onProperty rdf:resource="#has_inherent_3-D_shape" />
	             <owl:hasValue
	rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">false</owl:hasValue>
	       </owl:Restriction>
	   </rdfs:subClassOf>
	   … …
	</owl:Class>

Supp 8 – Converting class *Body_substance* of the FMA into class in OWL DL

The class *Esophagus* has attributed slot *ATTRIBUTED_PART* and one of the values for this slot is the instance *fm-live_10718* [10], for which a nested class is generated in OWL, as shown in Supp 9. We constructed such nested classes following the same conversion rules for classes.

Esophagus in CLIPS	Esophagus in OWL DL
Instance definition:	<owl:Class rdf:ID="Esophagus">
	<rdfs:subClassOf>
	<owl:Restriction>
([Esophagus]	<owl:onProperty rdf:resource="#attributed_part" />
(attributed_part	<owl:someValuesFrom>
[fm-live_10718]	<owl:Class rdf:ID="fm-live_10718"> *<!--nested class for instance [fm-live_10718]-->*
[fm-live_10719]	<rdfs:subClassOf
…)	rdf:resource="#Organ_subdivision_part_of_relationship_value" />
…	<rdfs:subClassOf>
)	<owl:Restriction>
Instance definition of instance fm-	<owl:onProperty rdf:resource="#related_part" />
live_10718:	<owl:someValuesFrom rdf:resource="#Wall_of_esophagus" />
	</owl:Restriction>
	</rdfs:subClassOf>
([fm-live_10718]	<rdfs:subClassOf>
of Organ_subdivision_part_of_relationship_value	<owl:Restriction>
(related_part Wall_of_esophagus)	<owl:onProperty rdf:resource="#anatomical_arbitrary" />
(anatomical_arbitrary Anatomical)	<owl:hasValue rdf:resource="#individual_Anatomical" />
(partition Partition_1)	</owl:Restriction>
(shared_unshared Unshared)	</rdfs:subClassOf>
)	… …
	</owl:Class> *<!-- end of nested class for instance [fm-live_10718] -->*
	</owl:someValuesFrom>
	</owl:Restriction>
	</rdfs:subClassOf>
	… …
	</owl:Class>

Supp 9 – Converting class *Esophagus* of the FMA into class in OWL DL

[10] Note that instances in Protégé are composed of two groups, one is classes modeled both as instances and metaclasses such as *Esophagus*, and the other is “pure” instances without meaningful names such as *fm-live_10718*.

Defining classes in OWL DL by necessary and sufficient conditions

The class Cell is defined as shown in Supp 10, with its taxonomic relation and all constitutional parts in one necessary and sufficient condition. The shadowed part in Supp 10 corresponds to the necessary conditions of Cell (global axioms).

Cell in CLIPS	Cell in OWL DL

Metaclass definition:
 	
(defclass Cell
    (is-a Anatomical_structure) …)
	<owl:Class rdf:ID="Cell">
	        <owl:equivalentClass>
	          <owl:Class>
	            <owl:intersectionOf rdf:parseType="Collection">
	            <owl:Class rdf:about="#Anatomical_structure" />
	            <owl:Restriction>
	              <owl:onProperty rdf:resource="#constitutional_part" />
	              <owl:someValuesFrom rdf:resource="#Plasma_membrane" />
	            </owl:Restriction>
	            <owl:Restriction>
	              <owl:onProperty rdf:resource="#constitutional_part" />
	              <owl:someValuesFrom rdf:resource="#Cytoplasm" />
	            </owl:Restriction>
	            <owl:Restriction>
	              <owl:onProperty rdf:resource="#constitutional_part" />
	              <owl:someValuesFrom rdf:resource="#Cell_nucleus" />
	            </owl:Restriction>
	            </owl:intersectionOf>
	          </owl:Class>
	        </owl:equivalentClass>
	        <rdfs:subClassOf>
	          <owl:Restriction>
	            <owl:onProperty rdf:resource="#bounded_by" />
	            <owl:someValuesFrom rdf:resource="#Surface_of_cell" />
	          </owl:Restriction>
	        </rdfs:subClassOf>
	        … …
	</owl:Class>

Instance definition:
 	
([Cell]
      of Anatomical_structure
    (constitutional_part
                Plasma_membrane
                Cytoplasm
                Cell_nucleus)
 	
    (bounded_by Surface_of_cell)
    (has_boundary TRUE)
    (has_physical_state Solid)
    (part_of
                Tissue
                Body_substance)
    (regional_part
                Apical_part_of_cell
                Basal_part_of_cell)
   …
)	

Supp 10 – Defining class Cell in OWL DL

Designating annotation properties in OWL DL

We manually designate slots of the FMA to become annotation properties in OWL DL, including *UWDAID* (typed STRING in Protégé) and *PREFERRED_NAME* (typed INSTANCE), shown in Supp 11.

```
<owl:AnnotationProperty rdf:ID="UWDAID" />
<owl:AnnotationProperty rdf:ID="Preferred_name" />
```

Supp 11 – Designating annotation properties in OWL DL

For INSTANCE typed slots such as *PREFERRED_NAME* whose value is an instance in Protégé, we wrap all the slot values in the instance into one data literal as annotation value in OWL DL, as shown in Supp 12.

Body_substance in CLIPS	Body_substance in OWL DL
Instance definition:	<owl:Class rdf:ID="Body_substance">
([Body_substance]	<UWDAID> 9669 </UWDAID>
of Material_physical_anatomical_entity	<Preferred_name>author("JOSE_MEJINO_MD") authority("Cornelius_Rosse")
(UWDAID "9669")	modification("Dec_20_1996__10046193PM") name("Body_substance")
(Preferred_name [KB_INSTANCE_08389])…)	</Preferred_name> *<!-- from [KB_INSTANCE_08389] -->*
	… …
Instance definition of instance	</owl:Class>
KB_INSTANCE_08389:	
([KB_INSTANCE_08389]	
of Concept_name	
(author "JOSE MEJINO, MD")	
(authority "Cornelius Rosse")	
(modification "Dec 20 1996 1:00:46:193PM")	
(name "Body substance"))	

Supp 12 – Annotation properties and values in class Body_substance in OWL DL