Multi-criteria retrieval of CAD assembly models

Katia Lupinetti [a,b,*], Franca Giannini [a], Marina Monti [a], Jean-Philippe Pernot [b]

[a] *Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes", Consiglio Nazionale delle Ricerche, Italy*
[b] *Arts et Métiers ParisTech, LSIS Laboratory, France*

Keywords:
Assembly retrieval
Multi-layered search
Partial retrieval
Assembly representation
Multi-modal descriptors
Signature extraction

A B S T R A C T

Being able to reuse existing design knowledge is of major interest to help designers during the creation of new products. This is true at the level of the parts and even more at the level of the assemblies of multiple parts. Meaningful information and knowledge can be extracted from existing geometric models and associated data and metadata, as well as from the processes followed to define them. This paper proposes a method to characterize and structure CAD assembly models to enable the retrieving of similar models from a database. A framework has been devised for the retrieval of globally and/or partially similar assembly models according to multiple user-specified search criteria. It is based on an assembly descriptor, called the Enriched Assembly Model, which is an attributed graph that encodes all the required data automatically extracted from the geometry and structure of the CAD models. The data are organized in four layers: structural, assembly interface, shape and statistic layers. Starting from a real CAD model or from an abstract query model, the algorithm retrieves models from the database by solving a matching problem. The matching between two assembly models is translated into the problem of finding a sub-isomorphism between two EAMs. The layered organization of the EAM allows partially defined queries, which can be further refined. The effectiveness of the proposed approach is illustrated with results obtained from the developed software prototype.

1. Introduction

CAD models have become mainstream in many industrial applications. They can be considered as digital product reference models stored within a Digital Mock-Up used and shared all along the Product Development Process (PDP) Falcidieno, Giannini, Léon, & Pernot, 2014. Most of the time, new products result from an adaptation of existing ones and from the combination of known technological solutions. Thus, having an easy access to models already designed and available from company databases is of major interest to rapidly prototype new products satisfying similar specifications and requirements. Being able to reuse existing information and data such as components, sub-systems, materials, process planning, manufacturing strategies, production costs, as well as the geometry of the 3D models (Yu-Shen, Fang, & Ramani, 2009; Zehtaban, Elazhary, & Roller, 2016), becomes a crucial differentiating factor for the industries whose competitiveness is driven by the well-known triptych cost-quality-delay. The ability to retrieve existing models, either parts or assemblies, can be useful to reach several objectives as reusing an existing assembly in new configurations, or providing access to existing design knowledge (e.g. simulation results, manufacturing strategies) related to similar products (Gupta, Cardone, & Deshmukh, 2006) or to identify similar configurations that could benefit from a standardization.

In this context, one challenge lies in the fact that the size of the databases has grew up exponentially in the last few years. Thus, it is increasingly challenging to handle the large amount of produced data and to develop new searching, browsing methods and tools (Roj, 2014). This is notably true for what concern the structuring, the access and the reuse of CAD model databases.

A retrieval process is not straightforward when considering CAD models potentially made of several hundreds of thousands of parts. Thus, specific systems with proper search algorithms have to be developed and optimized to be able to retrieve elements in a user-friendly way and within a reasonable time.

In simple text-based retrieval, users type a list of words or sentences that characterize the objects they want to find in the database. Such a strategy assumes that the data have been annotated in a proper way with textual information. Thus, some models may be not retrieved, because they have not the same text in their

Peer review under responsibility of Society for Computational Design and Engineering.

* Corresponding author at: Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes", Consiglio Nazionale delle Ricerche, Italy.

E-mail addresses: katia.lupinetti@ge.imati.cnr.it (K. Lupinetti), franca.giannini@ge.imati.cnr.it (F. Giannini), marina.monti@ge.imati.cnr.it (M. Monti), jean-philippe.pernot@ensam.eu (J.-P. Pernot).

annotations even if they are semantically related to the query. To overcome these limitations, search methods based on thesauri, i.e. collections of controlled vocabulary terms that use associative relationships, can be adopted. However, these techniques are not sufficient since annotations may not be present and there is no guarantee of compliance to name conventions. Moreover, they do not consider the shapes of the parts.

Similarly, in case of complex products made of several parts, a method based only on the shape is not sufficient for retrieving the target assembly model. Actually, 3D models with similar shapes can be assembled in different ways, involving different kinematic joints and then different relationships between their parts. For example, an assembly of two parts with 5 screws can be considered similar to the same assembly with 6 screws. In this case, at a higher level, what is important is that the two parts have been screwed whatever the number of screws. Thus, an advanced search method has to incorporate mechanisms working at different levels (e.g. geometry, structure, kinematic, annotation).

For the effective re-use of existing models, content-based methods should also allow queries without the specification of a CAD model as input. This possibility is particularly important since at the early design stage, the designer can be interested in expressing incomplete queries, e.g. simply by specifying some attributes of the assembly model, just to take inspiration from the available models. Therefore, the challenge is to find an assembly representation able to support user requests at different level of details. In addition, associated data and metadata should be automatically extracted to avoid tedious manual instantiations.

Therefore, it is crucial to provide a tool for the retrieval of assembly models, which can be tailored to the user needs. It should be able to consider multiple criteria related to the assembly and the part shapes, the interlinks between sub-assemblies and parts and other aspects that are implicitly stored in the 3D data.

In this paper, we propose a framework for the retrieval of globally and/or partially similar assembly models according to different search criteria that can be convenient for designers. It is based on an assembly descriptor, called the Enriched Assembly Model (EAM), which encodes all the required data automatically extracted by analyzing the geometry and structure of the CAD model. The matching between two assembly models is translated into the problem of finding a sub-isomorphism between two EAMs. It allows partially specified queries, which can be further refined and applied on search results. Section 2 reviews the related works. The EAM model and the complete framework are introduced in Section 3. Section 4 discusses some examples obtained from our prototype software and which consider either a real CAD model or an abstract query model as input of the retrieval process. The last section ends this paper with some conclusions and perspectives.

2. Related works

Researchers in computer graphics have largely addressed the problem of single three-dimensional object retrieval (Biasotti et al., 2003; Funkhouser et al., 2003; Giannini, Lupinetti, & Monti, 2017). To this aim, various shape descriptors have been defined and quite large overviews are provided by Iyer, Jayanti, Lou, Kalyanaraman, and Ramani (2005) and by Tangelder and Veltkamp (2008). These techniques focus on shape characteristics and do not take into account some other aspects, which are important when considering the similarity of assembly models. For example, the relationships between the parts are not used and it is therefore not possible to retrieve similar assembly structures or more simply a sub-assembly in another assembly.

To overcome these limits, more recently, efforts have been devoted to address the retrieval of assemblies. To deeper analyze the techniques, directly or indirectly addressing the identification of similarities in assembly models, we identified several criteria grouped into the following four macro-categories: *Context*, *Assembly characterization, Assembly descriptor*, *Query model*.

The *Context* includes the objectives of the work and the type of geometric representation (i.e. B-Rep, 3D mesh or point cloud) used to represent the assemblies. The *Assembly characterization* refers both to the type of data and knowledge (i.e. geometric and/or topological characteristics) the authors use to typify the assembly model and to the way the information concerning the assembly relationships are obtained. More specifically, it indicates if the method assumes that the relationships between the components in the assembly are explicitly represented in the native CAD models, automatically derived from the assembly geometry, or manually specified by the user. The *Assembly descriptor* indicates at what level (assembly, part, or feature) the assembly is characterized. At the assembly level, an assembly is described by its parts and their relationships; at the part level an assembly is described only through the list of its constituting parts; and at the feature level, shape portions having specific assembly meaning are used to characterize an assembly. Moreover, this last category includes how the assembly descriptors are represented, e.g. as graph or vector. Finally, the *Query model* category specifies how the query is expressed: e.g., a single CAD assembly model, a set of CAD assembly models, a set of CAD positioned part models, a 3D mesh, or an abstract assembly descriptor.

Table 1 gathers the existing approaches and their positioning according to the above criteria. Only the methods that directly aim at the retrieval of similar assembly models are considered. Among them, Li, Zhou, Liu, Niu, and Kong (2016), Kazhdan, Funkhouser, and Rusinkiewicz (2003) stresses the importance of CAD retrieval to reuse existing solutions. Their method is applicable both to parts and to assembly models to find global as well as partial similarities. Anyhow, in case of assembly models, the retrieval system is strongly dependent on the structure defined by the designer, not recognizing as similar the same model but with a different sub-assembly structure. Other works that adopt a more comprehensive approach are (Chen, Gao, Guo, & Bai, 2012; Deshmukh, Banerjee, Gupta, & Sriram, 2008). Deshmukh et al. take into account many different aspects that play a meaningful role in the description of an assembly model (Deshmukh et al., 2008). The approach offers the possibility of using vague incomplete queries. Its main limit is in the required availability of several important information (such as the component orientation, component relationships and the joint constraints). Chen et al. (2012) propose a global approach, which aims to overcome this limitation. This work focuses on the product structure and on the relationships between the different parts of the assembly. The assembly descriptor presented in this work takes into account different information levels. It includes topological structure, relationships between assembly components, and geometric information. It also permits the use of rough and partial incomplete queries to allow a search adaptable to the designer requirements. Similarly to the ones presented in Chen et al. (2012), Deshmukh et al. (2008), our framework is based on an assembly model able to support user requests at different level of specification details. Differently from Deshmukh et al. (2008), it does not require the user to add manually some information. Differently from Chen et al. (2012), the mapping algorithm is not limited to the identification of assembly models with the same structure in terms of sub-assemblies. Moreover, our framework does not rely on a specific CAD native file format. It requires as input a STEP file describing the assembly model in which only geometric information are available, and allows the retrieval of assembly models, which are similar to the query model or contain a subset similar to it, or vice versa.

Table 1
Summary of works directly addressing assembly model retrieval.

Paper	Context		Assembly characterization		Assembly descriptor		Query model
	Input model	Work objective	Characterizing information	Availability of assembly relationships	Assembly descriptor level	Descriptor representation	Query model
Zhang, Xu, Li, Jiang, and Wei (2013)	B-Rep	Search for frequent similar sub-assembly models	Curvature, Model components, Mating constraints	Explicit in the CAD model	Assembly Part	Face Adjacency Graph	Assembly model
Hu, Wang, Yong, and Paul (2013)	3D mesh	Search for globally similar assembly models	Model components	NA	Part	Component Vector	Assembly model
Chen et al. (2012)	B-Rep	Search for globally or partially similar assembly models	Model components Mating constraints Degree of freedom Annotation	Partially extracted	Assembly Part	Hierarchical graph	Assembly model
Tao and Huang (2012)	B-Rep	Search for globally similar assembly models	Model components Surface properties Contact relations	Manually inserted	Assembly Part	Component attributed relation graph	Assembly model
Miura and Kanai (2009)	B-Rep	Search for globally similar assembly models	Shape characteristics Mating constraints	Explicit in the CAD model	Assembly Part	Attributed Assembly graph	Assembly model
Deshmukh et al. (2008)	B-Rep	Search for globally or partially similar assembly models	Model components Mating constraints Degree of freedom Annotation	Explicit in the CAD model	Assembly Part	Mating graph	Set of statistics Mating graph
Wang, Li, Zhang, and Yu (2016)	B-Rep	Search for globally similar assembly models	Shape characteristics	NA	Part	Component Vector	Set of part models
Li et al. (2016)	B-Rep	Search for globally or partially similar assembly models	Shape characteristics Kinematic equivalence	Explicit in the CAD model	Assembly Part	Hierarchical graph	Assembly model

To sum up, the retrieval of CAD parts is deeply studied and efficient solutions are proposed. However, quite few works have been suggested for the retrieval of CAD assemblies. As it can be understood from Table 1, most of the proposed techniques rely on high-level information that are not always available from the assembly models. The proposed solution aims at devising a search framework that includes methods able to provide the automatic extraction of important information allowing the retrieval of assemblies satisfying specific shape, parts' arrangement and/or mechanism characteristics.

3. The retrieval framework

CAD systems use different and proprietary file formats to store all the information specified by the user when modeling specific parts. The files content strongly relies on the type of functionalities provided by the specific CAD system, therefore building a generic retrieval system cannot trust on the presence of data that would be too specific to a CAD system. To overcome this limit, neutral file formats are generally used for the CAD data exchange. Thus, in our framework, we adopted the STEP standard format (ISO 10303-203 and ISO 10303-214) as representation format of the assembly models and to get access to the associated information. Theoretically, this standard supports the representation and exchange of assembly models including the kinematic relationships between their components and their constraints. However, most of CAD systems do not contain the latter ones and generate files that do not incorporate the kinematic relationships and constraints. Similarly, other information used all along the PDP may be not stored, or can be inaccurate due to their human nature. Consequently, our approach and the associated methods only rely on geometric data as well as on the hierarchical assembly structure of the CAD models.

The proposed framework is based on the so-called Enriched Assembly Model (EAM). An EAM is an assembly descriptor, which encodes all the required data automatically extracted from the geometry and structure of a CAD model. The problem is then to find in a database the CAD models that have an EAM similar to the one given as a query. Thus, the framework considers both real-time processes and batch processes to be executed in advance. The batch processes evaluate the EAM for all the models of the database. Real-time processes evaluate the EAM descriptor for the query and perform the comparison with those in the database. It must be noted that the EAM is a very rich model, including many information on the represented assembly. Some data are apparently redundant, but they offer the advantage to allow scalable queries. Therefore, a complete EAM version is computed only for the models of the database. For the query model, only the layers containing information assessable on the required detail are computed and exploited for the matching, thus reducing the complexity of the system.

3.1. The enriched assembly model descriptor

To allow an efficient and meaningful retrieval of CAD assembly models, a suitable description of the assembly should be provided. As previously described, an effective reuse of already existing models requires the possibility to perform search with queries using incomplete information and highlighting valuable characteristics, which might be not explicitly encoded in the CAD models. To this aim, the so-called Enriched Assembly Model descriptor is proposed. It organizes information in a multi-layer structure aimed at guaranteeing the search flexibility (Lupinetti, Giannini, Monti, & Pernot, 2016). It also permits search queries involving one or multiple criteria based on the characteristics of the different layers and combined in different ways. The considered layers are: structural layer, interface layer, shape layer and statistic layer. These layers are described in the following sub-sections, while Section 3.3.1 illustrates how the data stored in the EAM are extracted from the assembly CAD representation.

3.1.1. Structural layer

The structural layer of the EAM encodes the hierarchical assembly structure of the CAD model as specified by the designer. This partition, even if it is driven by some standard rules, is not unique and reveals the designer intents. For example, the assembly can be organized in a way that the forthcoming assembly simulation steps are eased, or it can be organized with respect to visualization

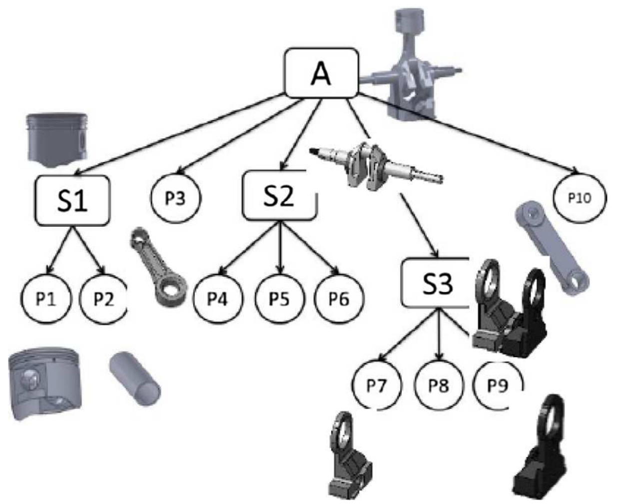


Fig. 1. Example of structural layer of an assembly model.

issues using an octree-based decomposition, or it can be decomposed according to criteria based on the constitutive materials. Such decomposition into parts, subassemblies or a single component resulting from the merging of all the parts and subassemblies also corresponds to the way designers may focus on a product. Actually, designers can consider the product either as a whole object with its characteristics (e.g. volume, gravity center), or focusing on subassemblies (e.g. for kinematics purposes) or parts (e.g. for manufacturing issues). Fig. 1 shows an example of the structural layer of an assembly model. The object is an engine formed by three sub-assemblies: a piston, a crank shaft and a mass (S1, S2, S3) and two linking parts (P3, P10).

The structure information is stored as a tree graph in which arcs indicate the relation part-of and nodes correspond to the assembly components. In particular, leaves represent the parts of the assembly model, the root the entire assembly model, while intermediate nodes represent sub-assemblies of the original model. Attributes are then associated to leaf nodes to indicate the type of the corresponding component: axis, bearing, c-clip, cylinder like, cube like, gear, key, linkage arm, nut, part of bearing, screw and bolt, spacer, sphere like, torus like and miscellaneous. Here, the rational is to classify parts in classes useful to distinguish elements likely corresponding to fasteners (e.g. screws, bolts and nuts) from others characterizing elements (e.g. bearings and gears).

Again, it is important to notice that the assembly decomposition depends on the context and, even if it represents a semantic organization, it is not unique. Therefore, assembly similarity cannot strongly require same structures if not specifically requested by the user. However, we believe that there is a level of decomposition under which multiple decompositions of a same product will remain similar, and would in this case correspond to the smallest common denominators. Such an understanding can be performed at the level of the parts and subassemblies, but also at the level of the joints between the elements constituting the global assembly. This point is not further discussed in this paper. Finally, all the elements in the tree are linked to data in the other information layers to fully characterize them.

3.1.2. Interface layer

This layer encodes the relationships between the different parts in an assembly model regardless the assembly structure. Therefore, it consists of arcs linking leaf nodes. Links between sub-assemblies and the rest of the assembly components can be obtained by simply considering the arcs that do not link components of the same sub-assembly.

The possible relationships between two parts can be grouped into contact, interference and clearance (Roj, 2014; Shahwan, Foucault, Léon, & Fine, 2014) as shown in Fig. 2.

Two parts are in contact, if they touch along low-level geometric entities such as surfaces, curves or points without any shared volume.

Two parts define an interference, if a common volume exists between them. Most of the time, this configuration does not exist between two real objects. However, such an unreal configuration can appear during the modeling phase, e.g. when considering idealized and simplified parts for simulation purposes, or when checking the interferences at an intermediate design stage where dimensions are not fully tuned. Sometimes, such configurations simply result from modeling errors. However, some interferences can be desired and designed on purpose. This is what happens for elastic seal parts or shrink-fitted parts that are often modeled in their initial configurations, i.e. before their deformation and assembly. Since interferences should not exist in a correct assembly model, they are not included in the EAM and, if present, are treated as contact.

Clearance occurs when the distance between two surfaces of two parts is meaningful for the considered assembly, i.e. it is a small non-null distance between two parts in the assembly. This case is rather ambiguous since the design intent can correspond to both non-contact and contact configurations (i.e. due to tolerance issues) and therefore, is currently not treated in our system.

The interface layer is itself a multilayered one (Lupinetti, Giannini, Monti, & Pernot, 2016). Interfaces and contacts are further detailed in terms of types of kinematic pairs between parts, i.e. number and types of geometric elements, which are mating. At the lowest level, there are the geometric entities (points, curves or surfaces) in contact between two parts. This information is stored as attributes in the previously introduced tree graph. In particular, for each contact, an attributed arc is inserted. The attributes specify the type of the involved entities and, in case of face contact, the corresponding degree of freedom (DOF) between the linked parts of the assembly. Table 2 shows the assigned DOF according to the surface type associated with the face, where R indicates a

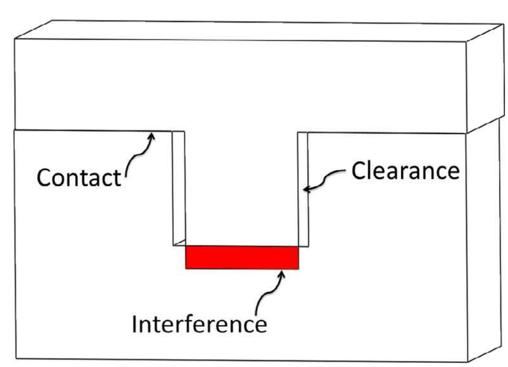


Fig. 2. Possible relationships between parts.

Table 2
DOF values according to the surface type.

Type	Parameters	DOF
Planar	n normal	R_n, T_u and T_v, where u and v are orthogonal to n
Cylindrical	u axis O origin	R_{u+O} and T_u
Conical	u axis O origin	R_{u+O}
Spherical	O origin	R_{u+o}, R_{v+o} and R_{n+o}
Toroidal	u axis O origin	R_{u+O}

rotation, T a translation, the subscripts u, v and n the vector along which the rotations/translations are allowed.

The contacts between two parts contribute to define the relative movements using the theory of mechanisms. For this computation, we assume that contacts do not change over the time and are preserved during the motion of the parts. Only the portion of the involved surfaces may change. No new contact can appear and no contact can disappear. This assumption is too restrictive to allow the treatment of some particular mechanisms, which will not be considered in this paper.

Under the above assumption, we classify the contacts and interfaces in two types, i.e. positioning and interlocking (Chan & Tan, 2003), which depend on the relative motions between the two parts. In particular, interlocking configurations are those that prevent any movement, i.e. with zero DOF. Fig. 3a shows two simple parts with a positioning interface. Indeed, there exists a direction along which the parts can translate while preserving their contacts. On the contrary, the parts in Fig. 3b cannot be moved in any direction without losing their contacts, thus it corresponds to an interlocking configuration.

3.1.3. Shape layer

High-level information, as kinematic interferences or semantic knowledge, are very efficient to describe assembly models. However, shape information is also important to discriminate among assemblies. Thus, in the proposed approach, the specification of the information related to the shape of the assembly components are included in the so-called shape layer.

Useful for comparison, this layer can also be interesting for the visualization. In particular, for each node of the structural layer (i.e. for both parts and sub-assemblies), two mesh representations are associated together with several shape descriptors. The first mesh is a rather precise model, whereas the second corresponds to its rough representation. On the one hand, these double representations allow an adapted visualization. On the other hand, they enable both a fast browsing of the objects and of the search results as well as the specification of both precise and rough queries with imprecise shapes. This is useful during the PDP, when the shape is being defined, and it is quite reasonable to search for possibly reusable products that share the main behavioral (e.g. degree of freedom) and overall shape characteristics.

As previously said, many shape-descriptors have been defined to compare geometric models. Iyer et al. highlighted that there is no a unique shape descriptors which suits all the possible shapes and comparison purposes (Iyer et al., 2005). On the contrary, depending on the type of object a specific shape descriptor can perform better than the others. For instances, shape distribution is not able to differentiate models with complex shapes, while the spherical harmonics are robust to compare solids of revolution.

In our framework, we consider descriptors related to both the overall component and its shape variation. Among the descriptors of the first type, we consider the volume and the surface area, which are size dependent and are appropriate for parts replacement and directly computable from the B-rep data. The other shape descriptors are the spherical harmonics and shape distribution, which perform well with prismatic parts and shapes of revolution of which are mostly composed the mechanical products we are considering (Iyer et al., 2005).

3.1.4. Statistic layer

To ease the filtering of large datasets, the statistic layer includes some numerical values which synthetize some of the data stored in the previous layers.

Statistics referring to the overall assembly or to sub-assembly nodes are the numbers of: (i) sub-assemblies, (ii) components of a specific type (e.g. axis, nuts, bolts), (iii) patterns of repeated components of a specific type (e.g. linear patterns, circular patterns). The inclusion of information related to the presence of patterns can support the search of similar models having similar manufacturing or assembly processes. Let us suppose the user is looking for a rolling bearing (Fig. 4a) in a dataset. This type of model is characterized by the presence of repeated balls arranged in a circular pattern. Using the statistic of the assembly, we are able to discard easily and rapidly models as the one illustrated in Fig. 4b, which does not own this characteristic.

Statistics also refer to the parts themselves and include: (i) percentages of specific type surfaces (i.e. planar, cylindrical, spherical, free form, toroidal) with respect to the overall area, (ii) numbers of maximal faces (i.e. adjacent faces sharing the same underlying surface characteristics are considered as a single face) of a specific type surface (i.e. planar, cylindrical, spherical, free form, toroidal). The use of such percentages allows for discarding directly the comparison between two shapes without the use of heavier descriptors, thus relieving the matching process. The number of maximal faces is used for more advanced searches for which classical shape descriptors difficultly achieve good results. Let us suppose the designer is looking for a model, whose overall shape is similar to the one depicted in Fig. 5a and which does not contain any fillets. This query can be translated in the fact that the percentage of cylindrical and freeform surfaces should be very small, or even null. As a result, even if their overall shapes are very similar, the model of Fig. 5b will be rapidly discarded since it has a percentage higher than the one in Fig. 5a.

Of course, based on this first list of statistics, additional descriptors can be used to enrich the characterization of the assembly model at different levels of the structural layer.

3.2. The framework architecture

The architecture of the framework is illustrated in Fig. 6. It shows the different modules (rectangles) as well as the way they communicate at the different levels (arrows). There are three main levels:

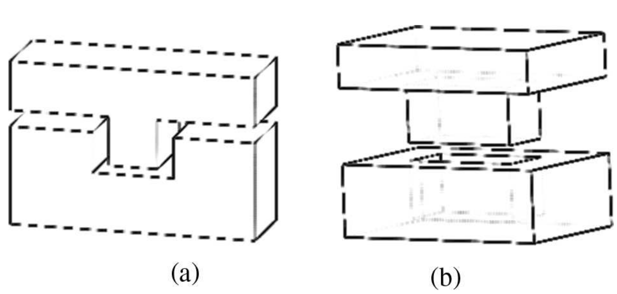


Fig. 3. Examples of positioning (a) and interlocking (b) configurations.

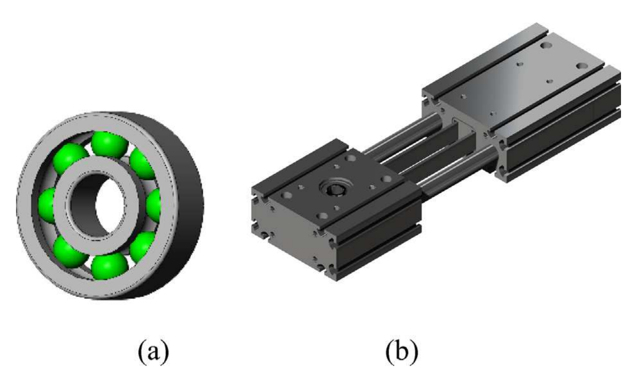


Fig. 4. Examples of assemblies with (a) and without (b) circular pattern.

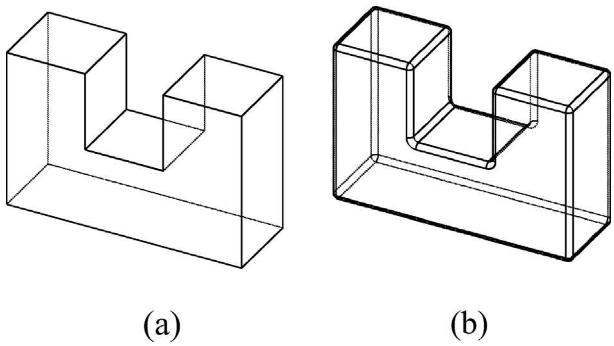


Fig. 5. Same part model with sharp edges (a) or fillets (b).

- The *user interface level* is responsible for the visualization aspects and for the handling of the user interaction activities. It provides a support for: (i) the visualization of the EAM; (ii) the search query specification; (iii) the browsing of the search results. Thanks to a dedicated graphic user interface, the designer easily specifies his/her queries, search criteria, filters for the visualization of the EAM and the dataset on which to perform the matching process. The EAM visualization module allows picturing an assembly model by means of a graph structure, whose nodes can be selected to enquire the content at the different levels of details. The geometry of the overall assembly, as well as the one of the intermediate nodes and leaves, can be previewed, thanks to the meshes stored at the shape layer. Finally, it allows the browsing and analysis and the matching results presenting the correspondences found and the degree of similarities achieved.
- The *functional level* contains the main modules of the framework, which are detailed in the next subsections. The creation module oversees the creation of the EAMs from the CAD assemblies contained in the original database, as well as the storing of those enriched models within a dedicated database. At this stage, explicit and implicit information of the CAD models are made available. Moreover, this level deals with the creation of the EAM query model using either an existing model (i.e. a STEP file) or an abstract query model specified by the user through the user interface layer. Finally, this level incorporates the matching module used to compare the EAM query model with the EAMs in the database.
- The *data level* provides the input for the functional level and contains the elaborated data.

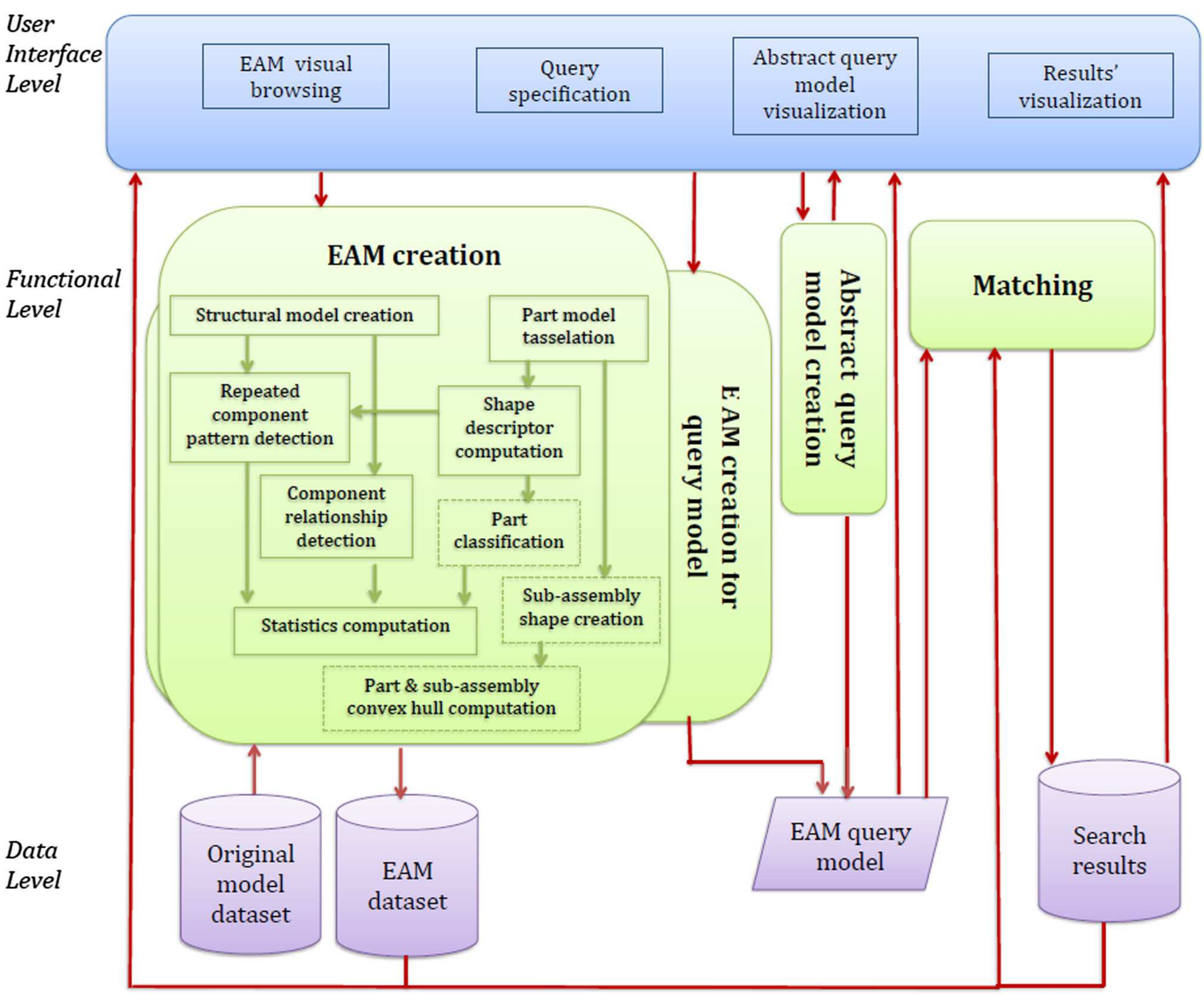


Fig. 6. The framework architecture.

From the operational point of view, it is necessary first processing the database in which models similar to the query should be found. Each assembly is processed by the "EAM creation" module. In this phase, a file is created where all the extracted information is archived. Then, two search scenarios can be possible. If the user wishes to retrieve all the models similar to an existing CAD model, the "EAM creation" module generates the corresponding EAM for the query model. Conversely, if the user is interested in finding CAD models having specific characteristics, the "abstract query model creation" module creates the corresponding EAM. In this case, the user can interactively specify a graph indicating several attributes, such as a rough component shape. The user can indicate if a node is a part or a subassembly and can also assign its attributes. The same happens for the arcs, which can express parental or contact relationships. Finally, when the dataset is entirely processed and the user has specified his/her own search criteria, the search process is managed by the "matching" module. This module communicates with the visualization module to display the results and associated matching scores.

3.3. Architecture module descriptions

This section details the modules of the functional level which can be considered as the core of our similarity detection framework. The EAM creation module allows the creation of the EAMs for all the CAD models of the original database (Section 3.3.1). It can also be used to define an EAM for a query model defined with a CAD model. If an abstract query model is to be preferred, a dedicated module is used (Section 3.3.2). EAMs are compared using the matching module (Section 3.3.3). Since the system has to manage assembly similarities according to multiple criteria, the matching procedure has to be as scalable as possible.

3.3.1. EAM creation

Since a STEP file does not contain all the information required to create an EAM, several geometric reasoning processes have to be performed on the assembly model to extract the desired information and generate the assembly descriptors.

The creation of the EAM results from the composition of several functions. Some of them can run in parallel, while others need the output of previous computations. The dependences of all the processes are illustrated in Fig.6.

The process starts with the reading of a STEP file. Nodes and arcs of the EAM structural layer are created by the ***structural model creation*** function. Later, for each created leaf, the part statistics are computed through the function ***part statistics computation*** and, at the same time, the function ***component relationship detection*** runs. Once the part statistics have been computed, patterns of repeated components are detected using the ***repeated component pattern detection*** module described in Lupinetti, Chiang, Giannini, Monti, and Pernot (2017). This module identifies linear translation (Fig. 7a), circular translation (Fig. 7b), circular rotation (Fig. 7c) and reflective (Fig. 7d) arrangements of parts. Generally, repeated components are explicitly indicated in the STEP file as multiple occurrences of the same part. When not explicitly revealed, components are considered repeated when presenting the same values for the surface area, volume and the associated statistics (i.e. number of faces of a specific type and related area percentage). Of course, such criteria do not fully characterize repeated components but represent necessary and easy check conditions to identify them.

The interface layer is created by the function ***component relationship detection*** by analyzing the relationships between components. This kind of information is not stored in the STEP file. Therefore, the detection of the possible part interactions require a reasoning on the geometric data available in the STEP file. We use functionalities provided by the API of the commercial system SolidWorks© for the detection of interferences and access to the faces (or edges or vertices) involved. The function for detecting and evaluating the relationships between parts includes the following steps:

(i) *Detection of interferences*. According to the description provided in the interface layer, we retain the intersections and the contacts between assembly components.
(ii) *Identification of parts in contact*. We identify the involved parts in each contact and volumetric interferences due to tolerances for which we can deduce the right configuration, as for example the intersection between a spherical surface and planar one, which is then treated as a punctual contact.
(iii) *Identification of contacts between parts*. For each pair of parts in contact, we compute the non-regularized intersection, i.e. overlapping portion.
(iv) *Identification of kinematic pairs*. Kinematic pairs constrain the relative motions of two parts. The analysis of the typology of the elements involved in non-regularized intersections allows the identification of kinematic pairs. For example, if a planar/cylindrical face is involved in a non-regularized intersection, it indicates that the two parts are in contact through two planar/cylindrical faces. In case of points and curves involved in a non-regularized intersection, additional geometric verifications are needed to identify the type of the faces which originate the contact. This additional checking is not included in the work presented in this paper.
(v) *Relative motion computation*. According to the theory of mechanisms, the association of several kinematic pairs defines the degrees of freedom between two parts.

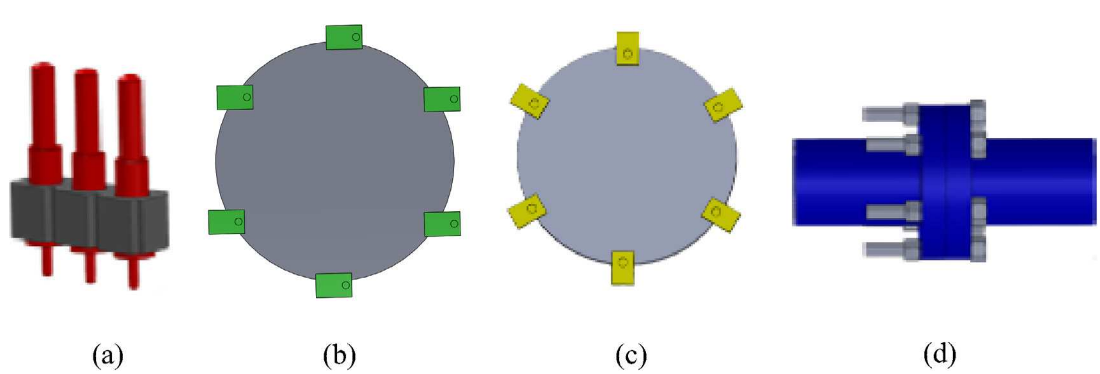


Fig. 7. Examples of patterns of repeated components.

(vi) *Identification of the type of contact*. The DOFs associated to kinematic pairs between two parts are further analyzed to detect whether they correspond to just a positioning relation or if the components are interlocked.

For each part in the assembly, the corresponding tessellation (if not already present in the dataset) is computed by the ***part model tessellation*** function. Not only the tessellation is used for visualization purposes, but it is also used to compute the associated shape descriptors not directly computable from the B-rep representation, namely spherical harmonics, D2 distance and shape distribution. These operations are performed by existing and available procedures (ISTI – CNR; Kazhdan et al., 2003; Osada, Funkhouser, Chazelle, & Dobkin, 2002; SOLIDWORKS; Yu-Shen et al., 2009).

The shape descriptors are the inputs for the ***part classification*** function, which associates a category to each assembly component. This module exploits a combination of the computed shape descriptors according to rules detected through a machine learning approach (Rucco, Giannini, Lupinetti, & Monti, 2017). This categorization is aimed at reducing the number of comparison between parts for shape similarity assessment, to possibly discard negligible parts (such as fastener) during the matching process and to better support the formulation of abstract queries. The classification is performed according to the following categories: axis, bearing, c-clip, cylinder like, cube like, gear, key, linkage arm, nut, part of bearing, screw and bolt, spacer, sphere like, torus like and miscellaneous. These classes have been selected for discerning elements possibly corresponding to fasteners (e.g. screws, bolts and nuts) with elements corresponding to important parts characterizing specific mechanisms, such as those involving speed and movement modification. These classes are not at the same level of specification, being some more geometry oriented (e.g. cylinder-like or torus-like) and others referring to the specific mechanical component type (e.g. gear or axis). Several reasons motivate this choice. First, parts can be designed at different levels of details depending for instance on the design stage or on the fact that the component is internally produced or acquired from third parties. Thus, a gear can be fully detailed or even designed as an engraved cylinder with a trough hole. Analogously many mechanical parts, which are themselves assemblies and normally acquired by third parties, such as bearings, are frequently available from online catalogues and included in larger assemblies as a single component potentially with a simplified shape. This motivates the inclusion bearing class also for single part components. Second, solids having the same shape may correspond to different part types and their real meaning can only be detected considering how they are used. Thus, we decided to include also the more generic shape oriented class. In fact, for the ground truth specification of the classes we used existing databases (Jayanti, Kalyanaraman, Iyer, & Ramani, 2006) and interviews with mechanical engineers and designers, when the shown object could correspond to different mechanical components it has been assigned to the most generic one. Knowing that similar objects might be classified as belonging to more than a single class, during the matching process elements classified as belonging to a given class are compared only with those of the same and of the equivalent classes.

At the end of the creation process, the EAM is represented as a graph, where all the extracted information is encoded as attributes of nodes and arcs. Fig. 8 illustrates an example of the graph structure created from a CAD model and enriched with semantic information. For readability purposes, only a part of the attributes is represented. The simply-circled nodes are associated with parts, while the double-circled nodes (S and N) are associated with a set of parts belonging to circular rotational patterns. The straight arc connects two components, which are in face contact, and the associated label corresponds to the allowed DOF, where R indicates a rotation, T a translation, the subscripts u, v and n the directional vector along which the rotations/translations are allowed in the local reference frame of each part. The wavy arcs indicate a line contact and according to the description of the interface layer, we do not consider the DOF between parts in contact by a vertex or an edge. Thus in those cases, we do not have labels specifying the corresponding degree of freedom.

3.3.2. Query EAM creation

As mentioned before, the graphic user interface provides functionalities to specify the query and to create the EAM, i.e. the EAM which is then compared to all the EAMs stored in the database. This process includes the creation of an abstract EAM model, in which the various layers can be fully or partially defined. In this process, some of the EAM characteristics and associated values can be specified by the user or automatically computed from a provided example, which can range from a precise CAD assembly model to an abstract assembly graph.

Actually, when the query corresponds to an existing CAD model, the EAM is created using the same "EAM creation" module as described in Section 3.3.1. If during the query specification, the user relaxes some characteristics, which he/she considers irrelevant, the corresponding evaluation functions are ignored.

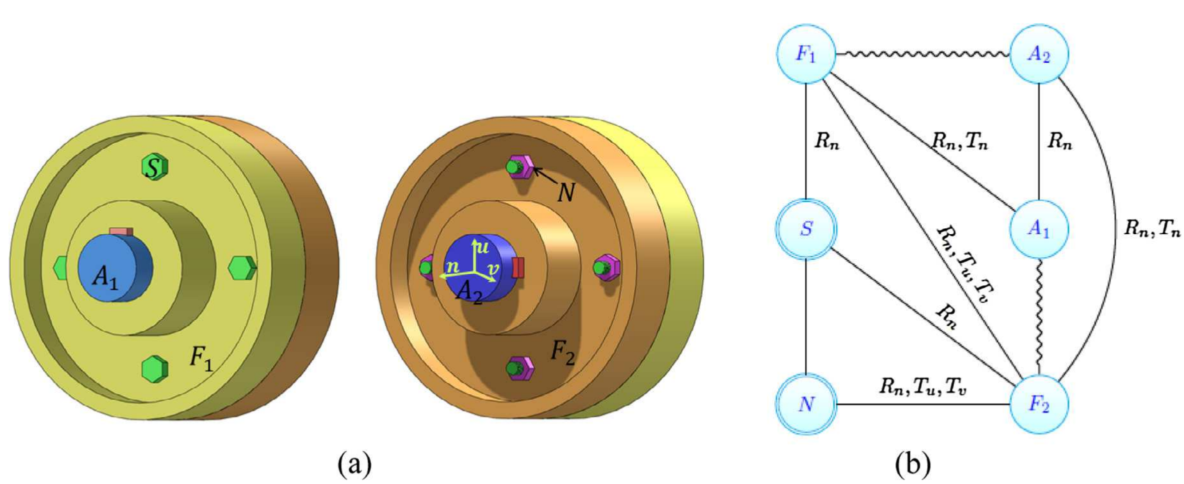


Fig. 8. Example of a CAD model (a) and a part of its EAM descriptor (b). The straight lines indicate face contacts and the wavy lines indicate line contacts between parts.

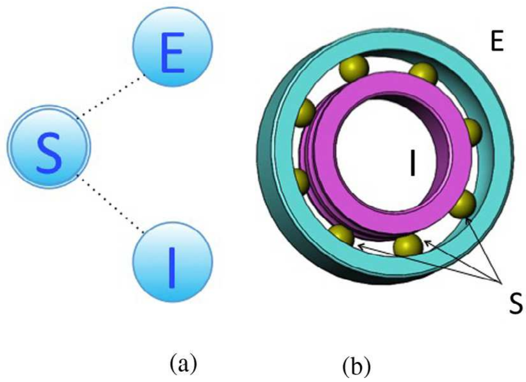


Fig. 9. Example of abstract query (a) and a possible corresponding CAD model (b).

In case of an abstract query, the user has to describe his/her query starting from scratch, i.e. without using a CAD model as a reference. For abstract queries, the mandatory information is the number of the constituting principal components and the related interface links. The abstract query creation is supported by a dedicated user interface. The user can add some nodes together with some associated attributes. He/she can link the nodes with arcs and can also specify the type of the arcs, i.e. a structural arc or an interface arc and in the last case with the label regarding the remaining DOFs. The attributes that the user can specify are those described in the different layers of the EAM descriptor.

Fig. 9 shows a possible abstract query example. Here, thanks to the dedicated user interface, the user defines a graph involving three nodes, which have been assigned a class: *part of bearing* for the nodes E and I, and *sphere like* for the node S. For the node S, the characteristic of belonging to a circular pattern is also specified. Then, the user inserts the relations between components, first selecting the concerned nodes and then he/she describes the contacts (and the type) between the parts. In this case, vertex contacts are selected (dotted lines). A simple abstract query graph as the one described in Fig. 9a can represent a model as the one displayed in Fig. 9b.

Optionally, for each value associated to the attributed graph a percentage of allowed variation can be assigned, which can be used to speed up the retrieval process. For instance, the user can decide to look for similar assemblies whose number of parts of a given category differs at maximum of the 50% to the query examples. It allows a pre-filtering of the candidate most similar models through the verification of the concerned statistic values.

3.3.3. The EAM matching

The matching problem is addressed at the different levels of the EAM in a top-down manner. If the user expresses ranges in which two assemblies are considered similar, e.g. allowed percentage of different components or relations, a filtering can be applied to reduce the number of models to be compared.

The problem of finding the matching between two assembly models is translated into the problem of finding a sub-isomorphism between two EAMs. An EAM can be seen as an attributed graph structure. Thus, a partial correspondence between two graphs corresponds to the problem of finding their maximum common sub-graph (MCS). Among the various techniques proposed for the identification of the MCS (Bunke, Foggia, Guidobaldi, Sansone, & Vento, 2002), our strategy is to identify the maximum clique (MC) Pelillo, 1999. To compute it, an association graph is constructed where nodes equivalent in the two attributed graphs are mapped into a single associative node. Two nodes are considered equivalent if they have the compatible values of the attributes specified by the user through the search criteria accessible from the interface. To limit unnecessary comparisons, if the search is looking for assemblies similar in all the aspects, at first the comparison is performed on the structural nodes and assembly interface layers, then the geometric matching on the part geometry is performed only on the returned candidates. In this case, nodes are then considered equivalent if they are associated to equivalent categories. Similarly, arcs in the associated graph are present when the corresponding nodes in the attributed graphs are connected in the same way. Arcs are considered equivalent when at the interface layer, corresponding arcs have the same classification and DOF.

The maximum clique corresponds to the maximum set of nodes all connected together of this newly defined association graph. In our system, the maximum clique finding problem is solved using a method, which exploits the simulated annealing technique (Giannini et al., 2017). Shape descriptors and statistics information at the node and interface layers are then used to adjust the similarity ranking. Since the assembly comparison is important at the various information layers, the matching process provides as result a vector of measures: the first measure refers to the identified cliques, the second to the interface statistics, the third is related to the component shape similarity.

For example, suppose we want to compare the model in Fig. 8 with the one in Fig. 10, seeking for parts with similar shape at 80% and that simply preserve the contacts without considering the type. The construction of the association graph starts matching the nodes in equivalent classes whose distance of the vectors representing the shape descriptors is less than 20%. Then, for each pair of nodes, the possible association arcs are investigated. During this phase, an association node for each pair of screws and nuts is created, while for the two main parts no nodes are created since their shape does not satisfy the requirement. According to the specified query, we check if the original nodes (defined by the pair of association nodes) in their original graphs are both in contact or not. Notice that the contacts between the screws and the nuts in the first model are threaded (i.e. inducing a volumetric intersection) while in the second they are simplified in the form of cylindrical contacts. Then, the two models are recognized as similar if the type of contact is neglected. On the other hand, comparing just the shape of the two models, the number of matched parts increases, since just the main flange is different while both screws and nuts are detected as similar.

4. Results

A prototype system has been devised to evaluate the proposed framework. The interface and the matching module have been developed using Microsoft Visual C# 2013. The information necessary for the reasoning process of the EAM creation are extracted exploiting the Application Programming Interface (API) of the commercial CAD system SolidWorks©. The final framework is included into SolidWorks as a plug-in.

In the following, we introduce the dataset used to evaluate our approach and some results achieved thanks to our multi-level assembly descriptor and its multiple search criteria.

4.1. Database of CAD assembly models

So far, no public database exists to evaluate and compare different retrieval methods on assemblies composed of B-Rep CAD models. This lack is due to two main difficulties. First, it is generally hard to get realistic CAD models. Most of online repositories do

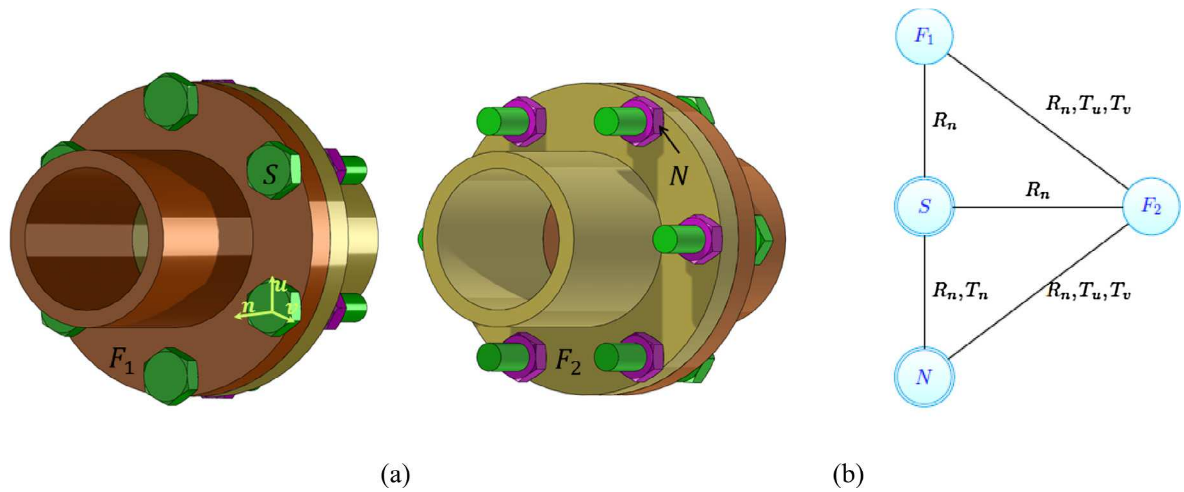


Fig. 10. Example of a CAD model (a) and a part of its EAM descriptor (b).

not provide complex assemblies and many of the available models are inaccurate, e.g. with unrealistic simplifications or with subassembly components collapsed in single parts, which makes the comparison quite insignificant. Sometimes, only the discrete representations are available. The second problem concerns the difficulties to create a meaningful ground truth, i.e. to identify the models that should be retrieved according to a specified query. If for a query defined by a single part, it is probably instinctive to identify similar models (even under multiple criteria), for an assembly it is not straightforward. The difficulty rises from the "partial" similarity, since in case of assembly models it makes sense seeking if the query is included in the assembly model or vice versa.

In this work, we collected 163 assembly models focusing on the quality of the models to minimize problems deriving from inaccuracies and unrealistic simplification. Table 3 illustrates the used dataset. In the future, to ease further comparisons with our work, we aim to make available the dataset with its own ground truth.

4.2. Filters and similarity criteria

To demonstrate that only a single criterion is not sufficient for a meaningful matching, we report in Table 4 a sample of the results obtained applying various criteria to the same query model. The model A is used as query and it has two main parts (one is the reflection of the other) and a set of screws and nuts, both arranged in a circular pattern. To easy the reading of the results, we include for each criterion a coarse similarity rate, defined as the number of matched nodes over the nodes in the query model. This kind of rate is too coarse to guarantee a good similarity assessment; it is just a simple way to give an idea about how many parts are matched. In the future, to provide a significant comparison, we intend to define different measures for the different criteria available in the framework, investigating how to weight all the criteria and how to combine (linear or not) them to provide a useful global assessment.

Table 3
Classification of CAD assemblies in our testing set.

Category	Number
Propeller mixer	18
Rotor wind turbine	22
Double rotor turbine	13
Hydraulic reduction	6
Bearing	36
Mill max	8
Linear actuator	10
Coupling flange	5
Landing gear	7
Hinge	4
Hydraulic rotor	6
Piston	5
Total	163

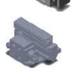
In the third column, we report the results using no criteria to limit the association between the nodes and creating the arcs if the numbers of rotation and translations in the joint level are the same. In this case, we can observe that all the models contain nodes with the same relative motion as in the query model. The model B has exactly the same contacts as in A, while M and N models have only planar contacts as the two main parts in A. The bolts and the nuts in the model C are not threaded inducing another kind of joint different from the one in the query model. This difference is underlined matching only 8 parts (4 screws and the 2 main parts) over 14 in the query model. The treated contacts in the query model make associable the screws with the balls in the bearing models D, E and F. Even if this kind of comparison is one of the most complex to compute, taken alone it does not distinguish enough the different models.

In the fourth column, we report the rates obtained by imposing conditions only on the component shapes and not on the joints (i.e. arcs) for which the Euclidean distance of the vector of the 3D spherical harmonics is lower than 0.35, which means that the matching parts have a shape similar greater than 65%. This comparison is computationally lighter than the previous one and we can observe that it discriminates more the results. Landing gear models (G and H) have shafts whose shape is similar to the screws. Again, according with our aim of partial matching, these models are retrieved since they include a portion of the query. The same situation occurs for the mill-max models (I and L), where the pins are similar to the screws.

The results using filters both on nodes and arcs is provided in the fifth column. Since the model B has the same contacts as the query, only the shape criterion is relevant. On the other hand, for the model C, even if the joint and shape criteria retrieve the same values, their combination gives a different result. Indeed, the joint criterion retrieves four screws, the two main parts and the two nuts not linked with the retrieved screws (8/14 = 0.57), while the shape criteria match the screws and all the nuts (8/14 = 0.57). The combination of these two criteria returns just the four screws and the two nuts disconnected (6/14 = 0.42).

The sixth column corresponds to the results obtained by posing as condition only the presence of same patterns of the query in the

Table 4
Similarity matching according to several filtering criteria.

 Query	Retrieved model	* Joint	Shape*	Shape joint	Circular patterns	Pattern, shape joint
A		1.00	1.00	1.00	Y	1.00
B		1.00	0.71	0.71	Y	0.71
C		0.57	0.57	0.42	N	0.00
D		0.29	0.00	0.00	Y	0.00
E		0.64	0.00	0.00	Y	0.00
F		0.57	0.00	0.00	Y	0.00
G		0.57	0.42	0.21	N	0.00
H		0.64	0.78	0.42	N	0.00
I		0.29	0.21	0.21	N	0.00
L		0.64	0.50	0.50	N	0.00
M		0.50	0.00	0.00	N	0.00
N		0.20	0.00	0.00	N	0.00

compared model. This condition is very simple to apply and makes an important filtering. This shows the importance of the statistic layer providing a basic but efficient way to discard inappropriate models. The contribution of these filters depends also on the context. For example, the pins of the two mill-max (i.e. I and L models) are arranged in two different ways, I in a linear way and L in a circular one. Thus, these models are not recognized as similar according to the pattern criterion.

From these results, it becomes clear that the adopted criteria are good for retrieving similar models. The proposed approach helps finding similar assembly while being closer to the user intent for the search. If the user really wants an assembly made of 4 screws, only such assemblies will be retrieved. However, if the user is interested in screwed assemblies, they will be retrieved whatever the number of screws. These examples also show that a single key is not sufficient to search efficiently, while combining several similarity criteria improves the results. Therefore, the next step is to define how to combine them to retrieve the models which best match the designer interest. Moreover, it is also important to choose an appropriate order in applying the filters to reduce as much as possible the complexity of the computation. A tentative to define such combination of criteria is proposed in the latest column, where first, the models with similar circular pattern are considered and secondly those among them with similar shapes and joints are retrieved.

4.3. Retrieval with abstract query

Thanks to the proposed framework, the user can define his/her own abstract query model as input of the retrieval process. Differently from what has been tested and summarized in Table 4, no reference CAD model is used as a query. The user directly specifies the structure of the graph as well as some attributes of the nodes and arcs. Here, we assume that the user is able to translate what he/she has in mind in the graph representation of the EAM. Fig. 11 shows some results using this feature. In the first row, the user sketches the abstract query while translating the idea of a bearing model in a graph composed of three nodes: two outer nodes as part of bearing and a middle node as a set of eight spheres arranged in a circular pattern. The nodes are linked by an arc that corresponds to a vertex contact. The system is able to manage this request retrieving bearing-like assemblies. Models from A to D are ball bearings, while the others (form E to H) incorporate parts matching the proposed query.

Another example is illustrated in the second row, where the user is seeking for a triptych of gear-bearing-shaft commonly used in epicycloidal speed reducers. The retrieved models incorporate the required elements modeled either with a detailed representation (models from A to D have gears with tooth) or with a simplified one (models from F to I have gears as a simple disk). Unfortunately, the skill of the proposed method to retrieve portions of the query can also provide some false positive, as the model E. This model is retrieved since it has three axes not linked among them. This limit suggests that not all the query subsets may be considered as a possible match and some filters should be applied. To overcome this limitation, the measure adopted for the similarity assessment should also consider the percentage of arcs in the retrieved subset, thus discarding cases as model E where the three shafts in the retrieved subset are disconnected.

5. Conclusions and future works

In this paper, a framework for the retrieval of similar CAD assemblies has been proposed. It is based on a three levels architecture and on the adoption of an assembly descriptor called Enriched Assembly Models. The EAM explicitly encodes the shape and liaison information, which describe the assembly constituent

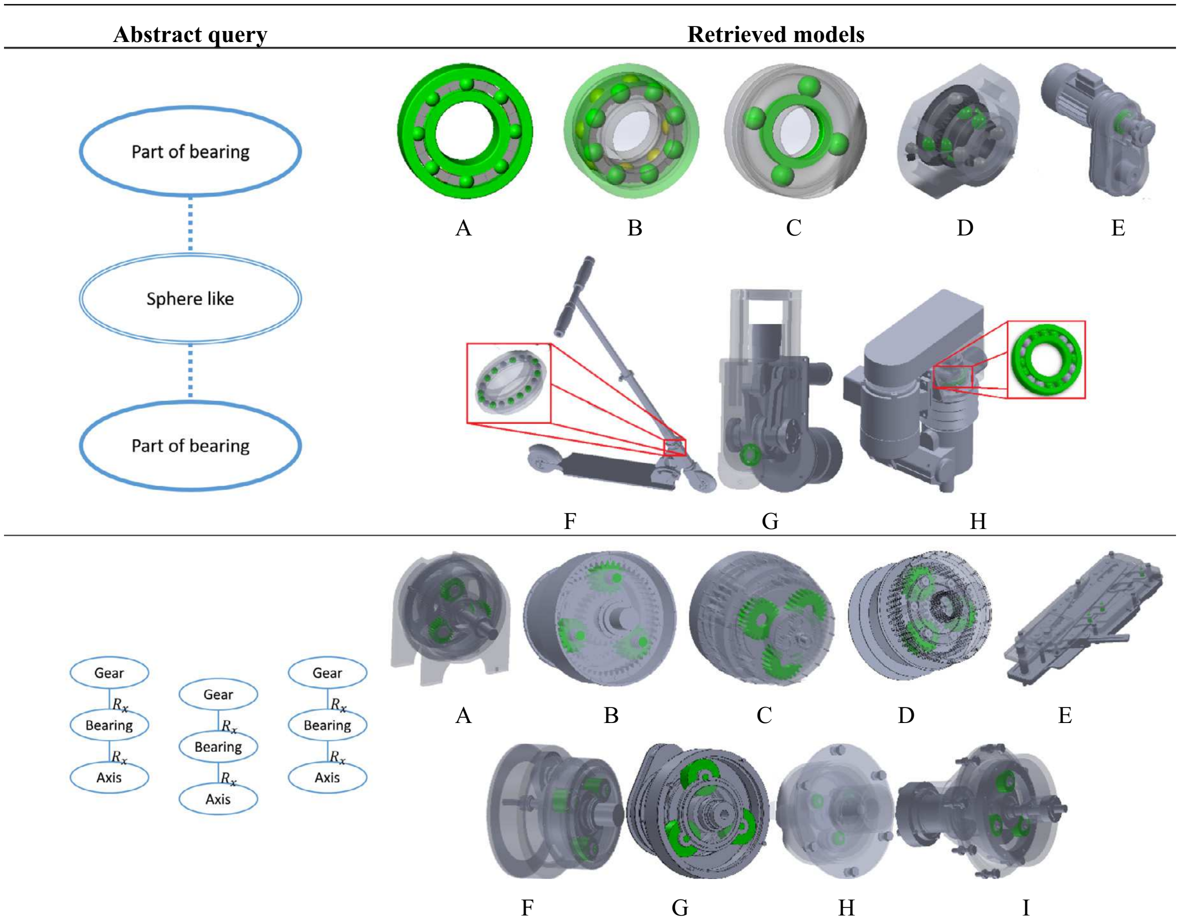


Fig. 11. Sample search with abstract query model.

elements and their arrangement, necessary for the model comparison. CAD models available in the database are pre-processed and enriched with information extracted from their geometric and topological data. Once enriched, the corresponding EAMs are available and ready for the matching step. Then, a query model is created by the user and compared to each EAM of the database using specific criteria. Both the creation of the query model and the criteria used for the comparison can follow different scenarios, which are reflected on the type of information considered for evaluating the model similarity.

Our approach has the following advantages. Differently from most of the systems available in literature, it allows the automatic computation of the joints between the components. The automatic pre-classification of the parts allows the use of query models not necessarily fully specified thus enabling partially defined queries, very useful in the early design phases where the product specifications are not fully known.

The proposed EAM is a first step toward the definition of a unified multi-level structure for CAD assembly description. Together with the definition of an advanced hierarchical matching process, it helps retrieving CAD assemblies in huge databases.

To make the proposed framework more functional, our future work includes the definition of several measures for the proposed multiple criteria as well as the combination of those measures to define meaningful global assessments. We intend to apply the approach also for the identification of interesting mechanism thus to allow a more semantic search of the models. Moreover, the information extracted and stored in the EAM can be used as signatures for the classification and indexing of CAD assembly models. Such an indexing can speed up the retrieval process while discarding from the matching procedure assemblies that do not belong to the same category.

Conflict of interest

Jean-Philippe Pernot as author of the paper and member of the Program Committee of the TMCE 2016 Symposium

References

Biasotti, S., Marini, S., Mortara, M., Patanè, G., Spagnuolo, M., & Falcidieno, B. (2003). 3D Shape Matching through Topological Structures. In I. Nyström, G. Sanniti di Baja, & S. Svensson (Eds.), *Discrete geometry for computer imagery. DGCI 2003. Lecture Notes in Computer Science* (Vol. 2886). Berlin, Heidelberg: Springer.

Bunke, H., Foggia, P., Guidobaldi, C., Sansone, C., & Vento, M. (2002). A comparison of algorithms for maximum common subgraph on randomly connected graphs". In *Structural, syntactic, and statistical pattern recognition* (pp. 123–132). Springer.

Chan, C. K., & Tan, S. T. (2003). "Generating assembly features onto split solid models. *Computer-Aided Design, 35*(14), 1315–1336.

Chen, X., Gao, S., Guo, S., & Bai, J. (2012). A flexible assembly retrieval approach for model reuse. *Computer-Aided Design, 44*(6), 554–574.

Deshmukh, A. S., Banerjee, A. G., Gupta, S. K., & Sriram, R. D. (2008). Content-based assembly search: a step towards assembly reuse. *Computer-Aided Design, 40*(2), 244–261.

Falcidieno, B., Giannini, F., Léon, J.-C., & Pernot, J.-P. (2014). Processing free form objects within a Product Development Process framework. In J. G. Michopoulos, C. J. J. Paredis, D. W. Rosen, & J. M. Vance (Eds.). *Advances in Computers and Information in Engineering Research* (Vol. 1, pp. 317–344). ASME-Press.

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., & Jacobs, D. (2003). A search engine for 3D models. *ACM Trans. Graph., 22*(1), 83–105. https://doi.org/10.1145/588272.588279.

Giannini, F., Lupinetti, K., & Monti, M. (2017). Identification of similar and complementary subparts in B-rep CAD models. *Journal of Computing and Information Science in Engineering*. https://doi.org/10.1115/1.4036120.

Gupta, S. K., Cardone, A., & Deshmukh, A. S. (2006). Content-based search techniques for searching CAD databases. *Computer-Aided Design and Applications, 3*(6), 811–819.

Hu, K.-M., Wang, B., Yong, J.-H., & Paul, J.-C. (2013). Relaxed lightweight assembly retrieval using vector space model. *Computer-Aided Design, 45*(3), 739–750.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., & Ramani, K. (2005). Three-dimensional shape searching: State-of-the-art review and future trends. *Computer-Aided Design, 37*(5), 509–530.

Jayanti, S., Kalyanaraman, Y., Iyer, Y., & Ramani, K. (2006). Developing an engineering shape benchmark for CAD models. *Computer-Aided Design, 38*(9), 939–953.

Kazhdan, M., Funkhouser, T., & Rusinkiewicz, S., (2003) Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing*. 2003. pp. 156–164. Symposium on geometry processing. Vol. 6.

Li, Z., Zhou, X., Liu, W., Niu, Q., & Kong, C. (2016). A similarity-based reuse system for injection mold design in automotive interior industry. *The International Journal of Advanced Manufacturing Technology, 87*(5), 1783–1795.

Lupinetti, K., Chiang, L., Giannini, F., Monti, M., & Pernot, J. P. (2017). Regular patterns of repeated elements in CAD assembly model retrieval. *Computer-Aided Design and Applications, 14*(4).

Lupinetti, K., Giannini, F., Monti, M., & Pernot, J. P., (2016). CAD assembly descriptors for knowledge capitalization and model retrieval. In (Imre Horvath, Jean-Philippe Pernot, Zoltan Rusack, Eds.) *Proc. Tools and Methods of Competitive Engineering*. Selected papers from the eleventh Symposium on Tools and Methods of Competitive Engineering – TMCE 2016, May 09–13, Aix-en-Provence, France, pp. 587–598.

Lupinetti, K., Giannini, F., Monti, M., & Pernot, J. P. (2016). Automatic extraction of assembly component relationships and their application for assembly model retrieval. *Procedia CIRP, 50*(2016), 472–477.

Miura, T., & Kanai, S., (2009), 3D shape retrieval considering assembly structure. In *Proceedings of Asian Symposium for Precision Engineering and Nanotechnology 2009* (ASPEN 2009).

Osada, R., Funkhouser, T., Chazelle, B., & Dobkin, D. (2002). Shape distributions. *CM Transactions on Graphics (TOG), 21*(4), 807–832.

Pelillo, M. (1999). Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation, 11*(8), 1933–1955.

Roj, R. (2014). A comparison of three design tree based search algorithms for the detection of engineering parts constructed with CATIA V5 in large databases. *Journal of Computational Design and Engineering, 1*(3), 161–172.

Rucco, M., Giannini, F., Lupinetti, K., & Monti, M. (2017). Automatic part classification using machine learning technologies. IMATI Report Series, January, Number 17–06.

Shahwan, A., Foucault, G., Léon, J.-C., & Fine, L. (2014). Deriving functional properties of components from the analysis of digital mock-ups. *Engineering Computations, Emerald*, 16.

SOLIDWORKS Application Programming Interface (API) <http://help.solidworks.com/2017/English/api/sldworksapiprogguide/Welcome.htm>.

Tangelder, J. W., & Veltkamp, R. C. (2008). A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications, 39*(3), 441–471.

Tao, S., & Huang, Z. (2012). Assembly model retrieval based on optimal matching. In *Software engineering and knowledge engineering: Theory and practice* (pp. 327–336). Springer.

The visual computing lab – ISTI – CNR: Visual computing library: a gpl c++ library for mesh processing <http://www.meshlab.net/>.

Wang, P., Li, Y., Zhang, J., & Yu, J. (2016). An assembly retrieval approach based on shape distributions and Earth Mover's Distance. *The International Journal of Advanced Manufacturing Technology, 86*(9), 2635–2651.

Yu-Shen, L., Fang, Y., & Ramani, K. (2009). IDSS: Deformation invariant signatures for molecular shape comparison". *BMC Bioinformatics, 10*(1).

Zehtaban, L., Elazhary, O., & Roller, D. (2016). A framework for similarity recognition of CAD models. *Journal of Computational Design and Engineering, 3*(3), 274–285.

Zhang, J., Xu, Z., Li, Y., Jiang, S., & Wei, N. (2013). Generic face adjacency graph for automatic common design structure discovery in assembly models. *Computer-Aided Design, 45*(8–9). 1138-115.