The FMA in OWL 2

C. Golbreich[1], J. Grosjean[2], S.J. Darmoni[2]

[1] LIRMM CNRS 5506, Montpellier & University Versailles Saint-Quentin, France
[2] CISMeF & TIBS, LITIS EA 4108, Rouen University Hospital, Rouen, France

cgolbrei@gmail.com, Julien.Grosjean@chu-rouen.fr, Stefan.Darmoni@chu-rouen.fr

Abstract. Representing the Foundational Model of Anatomy (FMA) in OWL 2 is essential for semantic interoperability. The paper describes the method and tool used to formalize the FMA in OWL 2. One main strength of the approach is to leverage OWL 2 expressiveness and naming conventions of the native FMA to explicit some implicit semantics, meanwhile improving its ontological model and fixing some errors. A second originality is the flexible tool developed. It enables to easily generate a new version for each Protégé FMA update. While it provides one 'standard' FMA-OWL version by default, many options allow for producing other variants customized to users applications. To the best of our knowledge, no complete representation of the entire FMA in OWL DL or OWL 2 existed so far.

Keywords: Ontology, OWL, Life Sciences, Health, Anatomy.

1 Introduction

The Foundational Model of Anatomy (FMA) is "a reference ontology about human anatomy" [1-2]. The FMA is intended to model *canonical* human anatomy that is, "the ideal or prototypical anatomy to which each individual and its parts should conform" [1]. It contains more than 85,000 classes, 140 relationships connecting the classes and over 120,000 terms. Most entities are anatomical structures composed of many parts interconnected in complex ways, described in terms of their regions, constituents, innervations, blood vessels, boundaries etc. For example, a Heart has two regions – its *left and right side* -, several constitutional parts – *Wall of Heart, Interatrial, Interventricular*, and *Atrioventricular septum, Mitral Valve*, etc. -, is innerved by the *Deep cardiac plexus, Right and Left coronary nerve plexus*, etc. Thus the FMA is a very large and perhaps one of the most complex ontology in the biomedical sciences.

OWL 2 is the W3C standard for ontologies on the Semantic Web [8]. OWL 2 provides several advantages for Life Sciences ontologies: interoperability, semantics, reasoning services. (1) *Interoperability* is important for shared use across different domains. Once converted to OWL 2, ontologies become easier to be connected or combined with other ontologies. (2) *Semantics* (meaning) of terms is formally specified thanks to the underlying description logics. (3) Another practical benefit is that it allows to exploit the multitude of existing OWL tools, in particular powerful *reasoners.* Furthermore OWL 2 higher *expressiveness,* in particular its new metamodeling abilities, is of major interest as shown next.

The objective of the work is to represent the FMA in OWL 2, in order to make it interoperable with the increasing number of OWL ontologies available. Formalizing the FMA in OWL 2 provides a precise and rigorous meaning to the anatomical entities, crucial for example to share annotated resources. Making an OWL 2 version available is also an indispensable step for being able in the future to assist the FMA maintenance and to assure its quality thanks to OWL reasoning services and tools. The aim of the work is not to simply convert the FMA (for example by a script) from a format to another one, but to leverage OWL underlying description logic for enriching the FMA entities with formal definitions and axioms having a sound anatomical meaning.

A first strength of the presented approach consists in exploiting naming conventions and lexical patterns of the native FMA to explicit the implicit semantics (meanwhile improving its ontological model and fixing some errors). A second originality is that the tool developed makes it possible to generate a new version each time the Protégé FMA is updated by its authors. As they are not very familiar with OWL and may prefer to continue to use existing Protégé frame editor, this friendly and easy to use converter is very useful for them to automatically create the OWL conversion. Additionally, while it is possible to provide one ‘standard’ FMA-OWL version, many options allow for producing by a simple clic other variants customized to users applications, if needed. The next sections describe the method and tool achieved for representing the FMA in OWL 2 and presents the results obtained so far.

2 Method

The FMA ontology is implemented in Protégé frames[1] and stored in a MySQL database backend. Transforming it into OWL 2 is not a simple translation. It requires to specify the meaning of its terms in logics and to express by logical statements (axioms) some knowledge about the anatomical entities, which is not explicit in the native FMA. This raises several issues. The first one is that different types of information are embedded in Protégé FMA. Indeed, apart from the domain knowledge concerning the anatomical entities, the FMA also includes meta-level knowledge. The problem is that interpreting both knowledge in the same model might lead to undesired consequences because of their interactions. Two solutions are proposed thereafter: an OWL 1 DL ontology *without* metaclasses and an OWL 2 ontology *with* metaclasses (§2.1). The second challenge is to guarantee that the formal definitions and axioms created are semantically correct from an *anatomical* viewpoint. The idea for it is to use lexical patterns (§2.2). The third issue is that, given the large size of the FMA, it is essential to *automatically* generate the OWL axioms. A friendly tool (§3) has been achieved that for. An interesting feature of this tool is that it enables to

[1] the frame-based system developed by Stanford Center for Biomedical Informatics Research

create by default a 'standard' ontology from the FMA native frame version, and also other customized variants useful for specific applications, if wanted.

2.1 Metamodeling

In FMA Protégé frames each anatomical entity is modeled both as a class and a metaclass[2]. At the domain level, classes describe the anatomical entities. At the meta-level, metaclasses serve several purposes. They associate metadata to the anatomical entities, for example they attach to the class Heart its author 'JOSE MEJINO, MD', preferred-terms 'Heart' in English, 'Cor' in Latin, Non-English equivalent 'coeur' in French, its definition, synonyms, FMAID, etc. Metaclasses are also used to define 'templates' for some given types of entities. For example, the metaclass *Organ With Cavitated Organ Parts*, is intended to specify the common template of all the organ types (species) that have cavitated organ parts. Metaclasses are organized into a subclass hierarchy. The metaclass *Heart*, is a subclass of *Organ with cavitated organ parts*, itself subclass of *Organ*, of which it inherits the slots, facets, etc., e.g.; *bounded by* with range *Surface of organ*, *arterial supply* with range *Artery, Arteriole, Arterial plexus* etc. On the opposite, at the class level, the own slots, e.g.; *part of, bounded by, arterial supply*, are assigned particular values. Thus, the structure of an anatomical entity, e.g.; a canonical *Heart,* can be specified as being an *Organ With Cavitated Organ Part*, having a *Right atrium*, *Left atrium, Right ventricle, Left ventricule* as parts, being *bounded by Surface of heart,* having *Right coronary artery* and *Left coronary artery*, etc., as *arterial supply*.

To avoid undesired effects caused by interpreting both knowledge in the same model, it is offered to have (a) an OWL 1 (2) DL ontology *without* metaclasses but capturing their knowledge otherwise, or (b) an OWL 2 ontology *with* metaclasses: (a) An OWL 1 Ontology *Without Metaclasses* was initially proposed earlier [3] before OWL 2, because an OWL 1 DL ontology requires the deletion of the FMA higher order structure. To still capture the information embedded at metaclasses, metaclass instantiations are replaced by subclass axioms and metaclasses are transformed into ordinary OWL classes [3]. This did not introduce significant change, because "*all concepts in the Anatomy Taxonomy are subclass of a superclass and also an instance of a metaclass*". As metaclasses specify a given "template" of classes and classes specify the structure of their instances, property restrictions at metaclasses are interpreted as ako closure axiom and approximated by universal restrictions, while restrictions at classes are translated into existential restriction. Now, thanks to OWL 2 metamodeling new features, punning and enhanced annotations [9], it is possible to have an OWL 2 Ontology *With Metaclasses,* which (partly) better reflects the FMA authors design. Indeed, while OWL 1 DL required a strict separation between the names of classes and individuals. OWL 2 relaxes this separation [9]. Now punning makes it possible to represent metaclasses in using the same term to refer to a class and an individual, while retaining decidability: the name `Heart` can be used both for the metaclass `Heart` and for the class `Heart`, instance of `Organ with`

[2] In FMA frames, each anatomical entity is modeled both as a metaclass and as a class. "… *for enabling the selective inheritance of attributes*" [1 2].

`cavitated organ parts`. Thus, removing instantiation links is no more mandatory and using metaclasses that reflect more accurately the FMA templates is possible. On the other hand OWL 2 enhanced annotations are used for representing the metadata attached to the FMA entities. While OWL 1 allowed extralogical annotations, such as a label or a comment, OWL 2 additionally allows for annotations of axioms and of annotations themselves. In FMA frames, properties such as preferred name, synonyms, non-English equivalents, etc. are modeled by slots assigned with individuals of the Concept name class as values. As they concern metadata and not data the domain of anatomy, using OWL 2 annotations of annotation is more appropriate than FMA metaclasses.: the domain and meta-level data are no more confused and do not interact. Besides, a huge number of individuals are thus removed. For example, the class `Heart` (1) is annotated by the label `"Coeur"@fr` (4), the labeling itself being annotated (2) by its creator JOSE MEJINO MD (2), date (3), FMAID "217079" (4), publisher, etc.

```
(1) Declaration(Class(:Heart))
(2) AnnotationAssertion(Annotation(dc:creator "JOSE MEJINO MD"^^xsd:string)
(3) Annotation(dc:date "Thu May 12 142434 GMT-0800 2005"^^xsd:date)
(4) Annotation(:FMAID "217079"^^xsd:string).. rdfs:label :Heart "Coeur"@fr)
```

2.2 Formal semantics

The second main challenge is to enrich the FMA with formal definitions and axioms that have a sound anatomical meaning. The formalization is achieved in two steps. The first step focuses on the transformation of the FMA frames syntax and the second step on the FMA anatomical entities semantics. While the first transformation closely mirrors the FMA native model, the latter pushes the logical formalization further: new definitions and axioms are added that express some knowledge which was not explicitly stated in frames. Partly for historical reasons (OWL 2 did not exist before), the first step transforms the FMA ontology from frames to OWL 1 DL (*FMA-OWL v1*), the second step brings it to OWL 2 (*FMA-OWL 2*).

The transformation of the frames syntax in OWL reuses the 2005 rules defined in [3]. In short, Protégé classes and slots are converted into OWL classes and properties, with the specified domain and range. Slot characteristics (inverse, symmetric, functional) are translated using corresponding OWL constructs. Values of own slots of classes are converted either into OWL values of annotation properties or into existential property restrictions. As said above, property restrictions defined at metaclasses or classes are respectively transformed into universal or existential property restrictions and metaclass instantiation is replaced by a subclass relation.

At the second step the logical formalization is pushed forwards and the FMA ontology is enriched in several ways described in more details thereafter : (a) classes definitions are automatically generated from lexical patterns; (b) meanwhile numerous related axioms are automatically created or moved (c) new properties characteristics are added; (d) OWL annotations of annotation are used for metadata (as said §2) (e) OWL 2 metaclasses are created, but they can be omitted on demand.

(a) Class definitions. An important shortcoming of the 2005 ontology was its class definitions. Class expressions were built from one uniform property, e.g.;

constitutional part. However, all anatomical entities cannot be uniformly defined from the same properties [3]. New formalization rules are now defined that provide safe definitions. The key idea is to exploit *lexical* patterns of the FMA vocabulary and implicit properties omitted in such names (joined to the inference power of OWL). For example, it is very likely that the pattern Left_A (e.g., Left_Hand) denotes all A (Hands) that have left *laterality*, that Left_superior_cervical_ganglion means all the left and superior cervical_ganglion, Region_of_cytoplasm all the regional parts of cytoplasm etc. As the new rules create different forms of definition depending on each pattern, the patterns are basically unambiguous and moreover, their meaning was checked with FMA authors[3], all class definitions and axioms introduced in this manner are fully reliable. At the moment, two types of patterns are supported: (i) Pattern P_A denoting symmetrical siblings with an opposite anatomical_coordinate, e.g., Left_A/Right_A, Anterior_A/Posterior_A, Inferior_A/Superior_A etc., or an opposite gender, e.g.; MaleA/FemaleA and (ii) Pattern A_of_B denoting parts of entity, e.g., Lobe_of_Lung. Classes are incrementally defined as follows.

• Pattern P_A. At first, the Anatomical_coordinate subclasses are defined. Primary_Anatomical_coordinate are specified via property value restrictions, for example, axiom (1) states that *Left* denotes all objects with left laterality. Binary_Anatomical_coordinate are defined as an intersection of Primary_Anatomical_coordinate classes. For example axiom (2) states that Left_superior refers to all objects having a left and superior anatomical_coordinate. Entities of pattern P_A, where P is a Primary_Anatomical_coordinate subclass, are then provided definitions. For example, axiom (3) states that Left_Hand (resp. Right_Hand) denotes all hands having left laterality.

```
(1) EquivalentClasses(:Left ObjectHasValue(:laterality :individual_Left))
(2) EquivalentClasses(:Left_superior ObjectIntersectionOf(:Superior :Left))
(3) EquivalentClasses(:Left_Hand ObjectIntersectionOf(:Hand :Left))
(4) EquivalentClasses(:Lobe_of_Lung
       ObjectIntersectionOf(:Anatomical_Lobe
       ObjectSomeValuesFrom(:regional_part_of :Lung)))
(5) EquivalentClasses(:Region_of_cytoplasm
       ObjectIntersectionOf(:Region_of_cell_component
       ObjectSomeValuesFrom(:regional_part_of :Cytoplasm)))
(6) SubClassOf(:Hand ObjectExactCardinality(1 :laterality
       ObjectOneOf(:individual_right :individual_left)))
```

• Pattern A_of_B. In most cases a name A_of_B is a contraction formed from A and B, that omits some property *p* relating the entities A and B. The idea for providing semantics to entities A_of_B is to build a class expression from that relation. The missing property is recovered from the list of property restrictions attached to the class. For example, axiom (4) expresses that Lobe_of_Lung refers to all *anatomical lobe* that are a *regional_part_of* some *lung*. A particular process is defined for A_of_B where A is Region, Zone, Segment, Subdivision.

[3] In a very few cases, ambiguity was solved via discussions with FMA's authors

From FMA authors, all 'region' classes of the FMA denote *regional parts,* further distinguished on the types of boundary used to define the region, for example *Organ segment* is an organ region with one or more anchored fiat boundaries, *Organ zone* is an organ region with one or more floating fiat boundaries. At the moment, the p handled are only the `part_of` properties and subproperties (e.g.; `regional_part_of`) but this will next be extended to other relationships.

(b) *Axioms. The lexical patterns are not only used for class definitions, but also for handling - creating/removing/moving - axioms:*

Disjointness and subclass axioms. While the sibling symetrization process provides semantics to classes of pattern `P_A`, it operates other tasks at the same time: 1° it adds relevant subclass axioms. 2° it detects and repairs errors or omission in the native FMA (for details see Algorithm 1). For example, while the meaning of `Left_Hand` is formalized by the equivalent class axiom (3) meanwhile, several subclassOf axioms are added: for example axiom (6) asserts that each hand necessary has exactly one left or right laterality and the axiom `DisjointClasses(`*`Left Right`*`)`, states that nothing can be both left and right. In fact, for each modality, only one single disjointness axiom is created to state that nothing can have two opposite modalities. Hence, all `Left_A` and `Right_A`, e.g.; left and right hands are inferred to be exclusive, and much less axioms are used. The algorithms implemented for each pattern are quickly sketched below.

Algorithm 1. The process for symmetrical siblings first parses all names of classes to get the terms matching a specific prefix `P_` where `P` is a subclass of `Primary_Anatomical_coordinate` (e.g. `Left`). For each class `P_A`, (e.g. `Left_A/Right_A`), if `A` exists and `A` (or `Anatomical_A`) is a direct superclass of `P_A`, then several axioms are created respectively for `P_A`, its sibling and its father, according to the following rules: (1.1) each time A has a child `P_A`, A should have the pair as children, unless exceptions; (1.2) each time `A` has two symmetrical children, e.g.; `Left_A` and `Right_A`, and `A` has an existential restriction on a *part* property or subproperty, the two siblings should have symmetrical restrictions (modulo symmetry); (1.3) if a (symmetrical) restriction is present in two symmetrical siblings but not in their direct superclass, the relevant abstracted restriction is added to it. For example, as `Left_Hand` and `Right_Hand` have restrictions `ObjectSomeValuesFrom(:`*`constitutional_part: Investing+fascia+of+left+hand`*`)` (resp. *`Investing+fascia+of+right+hand`*), the missing axiom `subclassOf(`*`Hand`* `ObjectSomeValuesFrom(:`*`constitutional_part`*`:`*`Investing+fascia+of+ +hand`*`))` is created; (1.4) as explained above, for each `P_A`, two axioms are created: a class `axiom EquivalentClasses(:`*`P_A`* `ObjectIntersectionOf(:P :A))` and a subclassOf axiom like (6) for example, which asserts that each A necessary has exactly one left or right laterality `SubClassOf`*`(:A`* `ObjectExactCardinality`*`(1 :laterality`* `ObjectOneOf`*`(:individual_Left :individual_Right)))`.

*Algorithm 2***.** Similarly, the process first parses all names of classes to get the terms that match the pattern `A_of_B`. The `EquivalentClasses(:`*`A_of_B`* `ObjectIntersectionOf(`*`:A`* `ObjectSomeValuesFrom`*`(:p_of :B)`*`))` axiom is created in any of the following cases: (2.1) if the direct superclass of `A_of_B` is `A` or `Anatomical_A` and `A` has a restriction on a *part_of* property or subproperty `p_of`: `SubClassOf(:A ObjectSomeValuesFrom`*`(:p_of :B'))`* with `B'` direct superclass of B (e.g. Ganglion_of_cranial_nerve). (2.1b) if B' is not a direct superclass of B (it may be a distant ancestor) but A or Anatomical_A exists and B' has a restriction for the *inverse* p of p_of: SubClassOf(:B' ObjectSomeValuesFrom*(:p :* A_of_B')); (2.2) if the direct superclass of A_of_B is A or Anatomical_A and A_of_B has a restriction SubClassOf(:*A_of_B* ObjectSomeValuesFrom*(:p_of : B*)) (e.g. Tendon_of_biceps_femoris). For example, as the direct superclass of Lobe_of_Lung is Anatomical_Lobe and Lobe_of_Lung is a subclass of regional_part_of some Anatomical_Lobe the axiom EquivalentClasses*(:*Lobe_of_Lung ObjectIntersectionOf*(:*Anatomical_Lobe ObjectSomeValuesFrom*(:regional_part_of: Lung)*)) is created. (2.3) A specific process handles classes A_of_B where A is *Region_of, Zone_of, Segment_of, Subdivision_of* (1273 classes). It defines A_of_B as regional_part of B, like axiom (5) for Region_of_cytoplasm.

- Completing or compacting axioms. In canonical anatomy, if an entity A has some part B, then reversely B should also have some part A (which is not logically equivalent). 669 missing subclassOf axioms expressing such 'symmetrical' restrictions are created. On the other hand, based on inference, several axioms are removed: if all the subclasses of A have a same existential restriction, it is removed from the subclasses and moved up to A.

(c) Properties characteristics. OWL 2 allows new characteristics of object properties. According to FMA authors, *part, regional_part, constitutional_part, systemic_part, member* and their inverse are asserted to be transitive, irreflexive, asymmetric, *continuous_with* and *connected_to* are symmetric, and *continuous_with* is reflexive.

3 The FMA-OWLizer Tool

The third issue to tackle was to offer a formalization tool that can deal with the sheer size and the frequent updates of the FMA. FMA-OWLizer is a friendly and easy to use tool that automatically generates on a simple clic a 'standard' FMA ontology in OWL[4]. It can process *all* existing public FMA versions, FMA 2005 version, FMA3.0 (2008), April 2010 FMA 3.1 update. It is highly flexible, allowing providing also a customized ontology adapted to the users' needs for and their application. The main parameters are selected via a friendly *graphical user interface* (http://www.lirmm.fr/tatoo/IMG/pdf/FMA-OWLizer.pdf), while the other ones are configured in configuration files. For example, the file 'classes_to_delete.txt' state the classes to be removed. FMA-OWLizer includes many options. It is possible to select the chosen source file as input, to have metaclasses or not, to choose the properties to be included, to customize the class and property axioms in various ways: to supply particular class definitions by designating the properties, e.g.; *constitutional_part,*

[4] Which one should be the 'standard', the ontology with or without metaclass, depends on the native FMA, mainly of the future improvements of its templates.

bounded_by etc for the equivalent classes axioms, to include/remove all the subclass axioms (e.g. for performance tests), to configure properties characteristics. For example, to get an OWL 2 DL ontology that reasoners can process, it is recommended to select 'ignore irreflexive and asymetric'. Otherwise, as the properties *part* and their inverse are transitive, asymmetric and irreflexive, the ontology would violate the OWL 2 restriction that only simple roles can be used in asymetric and irreflexive object property axioms. It is also possible to choose the concrete syntax used to store the ontology (RDF/XML, OWL/XML, Functional Syntax), to select French or English for the GUI. FMA-OWLizer is a local Java program designed and developed specifically for the FMA. All processes are performed via the OWL API 3.0, benefiting of its functionalities. The GUI is achieved with the Swing/AWT Java graphics libraries and is multilingual support (bundle files) thanks to the CISMeF Utils platform.

4 Results

Table 1: Metrics of FMA-OWL ontologies

File	Size	Classes	Class axioms	Expressivity
FMA-OWL 1 from FMA 2005				
#1 without N&S	41,6	41648	236208	$\mathcal{ALCOIF}(\mathcal{D})$
#2. with N&S	40,8	41648	230690	$\mathcal{ALCOIF}(\mathcal{D})$
FMA-OWL 2 from FMA 3.0 2008				
#3. without MTC.	256	85005	263389	$\mathcal{SROIQ}(\mathcal{D})$
#4. with MTC.	314	85005	261331	$\mathcal{SROIQ}(\mathcal{D})$

Complete representations of the entire FMA are now available in OWL 2. An OWL 2 ontology[5] without metaclasses (*FMA-OWL*2_noMTC Table 1 #3) has been generated from FMA 3.0. It includes all FMA classes and properties (except *homonym_of* and *homonym_for,* discarded in agreement with FMA's authors). The new class definitions and axioms, retain transitivity but voluntarily ignore irreflexivity and asymetry. This ontology offers 15 084 new definitions of classes, 16,113 disjointness axioms; 85,467 initial axioms are removed and replaced by one single axiom (next inherited), 15 subproperties axioms and 228,263 annotations. 7664 class definitions are obtained from the pattern `A_of_B`, while 7333 from the pattern `Left_A/Right_A`. Another OWL 2 ontology *with* metaclasses[6] is also available (*FMA-OWL*2_noMTC Table 1 #4). The FMA-OWLizer tool (§3) can generate both a standard FMA-OWL ontology and other ontologies of various size and complexity to fit specific applications needs. For example, *FMA-OWL v1* (Table 1) are OWL 1 partial (smaller) ontologies (41 Mb) issued from FMA 2005, of which left/right leaves are cut, without (#1) or with (#2) class definitions built from the *constitutional_part* property. Ontologies obtained from the FMA 3.1 update have also been generated. As the FMA

5 http://gforge-lirmm.lirmm.fr/gf/download/docmanfileversion/214/747/FMA_3.0_noMTC_100702.owl.zip
6 http://gforge-lirmm.lirmm.fr/gf/download/docmanfileversion/215/748/FMA_3.0_MTC_100701.owl.zip

is being incrementally developed and repaired, the FMA-OWLizer tool is highly helpful for generating FMA-OWL corresponding updates.

5 Discussion

There has been several efforts since 2005 for translating the FMA in OWL [3] [4] [5]. But earlier conversions to OWL were no fully satisfying. A main limitation of [4] [5] is the metaclasses representation. The difficulty is that translating FMA metaclasses into OWL 1 leads to OWL Full. [4] offers two components: an OWL DL and an OWL Full component. The former is obtained in omitting the metaclasses to remain in OWL DL, thus it is incomplete. The latter is a complete representation that imports the OWL DL module, but is an OWL Full component. In contrast, the FMA ontology [3] is OWL DL, while including the metaclass knowledge. However it had different limitations: the left/right leaves were cut for memory and tools limitation reasons, thus it was incomplete; the class definitions were not "semantically" satisfying for all classes: all anatomical entities cannot be uniformly defined solely in terms of their constitutional parts. The present approach offers more satisfying alternatives: an OWL 1 (2) DL ontology without metaclasses, which captures the metaclass knowledge like [3] but is now complete or a complete OWL 2 ontology where metaclasses are better represented via OWL 2 new metamodeling features. Second, the formalization is pushed much forwards, eliciting further the FMA underlying semantics. Presently, as they are based on lexical patterns, the class definitions and axioms created are semantically correct and reliable from an *anatomical* viewpoint. Besides, while the earlier conversion program [3] did not scale up and was not robust, the new implemented mapping of the syntax now handles the *entire* FMA and can overcome the changes of FMA successive updates. Regarding the pattern approach, a few patterns are shared with [11] but the goal of Abstracting and Generalizing is clearly different, it is to reduce the FMA size or abstract it. Automatically generating axioms is partly shared with [6], but our approach is more general.

However the ontologies produced are still unstable and exhibit errors that should be fixed. Though it was not the objective of this work, we made an attempt to check the FMA-OWL ontologies with a reasoner. But reasoning with FMA-OWL proved to be a real challenge. The FMA-OWL ontologies are perhaps the largest and most complex OWL ontologies available. Firstly no reasoner could classify them. Recently, the special 'core blocking' strategy of HermiT [7] that has been developed for FMA like ontologies with lots of unsatisfiable classes, finally succeeded to process them in a reasonable time. *FMA-OWL* 2 (Table 1 #3 - 2010-03-11) has 65,753 unsatisfiable classes out of 85,005. The time for classification, including loading and preprocessing was 58m 12s 929ms (by Birte Glimm*)*. *FMA-OWL* v1 with constitutional-part for N&S (#1) had 33,433 unsatisfiable classes out of 41,648, and the time for classification was 33m 46s 55ms. At the time of this work, no explanation tool compatible with OWL 2 and the OWL API 3.0 was available. Due to the large size of the FMA-OWL, it was difficult to go further in debugging it. This is obviously an interesting perspective in the future, as soon as friendly OWL reasoning and explanation tools will be available for large ontologies like the FMA.

6 Conclusion

We have presented a method to represent the FMA in OWL 2 and a friendly and flexible tool. Complete representations of the entire FMA in OWL 1 or OWL 2 DL are now available and over 15,500 FMA classes have a reliable logical definition. The tool allows for automatically producing new FMA-OWL updates and variants customized to applications on demand. As the FMA describes anatomical structures composed of complexly interconnected parts, automatic procedures that correctly encode the semantics without risking errors and that DL reasoners can process, is a real challenge. This work is an important step forwards. Future perspectives are to improve and push the formalization in OWL 2 or its extension [12] further, to exploit OWL reasoners and explanation tool to improve the FMA design and maintenance. The pattern approach is an interesting option to be applied also to other ontologies.

References

1. C. Rosse, J.L. Mejino Jr, The Foundational Model of Anatomy Ontology, In Burger A et al. (editors): Anatomy Ontologies for Bioinformatics: Principles and Practice, Springer,.59-118 ISBN 978-1-84628-884-5. 2008, New York.
2. C. Rosse, J.L. Mejino Jr, A reference ontology for biomedical informatics: the Foundational Model of Anatomy, J. Biomed. Inform. 36 (6) (2003) 478–500.
3. C. Golbreich, S. Zhang, O. Bodenreider, The Foundational Model of Anatomy in OWL: experience and perspectives, Journal of Web Semantics, Web Semantics: Science, Services and Agents on the World Wide Web, 4 (3) (2006) 181–195.
4. O. Dameron, D.L. Rubin, M.A. Musen, Challenges in converting frame based ontology into OWL: the Foundational Model of Anatomy case-study, in: AMIA Annual Symposium, Washington DC, 181-185, 2005.
5. Natalya F. Noy, Daniel L. Rubin, Translating the Foundational Model of Anatomy into OWL, Journal of Web Semantics, Web Semantics: Science, Services and Agents on the World Wide Web, 6 (2008) 133–136.
6. O. Dameron and Julie Chabalier. Automatic generation of consistency constraints for an OWL representation of the FMA 10th International Protégé Conference, 2007.
7. Birte Glimm, Ian Horrocks, and Boris Motik, Optimized Description Logic Reasoning via Core Blocking. In Jürgen Giesl and Reiner Hähnle, editors, Proc. of the 5th Int. Joint Conf. on Automated Reasoning (IJCAR 2010), LNCS 6173, pp. 57–471,. Springer.
8. W3C OWL Working Group, OWL 2 Web Ontology Language Document Overview W3C Recommendation 27 October 2009, http://www.w3.org/TR/owl2-overview/.
9. C. Golbreich and Evan K. Wallace, OWL 2 Web Ontology Language New Features and Rationale W3C Recommendation 27 October 2009, http://www.w3.org/TR/owl2-new-features/
10. Darmoni, SJ; Pereira, S; Névéol, A; Massari, P; Dahamna, B; Letord, C; Kedelhué, G; Piot, J; Derville, A & Thirion, B. French Infobutton: an academic and business perspective. AMIA Symp., Pages 920, IOS Press, 2008.
11. Eleni Mikroyannidi , Alan Rector, and Robert Stevens, Abstracting and Generalizing the Foundationql Model of Anatomy ontology, Bio-ontologies 2009.
12. B. Motik, et al. Representing Ontologies Using Description Logics, Description Graphs, and Rules. *Artificial Intelligence*, 173(14):1275-1309, 2009.

Acknowledgments. We are grateful to C. Rosse and O. Mejino for their advice and support and to Oxford team for their specialized classification algorithm, special thanks to B. Glimm.