Deriving Functionality from 3D Shapes: Ontology Driven Annotation and Retrieval

Elena Camossi, Franca Giannini and Marina Monti

Istituto di Matematica Applicata e Tecnologie Informatiche – CNR, camossi(giannini,monti)@ge.imati.cnr.it

ABSTRACT

3D shapes are of crucial importance in several applications but capturing, representing and processing the complexity of the knowledge a shape may reveal is not an easy task. Therefore it becomes crucial to study which form characteristics may have a specific meaning in order to make explicit, sharable and, above all, machine understandable, the knowledge embedded in 3D shapes. This may be done from different perspectives due to the increasing diversity of potential users. This paper illustrates a knowledge-based (KB) system supporting the functionality driven annotation and retrieval of 3D models. Kernel of the system is the *Form, Functionality and Behavior ontology*, which provides a representation of the knowledge needed to infer object functionality and behavior from the shape of models and their components.

Keywords: Shape semantics; ontology; 3D shape-based search; function-based shape reasoning; form, functionality and behavior.

1. INTRODUCTION

Nowadays also non-specialist users demand easier access to and use of 3D digital content and related technologies, wishing to use collaborative design, 3D multimedia, or on-line training and documentation. Because of the recent improvements in 3D object acquisition, visualization and modeling technologies, the number of 3D models available on the web is more and more growing, and there is an increasing demand for tools supporting the automatic search for 3D objects in digital archives, and in general for more effective and easier to use 3D shape processing tools. Shapes are then taking a central role in several key areas, but representing and handling a complex shape is not a trivial task also because of the complexity of the knowledge a shape may reveal. The development of user-friendly tools based on more powerful and flexible knowledge technologies seems to be the right way for making digital shapes machine-understandable. Among the several related research initiatives is worth to mention the AIM@SHAPE Network of Excellence [1,7] whose overall objective is the introduction of knowledge management techniques in shape modeling, with the aim of making explicit and sharable the knowledge embedded in multi-dimensional media, with focus on 3D content. On the one hand, this requires the development of automatic or semi-automatic tools able to get the semantics of 3D models; on the other hand it is necessary to build a common framework for reasoning, searching and interacting with the semantic content related to the knowledge domain.

In this paper, among the various complex semantic information a shape may provide, we focus on the functionality of objects, as it may be deduced from its shape, or from the shape of some of its parts. The importance of including into a product model the information concerning its functionality and the functionalities of its main components is generally recognized. In the literature, several product models have been proposed aimed at preserving the design intent in terms of product/component function and behavior throughout the whole product lifecycle [8,10]. In [8] it is summarized how the Core Product Model initially developed at NIST for a number of in-house research projects gives equal status to three main aspects of a product or artifact: its *function* (i.e. what the artifact is supposed to do), its *form* (i.e. the proposed design solution for the design problem specified by the function), and its *behavior* (i.e. how the artifact's form implements its function).

Gero and Kannengiesser in [10] present the so named *FBS framework*, in which the knowledge of the design agent is grounded in its experience and its interaction with the environment: they identify three different aspects of a design object: the *structure*, which describes the components of the object and their relationships; the *function*, which describes what the object is designed for; and the *behaviour*, which describes what the object does. Kitamura and Mizoguchi in [11] propose a framework for the systematization of functional knowledge and its application to engineering knowledge management through ontological engineering.

In the field of computer vision and image understanding several works address the problem of the automatic recognition of the object through the reasoning about the functionality of its parts, based on the idea that the 3D shape of an object implicitly provides indications about its function. Bogoni and Bajcsy in [6] propose a representation for object functionality and define a methodology for retrieving it relying on the observation of the interaction with the objects. In [16,17] an approach for integrating a 3D shape reasoning module for object recognition is proposed. The function-based shape reasoning module analyzes B-Rep models to determine if a shape satisfies the functional requirements of some categories of objects. In [14] shape and function from 2D and 3D images are recovered from functional parts, by combining a set of functional primitives and their relations with a set of abstract primitives and corresponding relations. Pechuck et al. in [13] suggest a process scheme for the function-based classification of 3D images based on a hierarchical description of object classes in terms of functional components.
The objective of the research presented in this paper is the definition of a knowledge-based (KB) design environment supporting on the one hand the retrieval of 3D models verifying the suitability of their shape to perform a specific functionality; on the other hand, in a virtual simulation environment the system supports the automatic annotation of 3D objects, candidates to perform specific functions. Both the scenarios apply to several application contexts ranging from industrial design, to animation for simulation or entertainment. The framework integrates several tools which are the result of different research activities carried out within the geometric modeling group of our institution [3].
The paper is organized as follows: Section 2 details the framework architecture, while Section 3 describes the Form, Functionality and Behavior ontology in which we have represented and formalized the related domain knowledge. Section 4 describes the user scenarios of reference, i.e. the retrieval of 3D shapes and the semantic annotation in a virtual simulation during the design process. Finally, Section 5 ends the paper with the final remarks and outlining future work.

2. AN ENVIRONMENT FOR FUNCTIONALITY DRIVEN ANNOTATION AND RETRIEVAL OF 3D SHAPES

The system described in this paper aims at supporting users in retrieving and automatically annotating 3D shapes in different application contexts. The general architecture, as illustrated in Figure 1, relies extensively on the domain knowledge formalized through the *Form, Functionality and Behavior (FFB) Ontology*, which provides a formal description of the various shape characteristics an object usually fulfills to perform a specific function.

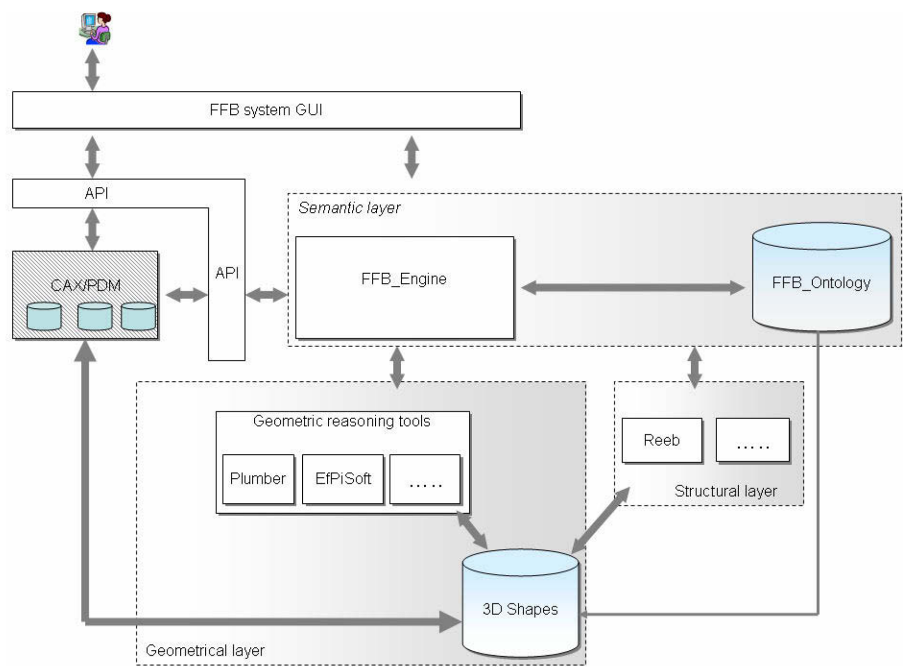

Fig. 1: The architecture of the Functionality-driven Annotation & Retrieval Environment.

Figure 1 shows the main components of the annotation and retrieval environment, which supports the complete metadata creation and includes tools for extracting, storing and managing the knowledge related to the *FFB* context. The represented knowledge concerns with *geometry* (the spatial extent of objects), *structure* (part-whole decomposition) and *semantics* (meaning in the specific context) of 3D models.
The framework architecture of Figure 1 is conceived to be used for the retrieval of 3D models during the design phase and their semantic annotation during the virtual simulation process, driven by functional characteristics of the shape. Both user scenarios are described in details in Section 4. The main elements of the system are the following:

***FFB system GUI*:** is the Graphical User Interface which enables the user to access the annotation & retrieval environment.

3D Shapes: is the 3D models repository; the geometric and structure reasoning tools work on 3D meshes, thus we consider that all 3D models in other representations are automatically converted into a 3D mesh.

***CAX/PDM*:** is the Computer Aided Design tool possibly integrated in a Product Data Management (PDM) system which handles the 3D models.

API: is the Application Programming Interface to enable the interaction of the CAX/PDM module and the FFB system GUI and the FBB Engine.

***FFB Ontology*:** is the knowledge base of the system. The *FFB ontology* gives a representation of the knowledge of the Form, Functionality and Behavior domain, to enable its effective sharing, reuse and analysis, and represents the information necessary to infer object functionality from the shape of models and their components. Differently from traditional DBMS, ontology, as other semantic web technologies, provides effective reasoning capabilities which greatly enhance the knowledge exploitation. The aim of the *FFB ontology*, in particular, is the full exploitation of the relationships and dependences among the aspects of form, functionality and behavior. The formalization provided by the *FFB Ontology* as described in this paper is intended to provide a proof of concept for our methodology, but the extension of the knowledge herein represented is straightforward. For instance, to add a new geometric tool it is sufficient to insert a new instance in the corresponding class with the correct property values; by contrast, a set of functionalities for a different family of objects (e.g. mechanical products) requires the insertion of new concepts, and a suitable setting of the relationships among them and with the legacy representation, in particular with the low level features which represent forms, positions, etc. The ontology will be illustrated in Section 3.

FFB Engine: is the software module, activated through the *FFB System GUI*, which interacts with the ontology to handle the functionality driven semantics and activates the appropriate tools to handle the 3D models.

Currently the framework includes two types of tools: *geometric reasoning tools,* which enable the automatic identification of 3D shapes with specified characteristics, and *structural reasoning tools*, which provide a conceptualization of a 3D shapes which may be used to find similar sub-parts. In particular, the tools supported so far are:

- *Plumber* [12], which classifies the features of a 3D object represented by a triangle mesh, segmenting a surface into connected components that are either “body” parts or tubular features, that is, handle-like and protrusion-like features, together with their concave counterparts, i.e. narrow tunnels and wells.

- *EfPisoft* [4], which performs a hierarchical face clustering of 3D triangle meshes, fitting primitives belonging to an arbitrary set. The method adopted generates a binary tree of clusters, each of which fitted by one of the primitives employed, currently planes, spheres and cylinders. Initially, each triangle represents a single cluster; at each iteration, all the pairs of adjacent clusters are considered, and the one that may be better approximated by one of the primitives forms a new single cluster.

- *Reeb Graph based geometric similarity tool* [5], which gives a method for partial shape-matching able to recognize similar sub-parts of objects represented as 3D polygonal meshes. The geometry and the structure of the shapes are coupled in a descriptor that provides a flexible coding, grounded on solid mathematical theories and that may be adapted to the user's needs and to the context of applications.

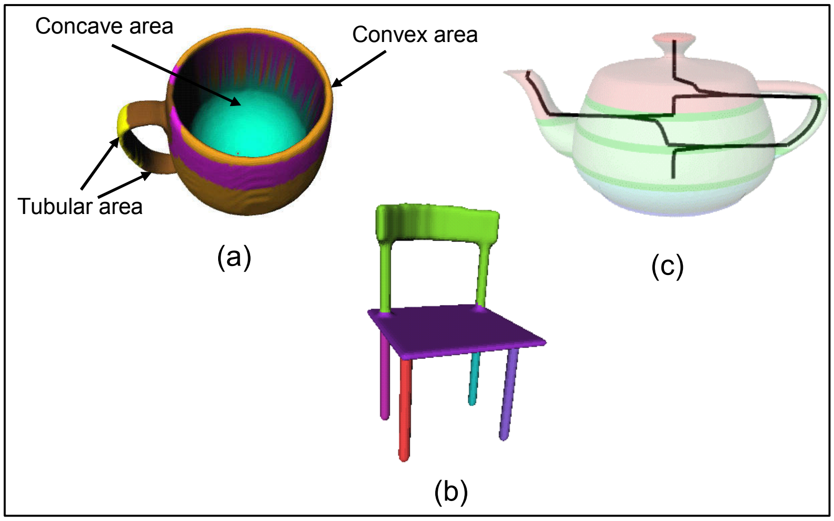

Fig. 2: Example of application of Plumber (a), EfPiSoft(b) and Reeb Graph based geometric similarity tool (c).

Through these tools, we are able to recognize analytical forms like sphere, plane, cylinder, and sub-parts of a model with such types of curvature; furthermore, we may distinguish also form characteristics like convexities and concavities, saddles, cylindrical and conical tubular forms, and splits. In Figure 2 examples of the application of the three tools are depicted. In Figure 2(a) light blue areas are concavities; orange areas are convexities and planes; yellow and brown highlight tubular areas. In Figure 2 (b), different colors highlight different segments, or components, of the 3D shape. In Figure 2(c), the Reeb Graph representing the structure of the teapot is depicted in black.

3. THE FFB ONTOLOGY

The *FFB ontology* is currently represented in OWL DL [18], the Description Logic sublanguage of the Web Ontology Language designed by the W3C Web Ontology Working Group to improve machine readability of web content. The core entities involved by the functionality driven annotation & retrieval of 3D shapes environment are depicted in Figure 3.

In the represented domain, a *product*, (elsewhere referred to as artifact) belongs to a unique product family and may be logically and physically subdivided into several *product components*. Each component may be made of one of more materials.

Given a product, the set of functionalities it fulfils are identified. Such functionalities are specified as "what the object is intended for" [8,10]. Each functionality may be carried out by one or more *functional area*, i.e. a physical area in the product, possibly related to a specific product component, with ergonomic characteristics specifically designed to accomplish the corresponding functionality. For instance, in a cup we may easily identify two functional areas: one for containing liquids, and the handle, which enables grasping. The set of functionalities related to a functional area in the product are a subset of the functionalities of the product. Moreover, a functionality may be accomplished by one of the *materials* employed to realize the product. For instance, the containment hot liquids may be accomplished by containers made of glass (but not made of soft plastic).

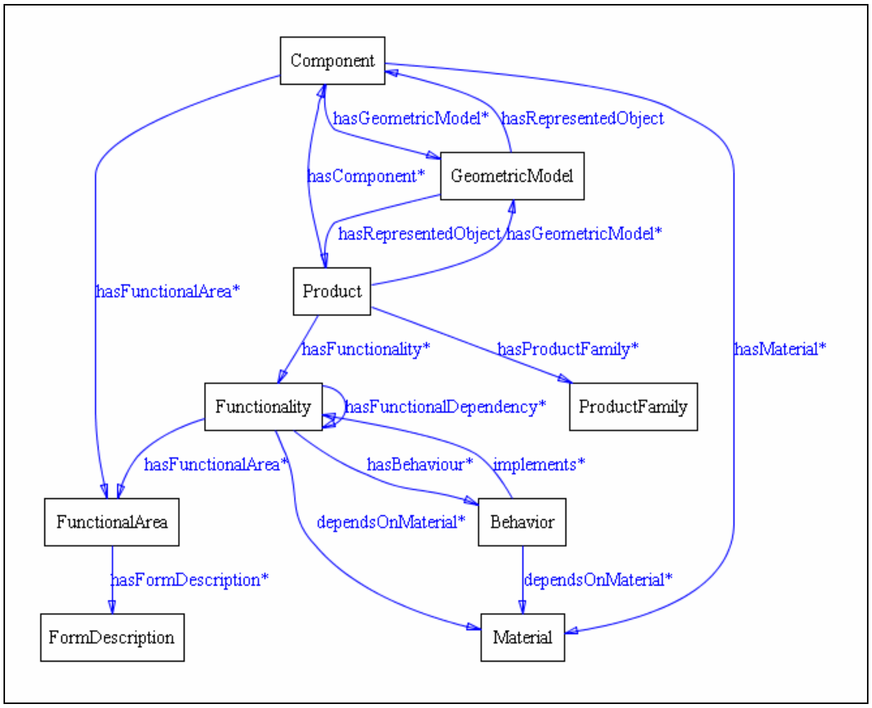

Fig. 3: The FFB ontology.

To fulfill a given functionality, components and consequently functional areas in a product may be required to be at a specified mutual position, and may have requirements on the relative dimension (cf. Figure 4). For instance, the seat and the back of a chair are *quasi* orthogonal at their borders, and are located at a distance in a given range. At the same time, the legs of the chair are required to be coplanar, in order to realize the functionality *stability*.

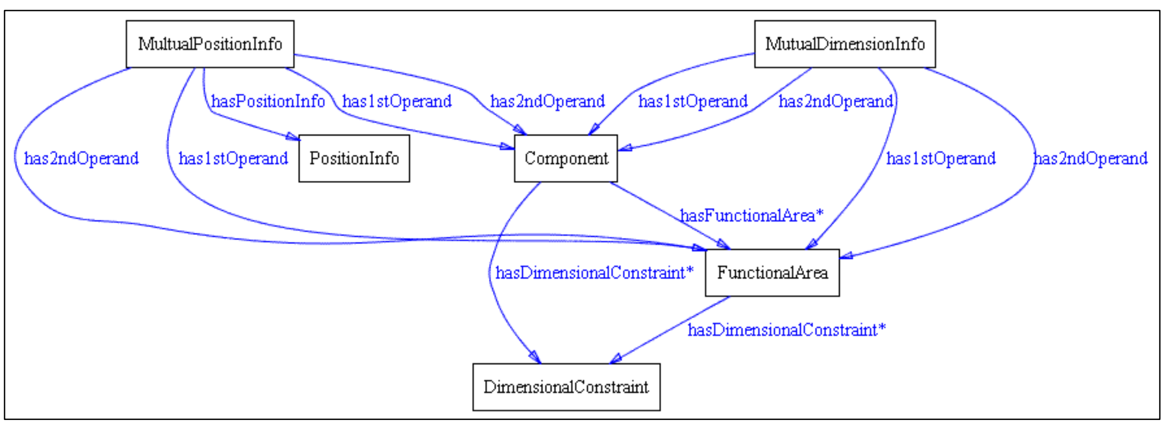

Fig. 4: Mutual constraints on position and dimension for components and functional areas.

We assume that not all the functionalities of a product or a component may be associated to a functional area, but there exist functionalities which are associated to the whole product. Given for instance a knife, it may be decomposed into two parts: a blade and a handle, each one representing a functional area of the object. The functionality of the blade is *to cut*, while the handle has *to be handled*. The combination of the functionalities results in the whole

functionality of the knife, which is *to cut with a particular (horizontal) movement*. Note that the functionalities of cutting and handling are both required to fulfil the functionality of cutting with a horizontal movement. In this case we specify that a *functional dependence* [9] exists among the three functionalities.
To enable the annotation and the retrieval of 3D shapes, a set of functionalities of interest are represented in the ontology. For this preliminary work, we focus on functionalities related to every day activities like seating, grasping, collecting, and cutting. The taxonomy of functionalities we consider so far is depicted in Figure 5.

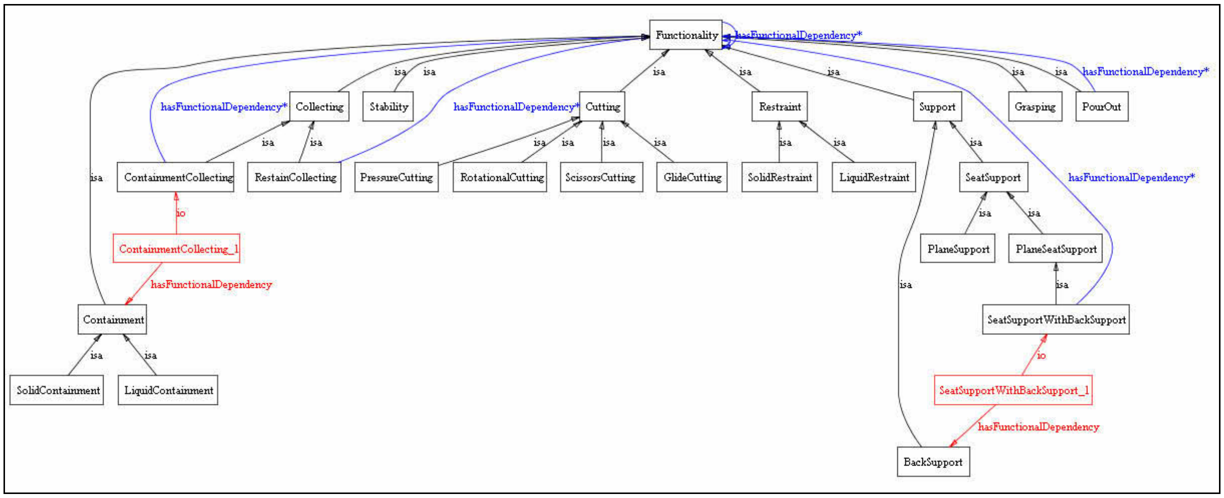

Fig. 5: Taxonomy of functionalities.

Each functionality is implemented by a set of possible *behaviors* (cf. Figure 8), which represent "what the object does" [8, 10], and specify the possible interactions with the object. For instance, in the virtual reality literature, different kinds of grasping are distinguished, to discriminate among the modalities a virtual human may obey the perform the grasping action [15].
The representation of the behaviors implementing a given functionality enables a full editing of an object in a virtual simulation scene. Once functionality is associated to an object, a set of behaviors that may be applied to the object is retrieved from the ontology. This set of behaviors may be further extended by the user.
Given a functional area, a set of potential *form descriptions* one may employ to accomplish the corresponding functionality is provided. A form description gives information about a form of reference, the position and the minimum dimension needed to accomplish the corresponding functionality, and information about requirement on stability, accessibility, symmetry, presence of holes and patterns. For instance, the functionality of support for seating has associated a functional area with a single form description, whose form of reference is a quasi-planar face, i.e. a plane with a tolerance in terms of convexity and concavity; minimum dimension requirements, i.e. the dimension sufficient to enable the seating of a human body; a position, in terms of height from the ground, given with respect to the dimension of a human body leg, and with a tolerance; requirements of accessibility, i.e. the form must be freely accessible, at list from one side, to enable the insertion of the human body.
The reference form is represented through its analytical equation, e.g. plane, cylinder, sphere, etc., and its shape characteristics, e.g. convexity, sharpness. Dimensions are specified as absolute values, or as proportion (e.g. with respect to the dimensions of the human body). Form, dimension and position are specified in the ontology with a tolerance, enabling to match a form description within a given threshold. Tolerances for dimension and position are given as percentage on the values specified. Tolerances of form are used by geometric recognition tools in order to relax the research of a given form with respect to its analytical equation. For instance, convex forms may be compared with planes whenever the convexity degree is low.
The forms represented in the ontology may be retrieved or recognized by using one or more geometric tools (cf. Figure 6). The specification of tools in the ontology has been inspired by the AIM@SHAPE [1] tool ontology, which has been designed for supporting an e-scientist in the use of shape processing tools. Currently, the geometric tools represented in the *FFB ontology* are Plumber [12], EfPisoft [4], and Reeb Graph based geometric similarity tool [5], but this set may be extended to other utilities which perform form recognition. Each tool recognizes one or more type of form,

among those represented in the ontology. The specific requirements of each tool for recognizing a given form are also represented. For instance, Plumber [12] requires the specification of the initial radius of the sphere employed to detect the intersection with the surface mesh.

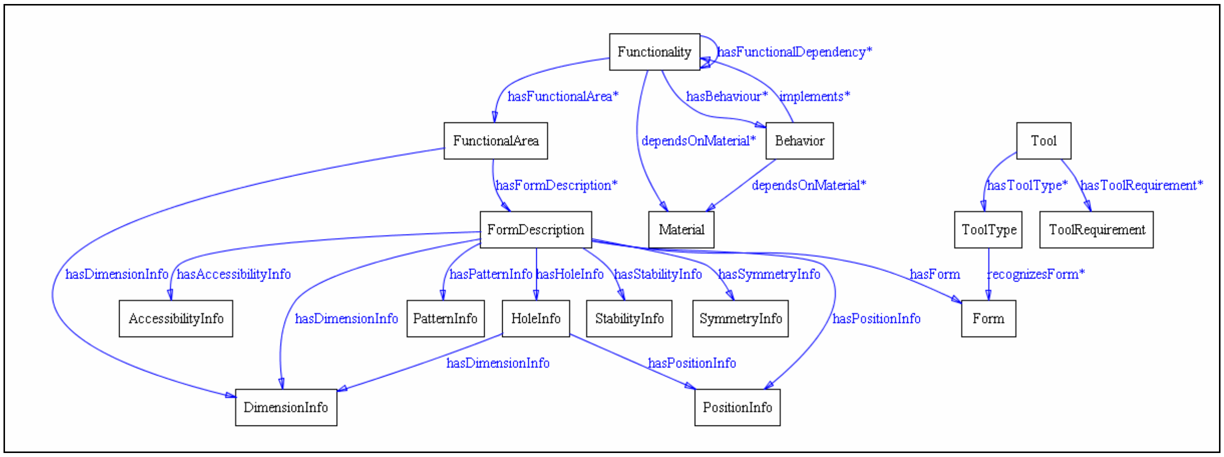

Fig. 6: Relationships among functionality, form, behavior and geometric tools in the FFB ontology.

4. ANNOTATION AND RETRIEVAL OF 3D SHAPES USER SCENARIOS

In this section, we describe the foreseen usage of the ontology described in Section 3 to support a user in the retrieval or the semantic annotation the 3D models of objects with shape characteristics accomplishing a specific function. The models may be stored in a repository, or embedded in a virtual simulation scene. In this second case, the system may analyze every model in the scene and annotate them according to their functional and behavioral characteristics.

The annotation process of the functional areas of a model can be speeded up by the knowledge already available, e.g. the type of the object under evaluation. The matching step is performed according to the knowledge embedded in the *FFB ontology*: the *FFB engine* extracts from the ontology the information about of the functionalities and their related shape characteristics; then, it gets from the ontology which geometric reasoning tools are appropriate to identify such forms; finally, a set of candidate models and functionalities are identified. The results of the elaboration are proposed to the user, which may confirm or reject them. If the user accepts the results, the models are annotated with respect to the identified characteristics of functionality and behavior. If the user rejects the objects or the functionalities proposed by the *FFB engine*, the retrieval or the analysis will continue until all the models and the functionalities in the knowledge base have been analyzed.

It may be possible to take advantage of *the Reeb Graph based geometric similarity tool* [5] for retrieving the models in the *3D shapes* repository which are geometrically similar to a prototype model with the required functional or form characteristics. The results are proposed to the user, which may accept or reject them, as in the previous case.

The method we propose includes also the application of a learning step, enabling the extension of the knowledge base: the information initially stored in the ontology comes from the conceptual design specification, but it is incremented whenever new models are inserted and annotated. Moreover, whenever the *FFB engine* proposes a model with a set of candidate functionalities to the user, he/she may complete or update the annotation identifying new functionalities and the corresponding functional areas, and the knowledge base is extended with the appropriate instances.

In the retrieval scenario illustrated in Figure 7, the user asks for models which are candidate to achieve a given functionality (e.g. liquid containment). The *FFB Engine* first searches for 3D models which may fulfil the functionality among those already semantically annotated. The models which satisfy the request are returned to the user, which may decide to accept the results, or to integrate them with other models in the repository. In this second case, the search module in the *FFB engine* queries the ontology for the shape characteristics associated to the requested functionality. Then, the *FFB engine* queries the ontology for the geometric reasoning tools suitable for identifying the retrieved shape characteristics. The tools are applied on the set of shape models not annotated, and a set of candidate models for the given functionality is retrieved and proposed to the user: among them, those accepted by the user are semantically annotated by the *FFB engine.*

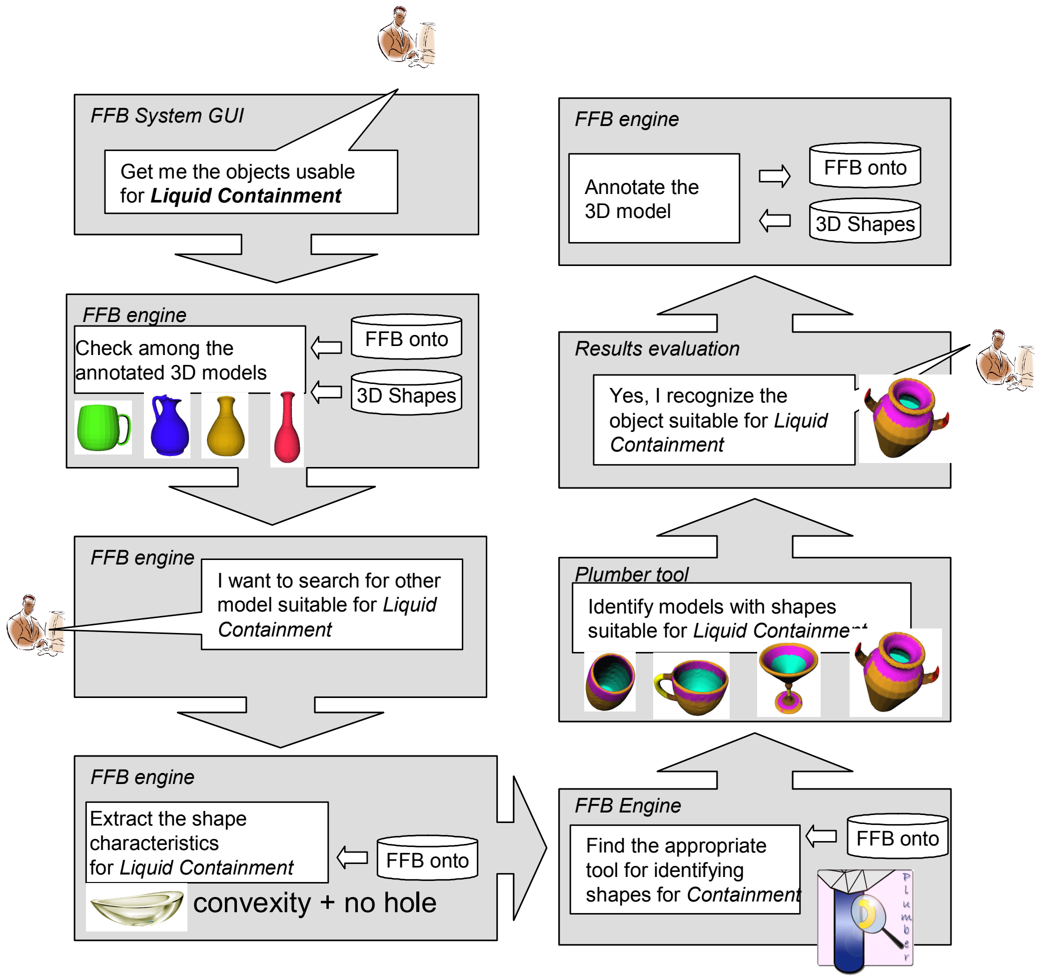

Fig. 7: Workflow for the retrieval of model candidates to perform a specific functionality.

As another example of application, we consider the simulation scenario for ergonomic evaluation of the car interior illustrated in Figure 8, where the interaction with the diverse equipments of the car interior has to be analyzed. The automatic identification of the components' areas suited to attain specific functionalities could ease the creation of the animated simulation. As schematized in Figure 8, each model is first elaborated by the segmentation tools, and the possible functional areas are identified; for each functional area, a form description is provided by applying the geometric reasoning tools available in the framework. Then, the potential functionality(ies) corresponding to the functional area is(are) identified. The matching is obtained comparing the form descriptions already stored in the ontology with the form description of each functional area identified in the scene. The form description may be associated to models already annotated or may simply characterize the functionalities specified in the ontology. For each form matching, the corresponding functionality (according to the knowledge base) is returned and proposed to the user as a candidate functionality for the object analyzed.

In this scenario, the annotation process might benefit of possibly available additional knowledge on the type of components involved that can be provided by the user or retrieved from the PDM data organization. In this case, the user may ask for the identification and annotation of the meaningful functional area depending on the considered component, e.g. the graspable areas in the car door interior or the steering wheel, which correspond to rather tubular parts. In this case, after the selection of a component in the scene, the applied workflow is similar to the one depicted in Figure 7, where the user indicates the functionality to consider and then the system directly applies the corresponding geometric reasoning tool as specified in the ontology.

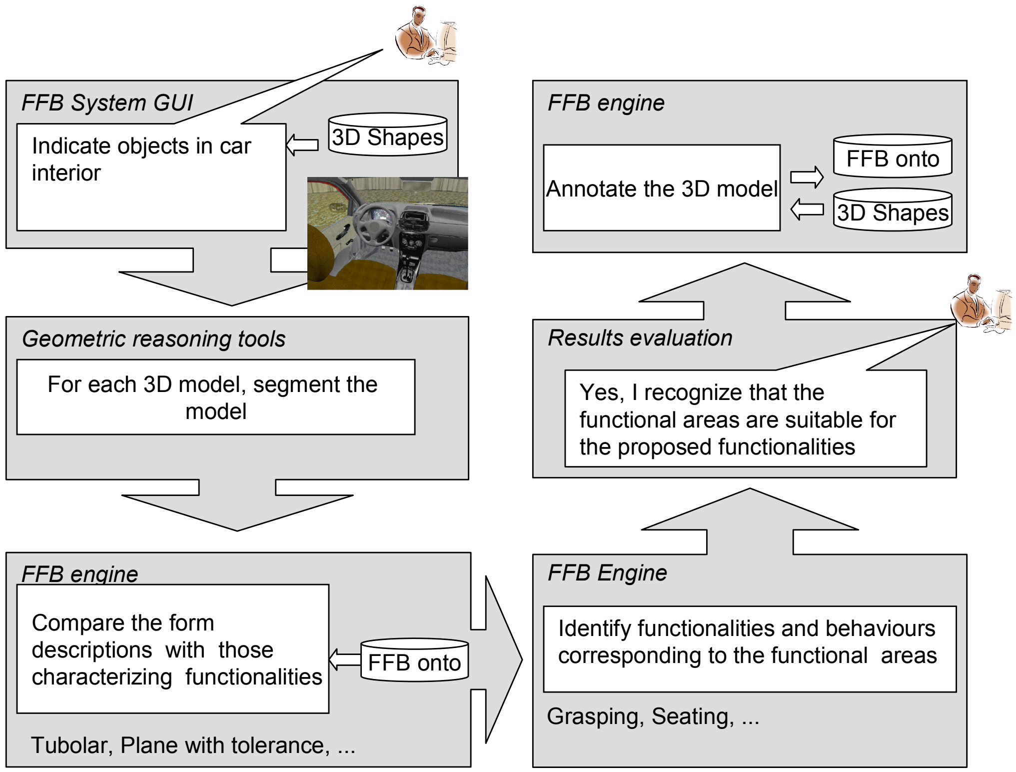

Fig. 8: Workflow for the annotation of 3D model in the virtual simulation process.

5. CONCLUSIONS

The paper has introduced a knowledge based framework for the annotation and retrieval of 3D shapes. The *FFB ontology* formalizes the implicit knowledge embedded in a 3D shape, which we argue may provide useful indications about the functionality of the represented object. The geometric recognition task involved by the architectural framework is performed through the application of a set of tools [3,4,5,12] developed within the geometric modeling group of our institution. The partial results we obtained indicates the validity of the approach.

Nevertheless, to assess the effectiveness of the FFB framework, further steps must be undertaken. The next pace will be the completion of the system through the integration of the geometric reasoning tools, and the development of the software module to perform the automatic annotation of 3D models. Moreover, we will integrate the geometric reasoning tools with new software for recognizing further forms and relationships. For instance, currently the framework misses a proper tool for the identification of sharp areas, which may accomplish the *cut* function.

Furthermore, we are planning the inclusion in the framework of an analysis tool for semantic similarity [2]. Such a tool has been developed within the AIM@SHAPE project to evaluate the semantic similarity among instances in an ontology, according to different contexts the user specifies. The tool, coupled with the Reeb Graph based geometric similarity tool, will enable the integration of the results obtained through the application of the FFB framework with models similar with respect to form, functionality and behavioral aspects.

6. AKNOWLEDGEMENTS

This work is partially supported by the AIM@SHAPE Network of Excellence (AIM@SHAPE) funded by the European Commission Contract IST 506766. In particular, the authors would like to thank Marco Attene, Silvia Biasotti, Simone Marini, Michela Mortara, and Giuseppe Patanè who have developed the involved geometric reasoning tools for their helpful support.

7. REFERENCES

[1] AIM@SHAPE project official web site, http://www.aimatshape.net.

[2] Albertoni, R; De Martino, M.: Semantic Similarity of Ontology Instances Tailored on the Application Context, ODBASE- OTM Conferences, LNCS 4275, 2006, 1020-1038.

[3] Attene, M.; Biasotti, S.; Mortara, M.; Patanè, G.; Spagnuolo, M.; Falcidieno, B.: Computational methods for understanding 3D shapes, Computers & Graphics, 30, 2006, 323–333.

[4] Attene, M.; Spagnuolo, M.; Falcidieno, B.: Hierarchical Mesh Segmentation based on Fitting Primitives, The Visual Computer, 22(3), 2006, 181-193.

[5] Biasotti, S.; Marini, S.; Spagnuolo, M.; Falcidieno, B.: Sub-part correspondence by structural descriptors of 3D shapes, Computer-Aided Design, 38, 2006, 1002–1019.

[6] Bogoni, L.; Bajcsy, R.: Functionality investigation using a discrete event system approach, Robotics and Autonomous Systems, 13(3), 1994, 173-196.

[7] Falcidieno, B.; Spagnuolo, M.; Alliez, P.; Quak, E.; Vavalis, E.; Houstis, C.: Towards the semantics of digital shapes: the AIM@SHAPE approach, European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies November 25-26, 2004, Royal Statistical Society, London.

[8] Fenves, S; Foufou, S.; Bock, C.; Sriram, R.-D.: CPM: A Core Model for Product Data, Journal of Computing and Information Science in Engineering, 5, 2005, 238-246.

[9] Gangemi, A.; Guarino, N; Masolo, C; Oltramari, A.: Understanding top-level ontological distinctions, Proc. of 17th International Joint Conference on Artificial Intelligence, Workshop on Ontologies and Information Sharing, 2001, 26-33.

[10] Gero, J.-S.; Kannengiesser, U.: The situated function-behavior-structure framework, in J.S. Gero (ed.) Artificial Intelligence in Design'02, Kluver Academic Publisher, The Netherlands, 2002, 89-104.

[11] Kitamura, Y.; Mizoguchi, R.: Ontology-based systematization of functional knowledge, Journal of Engineering Design, 15 (4), 2004, 327-351.

[12] Mortara, M.; Patanè, G.; Spagnolo, M.;Falcidieno, B,; Rossignac, J.: Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies, ACM Symposium on Solid Modeling and Applications,G. Elber, N. Patrikalakis, P. Brunet (Editors),2004, 339-344.

[13] Pechuk, M.; Soldea, O.; Rivlin, E.: Function-based classification from 3D Data via Generic and Symbolic Models, in AAAI, 2005, 950-955.

[14] Rivlin, E.; Dickinson, S.-J.; Rosenfeld, A.: Recognition by functional parts, Computer Vision and Image Understanding, 62, 1995, 164-176.

[15] Smith, L.-K.; Weiss, E.-L.; Lehmkuhl, L.-D.: Brunnstrom's clinical kinesiology, 5th Edition. F. A. Davis, Philadelphia, 1996.

[16] Stark, L.; Bowyer, K.-W.; Hoover, A.; Goldgof D.-B.: Recognizing object function through reasoning about partial shape descriptions and dynamic physical properties, Proceedings of IEEE, 84 (11), 1996, 1640-1656.

[17] Sutton, M.-A.; Stark, L.; Bowyer, K. -W.: Representing and reasoning about object functionality: towards an integrated approach, Intelligent Robots: Sensing, Modeling and Planning, Dagstuhl Workshop, September 1-6, 1996, 180-197.

[18] The Web Ontology Language (OWL), http://www.w3.org/2004/OWL/.