



**HAL**  
open science

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

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

► **To cite this version:**

Ahmad Shahwan, Jean-Claude Léon, Lionel Fine, Gilles Foucault. Deriving Functional Properties of Components from the Analysis of Digital Mock-ups. TMCE 2012 - 9th International Symposium on Tools and Methods of Competitive Engineering, Karlsruhe Institute of Technology (KIT); Delft University of Technology, May 2012, Karlsruhe, Germany. pp.739-751. hal-00645625

**HAL Id: hal-00645625**

**<https://inria.hal.science/hal-00645625>**

Submitted on 28 Nov 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## REASONING ABOUT FUNCTIONAL PROPERTIES OF COMPONENTS BASED ON GEOMETRICAL DESCRIPTIONS

**Ahmad Shahwan**

Grenoble University  
ahmad.shahwan@grenoble-inp.fr

**Gilles Foucault**

Grenoble University  
gilles.foucault@g-scop.inpg.fr

**Jean-Claude Léon**

Grenoble University / INRIA Rhône-Alpes  
jean-claude.leon@grenoble-inp.fr

**Lionel Fine**

EADS Innovation Works  
lionel.fine@eads.net

### ABSTRACT

*Digital Mock-ups (DMUs) are widespread and form a common basis for product description. However, DMUs produced by industrial CAD systems essentially contain geometric models and their exploitation often requires new input data to derive various simulation models. In this work, analysis and reasoning approaches are developed to automatically enrich DMUs with functional and kinematic properties. Indeed, interfaces between components form a key starting point to analyze their behaviours under operational reference states. This is a first stage in a reasoning process to progressively identify mechanical, kinematic as well as functional properties of the components. The overall process relying on the interfaces between components addresses also the emerging needs of conventional representations of components in industrial DMUs. Inferred semantics add up to the pure geometric representation provided by a DMU, to allow for easier exploitation of the model in different phases of a Product Development Process (PDP).*

### KEYWORDS

Designing and validating smart and intelligent products, DMUs, Geometric models, Assembly, Functional designation, Mechanics, Kinematics, Reasoning and knowledge representation

### 1. INTRODUCTION

As geometrical representations of a product, being an assembly of a number of solids, digital mock-ups (DMUs) provide engineers with powerful tools that allow for innovation and cut off production time.

As today's modelers provide user-friendly tools and visual environments to help the designer at the conceptualization phase of a product lifecycle, designers invest more time now applying their core engineering competencies to promote the quality of the model. Moreover, and with the existence of digital simulations that can predict the behaviour of the product being designed under operational circumstances, often interactively with the help of virtual reality devices, the designer can envisage the outcome of his work shortly after the model is conceptualized. Virtual and augmented reality techniques, varying from simple visualization to fully-immersive environments have been used in different areas throughout product's development process such as design and modelling, structural and behavioural simulation, and assembly/disassembly simulations and planning, to name only few [1][2][3]. However, all the corresponding simulation models need a fair amount of time to be generated from DMUs because complementary data must be interactively attached to each component, delaying the availability of a simulation model. This can result in delay increases up to a point where a simulation becomes useless because its output arrives too late in a PDP. Therefore, reducing the simulation

preparation time at the level of an assembly becomes a key issue to the efficiency of a PDP.

Because simulation algorithms are still too computationally heavy to allow for the direct processing of design models, especially with new emerging demands such as real-time interactivity, a moderately complex design model has to pass by a simplification stage before launching the necessary computations for simulation.

The simplification process, however, makes use of field expertise possessed by knowledgeable engineers and domain experts. Thus, this task is often done manually, despite efforts to automate it. The manual simplification is feasible to a certain extent, where model complexity and number of components are small enough to allow the modification to be done within the limits of available manpower. However, most industrial models exceed this extent by far, making the process uncomfortably time and resource consuming. It is also the purpose of the proposed approach to speed up the simplification processes for assemblies.

The rest of the document is presented as follows; we first address previous literature related to our work in Section 2. Next, we shed the light on what distinguishes our work, and highlight our contribution to the literature in Section 3. Section 4 **Erreur ! Source du renvoi introuvable.** addresses conceptual aspect and defines basic concepts central to our approach. In Section 5 we develop our approach in more details. Results are briefly shown and explained in Section 6. In the last section; Section 7, we conclude to summarize what have been done so far, and addresses 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 [4] have been paid in the field of features recognition (FR) in solid models. [5] 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 and are functionally significant.

In [4] 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 [6] addressed the problem of functional features extraction out of digital models, and classified existing solutions into human assisted approaches, feature based modelling, 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 [7], 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 of feature recognition shows a wide range of techniques that participate to the Computer Aided Process Planning (CAPP) automation. In [8] the feature recognition is integrated into the process of simplification as a preliminary step to prepare a tessellated model for finite element analysis. A technique to detect and simplify blending features to enhance the process of functional features detection is presented in [5] where the preservation of the topological properties of the underlying objects is taken into priority. Another approach, capable of handling more interacting shape features through an iterative approach is presented in [9], where form feature recognition techniques are used to detect features face-sets, and then the feature is removed before passing to the next iteration, where previously interfering features can be detected.

In [10] authors again tackle the problem of features interaction through a hybrid approach for feature recognition that is both graph and rule based.

## 3. CONTRIBUTION

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 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 standalone volume model is processed as an isolated entity. Assemblies, when addressed, are generally regarded as a collection of components processed with loose or no connections between them. Our work shares the same interest of anticipating functional properties of products knowing their geometrical representations. However, we are more interested in the identification of functional denomination of entire components with

vast variety of geometric configurations, and much higher complexity, either in the size of the model itself (number of components/solids), or in the size each component (number of geometric entities: faces, edges and vertices). Interactions between components in an assembly are also brought to focus, where those interactions vitally contribute to components functionalities.

Due to the wide diversity of geometric configurations that one functional class of components may possess, more informative features had to be looked for in their solid model to enable the extraction of functional behaviour rather than its mere intrinsic geometry.

#### 4. ASSEMBLY ANALYSIS

Approaches to DMUs' simplification still fail short to efficiently transform geometrical model entities mainly because of the lack of any functional descriptors of those entities. Our work comes to fill this gap, automatically enriching the plain geometrical representation with meaningful semantic annotations, as a preliminary step of the simplification process. To this end, we develop an algorithm that extracts some functional and kinematic features from product components as they are in their assembly configuration, to enable the inference of their *functional designations* (Section 4.4).

We briefly give an overview of our method of problem solving, before going into details of conceptualisation then design and development.

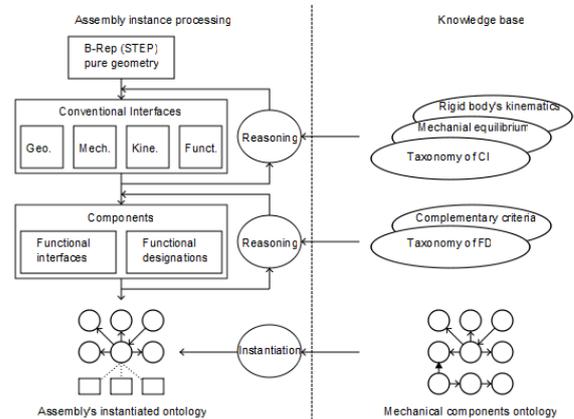
##### 4.1. Overview

The input to our algorithm is a pure geometric representation of a product. We first extract features that matter to our work out of such data, those features being the geometric interactions between adjacent components in the assembly.

Next, we enrich our knowledge about the assembly in hand, by narrowing the amount of doubt about mechanical, kinematic and functional properties. To this end, and to enable this clarification, more information is incorporated into our reasoning process. Such information is inspired by the domain knowledge of mechanical and industrial engineering.

We start with what geometric properties suggest, which is usually a vast collection of interpretations. Those interpretations are then reduced as a result of introducing vital pieces of information to our

knowledge base, such as mechanical equilibrium equations that hold truth all across the assembly, taking into consideration that its components are initially considered as rigid bodies. More knowledge is inferred as more information is considered.



**Figure 1** The overall processing of information

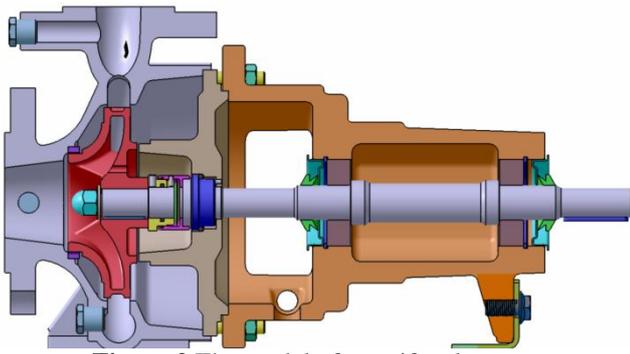
When enough knowledge about components in an assembly is gathered, an ontology describing functional designations and their properties is invoked to assign those designations to the assembly's components. The ontology serves as a reference to predict functional designations of components based on their previously inferred properties, and to provide knowledge repository enabling querying certain information about an assembly once instantiation is done; which is the assignment of model components to ontology classes.

Before concept of functional designation is made clearer, we address briefly *mechanical components* in a product assembly and their geometric representation in a DMU.

##### 4.2. Mechanical components

Mechanical components or mechanical parts are modular elementary units that are meant to deliver precise and well-defined functionalities. They are often required to meet certain geometric configuration, to enable the interfacing with other components, to assemble a functional product.

In the present framework, mechanical components are represented as solids (volume entity) in a DMU that represents the whole assembly. Those solids can also be grouped to form sub-assemblies, where sub-assemblies in this case build up the final product. Figure 2 depicts an example of an assembly of a centrifugal pump, showing different components.



**Figure 2** The model of centrifugal pump.

### 4.3. Component representation

The starting point of a DMU analysis contains the shape as a 3D representation of each of its components. This representation is often considered as equivalent to the physical component. However, the current practise in industry is to take advantage of components libraries such as TraceParts (<http://www.traceparts.com>) and to find compromises between the shape complexity of real component, the modelling time needed to produce its 3D digital model and a shape that can be easily processed at the subsequent steps of a PDP. As a result, real component representations and their 3D digital one may differ from each other (see Figure 3 and Figure 4).

Very often, the threaded part of a real component is simplified or idealized into a cylindrical area (see the difference between Figures 1 and 2). Similarly, teeth of gears are often removed in their digital models as an idealized representation.

The libraries collect 3D models of components as generated by the components providers. This means that they are not certified and may differ from each other even if the components are similar, e.g. a threaded hole of a bolt with a given nominal diameter may be found with different 3D models having different thread diameter. Handling the shape variants of components in libraries is not part of the present framework.

As a result, the idealizations of components influence the geometric interactions between them that form interfaces. In turn, using component shapes as a starting point of a DMU analysis can influence the inference of functional designations of components. Consequently, there is a strong dependency between shape-interface-function of components.

### 4.4. Functional designation

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

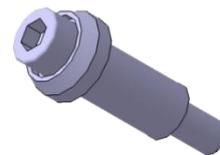
The functional designation decidedly determines the functional group of its component. One component can only have one functional designation, though it might have more than one function, indicated by the designation itself.

For instance, a screw whose shank is larger than its threaded part in diameter is usually referred to as shoulder bolt, shoulder screw or stripper bolt, and it has the functions of positioning and providing a pivot point at the same time (see Figure 3). “Shoulder Bolt” then is a functional designation that encapsulates two functionalities.

In this sense functional designations constitute equivalence classes that distinctly sort out all components in a digital mock-up.



**Figure 3** Shoulder bolt (courtesy Rabourdin Industrie).



**Figure 4** Shoulder bolt as represented in CAD systems (courtesy Rabourdin Industrie).

Functional designations are not matching the current designation of components in a bill of materials or as names of their digital model, e.g. ‘screw’ is part of current component names or designations in bills of materials. This designation is poor compared to the range of functions covered by this range of components and it is user-defined, which may not be uniform in a DMU and cannot be exploited in the current analysis process, because it is not reliable.

#### 4.5. Taxonomy of functional designations

In this work, we suggest a method to classify elementary components of a product through a taxonomy of functional designations. This is performed based on the geometrical description of different solids constituting the assembly, the interfaces between components and their neighbouring ones represented by the product's DMU and the component behaviour as it appears in the reference states associated with the DMU (see section 3.7). Then, it incorporates a functional meaning so that there is independency between each element to effectively form a taxonomy.

Different functional designations may share similar global functional behaviour, for example, screws are generally meant to fasten, and gears are normally expected to transmit moment, etc. As a result, the functional nomination can fit in a hierarchical structure whose final leaves are functional designations. We call this hierarchy the taxonomy of functional designations. Rooted by a label representing all possible mechanical components, the taxonomy provides more details about functional properties as one goes deeper in the hierarchy, until a leaf is reached which indicates an unambiguous definition.

Figure 5 shows a small portion of the functional designation taxonomy, showing the path to the functional designation of “Cap Screw”, amongst others.

Even though Figure 3 does not illustrate it, each leaf of the taxonomy contains also a geometric description of the component interfaces, their relative locations, mechanical and kinematic data as well, so that a connection can be set up between DMU geometry, mechanics, kinematics, and component functions.

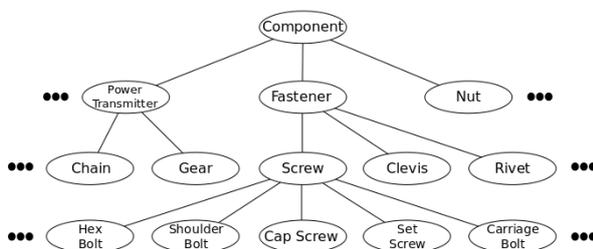


Figure 5 A subset of functional designation taxonomy.

Indeed, the functional designation taxonomy is the highest level one. Another taxonomy exists that is of lower level though more generic. It addresses the interfaces between components to express the possible functions that can be associated with the reference states. This taxonomy establishes a connection between the shape of an interface, its behaviour within each reference state and its function. Then, the functional designation taxonomy inherits from the interface one and forms a consistent framework incorporating geometry, mechanics, kinematics and functions covering reference states of the DMU.

#### 4.6. Conventional interfaces

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

A conventional interface is a broad concept that captures all aspects of the relationship between two neighbouring components in a product; it has geometric, mechanical, kinematic, 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 well defined, the goal shifts to deduce other properties to enable the mapping of each conventional interface into a meaningful functional interpretation. For example, our analysis may lead to the conclusion that a conventional interface having a cylindrical interference as geometric property (see Figure 7) transmits forces and moments in all directions, and allows neither translations nor rotations. This inference allows us to deduce that the concerned interface is a threaded link. We call such interpretation a *functional interface*.

Conventional interface form central concept in the core of our approach, around which the work can be divided into three distinguishable phases:

1. The geometric analysis to obtain geometric properties of conventional interfaces;
2. The interpretation of those geometric properties into functional interfaces;
3. The extraction of functional designation of components based on the functional properties of their conventional interface.

Despite the key importance of the first phase as a prerequisite to the second phase, and the third phase

as a final stage to obtain the sought functional designations, the core interest of our approach falls in the second one, where interface characteristics should be inferred in an efficient manner with a very small margin of error tolerated.

#### 4.7. Geometric interactions between components

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

We favour information offered by geometric interactions over mere geometric and topological properties of isolated components, and throughout the work we advocate the merit of this preference.

Geometric interactions are described by their interaction types and their interaction zones.

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

##### Contact

A contact between two solids defines one or more shared surface or shared curve, without any shared volume (see Figure 6).

The interaction zone of a contact is defined by the 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 two components touch each other.

Contacts provide very valuable information to our reasoning, as they usually help defining support points where forces can be transmitted. At the same time they work as motion barriers enabling the deduction of kinematic properties.

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

##### Interference

An interference between two solids defines a shared volume between them (see Figure 6).

The geometric zone of an interference is defined by the shared volume it creates.

Obviously, an interference is a non-realistic representation in the sense that two solids interfering in an assembly don't represent an overlapping

volume between the two corresponding components in a product, as this leads to physically impossible configurations. Nevertheless, interferences are often used to represent complex settings in a simpler manner. For instance, threaded links are most frequently represented as cylindrical interference volumes.

Due to its idealized nature, interferences are harder to interpret than contacts, however, they also provide worthy information to the process of reasoning.

##### Clearance

A clearance occurs when the minimal distance between two surfaces of two different solids is less than a defined threshold and conveys a functional meaning (see Figure 6).

The interaction zone of a clearance is the set of surfaces of each of the solids for which the minimal distance is smaller than the threshold.

Clearances are realistic representations in the sense that a clearance in the geometric model represents a guarding distance between the two corresponding components in the real product, though the accuracy of the distance may slightly vary.

Clearances are subjected to a parameter that is the play, this parameter varies depending on many factors like the overall size of the product and the accuracy of the design and the manufacturing tools. This makes the study of clearances quite perplexing. At the same time, clearances provide little information for the analysis process. For these reasons, clearance took minor attention while conducting our research. As a matter of fact, clearances were ignored when implementing our approach, considering the little advantage they bring compared to the overhead they entail.

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

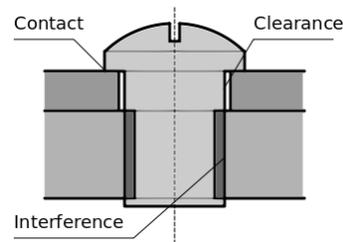


Figure 6 Demonstration of geometrical interactions.

#### 4.8. Reference states

We use a simple paradigm to reason about the nature of conventional interfaces based on solely their geometric properties. This paradigm suggests starting with a wide solution space dictated by the geometric model, then eliminating solutions that are unlikely to be encountered in a functional product. This unlikelihood envelops a solution that suggests non-physical configurations, or an assumption of either a design defect which is little tolerated in industry or an unjustified complexity or increase of cost.

An example of a non physical configuration is the assumption of friction-free contact leading to unbalanced forces for one of the underlying components. Whereas in the actual product, the contact is adherent, enabling the mechanical equilibrium of components. The non-physical friction-free proposal should be out-cast in favour of the realistic adherent contact solution.

An example of design defect is the solution that assumes a double contact, where two contacts have exactly the same direction, but are not produced with the same surfaces. Such a model is inefficient and hard to manufacture, because of inevitable machining error margins which it doesn't account for.

To the end of eliminating non-functional solutions, a set of criteria must be available to enable the judgement on their likelihood in an operational product. These criteria are grouped as sets of hypotheses that are assumed to hold truth all along the reasoning process. We refer to these sets of hypotheses as *reference states*.

We have so far recognized two of them; mechanical and kinematic reference states.

##### Mechanical reference state

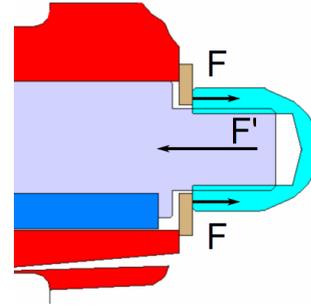
The mechanical reference state assumes that all components are rigid bodies, and that each component of the system in hand is at mechanical equilibrium; that is:

- 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 conventional interfaces of a component sum up to zero.

$$\sum_{c \in CI\{P\}} \{\vec{F}_c | \vec{M}_c\}_{/(\vec{0}, \vec{x}, \vec{y}, \vec{z})} = \{\vec{0} | \vec{0}\} \quad (1)$$

This is because conventional interfaces represent all the interactions of a component with its environment; thus, exhaustively incorporate all external forces and their moments.



**Figure 7** Zoomed cross section in the pump's assembly showing some geometric interactions.

Figure 7 shows a zoomed cross section in the centrifugal pump's model at the upper part of the shaft. Where planar contact between the nut and the washer can only be interpreted as planar support generating force  $\vec{F}$ , an opposite force  $\vec{F}'$  should be generated by the only other CI of the nut which is the cylindrical interference to enable mechanical equilibrium. This reasoning leads to elimination of a loose shaft connection interpretation of this CI.

##### Kinematic reference states

The kinematic reference states also adopt the rigid body assumption, however, it is based on rigid body 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 vectors reference to the same coordinate system and origin.

That can be otherwise stated as that the kinematic screws of all conventional interfaces forming a closed loop in the geometric model with respect to the same coordinate system and origin sum up to zero.

$$\sum_{c \in \{c_0, c_1, \dots, c_n\}} \{\vec{\Omega}_c | \vec{V}_c\}_{/(\vec{0}, \vec{x}, \vec{y}, \vec{z})} = \{\vec{0} | \vec{0}\} \quad (2)$$

This is because the relative motion of a rigid body with respect to itself is zero. By arbitrary choosing

one body of our loop, and then repeatedly applying Chasles equation starting by its first two neighbours, until the loop is closed, we conclude that the sum of relative motions (represented as kinematic screws) equals to the relative motion between the chosen object and itself, that is zero.

Besides assumptions made by references states, we also presume certain postulates that enable our reasoning.

### Model's consistency

Alongside the reasoning process, we consider the DMU, thus its geometric model to be consistent both from functional and conceptual points of view. Our design model should respect agreed-upon industrial standards, and provide a coherent pattern to enable the manufacturing of an operational product. This means that our product can be manufactured with available technologies. And that the final product won't fall apart.

This postulate, being central to our reasoning, allows for the forming of the following hypotheses among others:

- All pieces of the product are held tight together, which in turn leads to the mechanical reference state;
- A component with two parallel planar contacts that are not coplanar and share the same orientation indicates a design defect (a double contact situation);
- Unless justified by a functional kinematic chain, all internal motions in the model should reduce to only rotations. Unjustified translations signal a design defect.

### Time invariance

As an observation of industrial models and their kinematic behaviour, conventional interfaces are assumed to have global geometric properties that are invariant over time. That is despite the relative motion between two components; their geometric interaction (if any exists) maintains its nature with the course of time. However, the interaction zones may still change without leading to the rupture of a contact, release of an interference or break of a clearance.

This hypothesis emphasizes the importance of geometric interactions as not only a matter of volatile configurations, and allows the reasoning on those interactions to safely lead to permanent results.

## 4.9. Bottom-up approach

Our reasoning follows the bottom-up approach in that we start with a component at a time and study its conventional interfaces by going back to reference states and making our conclusions; which suggest a number of solutions that are consistent from the very local standpoint. Once reasoning is done at the level of individual entities, we take the results from there and move on to a larger perspective, taking into account neighbouring entities and their conventional interfaces, and checking our conclusions again against the reference states to refine them. This is done by eliminating solutions that became invalid in the way; when the system is looked at from a broader angle. We keep on going until the system as such is checked for consistency according to all reference states.

## 5. DESIGN AND DEVELOPMENT

As seen before, the work is divided into three major tasks, identification, interpretation, and matching. In this section we will address the technical details of each of these steps.

### 5.1. Identification

In this phase we undertake a pure geometric analysis to our model, in order to identify adjacent solids and define their geometric relationships in the frame of conventional interfaces.

The geometric interactions between components are then organized in a graph data structure called the *conventional interface graph (CIG)*, whose nodes are the model components, and whose edges are the conventional interfaces wrapping the geometric interactions.

We adopt the STEP file format [11][12] as a standardized representation of our geometric model. Although ISO 10303 has proposed notations to encompass functional and other semantic information in STEP, we consider our model to be purely geometric for the time being, ignoring other information, if any, since this information is neither reliable nor accurate because it is user defined.

We build upon Open CASCADE Technology (<http://www.opencascade.org>) software development platform to enable our geometric analysis.

We use a simple, yet efficient, approach that allows for the identification of most of the interactions that matter to our inference in later stages. In this

approach only *canonical surfaces* are considered; that is planar, cylindrical, conic, toroidal, and spherical surfaces. Those surfaces have high potentials to provide vital information that can be easily reasoned upon.

This approximation leads to an order of magnitude decrease in processing time when calculating geometric interactions, compared to traditional approaches, capable of handling free-form surfaces, like Boolean operations.

This simplification is not only justified by the radical boost in performance it brings, but also by the fact that most of our functional interfaces are indeed based on canonical surfaces. This is due to first manufacturing reason where canonical surfaces are easier to machine; and second, representation reasons, where free-form geometric details are avoided in a DMU. Observations show that though geometric interactions (mainly contacts and interferences) may occur between free-form surfaces, this kind of interactions are often irrelevant to our reasoning process. This makes the tradeoff worthwhile and the amount of loss in information insignificant.

Another simplification is the extensive use of enhanced bounding boxes that work as voxels enveloping the geometric entities. Simple bounding boxes are used to decide topological properties of primitive faces, particularly, their connectivity. While more complex structure of mutually disconnected bounding boxes are used to encapsulate maximal surfaces, allowing to more precisely represent discontinuous geometries.

### Maximal surfaces B-REP

The first step of this phase is the unification of representation. STEP represents the geometric model in a Boundary Representation (B-REP) format. Unfortunately, a B-REP encoding of a geometric object is not a unique one. That is; two STEP files may represent the same geometric configurations differently. This is due to the fact that one 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 modeling process.

A unified presentation is not only necessary for the sake of robustness, but also for efficiency

considerations. This is because the unified model with maximal surfaces contains less geometric entities than the original one, leading to a faster processing of the model.

To obtain the sought unified representation of one solid, we merge adjacent faces that belong to the same canonical surface into one entity; a maximal face. A maximal face is represented by its underlying oriented surface, along with a compound bounding volumes structure called multiple bounding boxes that envelops the original face with disconnected boxes parallel to the coordinate system unit vectors.

We call this unified representation the *maximal faces representation*. Though simple, it serves to generate geometric interactions between solids, specifically contacts and interferences in later stages.

### Geometric analysis

To estimate objects adjacency, we use simple bounding boxes to filter our pairs of objects that are unlikely to interact. The remaining pairs are then checked for geometric interactions.

For each surviving pair, maximal faces of one of the two objects 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.

For instance; two cylindrical surfaces with opposite orientations that share the same axis of cylinder and the same radius indicate a cylindrical contact. When the two radii differ, and if the normal of the surface with the larger radius is oriented outwards the inside of the cylinder, a cylindrical interference is reported in a first place. The case where the normal of the surface with the smaller radius is oriented outwards the inside indicates a cylindrical clearance if the difference between the two radii doesn't exceed the clearance distance threshold.

We call each configuration of cylindrical contact, cylindrical interference, and cylindrical clearance a *geometric interaction descriptor*. A geometric interaction descriptor is a well-defined unambiguous denomination of a geometric interaction. Other examples include planar contact, linear contact, circular contact, etc.

Each identified geometric interaction, labeled with its descriptor, is then encapsulated into a conventional interface connecting two components, that will later be attributed other inferred properties. The result is

then structured in the CIG, as the output of the identification phase.

## 5.2. Interpretation

After identifying the interactions between components in the 3D space, they are interpreted to induce their mechanical, kinematic, and functional signification.

As mentioned in the previous section, the assembly processing follows a simple bottom up approach in which we first associate each conventional interface with all the possible interpretations it may hold. Those interpretations are suggested by its geometric interaction properties. To this end a thesaurus has been set up, that provides those suggestions. The thesaurus is organized in a hierarchical structure according to the level of details of the geometric description of the interaction. The very first level under the root consists of three categories: contact, interference, and clearance. The leaves of the hierarchy bijectively map to all possible geometric interaction descriptors. Each of those leaves is associated with all possible functional interpretations that can be represented this way in the industry. We call this hierarchical structure *the taxonomy of conventional interfaces*.

The interpretation phase starts where the previous one ended, that is with the CIG. The first step in this phase is to match each conventional interface with its appropriate leaf in the taxonomy according to its geometric interaction descriptor. Once this is done, thanks to the bijective relation between leaves and descriptors, the interpretation suggested by the taxonomy is assigned to the underlying conventional interface as potential functional attributes.

Next, the reasoning process begins with the help of reference states postulates. As we have so far identified two distinguishable reference states, we have two, possibly overlapping, analyses to take place, namely, the mechanical analysis and the kinematic analysis. As stated before, the basic approach we follow here is the elimination of sets of interpretations of the conventional interfaces that are incompatible with either of the reference state postulates.

### Functional interface

Functional interpretations of a conventional interface 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 cinematic properties that allow the expected behavior. Examples are planar support, cylindrical support, pivot link, threaded link... etc.

In our approach, mechanical and cinematic properties are represented as screws, called mechanical and cinematic screws, respectively. 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, null, which is also regarded as a constraints.

### Mechanical analysis

Based on the mechanical reference state, this analysis highly relies on the mechanical equilibrium equation of a component (Eq. 1).

For each component this equation must hold truth; that is the screws representing all mechanical interactions exterior to the component being studied at all its conventional interfaces must sum up to zero. Considering that one conventional interface may be interpreted as more than one functional interface, thus have more than one possible mechanical screw, the analysis ends up with  $\prod_{i=1}^n |CI_i|$  different combinations; where  $n$  is the number of conventional interfaces 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, represented at a single point of the space, sum up to zero.

This study will reveal incoherent combinations (where a value is null and strictly positive at a time, for example). Those combinations are then suppressed, leading to the elimination of certain function interpretations of a  $CI$ ; thus, the reduction of  $|CI_i|$ .

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 interpretation 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 CIs assigned to it. The iterative process stops whenever all assembly components are checked.

The case where  $|CI_i|$  evaluates to zero at some point of our reasoning signals an incoherence. This means that none of the suggestions proposed by the geometry adhere to the reference state.

### 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 are stationary to each others.

This knowledge, along with the respective motion between one kinematic class and another, enables the deduction of important information about the functional kinematic chains in n assembly. Such information is then used to reason about the functional designations of components.

As mentioned earlier, kinematic reference state is based on rigid body's kinematics. In contrary to the mechanical analysis, and instead of studying one component at a time, this phase 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.

An important shortcoming of a DMU is that it still misses the temporal aspect; that is, the 3D model represents reality at a given instant  $t$ , with no least information about how the product will look like shortly after. However, and for studying dynamics and kinematics this kind of knowledge is vital.

For this reason, a minimal user intervention is solicited, 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 exist or not. Kinematic properties assigned to one object propagate automatically to all its kinematic class members. 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 construction of functional interfaces and kinematic classes are done, those information are integrated all together to serve ultimate major goal of our research; the deduction of functional designation of components.

Functional interfaces and kinematic classes are translated into functional designation with the help of function designation ontology that describes mechanical and kinematic properties a specific component should acquire before belonging to the class of components identified by a specific designation.

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 and joining two adjacent components, the component should also have the same kinematic class as the two adjacent components.

As mentioned earlier, one component can only have on functional designation. However, our analysis may end up with more than one valid suggestion for the same object. In this case eliminatory criteria are needed to make the final call. One criterion could be to outcast functional designation with too many functions that are unnecessary for the product's operability.

## 6. RESULTS

In this section we briefly address the preliminary results we have obtained, knowing that our research is still at initial phases of implementation.

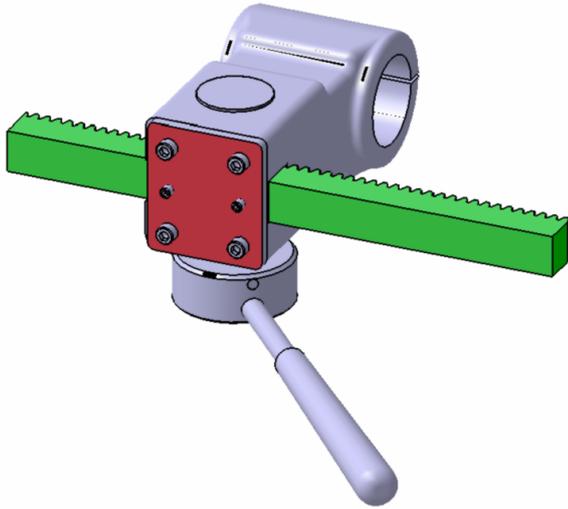
The implementation work was focused on the geometric analysis so far, to extract geometrical interactions of the assembly solid model in a timely manner.

To validate our results, we use a model of centrifugal pump that contains most of the geometric interactions we are concerned about (see Figure 2).

We also use a simpler model to compare our work to other methods only capable of handling relatively small models. This is the model of a drill support (see Figure 8).

Our algorithm is capable of detecting a subset of geometric interactions that we are interested in, this subset is easily extensible when new requirements

emerge. For the time being, the algorithm extracts accurately cylindrical, planar, linear and circular contacts, and cylindrical interferences. Those interactions provide a solid ground for the mechanical and kinematic study.



**Figure 8** Drill support model.

One advantage of such approach, besides the remarkable drop in execution time, is that interaction properties, such as axes and normal, which are important to later steps of inference, are seamlessly obtained. In contrary with Boolean operations that require further study of the obtained interaction zone to determine such properties.

In the following tables we show execution times for geometric analysis; that is the extraction of geometric interactions, but not the time taken to load the geometric model, as it is out of the scope of our work and completely independent of geometric interaction detection approach.

Table 1 compares the performance of our approach against different models of different complexities.

Example	Nb. Solids	Time in ms.	Nb. Contacts	Nb. Interf.
Drill Support	20	110	12	12
Centrifugal Pump	43	340	102	17

**Table 1** Execution time for different models.

Table 2 compares the performance of our approach against the basic Boolean operators algorithms provided by Open CASCADE, and augmented with basic bounding boxes early elimination technique. It has to be noticed that it is

applied to the drill support only since Open CASCADE operators failed to give a result on the centrifugal pump.

Algorithm	Time in ms.
Open CASCADE B.O.	82560
Our Approach	110

**Table 2** Execution time for different approaches applied to the drill support.

The remarkable drop in execution time is due to the avoidance of complicated accurate geometric computations and the exploitation of enhanced bounding volumes techniques and simple comparisons of geometric properties instead. This simplification leads to less precise information about the geometry of the interaction zone. However, and to fulfill the requirements of reasoning in later stages of our research, the obtained information, precisely directions and orientations, are just enough, while detailed quantitative values are unnecessary.

## 7. CONCLUSIONS

This work is a preliminary step towards an automated identification of components functional designation in a DMU based on their pure geometric description. In this document, we emphasized the motivation of our work, and formulated the theoretical framework upon which we build our algorithms and data structures. We also showed initial results of the very first phases of our work to validate the efficiency of the proposed approach.

The integration of components neighborhood information in the inference process was particularly suggested, presenting the concept of conventional interfaces that defines the interaction between one component and its adjacent ones. Starting with mere geometry, and passing through different other information, such as mechanical and kinematic assessment, we finally guess significant functional properties of the interaction. These suggestions are backed by the strong relationships between geometric configurations and internal forces at one hand, and geometrical configurations and kinematic properties at the other.

The work done so far shows that the method proposed has significant potentials to enable a fairly automated procedure of identification. It also points out the merit of the efforts still being paid in this particular direction of research.

In the light of proposed framework, more algorithms and data structures are still to be elaborated to materialize addressed theoretical studies. At the same time, the theoretical framework itself will continuously be revised, benefitting of the feedback of the development work.

## ACKNOWLEDGMENTS

This work is carried out in the framework of the ANR project ROMMA and the authors thank the ANR for its financial support.

## 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 International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, Munich, Germany.
- [4] 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.
- [5] H. Zhu and C. H. Menq, (2002), “B-Rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition”, *Comput. Aided Des.*, 34(2), 109—123.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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*, volume 61(7), 686—701.
- [11] ISO 10303-204:2002. STEP Part 204: Application protocol: Mechanical design using boundary representation. ISO, Geneva, Switzerland, 2002.
- [12] ISO 10303-204:2003. STEP Part 42: Integrated generic resource: Geometric and topological representation. ISO, Geneva, Switzerland, 2003.