Combining 3D Models and Functions through Ontologies to Describe Man-made Products and Virtual Humans: Toward a Common Framework

Federico Ulliana[1], Jean-Claude Léon[1], Olivier Palombi[1], Marie-Christine Rousset[2] and François Faure[1]

[1]Grenoble University - INRIA, federico.ulliana@gmail.com, Jean-Claude.Leon@grenoble-inp.fr, olivier.palombi@inria.fr, francois.faure@inria.fr
[2]Grenoble University, marie-christine.rousset@imag.fr

ABSTRACT

Products and virtual humans are commonly described using 3D models of their components and anatomical entities, respectively. Functions stand for symbolic information bringing a high level description of product subsets and apply also to human body anatomical entities. Ontology-based approaches bring new means to improve the efficiency of these digital models. Here, the purpose is to identify similarities and differences between existing ontology based descriptions of products and virtual humans. From this analysis, it is shown how compatible descriptions can be obtained and how a common framework can be derived where products and virtual humans can be both incorporated for various applications involving 3D models. The proposed framework contains an ontology-based description of products and virtual humans enabling the access to 3D models while accessing high-level semantic information through the use of inference mechanisms. Biomechanical simulations of virtual humans and structural behavior simulations of products is the context used to setup this common framework such that products and virtual humans can take part to these simulations where products and virtual humans are involved.

Keywords: virtual human, CAD geometry, mesh, ontology, RDF/RDFS.

1. INTRODUCTION

Modeling 3D complex products using functional information has been an initial target for CAD to support the design process [20, 33]. Similarly to design methodologies, functional information is often addressed in a top-down scheme with the main function of a product at the highest level that is progressively subdivided and refined into more technical and detailed functions related to a subset of a product [1, 11]. However, defining the relationships needed to get an efficient connection between functional information and 3D models of products and their components is still awaiting contributions. A simple example illustrating the use of functions with 3D models would be the ability to display sets of components in accordance with the product function they contribute to. Processing functional information for products has been essentially addressed in a top-down manner where it loosely connects to 3D geometry [28]. Recent advances showed that using reasoning processes connected to ontologies could bring new means to obtain more efficient product models [4] but propagating functional information up to the level of form features of components is not yet available.

Recently also, 3D technologies have brought new capabilities to describe virtual humans and perform various simulations using these models [3, 29]. Generating digital models of human bodies is still raising challenging issues to obtain consistent geometric models that contain a complete set of anatomical entities. Tuning these models with patient-specific data is another important issue. Anyhow, functional information related to 3D models has been barely addressed and it is part of the purpose of this work to show how recent advances in this field [23] can take advantage of 3D models and functional information through ontologies to develop new browsing and simulation capabilities.

In product representation as well as virtual human modeling, it is the purpose of the proposed contribution to analyze the main features of each of them to look for possible commonalities to define a more global infrastructure that can be useful for each category of 3D models as well as applications involving both 3D products and virtual humans.

The paper outline is as follows. Section 2 analyzes prior work with a particular focus on ontology modeling and the use of functional information. Section 3 highlights key concepts relating structure, functions and 3D models and corresponding issues and section 4 describes some purposes of these relationships. Section 5 details a framework to model virtual humans with ontologies and 3D models that involves functional information. This section addresses also the use of ontologies in product modeling where reasoning mechanisms contribute to add and process functional information. Finally, section 6 analyses the commonalities among virtual humans, products and functional information to specify the building blocks of a common ontological framework for virtual humans and products that can address efficiently simulation processes.

2. RELATED WORK

Essentially, assembly models have been proposed for design and manufacture applications [4, 26]. With the development of feature modeling, approaches set assemblies as components related to each other through geometric constraints [34]. These approaches share a common denominator where assemblies are described as geometric models enriched with technological information, e.g., component names, component materials, sub-system names, ... and functional data, e.g., component function, sub-system function, ... However, the closer to functional information, the higher the requirements to obtain information external to CAD environments and the greater the need of user's interactive input during a design process. Linking 3D geometric models with functional and other technological information is a key requirement to process efficiently assembly models for mechanical simulations [5, 30]. Knowledge-Based Engineering (KBE) approaches [8, 17] formalize engineering knowledge to automate some design tasks. KBE concepts take advantage of artificial intelligence techniques [20, 32] and strongly rely on language-based approaches. As such, they follow top-down approaches based on enterprise 'best practices' and address preliminary design stages rather than embodiment or detailed design ones, which is the product geometric description addressed here with DMUs (Digital Mock-Ups). This is hardly applicable to other companies and even more to other products. Using a bottom-up approach to determine precisely the connections needed components' 3D models and symbolic information, i.e. technological, functional, is the proposed approach addressed here based on the results of previous work [5, 30].

The development of ontologies in medicine has produced a wide range of contributions in the past years [12, 22] though few of them connect symbolic information with 2D or 3D geometric models [19, 22]. At present, these approaches address slice-based navigation in volumetric data, i.e. the user's viewpoint is restricted to the translation of the scanning device. However, these contributions exemplify the strong impact of coupling symbolic information with navigation within geometric models with new capabilities to query geometric models using topological relationships [22]. Ontology development in medicine has also addressed the extension of structural description of biological structures with functional information [6, 13]. Up to our knowledge, there is no approach providing a full 3D navigation environment where the user can modify arbitrarily the viewpoint, i.e. in rotation and translation like in any CAD environment, navigate using symbolic information to select 3D entities based on the biological structure of the 3D model or the functional meaning of its entities. Based on prior work, we focus on human body anatomy [23, 24] to set up these new capabilities and provide a framework that shares similarities with digital product processing as described above. Now, the purpose is to analyze the ontological framework, the 3D models and their interactions with functional information of both virtual humans and digital products to specify a common framework where all 3D models and ontologies can be inserted and interact with each other and with tasks such as simulation ones.

3. ANALYZING RELATIONSHIPS BETWEEN STRUCTURE-FUNCTIONS-3D MODELS FOR PRODUCTS AND VIRTUAL HUMANS

Prior to focus on the relationships involving functions of products or human beings anatomical entities, it is mandatory to rely on a common function definition so as to incorporate in the same environment digital representations of products and human beings.

3.1. Synthesis of Function Definitions for Products and Virtual Humans

Whether applied to products or to human beings, there are several definitions of functions [9, 15] and it is important, in the current context, to analyze and synthesize these definitions to reach a common point as required to set the required knowledge framework.

Considering biological systems [6, 13, 31], the definition adopted consider functions as the abstraction of biological processes or other entities toward a goal: when X has the function Y with the goal Z, then X is supposed to cause or otherwise bring about the state of the world Z, thus realizing Y. For example, it may be the case that a red blood cell transports oxygen. But the statement that "the function of the red blood cell is to transport oxygen" adds a goal or purpose to this description: the red blood cell is *supposed* to transport oxygen.

In our context, the ontology can incorporate the concept of function with a minimal structure expressing the relation between X and Y. This way, X and

Y are related to each other to focus on *what* is the consequence of Y.

Now, in the context of products, a diversity of function definitions exists also and can be synthesized with respect to different viewpoints [9]. Among them, the performance viewpoint states that a function is viewed as an abstraction of physical behavior. For example, consider a mechanical product which performs specified behaviors in specified situations (working conditions) and these achieve the same results. The set of behaviors define a functional class and the results are its function.

This definition differs from the previous one in the sense that it also refers to *how* the function is realized because it uses explicitly the concept of behavior. This is closely linked to the product design process [1, 9, 11] where the simulation of the product behavior is a critical task to ensure that a product fulfill the designer's prescription.

In our ontological framework where the purpose is to relate structural entities, i.e. anatomical entities and/or components, to functions, it is not in the scope of the ontology to describe the behavior representing a function because it is left to dedicated simulation software. Dropping the behavior part of a function restricts it to its '*what*' part. Consequently, these definitions become equivalent and functional information for virtual humans and products can be uniformly stated as the action caused by X: a biological system or a mechanical one.

3.2. Relationships between Structure – Functions – 3D Models

3D models of products and humans differ in many aspects, ranging from the geometric models used for their description to the way components or anatomical entities are identified and functions assigned.

Products employ 3D models of B-Rep NURBS type to generate representations required for manufacturing purposes. They are described as an assembly of components by means of a tree structure forming a hierarchy, i.e. assemblies/sub-assemblies/components (see Fig. 1). The names of components appearing in this tree are not standardized [14], hence not reliable and cannot be used efficiently for search processes. The names of product functions are not standardized either and they are not available with the CAD model of an assembly. Indeed, the components of an assembly are named (or labeled) with identifiers assigned by the design engineers and/or using company encoding principles. The functional designation (see Section 5.2 for details about this concept) of a component is part of a common body of knowledge of the designers, but it is not a common practice to use it as component label. However, there is some informal connection between a component name and its function, e.g. a screw is meant to fit into some component using its thread. This is informal because the screw can contribute to an assembly or a calibration function and a set screw or a stop screw are more precise designations that relate the screw to its function, i.e. they are functional designations of screws.

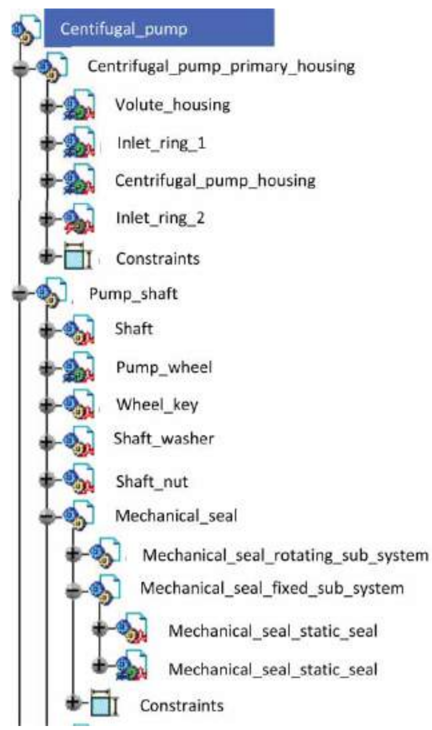

Fig. 1: Example of a subset of tree structure describing an assembly in a CAD system.

Often, the decomposition of a product into subsystems follows a graph structure rather than the tree structure prescribed by CAD systems. Further, there is a need for several decompositions of a product into subsystems and components depending on the user's needs during design or simulation processes (see Fig. 2), whereas CAD systems prescribe a unique tree structure for an assembly. For example, a hydraulic pump can be decomposed into hydraulic and mechanical subsystems (see Fig. 2b) to reflect some of its functions. Similarly, a kinematic simulation of this pump needs to refer to the various kinematic equivalence classes of components that define the movements of subsystems (see Fig. 2c) and the corresponding product structure is often a graph. Yet another decomposition can match component categories like standard components, standard sub systems and components that are specific to a product (see Fig. 2d). As such, there is no reference structure for a product. This is a consequence of the function – behavior – structure relationships [1, 11] where the analysis of a given behavior is derived from

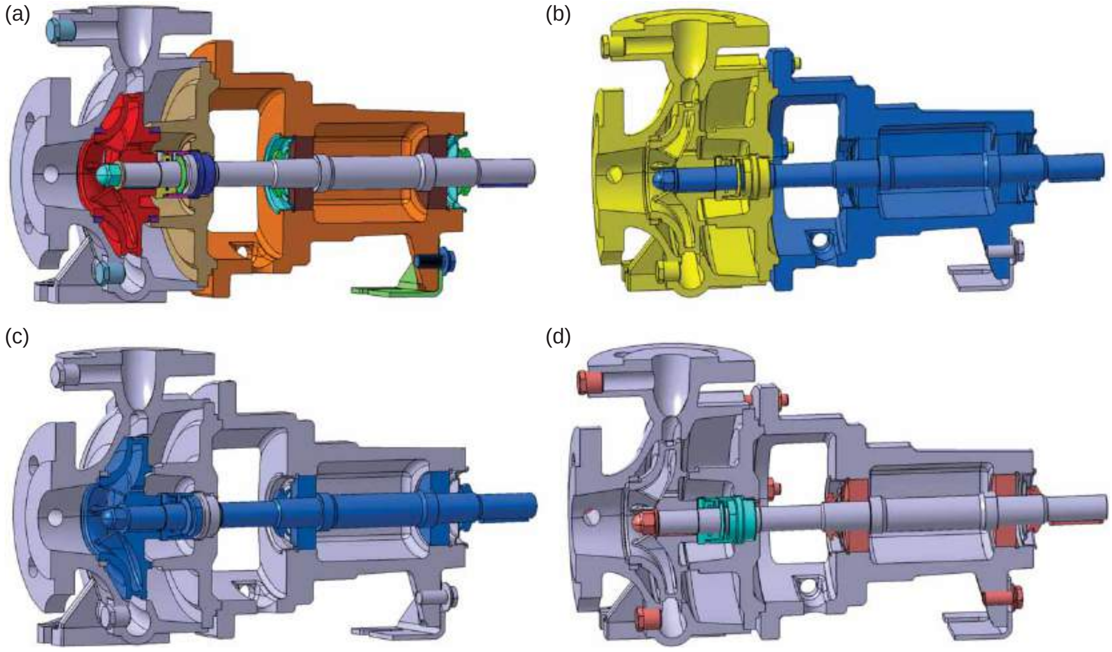

Fig. 2: An example of a hydraulic pump with various structures highlighted with different colors. A section view has been set up to display internal components. a) components are colored to ease the understanding of the product structure, b) component colors reflect the structure of the pump decomposed into a hydraulic subsystem (yellow), a rotational guiding subsystem (blue) and components not involved in the previous subsystems (gray), c) component colors reflect the kinematic structure of the pump: rotating components (blue), fixed components (gray), d) component colors reflect the category of a component set among: standard unitary components (pink), standard subsystems (cyan) and components specific to the product (gray).

a corresponding structure of the product to perform a function of this product.

As a complement to component functions, product functions satisfy a decomposition process into a taxonomy of overall functions, embodiment functions, and extrinsic/intrinsic functions [9]. However, this decomposition is not unanimously used and variants relates to primary, secondary functions [9]. Additionally, these decompositions are not precisely related to 3D models of components, which require analyses complementary to [5]. For these reasons, we don't take into account the function decomposition in the present work. This is a difference compared to virtual humans where it is referred to as a unique reference structure [16].

Virtual humans employ 3D models derived from NMR and/or tomography acquisitions through 3D reconstruction from slices. They are mesh-based 3D models generated independently of each other to produce anatomical entities that usually result into models inconsistent between them [35], i.e. two anatomical entities may interfere or can be disconnected rather than being in contact with each other.

Anatomical entities and functions are named in accordance with a standardized designation, i.e., the canonical anatomical terminology [7, 10, 16]. The structure of a virtual human is ideally identical for every virtual human and acts as a reference representation. Anatomical entities can be decomposed in a tree structure to express, for instance, how the skeleton is structured, without reusing twice the same anatomical entity. Anatomical entities participate in many functions of the human body. For instance, the knee participates both to the function of stability and movement of the body.

As we outlined, between products and virtual humans differences exist in terms of geometric models. Also, product's components and anatomical entities are not identified in the same way. Where a precise ontological taxonomy exists for anatomical entities through FMA (Foundational Model of Anatomy) [27], no such reference is available for product's components. This is perhaps the main difference between the two cases, in terms of classification. However, describing a human body anatomy and a product structure has some common features that can be used to set up a common framework. In both cases, the relationships between a product or a virtual human decomposition and a function are still open to a formal approach where many principal functions can be associated to an anatomical entity.

4. PURPOSE OF THE RELATIONSHIPS BETWEEN 3D MODELS – STRUCTURE – FUNCTIONS

Structural and functional information about products and virtual humans enhance the knowledge of the 3D models describing these mechanical and biological systems. When application software provides capabilities to display/select 3D models, its interactive behavior becomes a mandatory feature to manage the graphic entities attached to these 3D models. With such software, selection functions have been under focus to provide users with efficient means to reach the 3D content they are looking for (see the examples of Fig. 2). The most common approaches for multiple object selection include serial selection techniques that require the user to select objects one at a time, e.g. the ubiquitous ctrl + click (or shift + click) approach, and parallel selection techniques such as brushes, lassos, and selection shapes. However, as Lucas et al. [18] point out, each has certain limitations, especially in 3D. For instance, multiple objects may be difficult to distinguish, isolate, or even see due to occlusion, rendering size, environment clutter, and other display factors. Requiring the user to adjust the view can be tedious, cumbersome. From a user perspective, it is interesting to exploit functional information for visualization purposes. By associating a set of entities to a function, it is possible to retrieve and then display the set of entities needed by a given user, or even to split a given set of entities into subsets as needed by a given user like the examples of Fig. 2.

Similarly, for mechanical simulations there is the need to define sets of entities that contribute to a subset of a mechanical model as needed for a given simulation, e.g. extracting the set of anatomical entities that contribute to the knee flexion in the context of biological systems or extracting the set of components that are assembled with a given cap screw or set of cap screws in the context of a product. The same observation occurs also to define boundary conditions applied to each of these of these systems where other anatomical entities and components can be identified based adjacency properties rather than interactively with mouse clicks. Consequently, the generation of simulation models shares similarities with 3D scene management where the use of functional information can add efficiency to these tasks when each of them can rely on high-level queries.

Querying these informations results in a selection process, which is a common task to visualization as well as simulation preparation. Having a function-based selection process has already proved its interest [5, 30] but each category of 3D models differs in the type of 3D model used for their description. 3D products are commonly described with B-Rep NURBS CAD models whereas virtual humans are based on 3D mesh models, which significantly differ in terms of geometry processing but they stay rather similar when a selection process matters. It is the purpose of the next section to illustrate how complex selection processes can involve functional information.

5. ONTOLOGY-BASED ENVIRONMENTS FOR 3D PRODUCTS AND VIRTUAL HUMANS

We analyze now two frequent tasks addressed by products and virtual humans both: browsing and simulation preparation. Indeed, these tasks are frequent and can become very tedious in both cases. Navigating and selecting a component in either category of 3D models (product or virtual human) is tedious if performed classically with the (move + click) principle as well as other geometry-based selection principle [20]. Simulation preparation, as needed for products to study their structural behavior using Finite Element (FE) models or biomechanical simulations as needed to evaluate the mechanical behavior of virtual humans, also use FE-based or similar mechanical models to simulate movements. There, identifying the set of components needed for a given simulation, setting up the boundary conditions and adapting the components' shapes is a common denominator for both categories of models and these tasks are time consuming. Using functional information appeared to be an efficient means to speed up this preparation process for products [32] as well as virtual humans [11].

5.1. Structure of the Knowledge Base Describing a Virtual Human

As a case study, we present My Corporis Fabrica (MyCF) system. MyCF is an ontology-based tool for automatic reasoning and querying on complex anatomical models. Its structure can be outlined as follows. The current version of the ontology contains almost 74000 classes and relations describing canonical human anatomy, body functions, and 3D models. The system contains also a set of 11 inference rules used to semantically enrich the knowledge base; these are described next. The ontology is represented as an RDF database (www.w3.org/RDF). An RDF database is set of triples of the form $<$ subject, property, object $>$ that can also be seen as a graph. For example, anatomical knowledge about the fact that the *patella* is a *bone* is declared with the triple

$$< \mathrm{mcf : Patella, rdfs : subClassOf, mcf : Bone} >$$

MyCF data and rules are stored in a deductive RDF triple store build upon a Sesame RDF server (www.openrdf.org), and that can be queried with SPARQL (www.w3.org/TR/rdf-sparql-query) by remote-access facility via a web server (http://mycorporisfabrica.org/). The corresponding software architecture is illustrated in Fig. 3. The

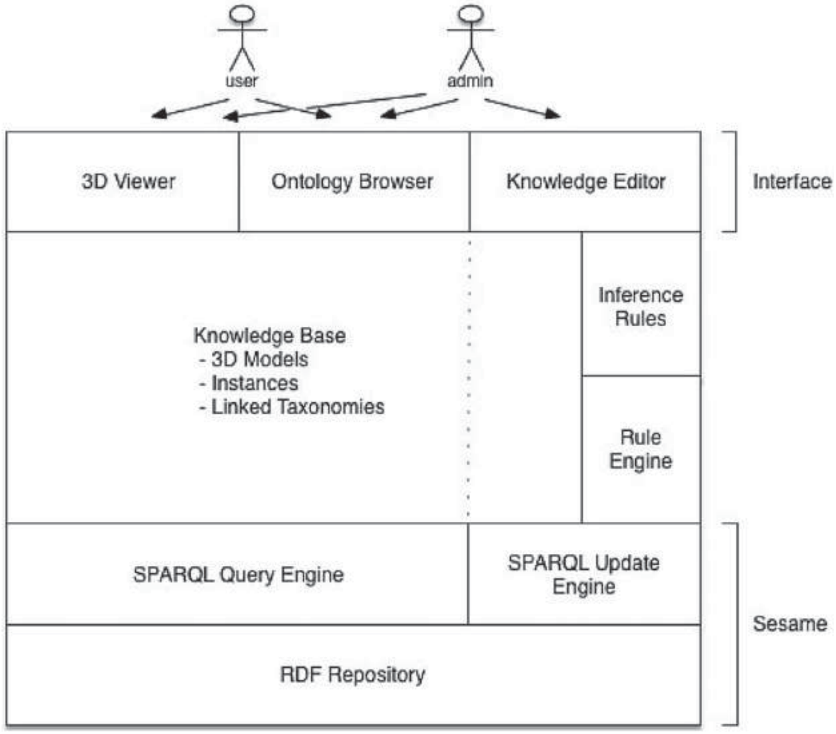

Fig. 3: System architecture of the anatomical ontology tool incorporating functional knowledge and inference mechanisms.

ontology can be edited using the Protégé editor [25] as part of the interface shown on Fig. 3.

With this web service in place, the ontology described by the knowledge base can be easily queried and visualized by other web applications. In phase of maintenance and evolution, it can be easily updated, just by entering or deleting triples and/or by modifying the set of rules, without having to change the reasoning algorithmic machinery used for answering queries. It is the strength of a declarative approach that allows a fine-grained domain-specific modeling and the exploitation of the result by a generic (domain-independent) reasoning algorithm.

The MyCF ontology is constituted by three interrelated taxonomies (see Fig. 4):

- The 3D models one that contains all the entities and relations required to display a 3D scene;
- The anatomical entity one which is a modified version of FMA that describes the human body anatomical entities and the structure of a human body;
- The body function one that structures the human body functions.

The anatomical core of the MyCF is based on FMA (Foundational Model of Anatomy ontology). The MyCF ontology introduces explicit links between anatomical entities, human body functions, and 3D graphic models of patient-specific body parts in order to form a whole knowledge base. Fig. 4 exemplifies the structure of MyCF ontology made of its three taxonomies. From left to right we have 3D models, anatomical entities, and functions.

The relations used to link the 3D taxonomy with anatomical entities and functions are *mcf:Describes, mcf:Displays* and *mcf:ParticipatesTo.* Fig. 4 illustrates how 3D objects are related to anatomical entities by the property *mcf:Describes.* 3D scenes (that are collections of 3D objects) are related to the body functions by means of property *mcf:Displays.* Finally, the fact that an anatomical entity is related with a function is stated by *mcf:ParticipatesTo.* As illustrated in Fig. 6, the taxonomy of 3D entities is constituted of two classes respectively called *mcf:3D-scene* and *mcf:3D-object.* The relation called *mcf:Contains* having the class *mcf:3D-scene* as domain and the class *mcf:3D-object* as range, defines a scene as a collection of objetcts. All concrete 3D scenes and objects are instances (or members) of these two classes. MyCF employs also four properties respectively called *mcf:Position*, *mcf:hasMesh*, *mcf:hasTexture* and *mcf:hasColour*, in order to possibly relate each

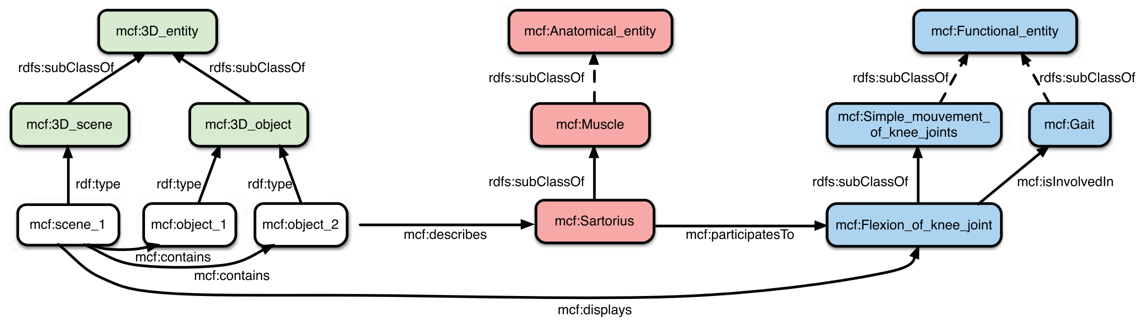

Fig. 4: Overall structure of the ontology and the three taxonomies contained in the knowledge base. Each color characterizes a taxonomy forming a subset of the ontology: 3D models (green), anatomical entities (pink), body functions (blue). White rectangles are instances of entities they are connected to. This figure contains only a small subset of the entities and properties contained in each taxonomy.

specific 3D-object to a position matrix, a mesh file, a texture file, and a color, respectively. For the sake of succinctness these are not reported.

Often, the use of ontologies is limited to the visualization of the basic domain-specific knowledge they contain. Yet, inferences allow a user to substantially enrich a knowledge base by adding implicit pertinent information. The MyCF system exploits reasoning aiming at expressing complex connections between anatomical entities and functions. The MyCF rules are then taken into account by an inference mechanism, which applies them on the base facts declared and stored as RDF triples in an iterative manner. This process is referred as *saturation.* Applying a rule means to instantiate the variables of its premises by linking them to explicit facts in the base data, and then by adding to the knowledge base new facts corresponding to the (appropriately instantiated) rule conclusion. The termination of this process is guaranteed by the form of the rules that are considered. MyCF uses Datalog rules. These rules are called *safe* since all the variables appearing in the conclusion of a rule also appear in its premises. It is worth noting that by exploiting inference rules the amount of facts a user would need to input in the knowledge base is reduced, which is thus added by inference. Rules can of course be added, removed or modified without impacting the inference mechanism.

Some rules in the current version of MyCF are the followings. Below, we range over variables with ?a, ?b and ?c.

(R1) If < ?a mcf:PartOf ?c > and < ?c mcf:PartOf ?b > then < ?a mcf:PartOf ?b > ;
(R2) If < ?a rdfs:subClassOf ?c > and < ?c mcf:InsertOn ?b > then < ?a mcf:InsertOn ?b > ;
(R3) If < ?a mcf:ParticipatesTo ?c > and < ?c mcf:IsInvolvedIn ?b > then < ?a mcf: ParticipatesTo ?b > ;

These rules allow us to present more in the detail the properties used in MyCF to describe virtual humains. The *mcf:PartOf* property is a complement to the classical *rdfs:subClassOf* property that is used throughout FMA ontology that prescribes only one tree structure of the human body anatomy. As an example, *joint* is a *part of* the *articular system* but joint is *not* a subclass of the articular system. This shows how a new human body structure can be described and navigated using the *mcf:PartOf* property. Rule R1 says that *mcf:PartOf* is a *transitive* property. To illustrate, because *joint* is a *part of* the *articular system* and the *articular system* is a *part of the Musculoskeletal_system, we conclude that joint* is a *part of the Musculoskeletal_system.*

The property *mcf:InsertOn* is used to specify attachment areas of anatomical entities. This knowledge is important in anatomy and also for biomechanical simulation purposes to express explicitly the connections among anatomical entities. For instance, the *distal tendon of right sartorius* is *inserted on* the Medial *part of proximal epiphysis of right tibia*, is expressed by adding the RDF triple:

<Distal_Tendon_Of_Right_Sartorius mcf:InsertOn Medial_part_of_proximal_epiphysis_of_right_tibia>.

Rule R2 says that if a given class representing an anatomical entity ?a (e.g., Sartorius) is a subclass of an anatomical entity ?c (e.g., Muscle) that is known to be inserted on an anatomical entity?b (e.g., Bone), then ?a is inserted on ?b (Sartorius inserts on a Bone).

Exploiting the relationships between anatomical and functional entities is decisive to retrieve the entities participating to some functions, and vice-versa. It is of key importance to be able to display/select interactively 3D geometric entities. To address this issue, the MyCF ontology presents two domain-specific relations, *mcf:hasFunction* and *mcf:participatesTo.* The *mcf:hasFunction* property is used to denote that an anatomical entity, as a whole, realizes a given function. The *mcf:participatesTo* property expresses that a given anatomy entity participates to a function. To refer to section 3, *mcf:hasFunction* designates the primary function of an anatomy entity whereas *mcf:participatesTo* expresses that a given anatomy entity may not perform the targeted function alone.

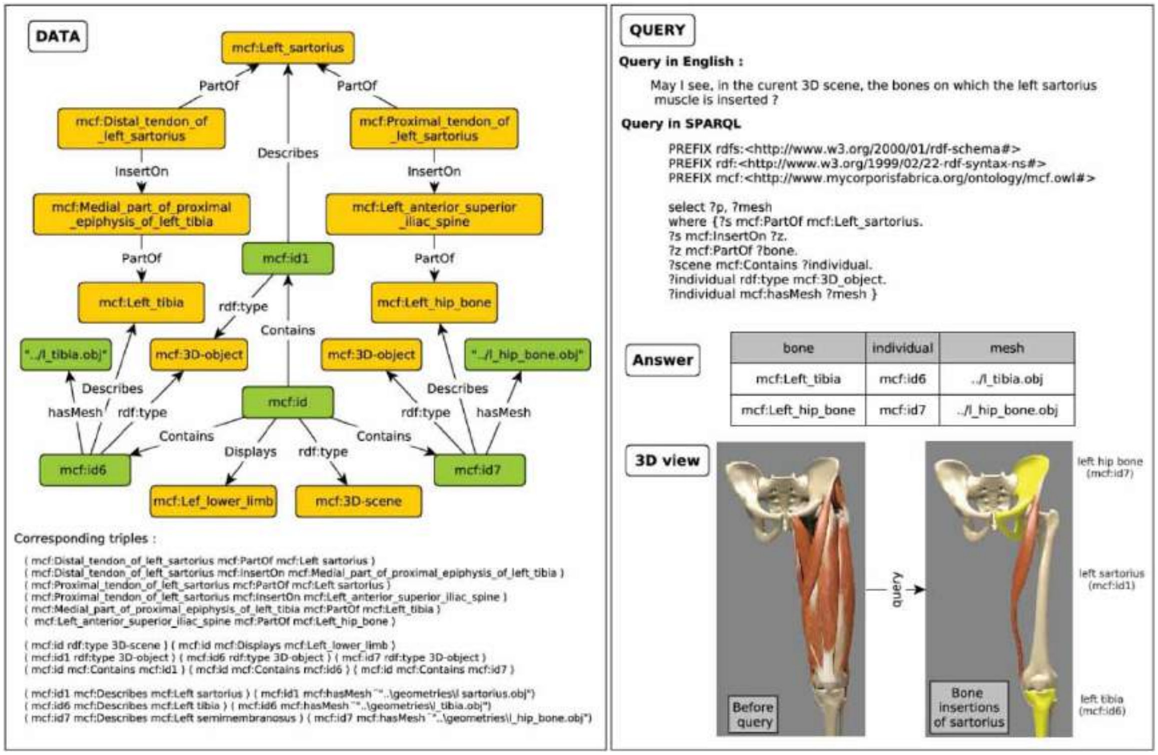

Fig. 5: Query example to select a set of anatomical entities for display or biomechanical simulation based on their connections with neighboring ones. The 3D view illustrates the effect of the query. The content of the knowledge base used to set up the query is depicted in the DATA rectangle.

The *mcf:IsInvolvedIn* property structures the anatomical functions to express how functions rely on each other, i.e. < ?F IsInvolvedIn ?F’ >, where F and F’ are functions both. This is comparable to *mcc:PartOf* where this property applies to anatomical entities only that expresses anatomical entities are constituted. For example, the *eversion of the foot* is involved in the *mobility of ankle joints*. For example, using rule R3, we can infer that the muscle *sartorius* participates to the function of *movement of knee* from the facts that *sartorial* participates to the *function knee flexion* and that the *function knee flexion* is involved in the function *movement of knee*.

With this ontology scheme, it is now possible to avoid the selection and display of these entities using purely geometry-based approaches. The 3D taxonomy aims at defining the smallest content enabling elementary tasks to display and select 3D objects. Of course, this can be enriched to refer to geometric criteria and other entities in this taxonomy.

We conclude this section by illustrating the contribution of this architecture with a query example showing its capability to let a user retrieve the anatomical entities useful to display or realize a biomechanical simulation (see Fig. 5). The objective is to select the subset of bones on which the *left Sartorius* muscle is inserted. We stress that the corresponding interactive selection with mouse clicks would be more tedious and error prone, especially if the anatomical entity has a complex shape and is hardly visible under some camera viewpoint. The DATA rectangle in Fig. 5 describes the subset of the knowledge base used for this query and how some of the properties described previously contributes to this query. There, the green rectangles are instances of classes whereas yellow ones are classes belonging to either of three taxonomies defining the ontology. The query expressed in SPARQL and the corresponding 3D views before and after querying the scene are depicted in the right rectangle of Fig. 5.

5.2. Structure of a Knowledge Base Describing a Product

Similarly, a digital description of a man-made product can take advantage of 3D geometric models and knowledge representation to improve the product description and the generation of simulations. Elements described here are based on prior work [5, 30]. Here, the purpose is to compare it with the previous virtual human description as a basis for the common framework addressed at section 6. From [30], the product description is set up in bottom-up way, starting with a geometric model of each of its components where component names are not regarded as reliable information. Other approaches [1, 28] have not stressed a tight connection between

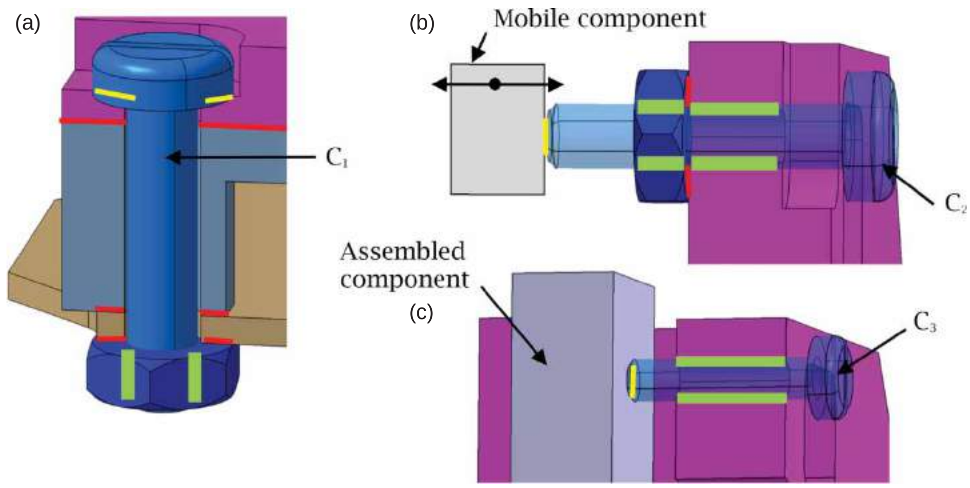

Fig. 6: Example configurations of functional designations of involving screws and their corresponding geometric interfaces: (a) a cap screw, (b) a stop screw, (c) a set screw. The location of geometric interfaces is indicated with colored line segments. Yellow lines indicates contact areas between the screw and its neighboring components, green lines locate interference areas between the screw and red ones locate contact areas between neighboring components of the screw that contribute to the function of the screw. In (b), the contact area at the extremity of the screw is highlighted though it is intermittent and, presently, not featuring the mobile component in contact with the screw.

3D geometry of components and functional information. The major contribution of [30] is the generation of Functional Designations (FD) of components. The ontology framework is based on Protégé [25] and OWL, FACT++ [33] and a specific development to perform some qualitative reasoning.

A FD of a component C refers to a text-based annotation T_C of C such that T_C uniquely identifies the principal function of C, independently of its dimensions and shape. For example, the functional designation of a screw component can be either cap screw, or stop screw of set screw (see examples in Fig. 6). This annotation T_C uniquely characterizes the set of functions of C because C may perform several functions. T_C is a member of the taxonomy of functional designations T_{fd} associated with a DMU. T_C designates a class of T_{fd}. T_{fd} contains a collection of FDs of interest for the analysis of this DMU. T_C is connected to the geometric entities of ∂C, the surface boundary of C, contributing to the geometric interfaces meaningful for T_C and hence, to the neighboring components of C contributing to T_C. This is applicable to a set of configurations where the function or the set of functions of C is derived from the nature and spatial locations of geometric interfaces that C shares with its neighboring components (see Fig. 6). Consequently, the FD uniquely identifies the set of functions of C. This illustrated in Fig. 6 with the same screw involved in three different FDs, i.e. cap screw (C_1), stop screw (C_2), set screw (C_3), where the corresponding geometric interfaces and locations are highlighted. Corresponding functions are respectively: tighten components (C_1), set component position (C_2) and tighten component using friction (C_3).

From a DMU input using a STEP file, the assignment of FDs is achieved through the following steps:

- Identification of geometric interfaces between the components of the DMU [15]. Depending on the category of configuration between two components, this interface can be of type contact, interference or clearance. [15, 30] have addressed some configurations of contact and interference with the objective of defining precisely the interfaces in the B-Rep model of each component. Because contacts and interferences often take place from simple surfaces (planes, cylinders, spheres, cones), it has been possible to set up a taxonomy T_{CI} of these configurations that are designated as Conventional Interfaces (CI) between components (see Fig. 7);
- Generation of Functional Interfaces (FI) from CIs. From T_{CI}, it is possible to derive a taxonomy of geometric interfaces with one or more functional meaning per CI. The functional meaning assigned to each FI is rather elementary because a CI reduces to a couple of simple surfaces, e.g. a CI defined as a cylindrical contact between two components. Consequently, the corresponding FI can be either a cylindrical loose fit or a cylindrical tight fit depending on the clearance between the cylinders. Most often, the clearance parameter is not explicit in a DMU, i.e. the cylinders in contact have the same diameter and no clearance exists in the geometric model of an assembly.

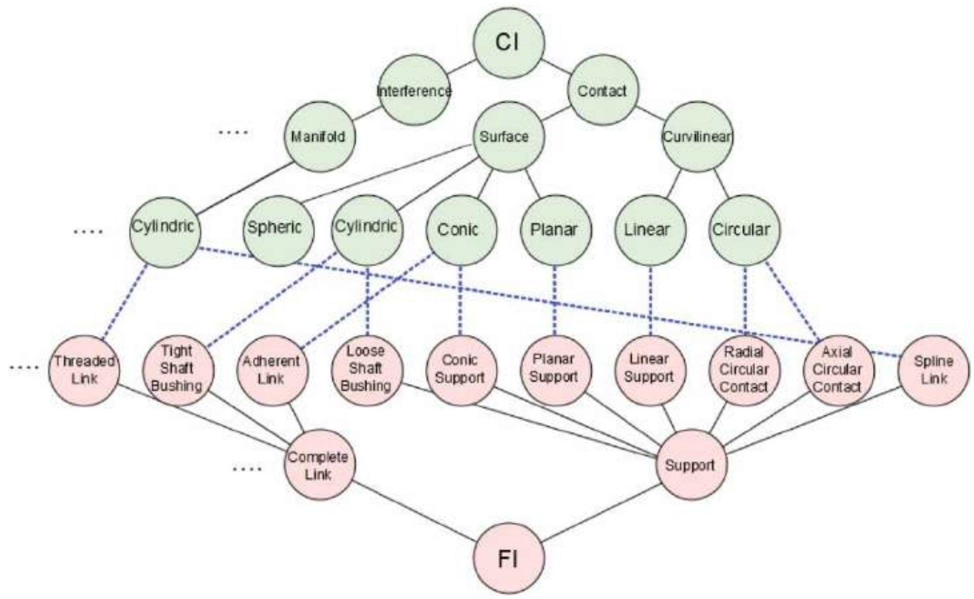

Fig. 7: Subsets of the taxonomies of conventional interfaces and functional interfaces. Dotted blue lines indicate properties between leaves of the CI taxonomy and leaves of the FI taxonomy characterizing all the functional interpretations that can be derived from each leaf of the CI taxonomy.

It is a consequence of the conventional representation of components in an assembly. In the example of Fig. 6, the interferences attached to C_1, C_2, C_3 produce two FIs: threaded link and spline link, because threads are not represented explicitly and a spline link can be idealized into two simple cylinders interfering with each other. Geometrically, FIs use the same entities as CIs. The purpose of FIs is to add functional information compared to CIs;

- Selection of one FI per CI through a qualitative reasoning process [30]. This reasoning process is not detailed here since it has been implemented using an interval-based arithmetic because it relies on mechanical equations, e.g. static equilibrium equations, that are not suited for an implementation using an ontological reasoner. The reasoning process refers to the concept of reference state of a component C to derive equations that refer to the geometric interfaces of C and may filter out some FI when multiple ones are attached to the same CI. In Fig. 6, this reasoning process would retain the threaded link as unique FI for each interference of C_1, C_2, C_3. If the reasoning process cannot retain a unique FI per CI, user's interactions are needed to select the correct FI;
- Having one FI per geometric interface, a reasoning process takes place that assigns a FD to each component. In [30], this is performed using the FACT++ reasoner that produces the following FDs: C_1 (cap screw), C_2 (stop screw), C_3 (set screw). The inference rules set up refer to all the FIs of C as well as the relative spatial position of these FIs, e.g.if $<$?C Has_a ?FI$_{th}$ $>$ and $<$?C Has_a ?FI$_{ps}$$>$ and $<$?FI$_{th}$ Is_orthogonal_to ?FI$_{ps}$$>$ then $<$?C Is_a *set_screw*$>$.Complementary rules are used to distinguish the cap screw from stop screw and the set screw.

Based on the above description, it appears that the FD of a component C cannot be regarded as a simple annotation, i.e. setting up a logical connection between a 3D model of C and a symbolic information (the FD of a component), because the FD relates to subsets of C through its FIs and their spatial relationships. Depending on the FD of C, its FD can also involve FIs of neighboring components of C (see Fig. 6a, b). As a result, a FD structures also the 3D model of C. Advantage can be taken from this observation to set connections between component or sub-assembly functions and component FD and use this overall structure through a function-based selection process [5]. This selection process has led to a template-based approach where a template refers to one or more functions to select components in DMU. The template addressed in [5] focuses on screw-based assembly functions and the user can add parameters to restrict the selection process to instances involving a given number of tightened components and can also be combined with geometric parameters like screw diameters for example. However, this template-based approach still requires an extension to larger range of functions to get a better insight of the relationships between functions and FDs of components.

3D viewing capabilities and geometry processing mentioned in the above description are based on OpenCascade library [21].

6. TOWARD A COMMON ONTOLOGY FRAMEWORK FOR 3D PRODUCTS AND VIRTUAL HUMANS

6.1. Framework Content

Based on the above description of ontology-based environments, now the purpose is to analyze and synthesize these descriptions to evaluate the applicability of a common framework where a virtual human ontology and a product ontology can be connected to each other to share a common framework for generating biomechanical or mechanical simulations, respectively. Additionally, such a common framework is of interest to be able to describe configurations where a virtual human would be equipped with prosthesis and biomechanical simulations would be derived from this configuration to evaluate the adequacy of this prosthesis.

We can observe the following common points between the virtual human and the product ontologies.

(Data models and Reasoning) The knowledge base generated from an RDF triplestore equipped with Datalog rules can be used both for representing virtual humans and man-made products.

As outlined in the paper, all facts and inference rules of the virtual human ontology are expressed using an RDF/RDFS framework accessed using SPARQL query and update engines. Within this framework, inference rules can be used to propagate new facts to all the relevant entities of the knowledge base. This saturation process leaves the knowledge base ready for querying any entity without requiring careful and complex update processes. Compared to OWL-based ontology description as in [5, 13, 22, 30] and reasoners such as FACT++ (or others) that have been used for managing man-made products, the use of RDF/RDFS is not regarded as a subset of OWL. Indeed, it has interesting non-first-order features, like the possibility of treating values both as constants and as classes or properties. Additionally, the RDF query language SPARQL has the capability to query at the same time the data and the schema and allows that variables stand for classes and properties. The virtual human ontology takes advantage of Datalog rules to perform inferences on top of RDF datasets. Datalog rules and Description Logics (DL), i.e. the logic frameworks used by OWL-based reasoners, are two orthogonal decidable fragments of first-order logics that have been extensively studied in deductive databases and knowledge representation. Datalog rules are regarded as easier to read and write for practitioners than DL-based rules. In addition, Datalog rules have a polynomial data complexity while allowing expressing complex interactions between properties and recursivity that have been exploited in the virtual human ontology. The rules expressed in DL/FACT++ for generating functional designations of product components can be expressed with Datalog rules and SPARQL queries. Thus, there is an effective alternative to the use of DL/FACT++ in the framework of product description. Finally, the RDF triplestore build using Sesame server is a low level layer in the software architecture that can host both the virtual human ontology and the product one.

(Basic Concepts: Anatomical Entities vs. Components) The concept of anatomical entity used in the virtual human ontology is subdivided into all the diversity of entities forming the human body and each of them must be clearly distinguished from the other. As a result, every named entity of the human body acts as an identifier of the corresponding physical entity and its corresponding 3D model. When relating a function to any anatomical entity, this function becomes clearly identified and geometrically located when referring to the corresponding 3D model of this entity.

In the context of a product, the elementary entity that can be equivalent to the anatomical entity is the component, i.e. the elementary item of a product. However, it is important that the component name be unambiguously related to its functions, as it is the case in the virtual human ontology. It is therefore mandatory that the component taxonomy be defined from the component FD rather than from the component name as it appears in a bill of materials.

Consequently, the qualitative reasoning and the inference processes that take place during the analysis of a DMU are required to obtain consistent connected sets of 3D geometric models and semantic information.

(Reference Taxonomies) The concept of product differs from that of the virtual human due to the fact that the structure of a product can vary significantly from one to another. On the one hand, the product structure influences the number and type of components it contains. On the other hand, a virtual human, even if patient-specific models are generated, is described with a constant anatomical structure. To take into account this difference, a product taxonomy must be added and connected to the taxonomy of components to take into the variability of a product structure and take advantage of product functions.

(Geometric Interfaces) The anatomical entities are linked to 3D models that describe their corresponding shapes. The connection between anatomical entities is symbolically expressed using the *mcf:InsertOn* property.

Now, considering a product model, the generation of simulations using Finite Element (FE) models has shown the benefit of having an explicit representation of geometric location of the connection between components, i.e. their geometric interface [5]. This geometric model is at the basis of geometric transformations needed to generate the shape of each component and to produce a consistent FE mesh. This interface can be also involved in the specification of specific mechanical models expressing the relative behavior between the components, e.g. friction. Therefore, the explicit geometric representation of interfaces between components is a necessary complement to the symbolic representation of the same concept in the ontology.

As observed previously for the virtual human model, the geometric description of the interface between anatomical entities is missing. However, it has to be considered that biomechanical simulations involving a subset of anatomical entities can rely on a library of mechanical models based on a reference human body model [35]. This library can be configured such that the biomechanical models can be generated consistently when taking into the effective dimensions and shape of the anatomical entities of the reference model. However, a limitation occurs if the virtual human ontology is applied to patient-specific models. Recently, the concept of anatomy transfer has been introduced that is able to adapt anatomical entities from an external skin to another one [2]. If the generation of consistent bio-mechanical models relies on a library of mechanical entities, they can be subjected to the use of an anatomy transfer to preserve the consistency of the mechanical for patient-based simulations.

Indeed, if a product is acting as prosthesis, geometric interfaces between the components of the prosthesis and anatomical entities will appear that is a new category of interfaces. Some features of this new category are addressed hereunder.

(Property Refinement) Regarding the properties added to the taxonomies' entities, the *mcf:PartOf* is clearly applicable to a product to define its structure. Additionally, a product structure can be addressed from different viewpoints (see Section 3), e.g. a product decomposition based on sub-systems as they as needed during an assembly process or a decomposition as it appears in kinematics when it is decomposed into kinematic equivalence classes. No such requirement has been observed for virtual humans. Thus, new properties or specialization of the *mcf:PartOf* may be required to process product models. Issues about the *mcf:InsertOn* property have been already addressed in the previous observations. Other properties like the *mcf:HasFunction* and others as they appear on Fig. 6 share similar meanings independently of the application context, i.e. virtual humans or products, and are generic enough to contribute to a common framework.

(Function-driven Visualization) At the abstract level, the concept of function is addressed rather similarly in the product design context and in the human anatomy, though the function designations differ significantly between a product and a virtual human (see Section 3). The ontology structure explicates *what* does a function. The product behavior associated with a function expresses *how* components perform to realize a function and requires complementary processing like the qualitative reasoning behavior mentioned at section 5.2. Again, this commonality can be exploited to contribute to a common framework that applies to browsing and simulation preparation needs. Browsing processes are improved through the use of functional information because it is a new means to characterize a group of 3D entities sharing a common concept that is meaningful from the user's point of view. Similarly, biomechanical simulation preparation is improved with functional information because simulation objectives focus on groups of anatomical entities that are meaningful with respect to a given body function. Consequently, selecting the appropriate entities helps preserving the consistency of the simulation and setting up the correct simulation parameters and boundary conditions.

(Geometric models) Anyhow, the framework must be able to process 3D digital models of components as well as 3D anatomical entities with their corresponding functional information. As described in section 5, virtual humans and products differ with respect to the category of 3D models they process. Anatomical entities are essentially described with 3D meshes, i.e. facetted representations whereas products are based on CAD models bounded by free-form surfaces expressed as B-Rep. However, displaying B-Rep models requires generating a visualization model that ends up being a mesh model similar to meshes of anatomical entities. In the context of CAD software, the reference B-Rep and the mesh models of a component live simultaneously when display and other geometry processing tasks are required. Consequently, product components can be assigned with the two categories of 3D models that can be used on purpose. This solution is acceptable as long as geometry processing focuses on either category of model. However, if it comes to process geometric interfaces between anatomical entities and components, this configuration requires further analysis that is left for future work.

6.2. Methodology and Tools

Achieving a common framework where products and virtual humans can be symbolically and geometrically represented and visualized is possible because of the strong commonalities between the two models, i.e., a human body can be seen as a structure of anatomical entities likewise a product is a structure of components. Here, it has been showed in Section 3.1 that a common concept of function could be applicable to virtual humans and products. Reconciling the differences that occur between them is necessary and requires however some efforts in the capitalization of product knowledge. Methods and tools needed for this goal are now discussed.

The knowledge representation framework constituted by RDF(S), Datalog Rules, and SPARQL queries is a formalism which is expressive enough to accommodate the description of both products and virtual humans, and their querying. From a practical point of view, this knowledge can be accessed with Semantic Web technology for RDF data such as Sesame Server or Jena that have already been proved to be effective.

The differences between the detailed description of the human body and that of products can be filled with an effort coming from the CAD and mechanical engineering communities in capitalizing the knowledge about assemblies and products. First, there is the need to provide a description of components (that are the atomic entities of an assembly) in terms of their function (see Section 5.2). Second, a macroscopic description of sub assemblies up to whole assemblies and products is also needed to describe complex 3D products. Finally, geometric interfaces between components that are crucial in simulations of a product should also find a correspondence in the ontology (see Section 5.2). These are essentially modeling tasks that should be carried by CAD and knowledge representation experts and geometry processing tasks to produce the geometric entities of components interfaces [15, 30] that still need further developments.

The visualization of products and virtual humans is possible in a common setting because generating a visualization model for a product ends up being a model similar to that of anatomical entities. This can be achieved with standard 3D visualization tools.

7. CONCLUSION

Given the digital descriptions of products with DMUs and of virtual humans, these two categories of models face similar issues among which, browsing and simulation preparation have been addressed to improve the efficiency of these tasks using functional information. The ontology-based framework set up for virtual human anatomy appears to be versatile enough to be adapted to digital products. The commonalities observed between 3D products and virtual humans in the context of browsing and simulation preparation and their adequacy with the ontology-based framework validates the interest of this approach to process symbolic information, like functional data, connected to 3D models.

Whereas the virtual human anatomy benefits of a well defined taxonomy of anatomical entities and functions, products must relate to the concept of functional designation to obtain an equivalent description. The assignment of functional designations requires qualitative reasoning processes and inference mechanisms that do not occur for virtual humans. However, the qualitative reasoning processes can be interfaced to the proposed ontological framework and the required inference mechanisms complement those acting at the level of the product ontology.

Ongoing work focuses on developing an ontology for biomechanical simulations that complement the virtual human ontology currently set up. Work is also performed to carry on the integration of the product ontology into the proposed ontology framework.

ACKNOWLEDGEMENTS

This work was partially funded by the advanced grant EXPRESSIVE from the European Research Council [ERC-2011-ADG_20110209].

REFERENCES

[1] Albert, A.; Burkardt, N.; Ohmer, M.: Contact and Channel Model for Pairs of Working Surfaces, Advances in Design, H. A. ElMaraghy, W. H. ElMaraghy, Springer, London, 511–520, 2006.

[2] Ali-Hamadi, D.; Liu, T.; Gilles, B.; Kavan, L.; Faure, F.; Palombi, O.; Cani, M-P.: Anatomy transfer, ACM Trans. Visualization, 32(6), 2013, 188 :1–8.

[3] Baek, S-Y.; Lee, K.: Parametric human body shape modeling framework for human-centered product design, Computer-Aided Design, 44, 2012, 56–67. http://dx.doi.org/10.1016/j.cad.2010.12.006

[4] Barbau, R.; Krima, S.; Rachuri, S.; Narayanan, A.; Fiorentini, X.; Foufou, S.; Sriram, R. D.: Ontostep: enriching product model data using ontologies, Computer-Aided Design, 44(6), 2012, 575–90. http://dx.doi.org/10.1016/j.cad.2012.01.008

[5] Boussuge, F.; Shahwan, A.; Léon, J-C.; Hahmann, S.; Foucault, G.; Fine, L.: Template-based geometric transformations of a functionally enriched DMU into FE assembly models, Computer-Aided Design and Applications, 11(4), 2014, 436–449. http://dx.doi.org/10.1080/16864360.2014.881187

[6] Burek, P.; Hoehndorf, R.; Loebe, F.; Visagie, J.; Herre, H.; Kelso, J.: A top-level ontology of functions and its application in the Open Biomedical Ontologies, Bioinformatics, 22(14), 2006, 66–73. http://dx.doi.org/10.1093/bioinformatics/btl266

[7] Burger, A.; Davidson, D.; Baldock, R.: Anatomy ontologies for bioinformatics: Principles and practice, Computational Biology, Springer, 2008. http://dx.doi.org/10.1007/978-1-84628-885-2

[8] Chapman, C.B.; Pinfold, M.: Design engineering – a need to rethink the solution using knowledge based engineering, Knowledge-based systems; 12, 1999, 257–267.

http://dx.doi.org/10.1016/S0950-7051(99)00013-1

[9] Deng, Y.-M.; Britton, G. A.; Tor, S. B.: A Design Perspective of Mechanical Function and its Object-Oriented Representation Scheme, Engineering with Computers, 14(4), 1998, 309–320. http://dx.doi.org/10.1007/BF01201762

[10] FCAT (Federative Committee on Anatomical Terminology): Terminologia Anatomica, International anatomical terminology, Thieme, 1998.

[11] Gero, J. S.: Design prototypes: a knowledge representation schema for design, AI Magazine, 11(4), 1990, 26–36.

[12] Hill, D. P.; Adams, N.; Bada, M.; Batchelor, C.; Berardini, T. Z.; Dietze, H.; Drabkin, H. J.; Ennis, M.; Foulger, R. E.; Harris, M. A.; Hastings, J.; Kale, N. S.; de Matos, P.; Mungall, C. J.; Owen, G.; Roncaglia, P.; Steinbeck, C.; Turner, S.; Lomax, J.: Dovetailing biology and chemistry: integrating the Gene Ontology with the ChEBI chemical ontology, BMC Genomics, 14(513), 2013. http://www.biomedcentral.com/1471-2164/14/513 .

[13] Hoehndorf, R.; Ngonga Ngomo, A.-C.; Kelso J.: Applying the functional abnormality ontology pattern to anatomical functions, J. of Biomedical Semantics, 1(4), 2010, 1–15.

[14] Iyer, N.; Jayanti, S.; Lou, K.; Kalyanaraman, Y.; Ramani, K.: Three-dimensional shape searching: state-of-the-art review and future trends, Computer-Aided Design, 37, 2005, 509–530. http://dx.doi.org/10.1016/j.cad.2004.07.002

[15] Jourdes, F.; Bonneau, G-P.; Hahmann, S.; Léon, J-C.; Faure, F.: Computation of components' interfaces in highly complex assemblies, Computer-Aided Design, 46, 2014, 170–178. http://dx.doi.org/10.1016/j.cad.2013.08.029

[16] Kostanjsek, N.: Use of The International Classification of Functioning, Disability and Health (ICF) as a conceptual framework and common language for disability statistics and health information systems, World Health Organization, Classifications, Terminology and Standards, Geneva, Switzerland, Vol.11 Suppl. 4, 2011. http://dx.doi.org/10.1186/1471-2458-11-S4-S3

[17] La Rocca, G: Knowledge based engineering: between AI and CAD. Review of a language based technology to support engineering design, Advanced Engineering Informatics; 26, 2012, 159–179. http://dx.doi.org/10.1016/j.aei.2012.02.002

[18] Lucas, J.; Bowman, D.; Chen, J.; Wingrave, C.: Design and evaluation of 3D multiple object selection techniques, In Proceedings of the ACM I3D, 2005.

[19] Milyaev, N.; Osumi-Sutherland, D.; Reeve, S.; Burton, N.; Baldock, R. A.; Douglas Armstrong, J.: The Virtual Fly Brain browser and query interface, Bioinformatics, 28(3), 2012, 411–415. http://dx.doi.org/10.1093/bioinformatics/btr677

[20] Ohsuga, S.: Toward intelligent CAD systems, Computer-Aided Design, 21, 1989, 315–337. http://dx.doi.org/10.1016/0010-4485(89)90039-0

[21] OpenCascade CAD software library; URL: http://www.opencascade.org/ , 2012.

[22] Osumi-Sutherland, D.; Reeve, S.; Mungall, C. J.; Neuhaus, F.; Ruttenberg, A.; Jefferis, G. S. X. E.; Douglas Armstrong, J.: A strategy for building neuroanatomy ontologies, Bioinformatics, 28(9), 2012, 1262–1269. http://dx.doi.org/10.1093/bioinformatics/bts113

[23] Palombi, O.; Ulliana, F.; Favier, V.; Léon, J-C; Rousset, M-C. : My Corporis Fabrica: an ontology-based tool for reasoning and querying on complex anatomical models, to appear in J. of Biomedical Semantics, 2014.

[24] Palombi, O.; Bousquet, G.; Jospin, D.; Hassan, S.; Revéret, L.; Faure, F.: MyCorporisFabrica: a unified ontological, geometrical and mechanical view of human anatomy, In: Lecture notes in computer science, LNCS, 5903, 2009, 207–219.

[25] Protégé, url: http://protege.stanford.edu/, 2012.

[26] Rahmani, K.; Thomson, V.: Ontology based interface design and control methodology for collaborative product development, Computer-Aided Design, 44(5), 2012, 432–44. http://dx.doi.org/10.1016/j.cad.2011.12.002

[27] Rosse, C.; Mejino, J. L. V.: A reference ontology for biomedical informatics: the foundational model of anatomy, J. of Biomedical Informatics, 36, 2003, 500. http://dx.doi.org/10.1016/j.jbi.2003.11.007

[28] Roy, U.; Pramanik, N.; Sudarsan, R.; Sriram, R. D.; Lyons, K. W.: Function–to-form mapping: model, representation and applications in design synthesis, Computer-Aided Design, 33(10), 2001, 699–719. http://dx.doi.org/10.1016/S0010-4485(00)00100-7

[29] Schmid, J.; Kim, J.; Magnenat-Thalmann, N.: Extreme leg motion analysis of professional ballet dancers via MRI segmentation of multiple leg postures, Int. J. Computer Assisted Radiology and Surgery, 6(1), 2011, 47–57. http://dx.doi.org/10.1007/s11548-010-0474-z

[30] Shahwan, A.; Léon, J-C.; Foucault, G.; Trlin, M.; Palombi, O.: Qualitative behavioral reasoning from components' interfaces to components' functions for DMU adaption to FE analyses, Computer-Aided Design, 45, 2013, 383–394. http://dx.doi.org/10.1016/j.cad.2012.10.021

[31] Smith, B.; Munn, K.; Papakin, I.: Bodily Systems and the Spatial-Functional Structure of the Human Body, Studies in Health Technology and Informatics, 102, 2004, 39–63.

[32] Smithers, T.: AI-based design versus geometry-based design or why design cannot be supported by geometry alone, Computer-Aided Design, 21, 1989, 141–150. http://dx.doi.org/10.1016/0010-4485(89)90068-7
[33] Tsarkov, D.; Horrocks, I.: Fact + + description logic reasoner: system description, In: Int. Joint Conf. on Automated Reasoning (IJCAR 2006). LNCS, 4130, 2006, 292–307.
[34] van Holland, W.; Bronsvoort, W. F.: Assembly features in modeling and planning, RCIM, 16, 2000, 277–94.
[35] Zygote; url: http://www.zygote.com/, 2014.