Author manuscript, published in "Lecture notes in computer science, LNCS 5903 (2009)"

inria-00438535, version 1 - 18 Sep 2010

My Corporis Fabrica : a Unified Ontological, Geometrical and Mechanical View of Human Anatomy

Olivier Palombi[1], Guillaume Bousquet[1], David Jospin[1], Lionel Revéret[1], and François Faure[1]

Anonymous

Abstract. A new anatomical database, My Corporis Fabrica (MyCF), is presented. It extends the reference anatomical ontology FMA (the Foundational Model of Anatomy) with the possibility to coherently integrate 3D geometrical data, as well as biomechanical parameters. The purpose of this extension is to allow the user to intuitively create a patient-specific 3D representation from a formal description of anatomical entities and also to automatically export this description to test a physical simulation. The main contribution of MyCF consists in the formalization of a comprehensive database structure implemented in MySQL, linking canonical description of anatomical entities with reality-grounded instances of such description in terms of geometrical data and physical attributes. The principle of ontology modeling inherited from FMA is maintained in order to guarantee a close consistency between the two databases. A detailed illustration is given for the knee. In this example, the canonical description of anatomical entities is developed, including the location of attachment of ligaments as well as their stiffness, using an intuitive 3D GUI. Finally, an instance of the knee based on this description is automatically augmented with the canonical part information and then exported to a physical simulator. This example shows the benefit of coupling an exhaustive description of the anatomical structures with a physical simulation to study the dynamic effect of ligaments on the stability of the knee articulation.

1 Introduction

Anatomical models of human body are required in an increasing number of domains, from the teaching of anatomy to biomechanical simulation. Typically, accurate 3D representation of human anatomy is needed in application such as surgery simulator, realistic prototyping of cloth or representation of avatars in 3D virtual communities. The 3D representation of internal anatomy entities is primarily accessible from medical imaging such as CT or MRI tomography. These imaging techniques provide a raw volume and need to be segmented to identify individual anatomical entities. Even if some powerful methods have been proposed, this segmentation step is still time consuming as it cannot be fully automatic. An alternative approach to volumetric segmentation is to use 3D

inria-00438535, version 1 - 18 Sep 2010

models designed by artists, but this involves a risk of loss in realism. Therefore, the need for a validated anatomical database of the human body linked with 3D representation is rapidly growing. This database should provide high-level information such as ontological relationships, as well as geometrical data for graphics applications and physical parameters for simulations. It should be extensible and anatomically relevant to integrate new conceptual entities, physical instances and characterization parameters. This is the goal of our approach with the design of the My Corporis Fabrica (MyCF) database. Specifically, MyCF extends FMA (the Foundational Model of Anatomy) [1] to provide additional data to create geometrical and mechanical models of anatomy. It is based on a novel database structure which considerably facilitates its extensibility. The remainder of this paper is organized as follows. In section 2, we review previous work in the domain. The design of MyCF database is presented in section 3. Implementation details are discussed in section 4. An overview of potential applications is presented in section 5. Concluding remarks are presented in section 6.

2 Previous work

Integrating patient-specific content into computational resources requires the use of formal and logical representation of anatomical knowledge. The field of computing science focused on structuring knowledge is called *ontology*. Ontologies of anatomical knowledge has been widely performed with a high level of details and complexity. The FMA can be presented as the most sophisticated anatomical ontology available today for the scientific community [2]. Nonetheless, the technical complexity involved in formalizing the anatomy can not handle general variability in any easy way. Specific approaches have to be imagined to address this issue. This lacking link is the key point in the emerging research field of *semantic 3D media* [3] and more specifically for medical applications [4, 5].

The Digital Anatomist Information System is based on FMA to link graphical and symbolic representations of anatomy [6]. A static 3D content (pictures of 3D models) is available to illustrate the FMA's concepts. But 3D data are not completely integrated in the ontology. Starting from FMA to embed ontology into specific computational resources seems to be relevant regarding the level of compliance with modeling principles used in FMA [7]. FMA is implemented with an open source ontology editor and knowledge-base framework called *Protégé* which is the only way to manage the ontology. Data are stored in a *MySQL* table but direct access to query is not intuitive. A querying agent exists for FMA but the flexibility of use is restricted [8].

Only structural anatomy has been considered in FMA. Topology, geometry and functions are required to build an efficient ontology for physical simulation. Strategy to extend FMA for a specific purpose is presented in [9]. An original approach, strictly limited to the muscoloskeletal system, has been proposed by Charbonnier et al. where FMA is extended to anatomical functional knowledge for physical simulation [10].

inria-00438535, version 1 - 18 Sep 2010

3 Database design

3.1 Generic concepts

An ontology is a specification of a conceptualisation [11]. Common components of ontologies include classes, attributes and relations. Classes are sets, abstract objects, entities, concepts. Classes have attributes or properties. Relations make links between classes. Each relation has a conceptual meaning. For instance in the following sentence: "The patella is a *part of* the knee and *is a* bone organ", relations (*is_a* and *part_of*) used between anatomical entities (patella, knee and bone organ) give a formal and logical description of canonical anatomy. *Is_a* and *part_of* are also called hierarchical relationships. The key point to build an ontology for specific application is the use of associative relationships (domain-specific) which must be clearly defined [12].

The general purpose of an ontology is to provide a mean of classifying individuals. Patient-specific data are *instances* of canonical anatomical entities stored in the ontology. Three kinds of binary relations can be used to manage efficiently anatomical knowledge and individuals: $< class, class >$ seen below, $< instance, class >$ and $< instance, instance >$.

MyCF is focused on the managing of individuals through an intuitive graphic user interface for non-expert users in ontology. The goal is to allow anatomists to contribute to the canonical part of MyCF. By this way, anatomical content of MyCF will be validated by experts. On the other hand, an user of MyCF can add individual content and get correct anatomical information as terminologies, functions, localizations, interactions and so on.

MyCF is based on FMA for the ontological description of anatomy. An original MySQL schema is proposed to ease sharing and extension, as explained in the subsection 3.2 and to manage individuals. A SQL query is enough to retrieve data from one or more tables, allowing efficient implementation in a specific application.

3.2 Description of the mySQL database

The figure 1 shows the logical structure of MyCF's database. The entity relationship diagram (ERD) can be divided in two main parts: canonical anatomical knowledge on one hand and individual data on the other hand.

MyCF's canonical part: All canonical classes of MyCF are stored in the main table called *Anatomical Entities*. An ontology id, used as a primary key, is given for each one. Definition and favorite name are available as well.

In an ontology, anatomical entities are organized into a hierarchy named taxonomy. A taxonomy is a directed acyclic graph satisfying some conditions [7]. Our database schema allows to represent generic graphs, including directed acyclic graphs. Each entry of *Relation_Type* defines a graph, which can be a taxonomy or a more general graph, where its nodes are contained in *Anatomical Entities* and its oriented edges in *Relation*.

inria-00438535, version 1 - 18 Sep 2010

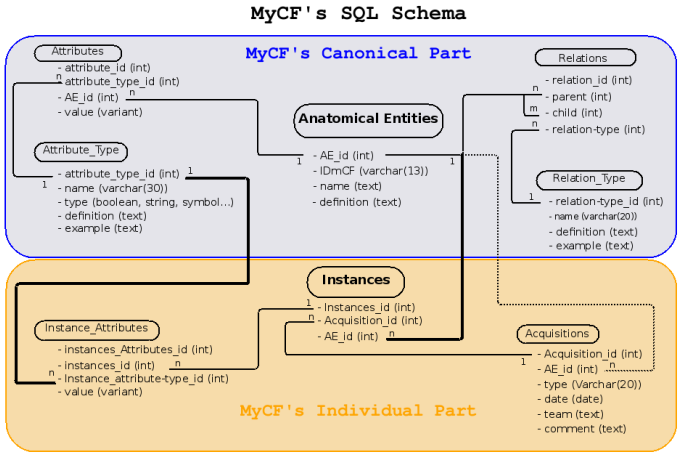


Fig. 1. Entity Relationship Diagram of MyCF's database. The canonical anatomy part is on the top and the individuals part is on the bottom. Black lines depict relationships (1:n) between tables. Two bold lines show the relationships between the canonical and the individual parts. The dotted line depicts a relationship used to force user to select an existing anatomical entity (restricted to anatomical segment) during insertion process of a new acquisition in the database.

In practical terms, the table called *Relation_Type* gives definition of relations. For instance, the relation *is_a* is defined here with text definition and examples. The binary relations between anatomical entities are stored in the table called *Relations* where parents and son are defined using anatomical entity ids and where the type of relation is given.

The attributes are managed also with two tables. The table *Attributes* defines the properties of *Anatomical Entities*. The table *Attributes_Type* ensures the genericity of the attributes.

With those five tables, it is possible to expand endlessly MyCF without adding tables. Furthermore, the simple organization allows users to write his own SQL queries to retrieve or update data.

MyCF's individual part: Instances are linked to anatomical entities. The 1:n relationship (bold line in figure 1) allows to have many instances of one anatomical entity. Instances have specific attributes, the position in a 3D scene for instance. The types of attributes are stored in the canonical part of the database. But the attribute's value of a given instance is saved in the table called *Instance_Attributes*. Furthermore, general information can be stored in the canonical part of the database as, for instance, the yellow color conventionally used to depict bones in anatomical drawings. If the color is not given by the user for an instance, color stored in the canonical part can be automatically retrieved. By this way, a priori anatomical knowledge can be used to complete

inria-00438535, version 1 - 18 Sep 2010

patient-specific content. The origin of patient-specific 3D data is a set of medical images (after segmentation). 3D models coming from a given acquisition will be used together to generate a 3D scene. The table called *Acquisitions* is used to manage it.

3.3 Instances

Geometry is one of the most challenging issues in anatomy modeling. Each anatomical entity can be represented using various types of geometric models (volumetric images, surface or volume meshes, analytical surfaces, implicit surfaces, etc.) at various resolutions or levels of detail. Consequently, we can not store one single geometrical description for each anatomical entity. So, we store, for each entity, a set of instances referenced by file names. We set no limitation on the file format and the type of geometric models. We have developed plugins for Blender [13], a free and open-source geometric modeler, to edit geometrical models of MyCF, see Fig 2. It handles surface meshes and can be used to modify shapes, to displace and resize objects and to label areas such as ligament attachment places.

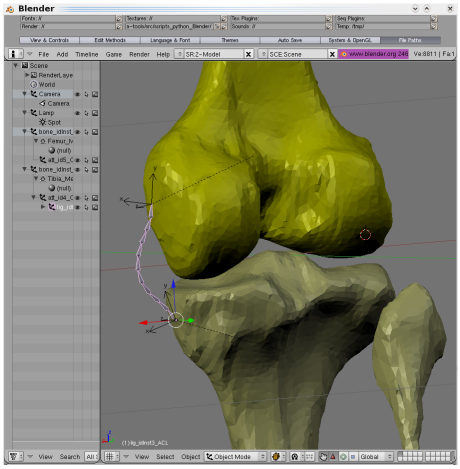


Fig. 2. Blender is used to edit individual geometrical models. Blender objects are created from information stored in MyCF (import script). Positions, insertions of ligaments are editable and modifications can be saved in MyCF (exportation script). The tibial collateral ligament (MYCF:00031) is under edition. The ligament has two extremities (the proximal and the distal extremity) which are defined in MyCF.

Geometry is often modeled based on medical images which represent several anatomical entities, which can be consistently viewed or simulated together. We therefore gather the anatomical entities modeled from a given set of medical

inria-00438535, version 1 - 18 Sep 2010

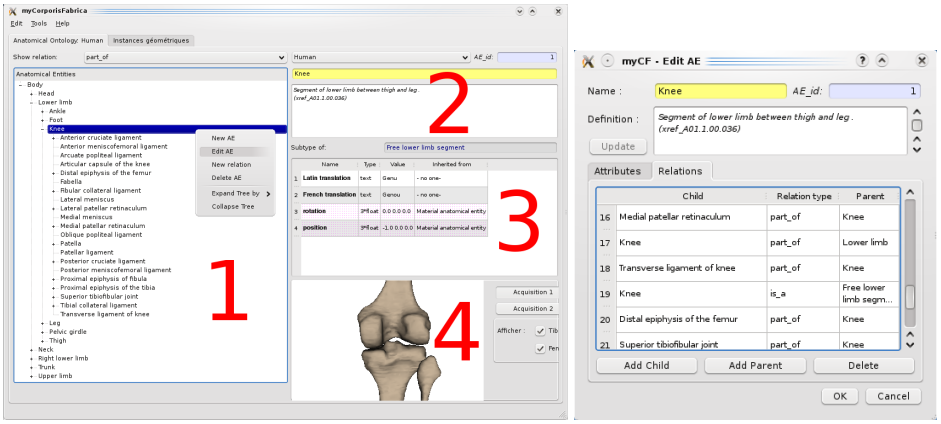
images in *acquisitions*, which allows us to retrieve all of them simultaneously from the database.

4 Implementation

To ensure flexibility and extension, the canonical part of the database is described by only five tables. The ontology contains many taxonomies, as defined in section 3.2. As a consequence, many graphs are spread into this five tables and information retrieval can be tedious. To easily exploit the *MyCF* database, we propose a library called *libmyCF*, written in python. It contains classes representing the anatomical entities, their relations and attributes and some useful algorithms are present.

In the canonical part, we automatically deal with the case of the attributes inheritance. So, when the attributes of an anatomical entity are required, all the attributes of this entity are returned, plus all the attributes of its parents, defined by the relation *is_a*, if the parsed attributes are not yet defined.

If the value of an attribute is modified and this attribute is inherited, a new attribute is created for this anatomical entity. Instance attributes are shown in the figure 3.



(a) (b)

Fig. 3. MyCF graphical user interface. **a)** Main window of MyCF. 1.the anatomical entities tree for the relation *part_of*, 2.information about the anatomical entity (AE), 3.attributes for this AE, 4.VTK visualization of an instance example. **b)** Edition of an anatomical entity. All the attributes and the relations of this anatomical entity are listed and editable here.

In the individual part, when the required instance does not exist, a new instance is created using the corresponding anatomical entity information. On the

inria-00438535, version 1 - 18 Sep 2010

same way, when the required instance attribute is missing, it is automatically created from the attribute of the same type of the anatomical entity corresponding to the instance.

For instance, let us consider a knee modeled by two bones (femur and tibia) and a ligament, see Fig 2. During the importation of the instances into Blender, we need to know where to insert the extremities of the ligament on the bones. To do that, we just have to request the attribute named *position*. Usually, the value of this instance attribute is retrieved and the extremity is placed. Then, we can modify the position of the extremity on the bone in Blender and export this new value into the database.

The first time the extremity is imported into Blender, the instance attribute is not set in the database. When the import script treat the instance 'extremity of the ligament', the position value is requested. As there is no instance attribute, a new one is automatically created from the attribute of the anatomical entity corresponding to the instance of the extremity. Then, modifications can be done in Blender and exported.

Thus, the link with the knowledge contained in the canonical anatomical model is automically managed by the library, and plugins using MyCF information are easy to write.

5 Applications

Gathering ontological, geometrical and mechanical data consistently in a database allows new applications. We sketch some of them in this section.

5.1 Teaching

An obvious application of the database is the teaching of anatomy. MyCF can be used by students to handle anatomical knowledge within a consistent presentation. The graphical representations of anatomical entities allow a more intuitive way for knowledge-based navigation and understanding, as illustrated in Fig 3. Each anatomical entity can be displayed in 3D, while its logical relations with the other entities appear in the interface. Variability can be seen as well through 3D instances saved in the database. The Digital Anatomist Interactive Atlases integrates the FMA with 3D graphical illustrations but without real 3D interaction [14].

5.2 Mechanical models

Beside geometry, we store mechanical parameters in the database, which allows to export an acquisition to physical simulation engines. The user can select the anatomical entities to export and the mechanical models to physically simulate them. For instance, to simulate a knee, the user can decide to export the bones and the main ligaments. Various mechanical models can be used to represent a given object, depending on the purpose of the simulation. For instance, studying

inria-00438535, version 1 - 18 Sep 2010

the kinematics of a given joint based on bones and ligaments requires rigid bone models. Alternatively, studying bone fractures requires deformable models such as finite elements. When the rigid model is applied to a given entity, its mass and inertia matrix are automatically computed based on geometry and density. Alternatively, the user can chose to model an entity as a Finite Element Mesh with Hooke material. In this case, the same process is used to compute the mass and stiffness matrices, depending on the geometry, the shape functions as well as material parameters.

Deformable models such as finite elements require volumetric meshes. Meshes are an important issue in simulation, and we can not propose all meshing algorithms using various complexity and quality parameters. When volumetric tetrahedral meshes are available in the geometric data, they can be easily exported to finite element simulation modules. The precision and the computation time of finite element simulations heavily depend on the geometrical quality and the resolution of the volumetric meshes. Nesme et al.'s grid-based finite elements use deformable regular grids embedding objects defined by arbitrary geometrical models [15]. The grids have arbitrary resolution, which allows to trade off accuracy for speed according to the application, and the geometrical quality is not an issue since all cells are cubes. This approach allows to create finite element models for any virtual object at desired resolution.

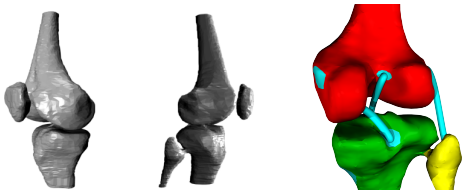
5.3 Completion of a model

A priori knowledge is useful to complete models where all anatomical entities are not defined. For instance, bone models may be built based on a CT-scan acquisition, whereas the connective tissues are not easily visible on the same data. In such a context, performing the mechanical simulation of a knee requires adding ligaments to the model, based on the knowledge of the ligaments properties and insertions locations on the bones.

MyCorporisFabrica allows the automatic creation of missing anatomical entities, using geometrical and material knowledge. The current implementation allows the creation of the main ligaments of the knee. The skeletal ligaments are modeled as one-dimensional mass-spring networks attached to the bones. To manage attach points, we have subdivided each ligament (using the *part_of* relation) in a proximal and a distal extremities. The areas on bones where skeletal ligaments are attached are clearly defined in the ontology. We use the relation *is_inserted_on* to inform where ligament's extremities are inserted. That exact position required for a given instance should be added manually. We're currenlty working on an automatic process to localise ligament insertions on instances using anatomical knowledge. Default values for spring stiffness and damping constants are stored in the database, as well as a default number of particles for the ligament. These values are modifiable by the user at creation time. Moreover, the 3D editor can be used to modify the attach points. Figure 4 illustrates this process.

Once the model is complete, it can be exported to a simulation engine, as illustrated in Figure 5. We use the SOFA simulation library [16] because it

inria-00438535, version 1 - 18 Sep 2010



(a) (b)

Fig. 4. Completion of a model. **a)** Knee bones from an acquisition. **b)** Ligament models not visible in the acquision are added using anatomical knowledge.

handles a variety of mechanical models and their interactions, it is free, open-source and highly customizable. However, export plugins to other simulation engines could be easily added to My Corporis Fabrica.

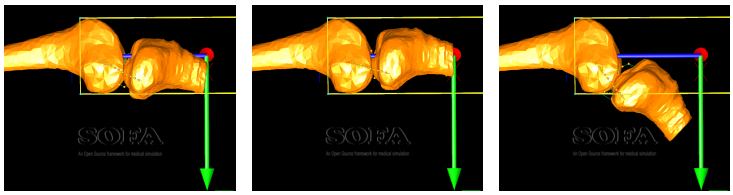


Fig. 5. Snapshots of a SOFA simulation of the model shown in Figure 4, with four ligaments.

6 Discussion

To the best of our knowledge, MyCF is the first multidisciplinary project focused on a close collaboration around anatomical physical simulations. The anatomits can find a formal frame for anatomical knowledge with interactvie 3D visualization and nonanatomist users can manage patient-specific content through confirmed anatomical concepts.

MyCF is a brand new project. The knee is presented as a case study and has been used to built the system. We plan to increase amount of anatomical enities in MyCF. To do so collaboration between users and curators must be strong and active. A collaborative web site around MyCF has been recently open for this purpose. Readers are invited to visit *www.mycorporisfabrica.org*. In the futur, MyCf sould be included in the OBO Foundry (*www.obofoundry.org*).

inria-00438535, version 1 - 18 Sep 2010

References

1. Rosse, C., Mejino, J.L., Modayur, B.R., Jakobovits, R., Hinshaw, K.P., Brinkley, J.F.: Motivation and organizational principles for anatomical knowledge representation: the digital anatomist symbolic knowledge base. J Am Med Inform Assoc **5**(1) (1998) 17–40
2. Rosse, C., Mejino, J.L.V.: A reference ontology for biomedical informatics: the foundational model of anatomy. J Biomed Inform **36**(6) (Dec 2003) 478–500
3. : http://www.focusedk3d.eu.
4. Magnenat-Thalmann, N., Schmid, J., Delingette, H., Guitian, J.I., Agus, M.: 3d anatomical modelling and simulation concepts. In: Eurographics 2009 Tutorial Notes. (2009)
5. : Webpage 3D Anatomical Human project http://3dah.miralab.unige.ch, Accessed March 2008.
6. Brinkley, J.F., Rosse, C.: The digital anatomist distributed framework and its applications to knowledge-based medical imaging. J Am Med Inform Assoc **4**(3) (1997) 165–183
7. Zhang, S., Bodenreider, O.: Law and order: assessing and enforcing compliance with ontological modeling principles in the foundational model of anatomy. Comput Biol Med **36**(7-8) (2006) 674–693
8. Mork, P., Brinkley, J.F., Rosse, C.: Oqafma querying agent for the foundational model of anatomy: a prototype for providing flexible and efficient access to large semantic networks. J Biomed Inform **36**(6) (Dec 2003) 501–517
9. Rosse, C., Kumar, A., Mejino, J.L.V., Cook, D.L., Detwiler, L.T., Smith, B.: A strategy for improving and integrating biomedical ontologies. AMIA Annu Symp Proc (2005) 639–643
10. Charbonnier, C., Gilles, B., Magnenat-Thalmann, N.: A semantic-driven clinical examination platform. In: In Surgetica'2007, Computer-Aided Medical Interventions: Tools and Applications, Chambry, France. (September 2007) 183–189
11. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. **5**(2) (June 1993) 199–220
12. Smith, B., Ceusters, W., Klagges, B., Khler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A.L., Rosse, C.: Relations in biomedical ontologies. Genome Biol **6**(5) (2005) R46
13. : http://www.blender.org.
14. Wong, B., Rosse, C., Brinkley, J.: Semi-automatic scene generation using the digital anatomist foundational model. J. Am. Med. Assoc. **AMIA '99 Symp.** (1999) 637–641 Suppl.
15. Nesme, M., Kry, P., Jeřábková, L., Faure, F.: Preserving topology and elasticity for embedded deformable models. ACM Trans. Graph. (aout 2009) Proceedings of SIGGRAPH'09.
16. Allard, J., Cotin, S., Faure, F., Bensoussan, P.J., Poyer, F., Duriez, C., Delingette, H., Grisoni, L.: SOFA - an open source framework for medical simulation. In: Medicine Meets Virtual Reality, MMVR 15, February, 2007, Long Beach, California, USA (2007) 1–6 http://www.sofa-framework.org.