HAL
archives-ouvertes.fr

Deriving Functional Properties of Components from the Analysis of Digital Mock-ups

Ahmad Shahwan, Gilles Foucault, Jean-Claude Léon, Lionel Fine

▶ To cite this version:

HAL Id: hal-00922958
https://hal.inria.fr/hal-00922958v2

Submitted on 10 Oct 2014

DERIVING FUNCTIONAL PROPERTIES OF COMPONENTS FROM THE ANALYSIS OF DIGITAL MOCK-UPS

Ahmad Shahwan
Grenoble University
France
ahmad.shahwan@grenoble-inp.fr

Gilles Foucault
Grenoble University
France
gilles.foucault@g-scop.inpg.fr

Jean-Claude Léon
Grenoble University / INRIA Rhône-Alpes
France
jean-claude.leon@grenoble-inp.fr

Lionel Fine
France
EADS Innovation Works
lionel.fine@eads.net

ABSTRACT

Digital Mock-ups (DMUs) are widespread and stand as reference model for product description. However, DMUs produced by industrial CAD systems essentially contain geometric models and their exploitation often requires user's input data to derive finite element models (FEMs). Here, analysis and reasoning approaches are developed to automatically enrich DMUs with functional and kinematic properties. Indeed, geometric interfaces between components form a key starting point to analyse their behaviours under reference states. This is a first stage in a reasoning process to progressively identify mechanical, kinematic as well as functional properties of the components. Inferred semantics adds up to the pure geometric representation provided by a DMU and produce also geometrically structured components and assemblies. Functional information connected to a structured geometric model of a component significantly improves the preparation of FEMs and increases its robustness because idealizations can take place using components' functions and components' structure helps defining sub-domains of FEMs.

KEYWORDS

Product simulation, design, DMUs, geometric models, assembly, functional designation, mechanics, kinematics, reasoning and knowledge representation

1. INTRODUCTION

As geometric representations of a product, digital mock-ups (DMUs) provide engineers with assembly and/or sub-assembly models that can be used as input for finite element (FE) analyses during a product development process (PDP). This is the specific context addressed here, even though assembly model processing takes place also at other stages of a PDP.

Virtual and augmented reality techniques, varying from simple visualization to fully-immersive environments, have been used at various stages of a PDP; such as design and modeling, assembly/disassembly (A/D) simulations and planning, to name only few [1], [2], [3]. There also, assembly models are central and their processing addresses issues similar to those addressed in this document.

Anyhow, a finite element model (FEM) derived from an assembly model needs a very large amount of engineers' interactions to be generated from its corresponding DMU. This causes a delay in the availability of such a model up to the point where the simulation becomes no longer useful as its output arrives too late in a PDP. For this reason, reducing the simulation preparation time at the level of assembly models becomes a key issue. Figure 1illustrates this configuration with a complex sub-

structure of an aircraft where FEM preparation is too tedious.
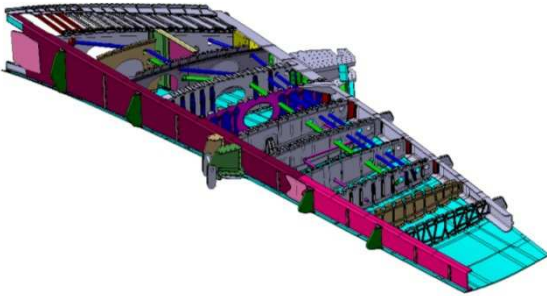
Figure 1 A complex assembly of an aircraft sub-structure (courtesy EADS IW).

In fact, assembly processing for FEM generation requires many engineers' interactions to produce interfaces between components that meet the objectives of FE simulation [4]. Interfaces are numerous, additionally; simplification processes must take place on some components. Thus, this task is often interactive, despite efforts to automate it [7], [8]. Its automation requires the identification of similarities among interfaces and components to set up algorithms. Often, engineers refer qualitatively to categories of components and connections between components to structure their FEM preparation, e.g. screws, bearings, bolted assemblies, etc. Indeed, this categorization can be formalized as a need to identify a component or a set of components from its/their function. DMUs conveyed through STEP files or even in native CAD modelers' format are hardly providing the desired information. Indeed, to process the geometric interfaces between components, there must exist some connection between the component designation and its geometric model. This suggests that the functional model of a component refers to a structured geometric model connected to its designation that stands for symbolic information characterizing its function. Hence, it is the purpose of the proposed approach to speed up the simplification processes of assemblies, by providing necessary semantic annotations and structured geometric models to their components through a bottom-up analysis of assembly models.

The rest of the document is presented as follows; we review prior contributions in Section 2. Next, we highlight our contribution in Section 3. Section 4 addresses new concepts related to interfaces and reference states that are central to our approach. In Section 5 we develop the process flow of our approach. Results are briefly shown and explained in Section 6. Section 7 concludes and discusses future works.

2. RELATED WORKS

The problem of bridging the gap between pure geometric representations and technical features of components has been frequently tackled in the literature. Efforts as early as [5] have been paid in the field of features recognition (FR) in solid models. [6] defines *features* (also referred to as form features or machining features) to be the representations of shape aspects of a physical product that can be mapped to generic shapes in a given context and are functionally significant.

In [5], a graph representation of the geometric model is generated before graph matching techniques are applied to extract form features, also represented as graphs.

Authors in [7] address the problem of functional features extraction out of digital models, and classify existing solutions into human assisted approaches, feature based modeling, and automatic feature recognition and extraction. Their proposed method falls in the last category and suggests a three stage solution that builds a hierarchical structure of part's shape in accordance to the level of details.

In [10], the author advocates an expert system approach to recognize application-specific features given the product's solid model as B-Rep.

A survey of recent approaches to feature recognition shows a wide range of techniques that participate to the Computer Aided Process Planning (CAPP) automation [7]. In [11] the feature recognition is integrated into the process of simplification as a preliminary step to prepare a tessellated model for FE analysis.

A technique to detect and simplify blending features to enhance the process of functional features detection is presented in [6] where topology is preserved. Another approach, capable of handling more interacting shape features through an iterative method is presented in [12], where form feature recognition techniques are used to detect features face-sets, and then a feature is removed before passing to the next iteration, where previously interfering features can be detected.

Ahmad Shahwan, Gilles Foucault, Jean-Claude Léon, Lionel Fine
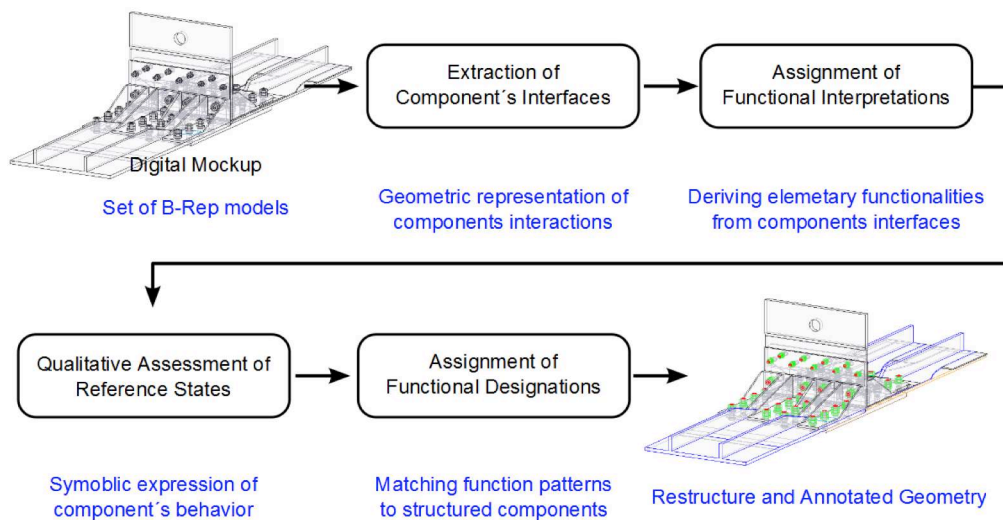
Figure 2 Overall scheme of processes

In [13] authors again tackle the problem of features interaction through a hybrid approach for feature recognition that is both graph and rule based.

The abovementioned solutions fall in the category of automatic feature recognition. Although such techniques aim at the extraction of functional information given the pure geometric model, they are still limited to a very small set of simple geometric configurations like holes, pockets, slots, rounds and fillets. Most of prior work fits into a bottom-up approach where features are extracted from low level geometric entities and a detached volume model is processed as an isolated entity. Whenever assemblies are referred they are generally regarded as a collection of components processed with loose or no connections at all between them.

In [14] authors do not only emphasize the connection between product's geometry and functional attributes, they also demonstrate with concrete examples the correlation between pairs of interfacing geometrical entities and the expected purpose of a product. It shows through industrial case studies how functionality is tightly coupled with the properties of geometrical interactions between surfaces belonging to different components that provide the desired—or even undesired—behavior.

The strong ties between geometry and semantic properties are again brought forward by [15] where authors analyze causal kinematic chains of a product based on its tessellated 3D model. A graph representation of product components and their contact characteristic is used to draw conclusions.

Alongside the reasoning process, reduced user input is solicited interactively.

Efforts are also paid to adapt design models to FEA purposes. Specific operations have been provided to handle contact zones generated by either B-Rep CAD models [16], or tessellated geometries [17], [18]. Those operations however didn't show any connection to the functional attributes of a geometric interaction, leaving an open question of how adequate those adaptations are to the simulation process.

Examining the literature shows that though it is possible to recognize some basic manufacturing features by merely considering local geometric properties of components, the detection of more complicated functional properties requires that the geometric model be regarded from a wider angle, that also covers the interaction between different components.

3. CONTRIBUTION

Approaches to DMUs' simplification still fail short to efficiently transform their geometrical models in a way that meets FEA requirements. This is because components are lacking functional descriptors and the latter are not connected to the geometric model of the components. Our work comes to fill this gap, enriching the plain geometrical representation with functional annotations and connecting them to structured geometric models as a preliminary step of the FEM preparation process. To this end, we develop algorithms that extract some functional and kinematic features from components as they are in
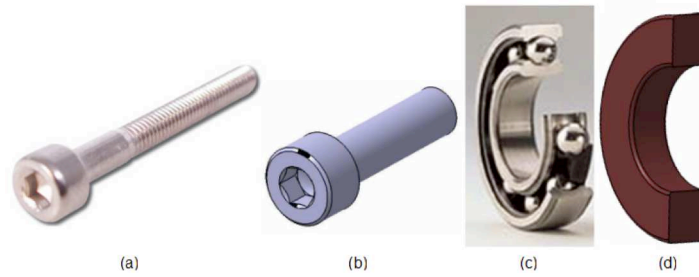
Figure 3 Difference between *real shape* and *digital shape*.
(a) Real cap-screw. (b) Digital model of cap-screw with threaded part smoothed out.
(c) Cross section in a real ball bearing. (d) Cross section in a simplified ball bearing.

their assembly configuration, to enable the inference of their *functional designations* (FD) (Section 4.3).

Being committed to simulation objectives; this doesn't only imply functional denomination of groups of geometric objects (i.e. components), but also requires a function annotation at different level of details, starting by elementary functions at the level of geometric interactions. This necessitates a restructuration of the original geometry.

Since functionality is attributed to mutual interactions between components and sub-assemblies in a product, the geometrical study of components as standalone objects is not sufficient to infer any functional knowledge. Components should be studied in their working environment instead. In this work we focus on geometrical interactions between components, rather than their isolated shapes.

A physical dimension is attached to those interactions in a qualitative sense to enable a functional behavioral study. This includes information such as internal forces, moments, and relative velocities. Thus, qualitative interactions between components in an assembly are a basis of a reasoning process that contribute to the characterization of components' functionalities as it is frequently operated with design methodologies [19], [20].

4. ASSEMBLY MODEL ANALYSIS

In this section we give an overview of our approach and briefly describe its central concepts.

4.1. FROM GEOMETRY TO FUNCTIONAL ANNOTATIONS

The input to our algorithm is a pure geometric representation of a product (see Figure 2). We first extract interesting features out of such input; i.e. the geometric interfaces between adjacent components in the assembly.

Next, we enrich the knowledge about the assembly. To this end, functional information is incorporated into our reasoning process using the domain knowledge of mechanical engineering.

We start with what geometric interfaces suggest; that is a set of functional interpretations. Then, those interpretations are filtered out to associate a unique function to each interface. This is achieved with the introduction of complementary rules to our knowledge base, such as mechanical equilibrium states that hold truth all across the assembly, considering that its components are rigid bodies. More functional information is inferred qualitatively as more states are considered.

After elementary functional information about components in an assembly is gathered, an ontology describing a hierarchy of FDs is invoked. This is used to classify components of the assembly. The ontology connects to a rule-based matching process to deduce FDs of components, and to provide a knowledge repository enabling querying functions. It is now the purpose of the FEM preparation process to take advantage of this information and of the newly structured components models.

4.2. MECHANICAL COMPONENTS AND THEIR REPRESENTATIONS

Mechanical components are meant to deliver well-defined functionalities through their geometric interfaces with their neighboring components. We can define the 'Real shape' of a component C as the real physical shape of C. A DMU is a set of components with 3D shapes. Here, it is hypothesized that these components are solids, i.e. volumes. They are represented using B-Rep models. Those shapes are located in 3D space, either globally, i.e.

Ahmad Shahwan, Gilles Foucault, Jean-Claude Léon, Lionel Fine

according to a global coordinate system, or relatively to other components, using geometric constraints. Geometric interfaces between components are not represented explicitly when using global locations, while they are ambiguously defined when using geometric constraints because the surfaces they involve may not belong to geometric interfaces.

DMUs represent assembly models whose components are hierarchically structured. This tree structure can reflect subsets called sub-assemblies; however, this structure may be absent. Consequently, this hierarchy is not a robust source of information and, hence, it cannot contribute to functional data. Figure 3 depicts an example of a DMU representing an assembly of a centrifugal pump.

Though solids in a DMU are supposed to reflect the shapes of their corresponding components, observations and design practices show the contrary. In fact, designers try to compromise the real shape of C with the speed of the design process, and the adequacy of solids for other stages in a PDP. This leads to the concept of digital shape. A 'Digital shape' of C is a volume model representing C in a DMU. Here, it is simply designated as a *shape*. Compared to the real shape of C, its shape derives from a simplification process also called idealization. Though real and digital shapes may be almost identical in one case, they may differ significantly in another (see Figure 3).
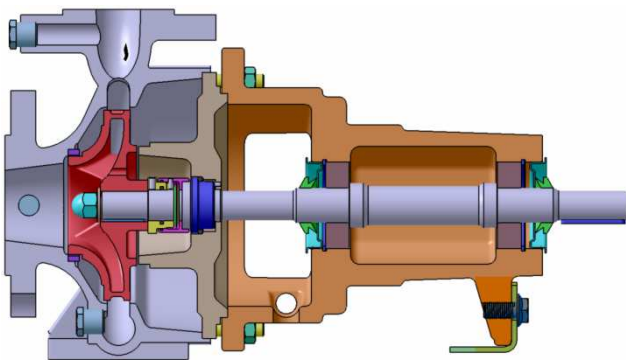
Figure 4 model of centrifugal pump.

Conventionally, the threaded part of a real component is simplified or idealized into a cylindrical area; see the difference between Figure 3 (a) and Figure 3 (b). Similarly, teeth of spline links are often removed in their digital shapes leading to interfering cylindrical areas. As a result, threaded areas and spline links produce the same geometric interfaces. Consequently, starting from this geometric interfaces leads to two different functional

links. This exemplifies the functional interpretations mentioned in the previous section.

Libraries, such as TraceParts [21], collect 3D models of components as generated by components providers. This means that they are not standardized and may differ from each other even if the components are similar. For example, a threaded hole of a bolt with a given nominal diameter may be found in different 3D models having different thread diameters [22].

Using component shapes as a starting point of a DMU analysis can influence the inference of FDs (Section 4.3) of components. Consequently, there is a strong dependency between shape, interface, and functionality of components as pointed out in design methodologies [19], [20].

4.3. FUNCTIONAL DESIGNATION

The functional designation of a component is *an unambiguous denomination that functionally distinguishes one class of components from another*.

The FD decidedly determines the functional group of its component. One component can only have one FD, though it might have more than one function, indicated by the designation itself. As an example, a '*nut*' is a designation relating to the threaded link function, a '*locking nut*' contains the addition of its locking function.

In this sense FDs constitute equivalence classes that distinctly sort out all components in a DMU.

FDs do not always match the assigned denomination of components in a bill of materials. Since component names are user-defined, it may not be uniform in a DMU and cannot be exploited in the current analysis process, because it is not reliable [23]. Even though standards exist [24], [25] that focus on threads, bearings, etc., their content contributes to describe standard functions but they are lacking FDs that are mandatory to generate an unambiguous set of component classes.

4.4. FUNCTIONAL TAXONOMIES

Different FDs may share a common generalized function. For example, screws are meant to fasten, and gears are expected to transmit moment, etc. As a result, the functional nomination can fit in a hierarchical structure whose leaves are indeed FDs. We call this hierarchy *the taxonomy of FDs*. Rooted by a label representing all possible mechanical
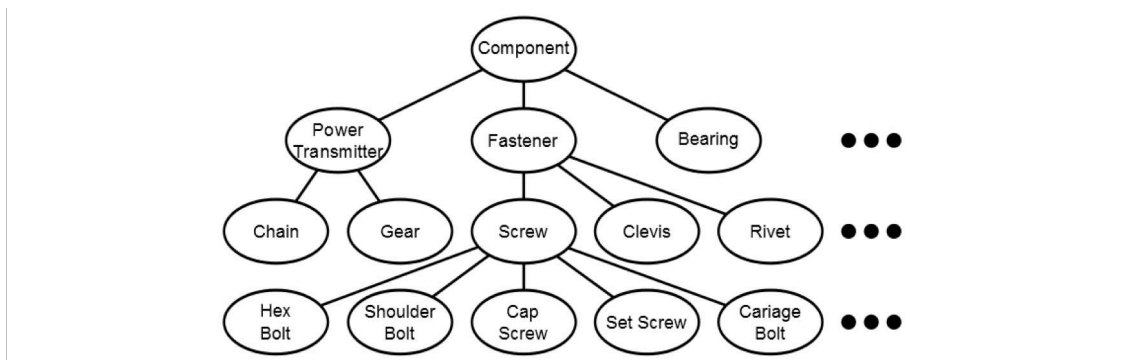
Figure 5 A subset of functional designation taxonomy.

components, the taxonomy increases details about functional properties as one goes deeper in the hierarchy, until a leaf is reached which indicates an unambiguous definition, i.e. a FD that characterizes a component and its set of functions.

Figure 5 shows a small portion of the FD taxonomy, showing the path to the FD of "*cap screw*", amongst others.

Representing a FD, each leaf of the taxonomy contains a geometric description of the component interfaces, their relative positioning, and their functional properties.

Indeed, the FD taxonomy is a high level one. Other, more generic, taxonomies of a lower level are part of the current approach. They refer to *functional interface taxonomy* and *shape interface taxonomy*. The former addresses the interfaces between components and suggests possible functions that can be associated. It also establishes a connection between the shape of an interface, its behavior, and its function using dualities between geometry and interaction forces or relative velocities [19], [20]. The latter is purely geometric and connects with the first one, it classifies the geometric interfaces.

4.5. CONVENTIONAL INTERFACES

Interactions between adjacent components reveal essential information that guides the identification of functional properties. We refer to such interactions as *conventional interfaces* (CI).

A CI is a broad concept that captures all aspects of the relationship between two neighboring components in an assembly; it has geometric, physical, and functional properties. The first step in our analysis is to extract geometric properties as the geometric model is our starting point. Once geometric interactions are defined, the goal shifts to

providing some physical parameter to eventually enable the mapping of each CI into a unique functional interpretation. For example, our analysis may lead to the conclusion that a CI geometrically described by a cylindrical interference (see Figure 7) transmits forces and moments in all directions, and allows neither translations nor rotations, as show corresponding physical parameters. This allows us to infer that this CI is a threaded link. We call such interpretation a *functional interface* (FI) (see Section 4.6).

CIs form a central concept in our approach, around which the work can be divided into three distinguishable phases:

1. The geometric analysis to obtain geometric properties of CIs;

2. Physical interpretations of those geometric properties into FIs;

3. The extraction of FDs of components based on the functional properties of their CIs.

These steps will unfold in the following text.

4.6. FUNCTIONAL INTERFACE

Functional interpretations of a CI are materialized in function interfaces. As the name reveals, a function interface describes a zone of interaction between two components that is supposed to deliver certain functionality. This is characterized by mechanical and kinematic properties that allow the expected behavior. Examples are planar support, cylindrical support, pivot link, threaded link, etc. This is much similar to concepts used in design methodologies [19], [20], though our current approach is a bottom-up one.

Ahmad Shahwan, Gilles Foucault, Jean-Claude Léon, Lionel Fine

4.7. GEOMETRIC INTERACTIONS BETWEEN COMPONENTS

The geometric interaction between two adjacent components determines the geometric properties of their CIs.

Geometric interactions are described by their interaction types and their interaction zones and are stored in the *shape interface taxonomy*.

The interaction type may be a contact, interference or clearance.

Contact

A contact between two components C1 and C2 defines one or more shared surfaces or shared curves, without any shared volume (see Figure 6).

The interaction zone of a contact is defined by this set of shared surfaces and curves, leading to potential non-manifold configurations.

A contact representation is usually realistic in the sense that a contact in the geometric model reflects the same configuration in the real product, where C1 and C2 are in touch.

Contacts provide very valuable information to our reasoning, as they usually help defining locations where resulting interaction forces can be transmitted. At the same time they work as motion barriers reducing components' degrees of freedom.

When a clearance between C1 and C2 becomes small enough, it may conventionally reduce to a geometric contact as well. Consequently, a cylindrical contact can be functionally interpreted either as a loose fit or a tight fit, as an example. Qualitative reasoning and reference states aim at selecting the correct interpretation.

In some conventions, however, a contact may represent an idealization of more complex settings, like threaded links or gears and rack-pinion links.

Interference

An interference between C1 and C2 defines a shared volume between them (see Figure 6). Obviously, an interference is a non-realistic representation in the sense that the two digital shapes of C1 and C2 interfering in an assembly don't represent overlapping volumes of C1 and C2 in a product, as this leads to non-physical configurations. Nevertheless, interferences are often the result of local shape simplifications often combined with rather complex settings of components' locations. For instance, threaded links most frequently results in cylindrical interference volumes.

Due to their idealized nature, interferences are harder to interpret than contacts; however, they also provide valuable information to the reasoning process.

Clearance

A clearance occurs when a distance between two surfaces of components C1 and C2 conveys a functional meaning (see Figure 6). Clearances can be organised into two categories. A first one is characterized by non-null distances that exist in the assembly model. The second one appears when clearances become small enough, i.e. an adjusted fitting of C1 and C2. There, the clearance is reduced to a geometric contact, as mentioned previously.

The distance value acting as threshold between the two categories is a matter of user's convention.

Clearances of category one depict an interaction zone between sets of surfaces of C1 and C2, this concept relates to functions. Therefore, it is not possible to address such clearances in a first place starting from the mere geometry of components. It is mandatory to have a FD of C1 and/or C2 to be able to determine the surfaces of C1 and/or C2 locating the clearance, e.g. on Figure 6, the clearance is located once the component is identified as a cap screw and then, its shaft can be located. Indeed, the FD of the cap screw structures the geometric model of this component with its FIs. Therefore, the shaft of the screw can be located, hence the location of the clearance. Section 6 shows how the identification of the FD of a component is connected to its shape structure.

Currently, the focus is placed on functional clearances represented as contacts. Consequently, geometric clearances are not addressed here.

Figure 6 illustrates different types of geometrical interactions on the example of two plates assembled together by means of a cap screw.
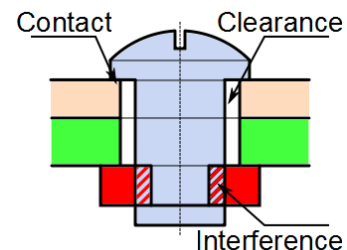

Figure 6 Geometric interactions of a bolt.

4.8. REFERENCE STATES

To evolve from pure geometry toward functions, as pointed at Section 4.5, it is mandatory to connect geometric parameters to other physical ones. FIs follow that path. Now, to connect to functions, the principle is to take advantage of the shape-behavior-function relationship often quoted in design methodologies [14], [19], [20] and functional design approaches [26], [27].

Referring to the concept of *behavior* is performed qualitatively using the concept of *state* of an assembly model. More details about the qualitative reasoning process can be found in [23]. A state of an assembly describes its physical and qualitative behavior through equilibrium equations. A behavior law is applied to each of its components where each interface is assigned a possible FI. This behavior law helps characterizing the physical objective of the state and dualities between geometry and mechanics are used to set the parameters of this behavior law from the FIs.

Because geometric interfaces can produce several functional interpretations, i.e. several FIs, the purpose of this qualitative reasoning is twofold:

- Filtering out FIs to reduce them to a single one per geometric interface;

- Identifying components contributing to a function.

It is assumed that assembly models input are consistent. To eliminate non-functional solutions, referential behavioral descriptions of a functional product must be available. Those descriptions are grouped in *reference states*. We have so far identified three reference states; static equilibrium, loads cycles, and kinematic chains.

Static equilibrium

The mechanical reference state assumes that all components are rigid bodies, and that each component of the assembly in hand is at mechanical static equilibrium; that is:

- The vector sum of all external forces is zero, and

- The sum of moments of all external forces around any axis is zero.

This can be otherwise stated as that the mechanical screws applied to all CIs of a component sum up to zero.

$$\sum_{c \in CI\{P\}} \{\overrightarrow{F_c} | \overrightarrow{M_c}\}_{/(\vec{0},\vec{x},\vec{y},\vec{z})} = \{\vec{0} | \vec{0}\} \qquad (1)$$

This is because CIs represent all the possible interactions of a component with its environment; thus, exhaustively incorporate all external forces and their moments.
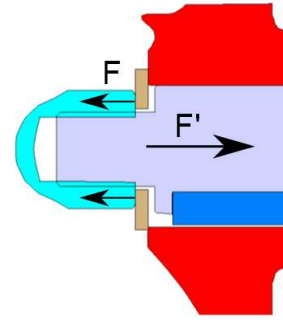
Figure 7 Cross section in pump assembly model.

Figure 7 shows a zoomed-in cross section in the centrifugal pump's model at the upper part of the shaft. Here, the planar contact between the nut and the washer can only be interpreted as planar support generating force $\vec{F}$, an opposite force $\overrightarrow{F'}$ should be generated by the only other CI of the nut, which is the cylindrical interference defining a threaded link to enable static equilibrium. This reasoning leads to eliminate the spline link connection interpretation of this CI because this FI cannot equilibrate the axial force $\vec{F}$.

Loads cycles

Based on the previous reference state, certain internal load generators can be recognized (such as springs and threaded links). Those internal loads propagate through force cycles in a functionally-coherent model. Detection of internal force cycles is necessary to group components and interfaces in functional groups, where they all together participate to fulfill certain functionality, such as fastening.

In Figure 7, the internal load generated by the threaded link between the nut and the shaft (blue and grey components respectively) is propagated through the washer then the impeller (brown and red respectively), before it bounces back to the shaft, as shown in Figure 4, forming a cycle.

Load cycles are of particular interest for FE analyses since they identify the component subsets that can be submitted to specific geometric transformations when preparing a FEM, e.g. interfaces between plates

 Ahmad Shahwan, Gilles Foucault, Jean-Claude Léon, Lionel Fine

tightened together in Figure 6 can be assigned friction behavior. This information can significantly help automate the FE preparation process because it structures the assembly model into functional subsets relevant from a FEA point of view.

Kinematic chains

The kinematic reference states also adopt the rigid body assumption, it is based on closed kinematic chains stating that the relative motion between two bodies A, and B equals to the sum of the relative motion between A and C and the relative motion between C and B, given that A, B, and C are rigid bodies, and that relative motions are expressed as rotational and translational velocities with reference to the same coordinate system.

That can be otherwise stated as that the kinematic screws of all CIs forming a closed cycle in the assembly model with respect to the same coordinate system sum up to zero.

$$\sum_{c \in \{c_0, c_1, \dots c_n\}} \left\{ \overrightarrow{\Omega_c} \middle| \overrightarrow{V_c} \right\}_{/(\vec{0}, \vec{x}, \vec{y}, \vec{z})} = \left\{ \vec{0} \middle| \vec{0} \right\} \qquad (1)$$

This is because the relative motion of a rigid body with respect to itself is zero. By arbitrary choosing one component of our closed cycle, and then repeatedly applying Chasles' equation starting by its first two neighbors, until the cycle is closed, we conclude that the sum of relative motions (represented as kinematic screws) equals to the relative motion between the chosen component and itself, that is zero.

Kinematic classes obtained from this state can connect with FEA hypotheses, e.g. to remove interfaces and merge components, to set contact and friction boundary conditions where relative motions are critical.

Besides assumptions related to each reference state, we also set complementary hypotheses that are valid across states and set the scope of our qualitative reasoning.

Model's consistency

Alongside the qualitative reasoning process, we assume the assembly, hence its geometric model, to be consistent from both functional and conceptual points of view. This means:

- All components of the product are held tight together, which in turn leads to the mechanical reference state;

- A component with two non-coplanar planar contacts that share the same orientation (thus parallel) indicates a design defect (a double contact configuration) because the component is rigid and manufacturing tolerances cannot ensure these multiple contacts;

- Apart from the component mobilities forming the kinematic chains of the assembly, all other mobilities, if any, in this assembly should reduce to rotations only. Unjustified translations signal a design defect because they create undesired dynamic effects.

Time invariance

Each of the reference states described above is subjected to their specific hypotheses. However, these hypotheses address the components and/or their CIs as they appear in the assembly model observed. Indeed, this observation contains another implicit hypothesis shared by all the states previously described: all the CIs of the assembly model studied are time invariant. This means the CIs extracted from the geometric model of an assembly stay unchanged during all the states expressing a set of its behaviors.

Considering different states of an assembly (rest configuration or operational state, action of a single component, among others) enriches the qualitative reasoning process. Unfortunately, an assembly model represents only a configuration at a given time, thus creating ambiguities in locating moving parts. To remove such ambiguities, user's inputs are mandatory though they must be as few as possible. To identify kinematic chains, user's input can be reduced to a relative movement expressing the start/end of a kinematic chain. Then, the qualitative reasoning process propagates this information throughout the assembly to identify the corresponding chain.

4.9. BOTTOM-UP APPROACH

Our reasoning follows a bottom-up approach in that we start from a pure geometric model of an assembly with no explicit functional information. Conventional representations of components produce geometric interfaces that initiate the generation of low level functional information, i.e. functional interfaces.

Then, the qualitative reasoning using static equilibrium can address individual components to start reducing the number of FIs per geometric one, wherever applicable. Increasing the extent of interfaces analyzed, functional information is gained using other reference states such as internal loads cycles and kinematic chains. Also, this leads to further eliminations of FIs if applicable. Then, a matching process using inferences takes place to assign FDs to categories of components, whenever possible, from the functional information gained through a range of behaviors analyzed using reference states (see Figure 2). As an example, cap screws can be identified once the static equilibrium and internal load cycles have been evaluated (see Section 5).

5. IMPLEMENTATION OVERVIEW

As seen before (Figure 2 and Section 4.5), the generation of functional properties is divided into three major tasks, geometric identification, functional interpretation, and matching. In this section, we address some technical details of each of them.

5.1. IDENTIFICATION

This first task performs a pure geometric analysis of an assembly model in order to identify adjacent components and define their interfaces to populate the shape interface taxonomy.

As a result, the interfaces between components are organized in a graph data structure called the *conventional interface graph* (CIG), whose nodes are the components, and whose arcs are CIs. More precisely, at this stage the CIs only contain the geometric model of interfaces (contacts and interferences).

We adopt the STEP file format [29], [30] as a standardized representation of our geometric model. Although ISO 10303 proposes annotations to include functional information in STEP [22], the assembly model is considered to be purely geometric since this information is neither reliable nor accurate because it is user defined [15], [31], [32].

The software is developed using Open CASCADE development platform [28] to set up this geometric analysis.

The current approach identifies interfaces based upon *canonical surfaces,* i.e. planes, cylinders, cones, tori, and spheres. Those surfaces are most frequently used to define functional surfaces forming the FIs. This

originates from manufacturing properties where canonical surfaces are easier to machine.
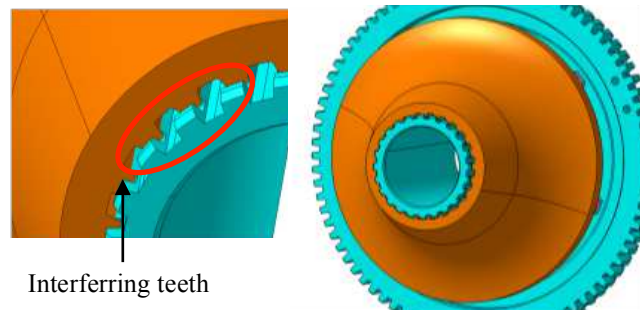
Interferring teeth

Figure 8 Approximate relative rotational position of components in a spline link (detailed view on the left, global one on the right) (courtesy ANTECIM).

Analyzing the geometry of an assembly is currently subjected to complementary hypotheses based on the analysis of industrial DMUs. Assuming that the tolerance of a geometric modeler describing B-Rep volumes in a CAD software is ε. $\varepsilon = max_{P_1 \in S_1} min_{P_2 \in S_2}(\|\overrightarrow{P_1 P_2}\|);$ where $\|\overrightarrow{P_1 P_2}\|$ designates the distance between two points P_1 and P_2 belonging to adjacent boundaries of patches S_1 and S_2, respectively. An order of magnitude of ε is 0.001mm for an industrial CAD software. Then, hypotheses can be stated and explained as follows:

- *Relative positions of components are accurate.* Indeed, it can be observed that the relative location of components can be far less accurate than the accuracy of a geometric modeler. One category of configurations illustrates this fact through Figure 8. There, the rotational relative position of the components forming the spline link exhibit interfering teeth rather than planar contacts. Generally, there is no strict constraint for the designer to set up a precise relative position between these components. If the kinematic chain does not incorporate any rotational indexing function, any approximate angular position is acceptable, which simplifies the task of the designer when setting up the digital model. Other similar configurations can be identified. Compared to ε, these configurations can produce deviations of several tenths of millimeters in magnitude;

- *Digital models of components are accurate.* This hypothesis derives from the observation that some surfaces of components may not be modeled as accurately as they are manufactured, i.e. the location of some surfaces significantly

Ahmad Shahwan, Gilles Foucault, Jean-Claude Léon, Lionel Fine

differs from their theoretical nominal position. These inaccuracies can be observed when these surfaces are not explicitly used to monitor a manufacturing process, e.g. teeth of gears can have their profile directly obtained from specific machine tools. Consequently, the tooth profile and/or the rotational symmetry of teeth can be less accurate than ε (see Figure 9). The loss of accuracy can be an order of magnitude greater than ε;

- *Digital models of components are homogeneously accurate and accurate at a level smaller or equal to ε.* The lack of homogeneity of component accuracy originates from the collaborative design activity often part of product design. Consequently, several companies using different conventional representations of components can incorporate digital shapes in the same assembly model, which can greatly influence geometric interfaces and illustrates the inhomogeneous accuracy that can be found in some assembly models. Another type of inhomogeneous accuracy originates from the multiple CAD modelers that can produce different component models belonging to the same assembly when several companies take part to a product design. If ε_i is the accuracy of one of these modelers, the geometric analysis of an assembly can be performed with = $\max_i(\varepsilon_i, \varepsilon)$. However, if $\alpha > \varepsilon$, geometric transformations required to produce an FEM need specific operators to robustly adapt component shapes. This issue is kept for future developments, which reduces the current approach to $\alpha \leq \varepsilon$.

Extending the current hypotheses to process industrial assembly models requires specific operators that are out of scope of the present description.

Maximal B-REP surfaces

STEP describes components' geometric models in a Boundary Representation (B-Rep) format. Unfortunately, a B-Rep encoding of a geometric object is not unique. That is; two STEP files may represent the same shape differently. This is due to the fact that an edge (then called a wire) can be represented as a set of topologically connected smaller edges laying on the same curve. The same applies to faces, where a face can be divided into smaller ones that share the same surfaces and are

topologically connected. This phenomenon originates from the component's modeling process where functional surfaces are often broken down into smaller pieces because of the constructive modeling process inherent to industrial CAD modelers.

Interferring teeth   Original shape
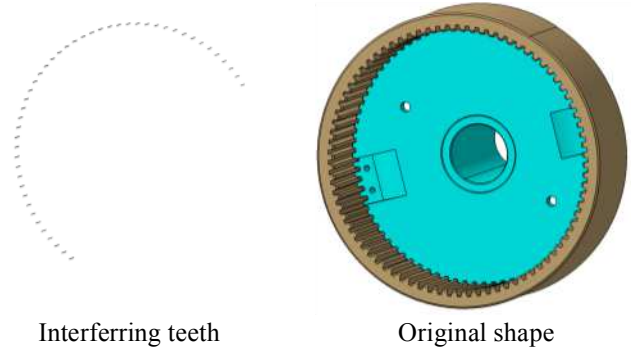
Figure 9 Effect of rotational symmetry inaccuracies. Not all the teeth are interfering whereas the components are exactly coaxial (courtesy ANTECIM).

Additionally, geometric modelers are subjected to topological and parameterization constraints [33]. This prevent the boundary decomposition from matching the real boundaries of a component, e.g. a cylindrical surface can be represented either with two half cylinders or a single cylindrical patch whose boundary contains a generatrix that is meaningless because it is not a boundary of the surface on the real component (see Figure 10).

The representation of real surface boundaries is mandatory to obtain a set of geometric entities that is consistent with the properties used in the qualitative reasoning process.

To obtain this representation of a component boundary, adjacent faces that belong to the same canonical surface are merged into one entity; a *maximal face*. A maximal face is represented by its underlying oriented, topologically connected faces. Edges are also grouped into *maximal edges* using the same criterion. As a result, a cylindrical face can end up with a boundary described by two closed edges without vertices. The corresponding data-structure uses hyper-graphs [34].

Geometric analysis

Bounding boxes are used to allow early filtering of non-adjacent solids. The remaining ones are then checked pairwise for geometric interactions.

For each pair, maximal faces of one solid are compared against those of the other. We adopt a simple, yet extensible approach to extract geometric

interactions, based on the comparison of the geometric parameters of surfaces.

The purpose is to produce the geometric domain describing each *shape interface*, e.g. a cylindrical contact, or a cylindrical interference. Other categories include planar contact, linear contact, circular contact, conical contact, etc. as organized in the shape interface taxonomy.

Each shape interface is then encapsulated into a CI connecting two components. The result is then structured in the CIG, as the output of this phase.
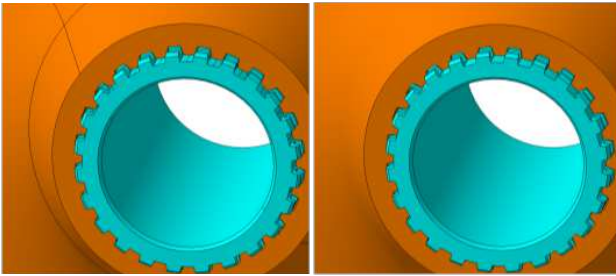
Figure 10 Effect of maximal faces and edges generation. Patch boundaries are marked with black edges. Initial boundary decomposition (left), boundary decomposition with maximal faces and edges (right) (courtesy ANTECIM).

5.2. Interpretation

Following a simple bottom up approach this second phase associates to each CI all possible functional interpretations it may hold. Interpretations are suggested by its geometric interaction properties (shape interface) and derive from the duality between geometry and internal forces and the functional conventional representations chosen. To this end, a thesaurus has been set up that provides the mapping between shape interfaces and their interpretations as FIs.

Once this association is done at the level of CIG, the qualitative reasoning process begins with the help of reference states. So far, there are three, possibly overlapping, analyses to take place, and they complete each other (see Section 4.8). The static equilibrium and kinematic chain analysis eliminate functional interpretations of CIs that are incompatible with either of them. Load cycles structure the CIG to identify the components contributing to a given function. Throughout this reasoning, physical properties are represented as screws. Those screws, however, do not hold scalar values, but qualitative constraints instead. Such constraints are: positive, strictly positive, negative, strictly negative, not null,

arbitrary, and one quantitative value that is null, which is also regarded as a constraint [23].

Mechanical analysis

For each component, Eq. 1 must hold truth; that is the screws representing all mechanical interactions exterior to the component being studied at all its CIs must sum up to zero. Considering that one CI may be interpreted as more than one FI, thus it has more than one possible mechanical screw. This analysis ends up with $\Pi_{i=1}^{n} |CI_i|$ different combinations; where n is the number of CIs of the underlying component, and $|CI_i|$ is the number of functional interpretations of CI_i.

For each of those combinations, the algorithm tests the possibility that all mechanical screws, sum up to zero. This study reveals incoherent combinations that are rejected, leading to the elimination of certain function interpretations of a CI; thus reducing $|CI_i|$.

Whenever possible, the goal of this analysis is to end up with only one functional interpretation per CI; that is $|CI_i| = 1$. This may not be achieved from the very first iteration on the components. However, the study of one component may lead to the elimination of some interpretations of an interface shared with a previously studied neighboring component. This adds up information that may in turn help eliminating further interpretations if the neighboring component is put to examination again. For this reason, the reasoning process is iterative. A component is checked once it is studied; however, it can be unchecked whenever an interpretation of one of its interfaces is suppressed, thus reducing the number of leaves in the taxonomy of FIs assigned to it. The iterative process stops whenever all assembly components are checked.

Kinematic analysis

This phase builds upon the kinematic reference state to define what we call *kinematic equivalence classes*. A kinematic equivalence class (or kinematic class, for short) is a set of components that share the same relative motion; that means that all members of a kinematic class can be regarded as a single rigid body.

The kinematic reference state is based on rigid body kinematics. It addresses closed loops of connections in the CIG. Eq. 2 is used along with kinematic screws that are properties of the functional interpretation to infer components' relative mobilities.

For this reason, a minimal user intervention may be needed, mainly to describe objects' motion after components are classified in kinematic classes. User's input specifically applies to rotational movements where the surfaces of revolution at t and $t+dt$ cannot bring information about whether a rotation exists or not. The kinematic properties help reducing further the number of leaves in the taxonomy of CIs assigned to each component.

Synthesis of functional designations

After the collection of mechanical and kinematic properties of components, and the generation of FIs, load cycles and kinematic classes; this information is integrated all together to deduce the FD of components.

FIs and load cycles are translated into FDs using a matching process based on inferences. A FD ontology describes the mechanical and kinematic properties of each component category. These properties are based on the type and relative positions of the FIs of each component, independently of its dimensions. For example, a component is classified as a "Cap Screw" when it has a threaded link and a planar support whose normal is parallel to the threaded link axis, with at least another planar support parallel to the first one, between the first one and the threaded link and joining two adjacent components. The candidate component should also belong to the same kinematic class as its two adjacent components.

The taxonomies and ontologies required to define FDs, FIs, etc. are set up using Protégé [35] and inferences are expressed with FACT++ reasoner [36].

6. RESULTS

To validate the proposed approach, we use a model of root-joint from the aircraft industry (see Figure 11). This assembly illustrates the wing-fuselage junction. We also demonstrate algorithm performances using a centrifugal pump model (Figure 4). Those models contain most of the geometric interactions we are concerned about.
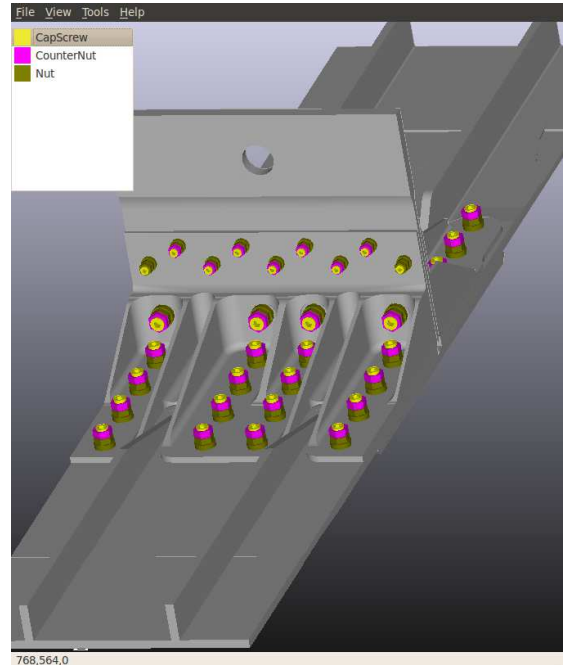

Figure 11 Model components colored according to their FDs.

Table 1 shows execution times for both assemblies. It is to be noticed that geometry loading time is the time that Open CASCADE takes to load the STEP file. This is out of the scope of our work, and no effort is made to enhance this time.

Ontology load time is managed by FaCT++. Network configurations have an impact here, since we use a client-server paradigm. Though this paradigm reduces performances compared to interfacing with a reasoner through its API, it enables high flexibility when choosing the reasoner (Pellet [37] has also been successfully tested).

Table 1 Execution time for root joint and pump models.

Example	№ Solids	Time (m.s.)						№ Cont.	№ Interf.
		Load Geometry	Load Ontology	Geometric Analysis	Qualitative Analysis	Semantic Reasoning	Total		
Root Joint	148	12234	1060	15733	1406	6320	36755	511	90
Centrifugal Pump	43	5261	2065	7134	7014	2431	23907	139	11
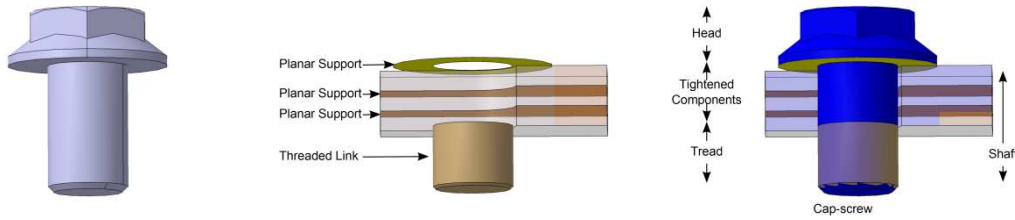
Figure 12 Restructured geometry of a cap-screw
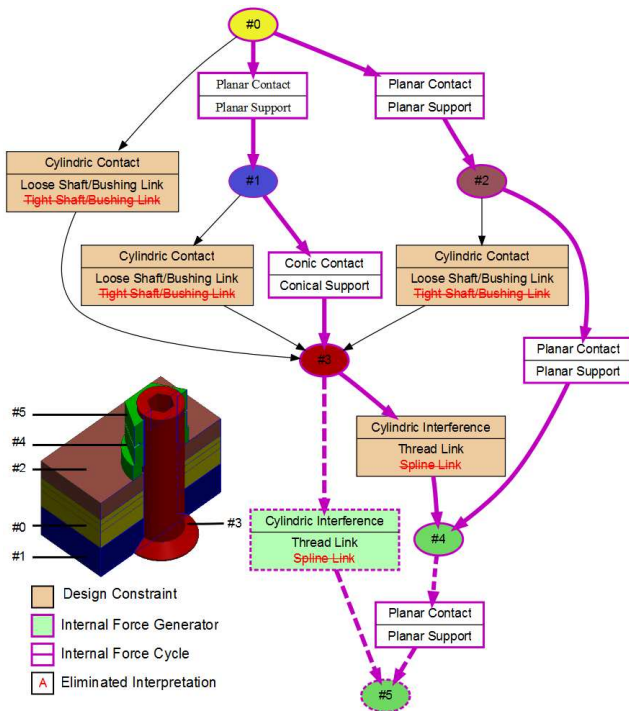
Figure 13 Sub-graph of CIG alongside its corresponding components in a DMU, showing FI elimination and detection of load cycles.

Geometric and qualitative analysis are indeed in the core of our interest in this research. Table 1 shows reasonable times considering the models' sizes.

Semantic reasoning is again performed by FaCT++ for the results shown in the table. However, care was taken to keep rules simple, and to minimize client/server communication, in order to speed up this phase of execution.

Figure 11 shows a snapshot of our application, after running the whole execution cycle on the root-joint example. The legend on the top left corner of the window shows detected FDs in the current model, alongside their respective colors in the geometric viewer part of the window.

A cross section into one of the detected bolted joint is shown in Figure 13. This figure also shows a sub-graph of the CIG, depicting how the number of FIs was reduced to one per CI using the qualitative

reasoning process. Figure 13 also shows load cycles, colored in magenta. Dotted lines refer to parts of the cycle that leads to the detection of a statically undetermined configuration, characterizing a possible function of a nut, i.e. a counter-nut.

Figure 12 shows how the initial geometry of a component gets structured according to its FIs once its FD is deduced (cap-screw in this case). The load cycles and the geometric model restructuring contribute directly to FEA requirements where load cycles help identifying sets of components related to physical phenomena, e.g. tightening effect of a bolt. FIs define how the geometric zones where specific boundary conditions, e.g. friction, contact, etc. can be applied in a FEM. All this information can significantly speed up the preparation process of a FEM.

7. CONCLUSIONS AND FUTURE WORK

This work is a step towards a semi-automated identification of components FD in a DMU based on their pure geometric description. Simultaneously with the FD, the geometric model of components gets structured based on their geometric interfaces with neighboring components. We also showed some results of the first phases of our work to validate the efficiency of the proposed approach and demonstrate how multiple interpretations can be reduced to unique ones using the mechanical equilibrium state analysis.

The integration of components neighboring interfaces in the inference process has a strong effect over the structure of a geometric model of a component. Starting with mere geometry, and passing through different other physical, parameters, such as mechanical and kinematic assessment, significant functional properties of component interfaces are derived. Then, the inferences referring to the relative position of the interfaces as necessary condition of a component to belong to a FD effectively subdivides the component with low level functional information.

Ahmad Shahwan, Gilles Foucault, Jean-Claude Léon, Lionel Fine

The content of the enriched assembly model is of great help for FEM preparation. The FD of C helps identifying the other components related to the function of C. The location of interfaces connected to the function of C can be used to partition C and the other components, set up shape transformations depending on the simulation objectives and desired boundary conditions at the interfaces related to the function of C.

Further work covers the insertion of complementary rules and states to identify a larger range of components; keys, bearings, gears are among the components to be addressed. Interfaces between two components that share similar geometrical properties will be grouped to enlarge the range of interface configurations that can be processed. Identifying punctual contacts and those along lines is another extension of the present work that will help cover a larger range of interface configurations. Higher level functional information will be derived to group components contributing to the same functional behavior. Operators will be developed to widen the hypotheses used to process assembly models for geometric interface detection.

ACKNOWLEDGMENTS

REFERENCES

[1] E. Blümel, S. Straßburger, R. Sturek and I. Kimura, (2004), "Pragmatic Approach to Apply Virtual Reality Technology in Accelerating a Product Life Cycle", Proc. of International Conference INNOVATIONS, Slany, Czech Republic, pages 199—207.

[2] V. Raghavan, J. Molineros, R. Sharma, (1999), "Interactive Evaluation of Assembly Sequences Using Augmented Reality", IEEE Trans. Robot. Autom., 15(3), 435—449.

[3] M. Schenk, S. Straßburger and H. Kissner, (2005), "Combining Virtual Reality and Assembly Simulation for Production Planning and Worker Qualification", Proc. of Int. Conf. on Changeable, Agile, Reconfigurable and Virtual Production, Munich, Germany, pages 411—414.

[4] F. Boussuge, J-C. Leon, S. Hahmann and L. Fine, (2012), "An analysis of DMU transformation requirements for structural assembly simulations",

Proc. of 8th Int. Conf. on Engineering Computational Technology, Dubronik, Croatie.

[5] S. Joshi and T. C. Chang, (1988), "Graph-based heuristics for recognition of machined features from a 3D solid model", Comput. Aided Des., 20(2), 58—66.

[6] H. Zhu and C. H. Menq, (2002), "B-Rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition", Computer Aided Design, 34(2), 109—123.

[7] A. Thakur, A. G. Banerjee, S. K. Gupta, (2009), "A survey of CAD model simplification techniques for physics-based simulation applications", Comput. Aided Des., 41(2), 65—80.

[8] J-C Léon, L. Fine, (2005) "A new approach to the preparation of models for F.E. analyses", Int. J. of Comp. Appl. in Technology, 23(2, 3, 4), 166—184.

[9] B. Falcidieno, F. Giannini, (1989), "Automatic recognition and representation of shape-based features in a geometric modeling system", Comput. Vision Graph. Image Process., 48(1), 93—123.

[10] A. L. Ames, (1991), "Production ready feature recognition based automatic group technology part coding", Proc. of the first ACM symposium on Solid modeling foundations and CAD/CAM applications, Austin, Texas, United States, pages 161—169.

[11] H. Date and S. Kanai and T. Kishinami and I. Nishigaki, (2006), "Flexible feature and resolution control of triangular meshes", Proc. of the sixth IASTED International Conference on Visualization, Imaging and Image Processing, Palma De Mallorca, Spain, pages 319—324.

[12] S. Venkataraman, M. Sohoni, (2002), "Reconstruction of feature volumes and feature suppression", Proc. of the seventh ACM symposium on Solid modeling and applications, Saarbrücken, Germany, pages 60—71.

[13] V.B. Sunil, R. Agarwal, S.S. Pande, (2010), "An approach to recognize interacting features from B-Rep CAD models of prismatic machined parts using a hybrid (graph and rule based) technique", Computers in Industry, 61(7), 686—701.

[14] A. Albert, N. Burkardt, M. Ohmer, (2006), "Contact and Channel Model for Pairs of Working Surfaces", Advances in Design, H. A. ElMaraghy, W. H. ElMaraghy, Springer, London, 511—520.

[15] N. J., Mitra, Y.-L., Yang D.-M., Yan W., Li, M. Agrawala, (2010), "Illustrating how mechanical assemblies work", ACM Trans. Graph., 29(4), 58:1—58:12.

[16] B. Clark, B. Hanks, C. Ernst, "Conformal assembly meshing with tolerant imprinting", in: Proc. 17th

Meshing Roundtable,Pittsburg, USA, October, 2008, pp. 267—280.

[17] R. Chouadria, P. Veron, (2006), "Identifying and re-meshing contact interfaces in a polyhedral assembly for digital mock-up", Eng. with Computers 22 (1), 47—58.

[18] R. Lou, J.-P. Pernot, A. Mikchevitch, P. Veron, (2010), "Merging enriched finite element triangle meshes for fast prototyping of alternate solutions in the context of industrial maintenance", CAD 42 (8), 670—681.

[19] A. Albers, A. Braun, P. J. Clarkson, H.-G. Enkler and D. Wynn, (2009), "Contact and channel modelling to support early design of technical systems", Proc. Int. Conf., ICED'09, August, Stanford, USA, pages 61—72.

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

[21] TraceParts, URL http://www.traceparts.com.

[22] G. Foucault, A. Shahwan, J-C. Léon, L. Fine, (2011), "What is the Content of a DMU? Analysis and Proposal of Improvements" In Colloque National AIP-PRIMECA, Mont-Dore, France.

[23] A. Shahwan, J-C. Leon, G. Foucault, M. Trlin, and O. Palombi, (2013), "Qualitative behavioral reasoning from components' interfaces to components' functions for DMU adaption to FE analyses", Computer-Aided Design, 45 (2): 383—394.

[24] ISO Technical Committee TC2, Fasteners, (1947), Berlin, Germany.

[25] ISO Technical Committee TC4, Rolling Bearings, (1947), Stockholm, Sweden.

[26] U. Roy, N. Pramanik, R. Sudarsan, R. D. Sriram, K. W. Lyons, (2001), "Function-to-form mapping: model, representation and applications in design synthesis", CAD, 33 (10): 699-719.

[27] U. Roy, B. Bharadwaj, (2002), "Design with part behaviors: behavior model, representation and applications", CAD, 34 (9): 613-636.

[28] Open CASCADE Technology, URL http://www.opencascade.org/.

[29] ISO 10303-204:2002. STEP Part 204: Application protocol: Mechanical design using boundary representation. ISO, Geneva, Switzerland, 2002.

[30] ISO 10303-204:2003. STEP Part 42: Integrated generic resource: Geometric and topological representation. ISO, Geneva, Switzerland, 2003.

[31] T. Ullrich and D. W. Fellner, (2011), "Generative Object Definition and Semantic Recognition", In proc. Eurographics Workshop on 3D Object Retrieval, pages: 1—8.

[32] R. Wessel and R. Klein, (2010), "Learning the Compositional Structure of Man-Made Objects for 3D Shape Retrieval", In proc. Eurographics Workshop on 3D Object Retrieval, pages: 39—46.

[33] K. Li, (2011), "Shape Analysis of B-Rep CAD Models to Extract Partial and Global Symmetries", PhD thesis, Grenoble University.

[34] G. Foucault, J-C. Cuillière, V. François, J.-C. Léon and R. Maranzana, (2008), "Adaptation of CAD model topology for finite element analysis", Computer-Aided Design, 20 (2): 176—196.

[35] Protégé (2012), URL http://protege.stanford.edu/.

[36] D. Tsarkov and I. Horrocks, (2006), "FaCT++ description logic reasoner: System description", In Proc. of the Int. Joint Conf. on Automated Reasoning, pages: 292—297.

[37] E. Sirin, B. Parsia, Bijan, B. C. Grau, A. Kalyanpur, Y. Katz, (2007), "Pellet: A practical OWL-DL reasoned", Web Semantics, 5 (2): 51—53.