Artificial Intelligence in Medicine 57 (2013) 119–132

Contents lists available at SciVerse ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

The Foundational Model of Anatomy in OWL 2 and its use

Christine Golbreich [a,*], Julien Grosjean [b], Stefan Jacques Darmoni [b]

[a] *Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, Centre National de la Recherche Scientifique 5506, Montpellier & University of Versailles Saint-Quentin, 78035 Versailles, France*

[b] *CISMeF, Rouen University Hospital & Information Processing in Biology & Health, Computer Science Information Processing, and Systems Laboratory, University of Rouen, 76031 Rouen, France*

ARTICLE INFO

Article history:
Received 9 October 2011
Received in revised form
12 November 2012
Accepted 30 November 2012

Keywords:
Semantic Web
Description Logic
Web Ontology Language
Ontology
Terminology
Health portal
Anatomy

ABSTRACT

Objective: The objective is to represent the Foundational Model of Anatomy (FMA) in the OWL 2 Web Ontology Language (informally OWL 2), and to use it in a European cross-lingual portal of health terminologies for indexing and searching Web resources. Formalizing the FMA in OWL 2 is essential for semantic interoperability, to improve its design, and to ensure its reliability and correctness, which is particularly important for medical applications.
Method and material: The native FMA was implemented in frames and stored in a MySQL database backend. The main strength of the method is to leverage OWL 2 expressiveness and to rely on the naming conventions of the FMA, to make explicit some implicit semantics, while improving its ontological model and fixing some errors. Doing so, the semantics (meaning) of the formal definitions and axioms are anatomically correct. A flexible tool enables the generation of a new version in OWL 2 at each Protégé FMA update. While it creates by default a 'standard' version of the FMA in OWL 2 (FMA-OWL), many options allow for producing other variants customized to users' applications. Once formalized in OWL 2, it was possible to use an inference engine to check the ontology and detect inconsistencies. Next, the FMA-OWL was used to derive a lightweight FMA terminology for a European cross-lingual portal of terminologies/ontologies for indexing and searching resources. The transformation is mainly based on a reification process.
Result: Complete representations of the entire FMA in OWL 1 or OWL 2 are now available. The formalization tool is flexible and easy to use, making it possible to obtain an OWL 2 version for *all* existing public FMA. A number of errors were detected in the native FMA and several patterns of recurrent errors were identified in the original FMA. This shows how the underlying OWL 2 ontology is essential to ensure that the lightweight derived terminology is reliable.

The FMA OWL 2 ontology has been applied to derive an anatomy terminology that is used in a European cross-lingual portal of health terminologies. This portal is daily used by librarians to index Web health resources. In August 2011, 6481 out of 81,450 health resources of CISMeF catalog (http://www.chu-rouen.fr/cismef/ – accessed 29.08.12) (7.96%) were indexed with at least one FMA entity.
Conclusion: The FMA is a central terminology used to index and search Web resources. To the best of our knowledge, neither a complete representation of the entire FMA in OWL 2, nor an anatomy terminology available in a cross-lingual portal, has been developed to date. The method designed to represent the FMA ontology in OWL 2 presented in this article is general and may be extended to other ontologies. Using a formal ontology for quality assurance and deriving a lightweight terminology for biomedical applications is a general and promising strategy.

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 more than 120,000 terms. Most entities are anatomical structures composed of many parts

* Corresponding author at: Department of Computer Sciences, University of Versailles Saint Quentin, 45 Av des Etats Unis, 78035 Versailles, France. Tel.: +33 467418653; fax: +33 467418500.
E-mail addresses: cgolbrei@gmail.com (C. Golbreich), Julien.Grosjean@chu-rouen.fr (J. Grosjean), Stefan.Darmoni@chu-rouen.fr (S.J. Darmoni).

0933-3657/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.artmed.2012.11.002

interconnected in complex ways, described, for example, in terms of their regions, constituents, innervations, blood vessels, and boundaries. For example, a *Heart* has two regions (its *Left side* and *Right side*), several constitutional parts (e.g., *Wall of heart*, *Interatrial*, *Interventricular*, *and Atrioventricular septum*, *Mitral valve*), and is innervated by the *Deep cardiac plexus*, *Right* and *Left coronary nerve plexus*. For this reason, the FMA is not only very large but it is, perhaps, one of the most complex ontologies in the biomedical sciences.

OWL 2 is the W3C standard for ontologies on the Semantic Web [3]. OWL 2 provides several advantages for Life Sciences ontologies: interoperability, important for shared use across different domains; semantics, meaning of terms is formally specified thanks to the underlying logic; and reasoning services. Once converted to OWL 2, ontologies are easily connected or combined with other ontologies. Another practical benefit is that OWL 2 allows for the exploitation of the many existing OWL tools, in particular powerful *reasoners*. Furthermore OWL 2's higher *expressiveness*, in particular its new metamodeling abilities, is of major interest.

The main objective of this work was to represent the FMA in OWL 2. A first motivation was to make it interoperable with the increasing number of OWL ontologies available. Formalizing the FMA ontology in OWL 2 provides a precise and rigorous meaning to anatomical entities, which is crucial to share or link multiple resources. Another motivation was to enable OWL 2 reasoning services and tools to assist the FMA design and its maintenance. Finally, another important motivation was to derive from the FMA ontology a lightweight, *quality-controlled* terminology of human anatomy for a cross-lingual portal of European terminologies [4]. Terminologies are fundamentally different from ontologies. While rich ontologies are useful for defining semantics and to check for consistency, lightweight terminologies are often enough for applications. In particular they are much faster to index and search for Web resources or data. While it is often the case that applications only use the derived terminology, the underlying OWL 2 ontology is essential to ensure that the lightweight derived terminology is reliable.

The FMA has specificities that make its formalization in OWL (DL) – a sublanguage of OWL designed to be decidable – a special and tricky task. Indeed, the FMA is huge and highly complex, due to its domain – human anatomical structures interconnected in complex ways – and to its Protégé frames source model, where each anatomical structure is modeled both as a class and metaclass. The presented work does not consist in simply converting (for example by a script) the FMA from one format to another, but it leverages the underlying description logic of OWL to enrich FMA entities with formal definitions and axioms having a sound anatomical meaning. A strength of the method consists in exploiting naming conventions and lexical patterns of the native FMA to make explicit the implicit semantics (meanwhile improving its ontological model and fixing some errors). A second originality is that this tool makes it possible to generate a new version each time the Protégé FMA is updated by its authors. Those who are not very familiar with OWL and prefer to continue to use the existing Protégé frame editor will find this tool easy to use to automatically create the OWL conversion. Additionally, while it is possible to provide one 'standard' version in OWL 2, with or without metaclasses, there are several options to easily produce customized applications if needed.

Reasoning with FMA is highly difficult. Since 2005 existing conversions were incomplete and led to performance problems with all existing OWL tools (e.g., editors such as Protégé and reasoners, such as Racer, FACT, and HermiT). Though it was not the objective of this work, we made an attempt to check the FMA ontology with a reasoner. After extracting smaller modules from the FMA with an OWL-based tool, it was possible to classify them, to display the inferred hierarchy with Protégé 4, and to detect and repair some errors in the FMA original design. But, reasoning with the FMA and explaining its errors remain a real challenge.

Finally, we present an application of the FMA OWL 2 ontology: the lightweight FMA terminology derived from it for a cross-lingual portal of European terminologies/ontologies used for indexing and search of Web resources.

2. Method

The FMA ontology is implemented in Protégé frames (the frame-based system developed by Stanford Center for Biomedical Informatics Research) and stored in a MySQL database backend. Transforming it into OWL 2 is not done by simple translation. Rather, it is necessary to specify the meaning of its terms in logic and to express, by logical statements (axioms), some knowledge about the anatomical entities that is not explicit in the native FMA. This raises several issues. First, different types of information are embedded in Protégé FMA. Apart from the domain knowledge concerning anatomical entities, the FMA also includes metalevel knowledge. Interpreting both types of knowledge in the same model can lead to undesired consequences because of their interactions. Two solutions are proposed here: an OWL 1 DL ontology *without* metaclasses and an OWL 2 ontology *with* metaclasses (Section 2.1). The second challenge is to guarantee that the formal definitions and axioms created are semantically correct from an *anatomical* viewpoint. The method proposed is based on lexical patterns (Section 2.2). Third, given the large size of the FMA, it is essential to *automatically* generate the OWL axioms. A friendly tool has been achieved for this purpose (Section 3) that enables it to create (by default) a 'standard' ontology of the FMA in OWL 2 (FMA-OWL) from the FMA frames and other customized variants useful for specific applications.

2.1. Metamodeling

The original representation of FMA in Protégé frames used an unusual representation in which each anatomical entity was represented both as a class and a metaclass ("... for enabling the selective inheritance of attributes" [1,2,7]). At the domain level, classes describe anatomical entities, while at the metalevel metaclasses serve several purposes. First, they associate metadata to anatomical entities. For example, they attach to the class *Heart* its author 'JOSE MEJINO, MD', preferred-terms, 'Heart' in English, 'Cor' in Latin, the non-English equivalent 'Coeur' in French, its definition, synonyms, FMAID, etc. Second, metaclasses also serve to define 'templates' for some classes, which specify 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. For example, the metaclass *Heart* is a subclass of *Organ with cavitated organ parts*, itself a subclass of *Organ*, of which it inherits the slots, facets e.g., *bounded by* with range *Surface of organ*, or *arterial supply* with range *Artery*, *Arteriole*, *Arterial plexus*, etc. At the class level, slots, e.g., *part of*, *bounded by*, *arterial supply*, are assigned particular values (e.g., *Surface of heart*). Thus, an anatomical entity, e.g., a canonical *Heart*, is specified as being an *Organ with cavitated organ parts* with a particular structure fulfilled by its individuals: any heart is composed of a *Right atrium*, *Left atrium*, *Right ventricle*, *Left ventricle*, is *bounded by* a *Surface of heart*, has a *Right coronary artery* and *Left coronary artery*, etc., as *arterial supply*. To avoid undesired effects caused by interpreting both types of knowledge in the same model, it is offered as having either an OWL 1 (OWL 2) DL ontology *without* metaclasses (but otherwise capturing their knowledge), or an OWL 2 ontology *with* metaclasses:

2.1.1. OWL 1 DL ontology without metaclasses

An OWL 1 DL Ontology *without metaclasses* was initially proposed earlier [5], before OWL 2. OWL DL requires the deletion of the FMA higher order structure. The solution that was adopted to remain at first order, while still capturing the information embedded in metaclasses, was to replace metaclass instantiations by subclass relations, which transforms metaclasses into ordinary OWL classes. 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*." [2]. As metaclasses specify a given "template" of classes while classes specify the structure of their instances, property restrictions at metaclasses are approximated by universal restrictions dedicated to limit the allowed types, while restrictions at classes are translated into existential restriction (for details and discussion see [5]).

2.1.2. OWL 2 ontology with metaclasses

Now, thanks to the OWL 2 metamodeling new features, punning, and enhanced annotations [3,6], it is possible to have an OWL 2 Ontology *with metaclasses*, which (partly) better reflects the FMA authors' original design. Indeed, while OWL 1 DL required a strict separation between the names of classes and individuals, OWL 2 relaxes this separation [3,6]. At present, punning makes it possible to use the same term to refer to a class and an individual, while retaining decidability. Thus, it is possible to keep metaclasses that reflect more accurately the FMA templates: the name `Heart` can be used both for the metaclass `Heart` and for the class `Heart` instance of `Organ with cavitated organ parts`. 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 also allows for annotations of axioms and of annotations. In FMA frames, properties such as preferred name, synonyms, and non-English equivalents, are modeled as slots, whose values are not strings but individuals of the Concept name class. As they do not concern data of the anatomy domain but metadata, using OWL 2 annotations of annotation is more appropriate than metaclasses. Thus, the domain and meta-level data are no longer confused and do not interact, and a huge number of individuals are removed. For example, the class `Heart` (Fig. 1 line 1) is annotated by the label "`Coeur"@fr` (Fig. 1 line 4) and the labeling itself is annotated by its creator JOSE MEJINO MD (Fig. 1 line 2), by its date (Fig. 1 line 3), FMAID "217079" (Fig. 1 line 4), publisher, etc. (See the Protégé display Fig. 12.)

2.2. Formal semantics

The second main challenge is to enrich the FMA with formal definitions and axioms that have an anatomically correct meaning. The formalization is achieved in two steps: first, the transformation of the FMA frames syntax and second, the transformation of the FMA 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 implicit 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 ($\mathcal{FMA} - \mathcal{OWL}\,v1$); the second step brings it to OWL 2 ($\mathcal{FMA} - \mathcal{OWL}\,2$).

The transformation of the frames syntax in OWL reuses the 2005 rules [5]. In short, Protégé classes and slots are converted into OWL classes and properties, with a specified domain and range. Slot characteristics (inverse, symmetric, functional) are translated using corresponding OWL constructs. Protégé frames uses so-called "own slots" [7] for representing information about a class that is not inherited by its subclasses or instances. This can be editorial annotations, in which case these values are converted to OWL annotation properties and values, or, in some cases, they can convey domain information, in which case they are converted to existential restrictions (existential retrictions are inherited by subclasses, but this does not raise an issue, cf. [5] for extensive discussion and example Section 4.1 (2)). Property restrictions, defined at metaclasses or classes, are transformed into universal or existential property restrictions, respectively. Metaclass instantiation is replaced by a subclass relation.

At the second step, the logical formalization is pushed forward and the FMA ontology is enriched in several ways: (a) classes definitions are automatically generated from lexical patterns; (b) numerous related axioms are automatically created or moved; (c) new properties characteristics are added; and as presented above Section 2.1; (d) OWL annotations of annotation are used for metadata and (e) OWL 2 metaclasses are created, which can also 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 [5]. Now new formalization rules are defined for creating the 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` denotes all the regional parts of cytoplasm, etc. The new rules create different forms of definition depending on the pattern. The patterns are unambiguous and moreover, their meanings were resolved in a few cases by checking with FMA authors. Therefore, 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., and (ii) Pattern `A_of_B` denoting parts of entity, e.g., `Lobe_of_Lung`. Classes are incrementally defined as follows.

- Pattern `P_A`. First, the *Anatomical_coordinate* subclasses are defined. *Primary_Anatomical_coordinate* are specified via property value restrictions, for example, axiom (1) Fig. 2 states that *`Left`* denotes all objects with left laterality. `Binary_Anatomical_coordinate` are defined as an intersection of *Primary_Anatomical_coordinate* classes.

```
(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)
```

Fig. 1. OWL 2 annotations.

```
(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)))
(7) DisjointClasses(Left Right)
```

Fig. 2. OWL 2 axioms created by the formalization process.

For example axiom (2) Fig. 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) Fig. 2 states that `Left_Hand` (resp. `Right_Hand`) denotes all hands having left laterality.

- 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 two entities A and B. The idea for providing semantics to entities A_of_B is to build a class expression from that implicit relation. The missing property 'p' is recovered from scanning the list of property restrictions attached to the class. For example, it is *regional_part_of* for `Lobe_of_Lung`. Axiom (4) Fig. 2 states that `Lobe_of_Lung` denotes all the *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 or Subdivision. From FMA authors, all 'region' classes denote *regional parts*, further distinguished on the type of boundary used to define the region, for example *Organ segment* is a region with one or more anchored fiat boundaries, *Organ zone* is a region with one or more floating fiat boundaries. At the moment, the properties handled are only the `part_of` properties and subproperties (e.g., `regional_part_of`) (Fig. 3), but this will next be extended to other relationships.

(b) *Axioms*. Numerous axioms are automatically generated or (re)moved. A first set is based on the pattern `P_A`, a second one is based either on anatomical specificities or on OWL semantics, as described below:

- Disjointness and subclass axioms. The lexical pattern `P_A`, not only serves for class definitions, but also for creating, moving or removing disjointness and subclass axioms. While the sibling symetrization process provides semantics to classes of pattern `P_A`, it achieves 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 *Process 1*). For example, while the meaning of `Left_Hand` is defined by the equivalent class axiom (3) Fig. 2, meanwhile several axioms are created: axiom (6) asserts that each hand necessary has exactly one left or right laterality. For each modality, a disjointness axiom is created stating that nothing can have two opposite modalities. For example, axiom (7) `DisjointClasses(Left Right)` states that nothing can be both left and right. Hence, from one single disjointness axiom, many classes are inferred to be disjoint: all `Left_A` and `Right_A`, e.g., left and right hands are inferred to be disjoint. Thus, many fewer disjointness axioms are asserted, since they can be inferred.
- Completing or compacting axioms. Other axioms are created as well. In canonical anatomy, if an entity A has some part B, then reversely B should also be a part of some A (which is not logically equivalent). 669 missing `SubClassOf` axioms expressing such 'symmetrical' restrictions are created based on the FMA `'part'` properties (Fig. 3). On the other hand, based on OWL semantics, several axioms have been removed, because they can be inferred: if all the subclasses of A have the same existential restriction, it is removed from the subclasses and moved up to A instead.

(c) *Property characteristics*. OWL 2 offers new property characteristics that are added in FMA-OWL. According to FMA authors,

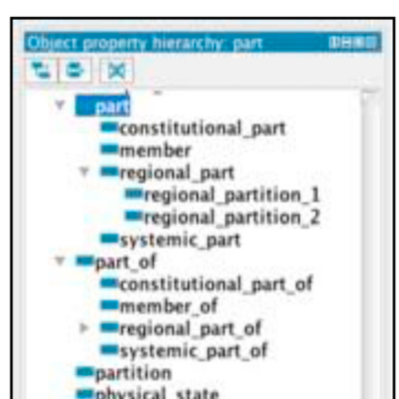

Fig. 3. Part-whole OWL 2 object properties.

part, *regional_part*, *constitutional_part*, *systemic_part*, *member* and their inverse are transitive, irreflexive, and asymmetric, *continuous_with* and *connected_to* are symmetric, and *continuous_with* is reflexive.

The process implemented for each pattern is outlined below.

Process 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*) (respectively *Investing+fascia+of+right+hand*), the missing axiom subclassOf(*Hand* ObjectSomeValues-From(:*constitutional_part:Investing+fascia+of+hand*)) is created;
(1.4) as explained above, for each P_A, two axioms are created: first a class axiom EquivalentClasses(:*P_A* ObjectIntersectionOf(:P:A)) and second, a subclass axiom like (6), which asserts that each A necessarily has exactly one left or right laterality ubClassOf*(:A* ObjectExactCardinality*(1:laterality* ObjectOneOf*(:individual_Left:individual_Right)))*.

Process 2.
Similarly, the process first parses all names of classes to get the terms that match the pattern A_of_B.
The class axiom EquivalentClasses(:*A_of_B*ObjectIntersectionOf(*:A* ObjectSomeValuesFrom*(:p_of:B)*)) 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 classesA_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.

3. FMA-OWLizer tool

The third issue was to achieve a formalization tool that can deal with the sheer size and the frequent incremental updates of the FMA. FMA-OWLizer is a friendly and easy to use tool that automatically generates, via a simple click a 'standard' FMA ontology in OWL (which one should be considered as the 'standard' ontology, that one with or without metaclass, depends on the native FMA, and mainly of the future improvements of its templates). It can process *all* existing public FMA versions, FMA 2005 version, FMA3.0 (2008), April 2010 FMA 3.1 update. It is highly flexible and provides a customized ontology adapted to the users' needs and 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' states 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*, *bounded_by*) for the equivalent classes axioms; to include/remove all the subclass axioms (e.g., for performance tests); and to conFig. property characteristics. For example, to get an OWL 2 DL ontology that reasoners can process, it is recommended to select 'ignore irreflexive and asymmetric' (Fig. 4). Otherwise, as the property's *part* and inverse are transitive, asymmetric, and irreflexive, the ontology would violate the OWL 2 requirement that only simple roles can be used in asymmetric and irreflexive object property axioms. It is also possible to choose which concrete syntax is used to store the ontology (RDF/XML, OWL/XML, Functional Syntax) and to select French or English for the GUI. FMA-OWLizer is a local Java program designed

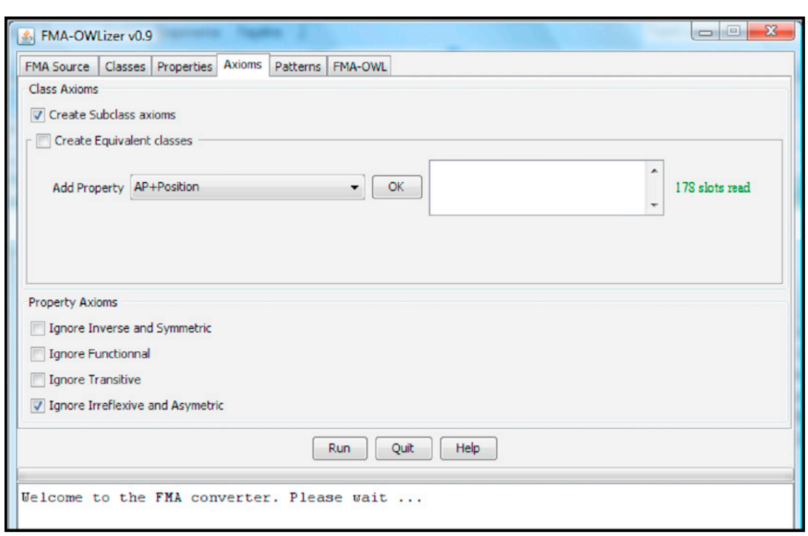

Fig. 4. FMA-OWLizer: selection of axioms.

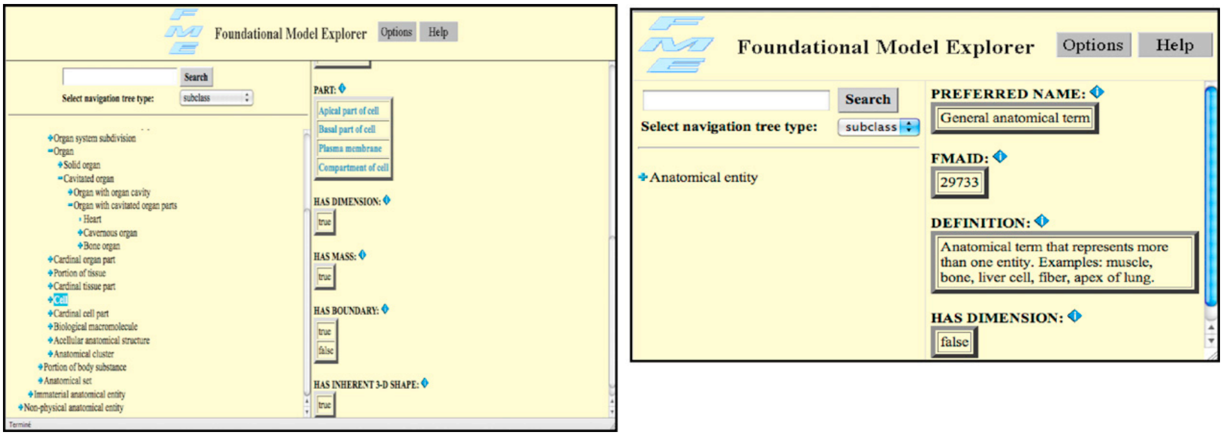

Fig. 5. FME display: (a) left part: Cell with two opposite values for 'has boundary' and (b) right part: General Anatomical Term with 'has dimension' value 'false'.

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. Reasoning with FMA-OWL

A first important benefit of FMA-OWL is that the ontology being formalized in OWL can now be checked with an automatic inference engine to detect its inconsistencies. Though it was not the objective of this work, an attempt was made to check the FMA-OWL ontology with a description logic reasoner. This proved challenging because of the large size and complexity of the FMA-OWL ontologies. At first, no reasoner could classify them. Then, the special 'core blocking' strategy of HermiT [8], which was developed for FMA-like ontologies with numerous unsatisfiable classes, succeeded in processing them in a reasonable time, but from the command line. $\mathcal{FMA}-\mathcal{OWL}\ 2$ (Table 1 #3 – 2010-03-11) has 65,753 unsatisfiable classes out of 85,005. The time for classification, including loading and preprocessing was 58 m 12 s 929 ms (performed by Birte Glimm). $\mathcal{FMA}-\mathcal{OWL}\ v1$ with constitutional-part for N&S (#1) had 33,433 unsatisfiable classes out of 41,648, and the time for classification was 33 m 46 s 55 ms. However, at the time of this work, it was not possible to run HermiT from Protégé 4.1, to visualize the inferred results and identify unsatisfiable classes.

Therefore, at a second step, it was decided to use FMA modules instead of the entire FMA, to incrementally identify and investigate the origin of unsatisfiable classes. The method consisted of extracting smaller or less complex modules with an OWL modules extractor tool [9] to locate errors. A module of type 'low module.' with the signature 'Organ,' was selected for a first study. (For details on different types of modules and the module extractor see [9].) It was then possible to classify it with HermiT in a reasonable time, and to visualize the inferred hierarchy and the unsatisfiable classes with Protégé 4.1. A sample of modeling errors found in this way and some classical schema of explanation are presented below.

First, it was found that a large number of unsatisfiable classes resulted from the assertion of reflexivity of the property `ContinuousWith` (FMA authors' private communication). Therefore, this feature was removed. The search for errors showed that most inconsistencies resulted from conflicting values of data properties, issued from different origins, as illustrated below.

4.1. Conflicting values at instance and class levels

A common cause of unsatisfiability is the conflicting values defined for a class at instance and class levels in the original FMA frames. A class having a boolean value defined in its own slot and the opposite value at its superclasses is unsatisfiable in OWL.

Example 1.

1. `Cell SubClassOf has_boundary value false`
2. `Cell SubClassOf has_boundary value true`
3. `Functional: has_boundary`

`Cell` is unsatisfiable (hence, so are all its subclasses) because of the two conflicting axioms 1 and 2:

Axiom (1) comes from the own slot `has_boundary` assigned with `false` in the FMA at instance (FMA.pins file), while axiom (2) comes from the value `true` asserted at its superclass `Material_anatomical_entity`. The property `has_boundary` is functional (3), because it was defined as a single-slot in FMA frames, hence the inconsistency. It can be noticed that the Foundational Model Explorer (FME) (http://fme.biostr.washington.edu/FME/index.html, accessed 29.08.12) displays this error online (Fig. 5a). To repair it, the value `false` asserted to the instance `Cell` should be removed. This repair is consistent with the axiom `Cell bounded by some Surface_of_Cell`.

4.2. Conflicting values of the `has_dimension` *property*

A frequent source of unsatisfiability is related to conflicting values of the data property `has_dimension`, which seems to be related to a modeling error in FMA regarding `General Anatomical Term` and `General Anatomical Term Template`, for which `has_dimension` has the value `false` (Fig. 5b) (respectively, as instance or subclass of `Non Physical Anatomical Entity Template)`. After removing these axioms, the number of unsatisfiable classes was reduced to 66517.

4.3. Conflicting global and local ranges

Another frequent case of unsatisfiability is due to conflicting values of the global range of a property and a local existential restriction: a property p that has an existential restriction to some class B (s1) for a class A, while the range of p is a class B' (s2) disjoint from B (s3) is unsatisfiable (s4) (Schema 1). Schema 1 is a basic schema of inference related to the range.

Schema 1
if
```
s1. p some B
s2. p range B'
s3. B DisjointWith B'
```
then
```
s4. p some B SubClassOf Nothing
```

```
1. First_metacarpal_bone SubClassOf contained_in some Thenar_compartment
2. Thenar_compartment SubClassOf Material_anatomical_entity
3. Material_anatomical_entity SubClassOf has_mass value true
4. contained_in Range Anatomical_space
5. Anatomical_space SubClassOf Immaterial_anatomical_entity
6. Immaterial_anatomical_entity SubClassOf has_mass value false
7. Functional has_mass
```

Fig. 6. Extract of the FMA axioms explaining the unsatisfiability of `First_metacarpal_bone`.

Indeed, from (s1) each individual of `p some B` is related by p to at least an individual of B, from (s2) each individual target of p is necessarily an individual of B', from (s3) B is disjoint from B', hence `p some B` is unsatisfiable and so are all its subclasses A.

Example 2.

1. First_metacarpal_bone SubClassOf contained_in some Thenar_compartment
2. Thenar_compartment SubClassOf Material_anatomical_entity
3. Material_anatomical_entity SubClassOf has_mass value true
4. contained_in Range Anatomical_space
5. Anatomical_space SubClassOf Immaterial_anatomical_entity
6. Immaterial_anatomical_entity SubClassOf has_mass value false
7. Functional has_mass

Fig. 6. Extract of the FMA axioms explaining the unsatisfiability of First metacarpal bone.

Unsatisfiability of `First_metacarpal_bone` includes several steps of inference. But as shown below, it basically relies on the above Schema 1, which leads to conflicting values of the data property `has_mass`. First, `contained_in some Thenar_compartment` is unsatisfiable. Indeed, the range of `contained_in` is `Anatomical_space` (line 4 Fig. 6) and `First_metacarpal_bone SubClassOf contained_in some The- nar_compartment` (line 1 Fig. 6). Hence according to Schema 1, since `Thenar_compartment` and `Anatomical_space` are disjoint, `contained_in some Thenar_compartment` is unsatisfiable, and so are all its subclasses.

Then it comes that `First_metacarpal_bone` is unsatisfiable because the property `contained_in` has an existential restriction to `Thenar_compartment` (line 1 Fig. 6), which is a descendant of `Material_anatomical_entity` (line 2 Fig. 6), for which `has_mass value true` (line 3 Fig. 6), while the range of `contained_in` is `Anatomical_space`, a subclass of `Immaterial_anatomical_entity` (line 5 Fig. 6), for which `has_mass value false` (line 6 Fig. 6). However, `has_mass` is functional (line 7 Fig. 6), hence it cannot be related to two different values.

4.4. Conflicting values from domain

Another case of unsatisfiability is issued from the domain asserted for a property, which is conflicting with other knowledge of the ontology. A basic inference related to the domain is that if a property p has an existential restriction to some class B for a class A (s5), and the domain of p is a class A' (s6) then A is inferred to be a subclass of A'(s7) (Schema 2).

Schema 2
if
`S5. A SubClassOf p some B`
`S6. p domain A′`
then
`S7. A SubClassOf A′`

Example 3. The explanation of `Portion_of_cytosol` unsatisfiability includes several steps of inference. But as shown below, it basically relies on Schema 2, which leads to conflicting values of the data property `has_inherent_3-D_shape`:

From lines 19 and 20 (Fig. 7), `surrounds` is the inverse of `surrounded_by`, which range is `Anatomical_structure or Serous_sac or Zone_of_heart`, thus its domain is inferred to be `Anatomical_structure or Serous_sac or Zone_of_heart (s6)`. On the other hand, from line 17 `Portion_of_cytosol SubClassOf surrounds some Cell_nucleus (s5)`. Hence, from Schema 2 it comes `Portion_of_cytosol SubClassOf`

```
1.  Functional: has_inherent_3-D_shape
2.  Anatomical_structure SubClassOf has_inherent_3-D_shape value true
3.  Cardinal_organ_part SubClassOf Anatomical_structure
4.  Organ_region SubClassOf Cardinal_organ_part
5.  Organ_zone SubClassOf Organ_region
6.  Zone_of_heart SubClassOf Organ_zone
7.  Organ SubClassOf Anatomical_structure
8.  Cavitated_organ SubClassOf Organ
9.  Organ_with_organ_cavity SubClassOf Cavitated_organ
10. Serous_sac SubClassOf Organ_with_organ_cavity
11. Cell SubClassOf constitutional_part some Compartment_of_cell
12. Compartment_of_cell SubClassOf constitutional_part some Cytoplasm
13. Cytoplasm SubClassOf attributed_constitutional_part some fm_live_14758
14. Portion_of_body_substance SubClassOf has_inherent_3-D_shape value false
15. Portion_of_cell_substance SubClassOf Portion_of_body_substance
16. Portion_of_cytosol SubClassOf Portion_of_cell_substance
17. Portion_of_cytosol SubClassOf surrounds some Cell_nucleus
18. fm_live_14758 SubClassOf related_part some Portion_of_cytosol
19. surrounded_by InverseOf surrounds
20. surrounded_by Range Anatomical_structure or Serous_sac or Zone_of_heart
```

Fig. 7. Extract of the FMA-OWL axioms (logically reorganized) justifying the unsatisfiability of `Portion_of_cytosol, fm_live_14758, Compartment_of_cell, Cell`, etc.

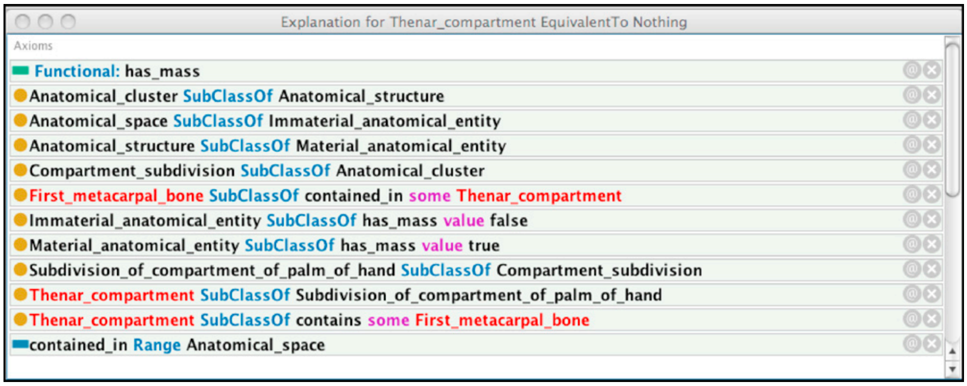

Fig. 8. Protégé explanation for `Thenar_compartment` unsatisfiability.

`Anatomical_structure or Serous_sac or Zone_of_heart (s7)`. However, from lines 14-15-16 `Portion_of_cytosol SubClassOf has_inherent_3-D_shape value false`, while `Anatomical_structure or Serous_sac or Zone_of_heart SubClassOf has_inherent_3-D_shape value true`. Indeed, from lines 7-8-9-10 `Serous_sac SubClassOf Anatomical_structure`, from lines 3-4-5-6 `Zone_of_heart SubClassOf Anatomical_structure`, thus `Anatomical_structure or Serous_sac or Zone_of_heart SubClassOf Anatomical_structure`, for which from line 2 `has_inherent_3-D_shape value true`. Thus, `Anatomical_structure or Serous_sac or Zone_of_heart SubClassOf has_inherent_3-D_shape value true`. Therefore, as the data property `has_inherent_3-D_shape` is functional (1) it cannot be related to two different values, hence `Portion_of_cytosol` is unsatisfiable and so are all its subclasses.

4.5. Propagations of unsatisfiability

A lot of classes are unsatisfiable either because they are a subclass of an unsatisfiable class or because they are a subclass of an existential restriction of a property pointing to an unsatisfiable class (Schema 3).

Schema 3
If
```
S8. A SubClassOf p some B
S9. B SubClassOf Nothing
```
then
```
S10. A SubClassOf Nothing
```

Example 4. `Thenar_compartment` is an example of propagation of unsatisfiability via Schema 3. `Thenar_compartment` is a subclass of `contains some First_metacarpal_bone` (1) and (Example 2) `First_metacarpal_bone` is unsatisfiable (2), hence according to Schema 3 `Thenar_compartment` is unsatisfiable.

1. `Thenar_compartment SubClassOf contains some First_metacarpal_bone`.
2. `First_metacarpal_bone SubClassOf Nothing`.

Example 5. According to Schema 3, since `fm_live_14758 SubClassOf related_part some Portion_of_cytosol` (line 18 Fig. 7) and `Portion_of_cytosol` is unsatisfiable (Example 3), so is `fm_live_14758`. Then, since `Cytoplasm SubClassOf attributed_constitutional_part some fm_live_14758` (line 13 Fig. 7) unsatisfiability is next propagated to `Cytoplasm` and similarly, since `Compartment_of_cell SubClassOf constitutional_part some Cytoplasm` (line 12 Fig. 7), it is propagated to `Compartment_of_cell` and finally to `Cell` (line 1 Fig. 7), which is in turn unsatisfiable.

As exemplified above, unsatisfiability is rapidly propagated from place to place via Schema 3 and to all the concerned subclasses. As FMA makes an extensive use of existential restrictions, this may partly explain why the FMA exhibits so many unsatisfiable classes.

The examples above have illustrated some patterns of recurrent errors in the native FMA. Many other FMA classes are similarly inferred to be unsatisfiable according to Schema 1 or Schema 2 and are next propagated via Schema 3 to other classes. Such modeling errors are typically errors that human designers, even experts, can easily make. Indeed, the FMA design has been in existence for more than 10 years, the ontology is huge, and several inference steps are often needed to detect errors.

While automatic inference tools, such as OWL reasoners, are helpful, they are not sufficient and an explanation tool is needed. At the time of this work, only a rudimentary justification tool was offered. No explanation tool compatible with OWL 2 and the OWL API 3.1 was available. Protégé 4 does not provide a real explanation but only a list of justifications (e.g., Fig. 7 or 8), which are difficult to understand. As underlined in the ISSUE-52 POSTPONED: specification of OWL equivalences and rewriting rules for explaining inferences http://www.w3.org/2007/OWL/tracker/issues/52, accessed 29.08.12): "The current version of Protégé 4 has functionality that identifies the relevant axioms involved in making an inference, but stops short of explaining how the entailments/consequences of these axioms can be chained together to create an explanation.". For instance, the complex explanation given in Example 3 to explain `Portion_of_cytosol` unsatisfiablity from the axioms listed Fig. 7 clearly illustrates this point. Therefore, it was difficult at this time to proceed further. Debugging and repairing the FMA are important tasks that await a user-friendly explanation tool for large ontologies. Repairing the FMA is critical for its use in applications.

5. Application

Ontologies and Terminologies both refer to vocabularies, but each model has a different concern and purpose. A formal ontology deals with *knowledge* describing *entities* (concepts) of a domain of interest. It provides the *meanings* of terms as *axioms* and *facts*. A terminology is of completely different nature. It does not describe the domain but specifies the language *terms* (words) used in the domain and provides information (data) about them, such as the preferred term, synonyms, and broader and narrower terms (though sometimes the distinction may be blurred or the two notions may overlap, e.g., SNOMED-CT [10]). While rich formal ontologies like FMA-OWL or SNOMED-CT are useful for defining the semantics and for checking consistency, terminology tools are usually much faster to index and search Web resources or patients records. It is often the case that applications only ever use the derived terminology,

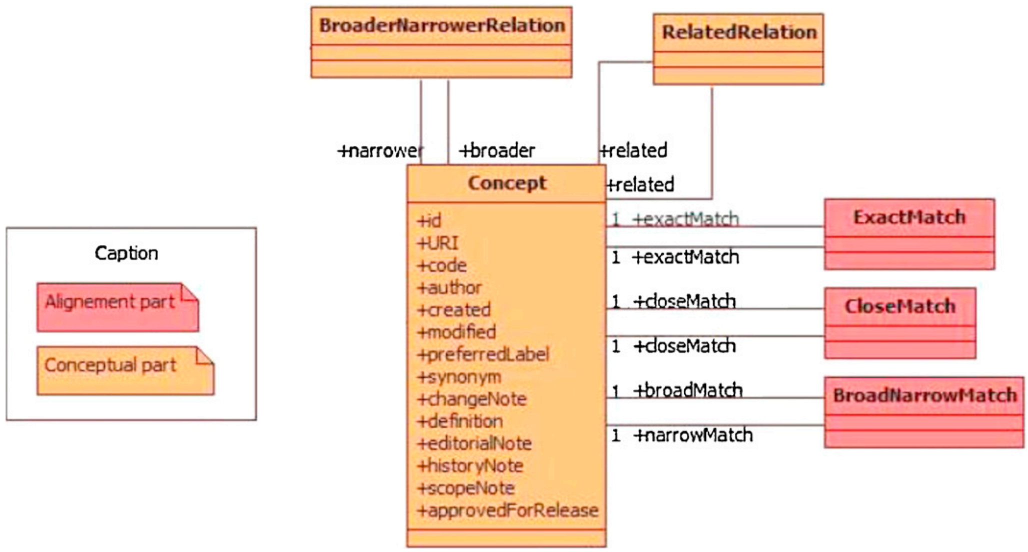

Fig. 9. The UMV2 conceptual model of the cross-lingual portal EHTOP.

which is often enough. A main benefit of the underlying ontology is to ensure that the lightweight derived terminology is well designed.

A first application developed from the FMA-OWL ontology is now operational: a lightweight terminology for human anatomy (FMA-TERM) has been derived from FMA-OWL and is included in a European cross-lingual portal of terminologies EHTOP [4] devoted to resources indexing and search (Section 5.1). FMA-TERM contains all concepts and relations, including all the repairs previously performed thanks to reasoning with FMA-OWL. EHTOP does not allow for repairs, inferencing or reasoning. It allows for searching and browsing among terminologies and ontologies, and looking for resources.

5.1. EHTOP cross-lingual portal

EHTOP [4] is a cross-lingual health terminology portal developed by the Rouen University Hospital to supply health professionals and students with the terminologies and ontologies available in Europe. EHTOP functionalities include the ability to search, browse terminologies, visualize hierarchies, and find resources indexed with terms.

EHTOP allows searching for terms or words in all portal terminologies. The search is performed among preferred terms, synonyms, definitions, and codes with three search modalities. The default modality is a *truncated* search. It searches for a full expression within terminologies from a truncated entry. For example, searching for 'Infarction,' EHTOP provides 15 answers from MeSH, including, among others: angina unstable, anterior wall myocardial infarction, cerebral infarction, heart rupture, post-infarction, infarction, and infarction anterior cerebral artery. The second modality is a *stemming* search that uses roots of words, removing usual prefixes and suffixes. Searching for 'cérébelleuses' without stemming provides 8 answers; with stemming, 43 terms are found, including terms such as 'cerebelleux.' The last modality is an *exact* search, which looks for the exact expression entered, for example, 'myocardial infarction.'

'Relations' (Fig. 10) offers a critical means for a user to navigate within and between terminologies, and thus to discover new information about relations between concepts. Intra-terminology relations provide the associations between concepts within a given ontology; inter-terminology relations offer the associations between concepts of different terminologies. The latter rely on mappings that are manually inserted by professionals or automatically obtained either from the UMLS or by natural language-specific techniques. It is possible to visualize the term position in the (multi-) hierarchy through the 'Hierarchy' tab. The 'Resources' tab displays all the resources indexed by the term, returned from various sources such as PubMed, Doc'CISMeF, and others.

While it puts a special emphasis on French, all main European languages are concerned, including those such as Greek and Russian that do not use the Latin alphabet. It is possible to query EHTOP either in a monolingual or bilingual mode, i.e., any language plus English. English has been chosen as the pivotal language because of its importance in Science. Currently, EHTOP includes 32 terminologies and ontologies, more than 1.1 million Descriptors, 1.3 million synonyms, 227,000 definitions, 402,000 BT-NT relations, 2.4 million other relations and 23 different languages.

Semantic interoperability relies on a generic model, the Unifying Model of Vocabulary (UMV2), and on the Web ontology standard OWL.

5.1.1. UMV2 conceptual model

UMV2 was developed to integrate terminologies and ontologies [4]. UMV2 defines a common model for all terms, whatever their terminology. It can be viewed as a meta-model or an upper ontology designed to support broad semantic interoperability between terminologies. The basic concept of UMV2 is `Descriptor`, which is quite similar to `skos:Concept` ('Concept' Fig. 9) of SKOS (Simple Knowledge Organization System http://www.w3.org/TR/2009/REC-skos-reference-20090818/, accessed 29.08.12). `Descriptor` has several attributes, for example, `label` refers to the preferred term used to name a descriptor in natural language, `synonym` refers to its synonyms in different languages, and `notation` refers to its identifier. Another key concept is `Association`, used to model a relation between two descriptors as a class. It is quite similar to UML association classes. UMV2 includes other general classes such as `BT-NT` for a hierarchical association, or `RT-RT` for "See Also" associations. UMV2 offers general properties that can be used to relate EHTOP descriptors. The properties BT and `NT` of a `BT-NT` association are used to assert that two descriptors have a hierarchical link. BT states the more general term ('broader') and `NT` the more specific ('narrower') term of the association. The property `RT` states that two descriptors have *some* relationship. UMV2 also provides 'mapping' properties (e.g., '`CloseMatch`',

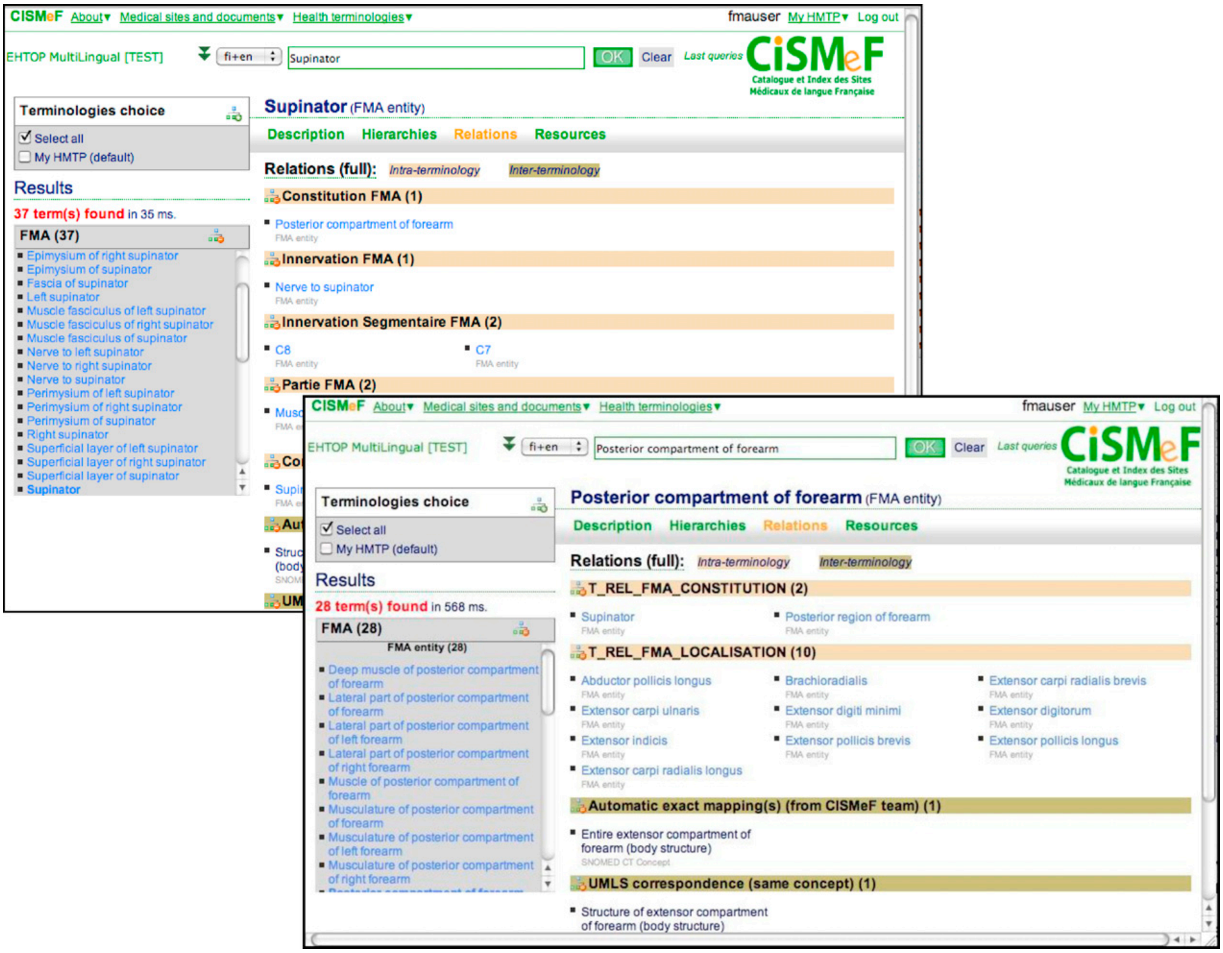

Fig. 10. Searching resources on the muscles located near by "Supinator" with EHTOP.

'ExactMatch' Fig. 9) used to align EHTOP descriptors, which are imported from SKOS: `skos:closeMatch`, `skos:exactMatch`, `skos:broadMatch`, `skos:narrowMatch` and `skos:relatedMatch`.

Each terminology T of EHTOP is built as an enrichment (UMV1-T) of UMV2. Each enrichment, e.g., UMV1-FMA, UMV1-ICD10, defines its own specializations of `Descriptor` and of `Association`, e.g., UMV1-FMA defines the descriptor `FMAEntity` and the associations `FMAinnervation`, `FMAdrainageveineux` (venous drainage).

5.1.2. OWL 2

We use OWL to share each terminology/ontology with different project partners. A bespoke parser is written for each terminology/ontology. It generates the OWL instance of EHTOP complying with the UMV2 model. A generic Java parser has been achieved for storing the resulting file in a relational DB.

5.2. From FMA-OWL to EHTOP

Going from the FMA-OWL ontology to a terminology (FMA-TERM) compliant with the UMV2 metamodel relies on a reification process transforming FMA-OWL Classes into FMA-TERM individuals representing terms, Object properties into `Association` subclasses. More precisely, each FMA-OWL class is mapped to an individual of the `FMAEntity` subclass, which IRI is generated from its FMAID in FMA-OWL, e.g; the individual `FMA_7088` for `Heart`, with the IRI http://www.chu-rouen.fr/smts#FMA_7088 generated from its FMAID (7088). `SubClassOf(fma:A fma:B)` axioms are transformed into `BT-NT` individual assertions, `SubClassOf(fma:A ObjectSomeValuesFrom(fma:R fma:B))` axiom into an individual and two object property assertions, FMA-OWL preferred-name annotations into labels, other annotations into datatypes. Object properties *R*, e.g; `bounded_by`, `venous_drainage`, `innervation`etc., are transformed into `Association` subclasses (named `FMAR`) and two object properties with `SubClassOf(A ObjectAllValuesFrom(R B))` and cardinality restrictions axioms. For example, `bounded_by` is mapped to the `FMAdelimitation` subclass of `Association` and two object properties `delimite and estDelimitéPar`are created with value restrictions.

Using HermiT for reasoning exhibited a modeling error in the initial implementation of FMA-TERM. All `Association` classes are re-classified as FMA entities. These undesired inferences resulted of wrong domain assertions of object properties. Instead of `FMAEntity` (#10) the domain should be an `Association` subclass (`FMA_7088-FMA_7167` is not an `FMAEntity` but an individual of `FMAdelimitation`).

As explained above, the transformation of FMA-OWL into FMA-TERM is a two-step process. First a specific Java parser achieved with the SAX API and Jena API generates the OWL file fulfilling the UMV2 model; next the resulting file is stored in a relational DB by a general software.

5.3. Use case scenarios

The use of the FMA terminology of EHTOP is briefly presented below through two use-case scenarios.

Scenario 1. Medical students are interested in searching information on the muscles located near by the "Supinator" and resources

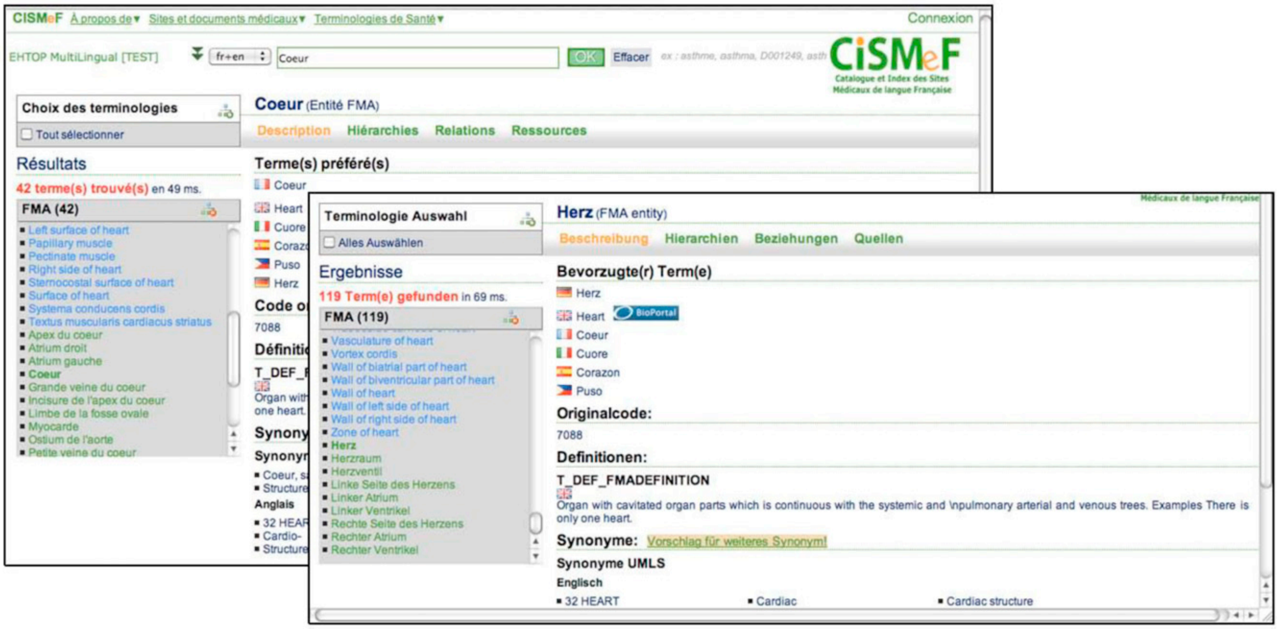

Fig. 11. Results returned by EHTOP for the query ‘Coeur’ in French + English and ‘Herz’ in German + English.

on them. Therefore, they start by entering “Supinator” (Fig. 10) and obtain 37 FMA entities.

By default, EHTOP displays information about the Descriptor “Supinator” itself (FMAID 38512) and provides a menu with three other tabs (at top): Hierarchy, Relations, and Resources. Selecting Relations provides four relations: *constitutional part of* (translated by ‘Constitution’ in French) pointing to the entity “Posterior compartment of forearm,” *nerve supply* (‘Innervation’), *segmental supply* (‘Innervation Segmentaire’), *member of* (‘Partie’). The medical students click on “Posterior compartment of forearm” and obtain the 10 muscles located in this particular compartment. Then, for each, they click on the Resources tab to obtain resources from several websites, including PubMed and CISMeF. An evaluation in practice of the tool has been achieved by medical students.

Scenario 2. A French user is interested in learning more about the Heart anatomy. He enters the query ‘Coeur’ -in EHTOP using the bilingual mode “fr+en”, i.e., French + English – (Fig. 11). EHTOP feedback is the list of 42 terms (left part of Fig. 11) found in French (green) or English (blue). Since he has chosen French as the “first” language, the search is done among the French labels and their synonyms in addition to the English labels and synonyms, and the display is also in French. The French user continues his investigation, selecting “Cardiac valve,” a term missing in French. Users are typically happy that the second language is always English, the ETHOP “pivot” language, as many more terms are available in English than in French. This French user can also see the translation of “Coeur” in five other languages. For instance, by a simple click on the German flag, he moves from French to German. The interface and the metadata are then automatically translated into German (right part of Fig. 11).

An advantage of French is that EHTOP incorporates two standards for French anatomical terms: Nomina Anatomica (NA) [11], published by the International Federation of Associations of anatomy in 1955, and a more recent translation of the Terminologica Anatomica (TA) [12]. TA was chosen as reference and is used for the French preferred terms, while NA is used for synonyms; for example, for the English terms “ulna” and “fibular” the French preferred terms are resp. “ulna” and “fibulaire” (from TA), while their synonyms are resp “cubitus” and “perone” (from NA). By using these two terminologies, EHTOP provides a bridge among users, specially among junior and senior staff.

6. Results

This section first reports results regarding the FMA-OWL ontology in OWL 2. Secondly, it provides results about the EHTOP FMA-TERM terminology.

6.1. FMA-OWL ontology

Complete representations of the entire FMA are now available in OWL 2 (Fig. 12). An OWL 2 ontology without metaclasses ($\mathcal{FMA}-\mathcal{OWL}$ 2_noMTC Table 1 #3) has been generated from FMA 3.0 ((http://gforge-lirmm.lirmm.fr/gf/download/docmanfileversion/214/747/FMA_3.0_noMTC_100702.owl.zip, accessed 29.08.12). It includes all FMA classes and properties (except *homonym_of* and *homonym_for*, discarded in agreement with FMA’s authors). The axioms retain properties transitivity but ignore irreflexivity and asymmetry. This ontology offers 15,084 new definitions of classes, 16,113 disjointness axioms; 85,467 axioms are removed and replaced by one single axiom that is inherited by all descendants, 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 (http://gforge-lirmm.lirmm.fr/gf/download/docmanfileversion/215/748/FMA_3.0_MTC_100701.owl.zip, accessed 29.08.12) is also available ($\mathcal{FMA}-\mathcal{OWL}$ 2_WithMTC Table 1 #4). The FMA-OWLizer tool (Section 3) can generate both a

Table 1
Metrics of FMA-OWL ontologies.

File		Size (MB)	Classes	Class axioms	Expressivity
$\mathcal{FMA}-\mathcal{OWL}$ 1 from FMA frames – 2005					
#1	Without N&S	41.6	41648	236208	$\mathcal{ALCOIF}$(D)
#2	With N&S	40.8	41648	230690	$\mathcal{ALCOIF}$(D)
$\mathcal{FMA}-\mathcal{OWL}$ 2 from FMA 3.0 frames – 2008					
#3	Without MTC	256	85005	263389	$\mathcal{SROIQ}$(D)
#4	With MTC	314	85005	261331	$\mathcal{SROIQ}$(D)

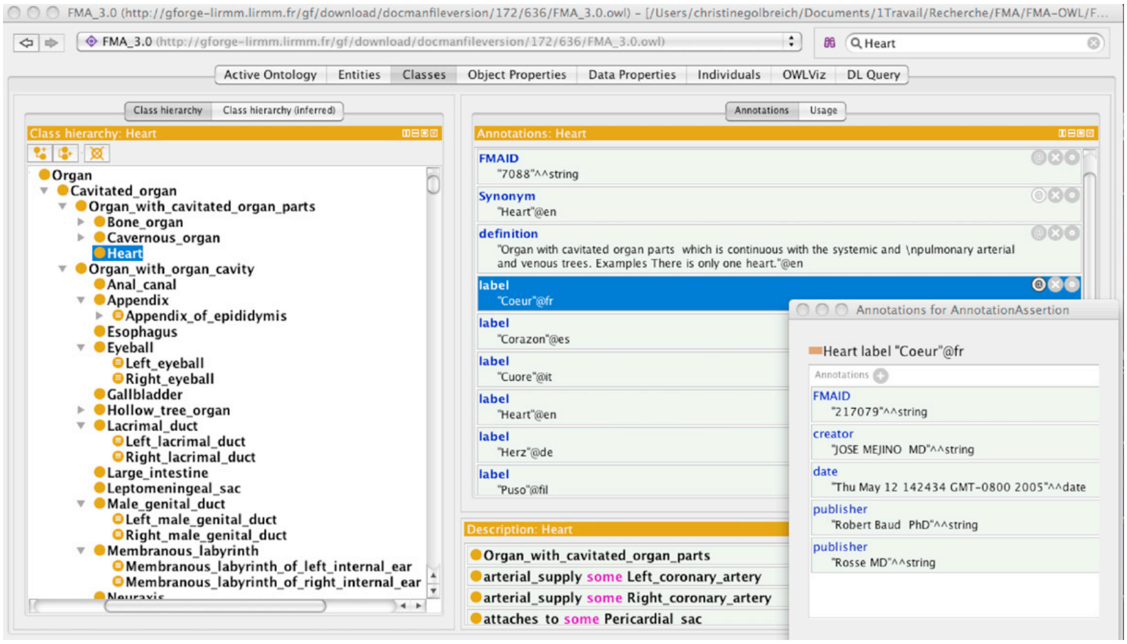

Fig. 12. $\mathcal{FMA}-\mathcal{OWL}$ 2 ontology in Protégé: the class Heart, its label in all the languages, and the annotation of the French labeling itself.

standard FMA-OWL ontology and other ontologies of various size and complexity that better fit specific applications needs. For example, $\mathcal{FMA}-\mathcal{OWL}$ *v1* are OWL 1 partial (smaller) ontologies (41 MB) issued from the FMA 2005, of which left/right leaves are cut, without or with class definitions (Table 1 #1 or #2) built from the *constitutional_part* property. Ontologies obtained from the FMA 3.1 update have also been generated.

We tried to check the FMA-OWL ontology with HermiT (see. results reported Section 4); however, it was not possible to visualize the inferred results and identify the unsatisfiable classes with Protégé 4.1. Therefore, we moved to an incremental approach that consisted of extracting smaller or less complex modules with an OWL-based extractor [9]. We classified and visualized the inferred inconsistencies and hierarchy for a large module extracted from FMA-OWL (Section 4). The time for classification was about 2 h. This process led to the identification of recurrent patterns of errors in the original FMA, for example, those related to the domain or range of some data properties. Due to the large size and complexity of the FMA-OWL, and present tools limitations, it was difficult to proceed further.

6.2. FMA-TERM terminology

The FMA-TERM terminology derived from FMA-OWL has been included in EHTOP (http://cispro.chu-rouen.fr/ehtop_site/connexion.html?lang=en,id=fmauser;password=fmapass, accessed 29.08.12), and is now operational. FMA-TERM is a lightweight $\mathcal{ALN}$(D) ontology with 39 classes, 41 object properties, 19 data properties, 136 subclass axioms and 218419 individuals. It includes 81042 FMA descriptors, 52040 English synonyms, 4436 French terms and 139 French synonyms. Among all the FMA-OWL properties, only the most important relationships (22) were selected for FMA-TERM. 52447 relations connect 39787 distinct classes and there are 81,224 hierarchical BT-NT relations that mirror the original FMA.

The EHTOP FMA terminology is available in all the languages provided in FMA-OWL and originally offered by the FMA: totally for English (81,042 FMA descriptors and 52,040 synonyms), partially for French ($n = 4436$), Latin ($n = 876$), German ($n = 354$), Spanish ($n = 488$), Filipino ($n = 72$), and Italian ($n = 49$). Since its integration into EHTOP, the FMA terminology has been increased: 9724 French translations were manually added to the already existing 4436 French terms (i.e., +219%).

Since January 2011, EHTOP has been used daily by librarians to index French health resources in the CISMeF catalog [13]. As MeSH, the pivotal terminology for the CISMeF catalog, was lacking precise terms of anatomy, FMA is now a central terminology to index and search anatomical resources. In August 2011, 6481 out of 81,450 CISMeF health resources (7.96%) were indexed with at least one FMA entity.

The FMA-TERM terminology of ETHOP was qualitatively evaluated by 32 second-year medical students. The finding showed 58% of satisfaction for its user interface and 76% for its functionalities and content. Consequently, a cross-lingual multi-terminologies and ontologies portal will be part of the medical informatics program for medical students from October 2011. The primary use will be anatomy using the FMA.

7. Discussion

There have been several efforts since 2005 to translate the FMA into OWL [5,14,15]. But earlier conversions to OWL were not fully satisfying. A main limitation of the Dameron et al. [14] and Noy and Rubin [15] studies is the metaclass representation. Translating FMA metaclasses into OWL 1 leds to OWL Full. Dameron et al. [14] offered two components: an OWL DL and an OWL Full component. The former omits the metaclasses to remain in OWL DL and thus is incomplete. The latter is a complete representation that imports the OWL DL module, but it is an OWL Full component. In contrast, the FMA ontology [5] is OWL DL and includes metaclass knowledge. However, it is also incomplete as the left/right leaves were cut for memory and tool limitations. Also, 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 solutions: the FMA-OWL (DL) version without metaclasses also captures metaclass knowledge but it is now complete; the FMA-OWL 2 (DL) ontology with metaclasses keeps the metaclass structure thanks to new OWL 2 metamodeling features. Second, formalization is pushed forward, eliciting further FMA underlying semantics. As these 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 [5] did not scale up and was not robust, the new mapping of the syntax now handles the *entire* FMA and can overcome the changes of successive updates.

Regarding the pattern approach, a few patterns are comparable with prior research [16], but the goal of abstracting and generalizing is clearly different; the goal of [16] is to reduce the FMA size or abstract it. Automatically generating axioms is partly shared with other work [17], but our approach is more general. The pattern recognition is an exact procedure based on a lexical match. The lexical pattern-based formalization used to create definitions makes sense because the FMA authors have defined the class names in a systematic way. For instance, FMA authors systematically use `A_of_B` (e.g., `Ganglion_of_cranial_nerve`) as a label, and define the other names as synonyms (e.g., `Cranial_nerve_ganglion`). However, there are a few exceptions that do not follow the naming convention above, for example, `Lung_parenchyma`. Either it is deliberate or this might be changed by FMA authors in the future. At the moment, the processes remain simple. There is no complex recursion and inferences are not considered (for efficiency). To be sure that the definitions provided are correct, they specify restrictive conditions. For example, *Process 2* only handles the cases `A_of_B` where the *direct* superclass is `A` or `Anatomical_A` and there exists a specific restriction (see 2.2). This is why for instance (as `Left_side` is not a FMA class), the classes `Left_side_of_B` are not processed. Nevertheless *Process 2* can recover some cases that do not meet exactly meet the required restriction, e.g., `Ganglion_of_cranial_nerve`. For example, while there does not exist a restriction `regional_part_of some Cranial_nerve`, there exists anoher restriction, `regional_part_of some Neural_tree_organ`, and `Cranial_nerve` is a *direct* subclass of `Neural_tree_organ`. Therefore, *Process 2* is applied and the definition generated for `Ganglion_of_cranial_nerve` is `Ganglion and regional_part_of some Cranial_nerve`. In contrast, there is no definition created for `Ganglion_of_vagus_nerve` because `Vagus_nerve` is a descendant but not a *direct* subclass of `Neural_tree_organ`. Similarly, *Process 1* does not create a definition for `Left_acetabulum` because there is no class named `Acetabulum` or `Anatomical_acetabulum` in FMA.

At the moment only two patterns are processed. They can still be improved and the approach extended to other patterns. The obtained ontologies in OWL are still unstable and contain a number of problems, which were originally present in the native FMA. Carrying forward the work on patterns and on understanding and correcting the unsatisfiable definitions is important to improve the FMA ontology design and consequently the FMA derived terminology.

8. General guidelines and lessons learned

Although the FMA has specificities, which makes its formalization in OWL DL special, it is possible to draw general guidelines and lessons from this work for transforming an ontology into OWL 2. Following are some examples.

1. Most often, transform property value restrictions of frames into subclass axioms with existential restrictions (see previous author's papers [5]).
2. To convert metaclasses, first carefully analyze their role. Use OWL 2 metamodeling ability only if they really carry metalevel knowledge; otherwise use annotations.
3. Do not confuse domain data and metadata. A good practice is to first model them in separate ontologies. If metadata are embedded in the source domain ontology, do not create domain individuals; rather, use OWL 2 annotations and annotation of annotations to state facts about metadata.
4. A good practice is to exploit both the syntax and the semantics encrypted in the names of the source model for the formalization.
5. Most ontologies exhibit specific lexical patterns. Exploit such patterns to provide semantics and obtain safe classes logical definitions (for instance, this approach might be applied to SNOMED, which awaits repairs and restructuring of anatomy, among others).
6. Exploit the patterns to create not only class definitions but also other axioms at the same time (e.g., disjointness).
7. Extracting modules is a fruitful strategy to detect errors and incrementally repair errors in large ontologies.
8. An underlying rich ontology in OWL is useful to ensure that a lightweight terminology is well designed.

9. Conclusion

We have presented a method to represent the FMA ontology in OWL 2 and developed a user friendly and flexible tool for it. As a result, complete representations of the entire FMA in OWL 1 or OWL 2 DL are now available and more than 15,500 FMA classes have a reliable logical definition. Additionally, the tool allows for automatically producing new FMA-OWL versions for each FMA update and other customized variants on demand. Being represented in logic, we could check the FMA-OWL ontology with an automatic inference engine to detect inconsistencies, proving that the native FMA still exhibits recurrent errors that are rapidly propagated from place to place. Though many classes are still unsatisfiable, this work is an important step forward and a major achievement. The results of this study indicate that it is worthwhile to pursue the approach presented in this paper, to make the FMA more coherent in the future, once an explanation tool is available for the OWL 2 API.

We also applied the FMA-OWL ontology to produce the FMA-TERM terminology, used in the EHTOP cross-lingual portal of European terminologies/ontologies. This portal is daily used to index Web resources. Without first converting the FMA ontology to OWL, it would not have been possible to ensure the quality of the FMA terminology and of the resources indexing.

We assert that terminologies and formal ontologies are fundamentally different. An OWL ontology is a domain model that brings a clear and unambiguous semantics. Highly optimized, sound, and complete reasoners exist for OWL. These reasoners can be used to classify an ontology and detect inconsistencies in class definitions, which is valuable during ontology development and for maintenance. In contrast, a terminology is a metamodel of a domain ontology that states the language terms used in the domain. Efficient and powerful tools exist for terminologies that are useful for fast indexing and searching. We advocate use of an OWL ontology for clear semantics and reasoning and to derive from it a lightweight terminology for resources indexing and searches in applications with large datasets. The underlying rich ontology is essential to ensure that the lightweight derived terminology is quality controlled and well designed. Reliability and correctness are particularly important for biomedical applications, where safety is critical.

Future goals are to push further the formalization in OWL 2 or extensions [18] and to exploit OWL reasoners and explanation tools to improve the FMA. Another interesting direction is to develop a general service for deriving lightweight terminologies from rich, underlying OWL 2 ontologies. Finally, the 'lexical pattern' based definition of OWL classes and axioms, and the global strategy presented – pattern-based formalization, modules extraction, inconsistencies explanation and repairs – are general. It may be worthwhile to apply it for quality assurance of other large reference ontologies, such as SNOMED-CT.

Acknowledgments

We are grateful to Cornelius Rosse and Onard Mejino for their advice and support, to Oxford team for their specialized classification algorithm, and to Birte Glimm that helped us in using HermiT at its beginning. We want to thank the anonymous reviewers for their comments, very helpful to improve that document. The authors thank Leslie Block for her valuable advice in the editing of this manuscript.

References

[1] Rosse C, Mejino Jr JL. The Foundational Model of Anatomy Ontology. In: Burger A, et al., editors. Anatomy ontologies for bioinformatics: principles and practice. New York: Springer; 2008. p. 59–118.
[2] Rosse C, Mejino Jr JL. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. Journal of Biomedical Informatics 2003;6: 478–500.
[3] W3C, OWL. Working group. OWL 2 Web Ontology Language Document Overview W3C Recommendation October 27; 2009. URL: http://www.w3.org/TR/owl2-overview/, [accessed 02.08.11].
[4] Grosjean J, Merabti T, Dahamna B, Kergourlay I, Thirion B, Soualmia LF, et al. Health Multi-Terminology Portal: a semantics added-value for patient safety. Studies of Health Technologies and Informatics 2011;66:129–38.
[5] Golbreich C, Zhang S, Bodenreider O. The Foundational Model of Anatomy in OWL: experience and perspectives. Journal of Web Semantics 2006;4: 181–95.
[6] Golbreich C, Wallace EK. OWL 2 Web Ontology Language New Features and Rationale W3C Recommendation October 27; 2009. http://www.w3.org/TR/owl2-new-features/, [accessed 29.08.12].
[7] Noy NF, Musen MA, Mejino Jr JLV, Rosse: C. Pushing the envelope: challenges in a frame-based representation of human anatomy. Data and Knowledge Engineering 2004;48(3):335–59.
[8] Glimm B, Horrocks I, Motik B. Optimized Description Logic Reasoning via Core Blocking. In: Giesl J, Hähnle R, editors. IJCAR 2010. LNCS, vol. 6173. Heidelberg: Springer; 2010. p. 457–71.
[9] Jimenez-Ruiz E, Cuenca Grau B, Schneider T, Sattler U, Berlanga R. Safe and Economic reuse of ontologies: a logic-based methodology and tool support. In: Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, Manolis Koubarakis, editors. The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1–5; 2008. *Lecture Notes in Computer Science* 5021. Springer 2008; 2008. p. 185–99.
[10] Stearns MQ, Price C, Spackman KA, Wang AY. SNOMED clinical terms: overview of the development process and project status. In: Proceedings AMIA Annual Full Symposium. 2001. p. 662–6.
[11] Committee IAN. Nomina Anatomica: approved by the Eleventh International Congress of Anatomists at Mexico City 1980 Together with Nomina Histologica and Nomina Embryologica. Williams & Wilkins; 1983.
[12] Fabry P, Baud R, Burgun A, Lovis C. Amplification of Terminologia anatomica by French language terms using Latin terms matching algorithm: a prototype for other language. International Journal of Medical Informatics 2006;75(7):542–52.
[13] Darmoni SJ, Leroy JP, Baudic F, Douyère M, Piot J, Thirion B. CISMeF: a structured health resource guide. Methods of Information in Medicine 2000;39(March (1)):30–5.
[14] Dameron O, Rubin DL, Musen MA. Challenges in converting frame based ontology into OWL: the Foundational Model of Anatomy case-study. In: AMIA annual symposium: Washington DC. 2005. p. 181–5.
[15] Noy Natalya F, Rubin DL. Translating the Foundational Model of Anatomy into OWL. Journal of Web Semantics 2008;6:133–6.
[16] Mikroyannidi E, Rector A, Stevens R. Abstracting and generalizing the Foundational Model of Anatomy ontology. In: Proceedings of the *bio-ontologies* 2009 conference. 2009.
[17] Dameron O, Chabalier J. Automatic generation of consistency constraints for an OWL representation of the FMA. In: Proceedings of 10th international Protégé conference. 2007.
[18] Motik B, Cuenca Grau B, Horrocks I, Sattler U. Representing Ontologies Using Description Logics, Description Graphs, and Rules. Artificial Intelligence 2009;14:1275–309.