SN Computer Science (2022) 3:19
https://doi.org/10.1007/s42979-021-00853-9

ORIGINAL RESEARCH

A Web-Based Solution Supporting CAD Assembly Model Exploration and Analysis

Katia Lupinetti[1] · **Daniela Cabiddu**[1] · **Franca Giannini**[1] · **Marina Monti**[1]

Received: 1 December 2020 / Accepted: 6 September 2021 / Published online: 26 October 2021

Abstract
In the last decades, digital 3D models have substituted 2D technical drawings for the design and development of mechanical products. This results in a huge amount of CAD (computer-aided design) models available in legacy and Web repositories. 3D CAD models are now the reference product specification, and are frequently exchanged with customers and companies collaborating in the product development. In addition, this great amount of data can be exploited by scientists to test and evaluate their research results. Therefore, the visualization and analysis of CAD models from everywhere and through any device are particularly useful for both engineers and researchers. To answer this need, this paper presents a web-based solution exploiting the X3D capabilities for the direct visualization in the Web browser of CAD assemblies and of some of their characteristics remotely computed by the application on user demand.

Keywords 3D web application · CAD data visualization · 3D visualization · Online service

Introduction

Recent advances in internet capabilities for delivering big amount of data together with the possibility of connecting with no time and location restrictions, make the development and use of Web applications more and more attractive. Contextually, as argued by Mounton et al. [30], Web applications have major benefits comparing to desktop applications since they are available for all major platforms and their use does not require the installation of any software or libraries (other than the Web browser). Anyhow, despite the growing interest in visualizing and processing 3D content on the Web, "current browsers are not designed to manage the data-intensive and interaction-packed throughput required by Web 3D applications" [36]. In the engineering domain, the increased number of 3D acquisition and authoring tools resulted in a large amount of available 3D models that from one hand can be reused by companies, thus avoiding the time-consuming recreation of useful assets, and from the other hand can be exploited by scientists to simulate and test their methods and algorithms. Most of the available repositories of 3D data content allow the 2D and/or 3D previews of parts to provide some more insight on the 3D model, before downloading it, to facilitate the model selection. Unfortunately, in general, such previewers just provide the 3D shape and only some commercial tools also specify indications of the salient parameters characterising the product. Thus, they are informative only for simple parts. For complex 3D assembly models, possibly made of up to several hundreds of parts, simple 3D visualisation is not enough to understand what are the parts that make up the model and how they combine with each other. For such models, one important issue is the ability to visualize individual hidden components [18], which may be arbitrarily complex, and may occlude one the others; see as an example Fig. 1a where the cover and the cage hide the enclosed parts.

To allow the inspection of the internal parts, several solutions have been proposed, such as playing with transparency values of some components to allow to see through them (see Fig. 1b). An alternative method consists in defining sections of the product by selecting one, or more planes (see

This article is part of the topical collection "Advances on Signal Image Technology and Internet based Systems" guest edited by Albert Dipanda, Luigi Gallo and Kokou Yetongnon.

✉ Katia Lupinetti
katia.lupinetti@cnr.it

1 Italian National Research Council (CNR), Institute for applied mathematics and information technologies "Enrico Magenes" (IMATI), Via De Marini 6, 16149 Genova, Italy

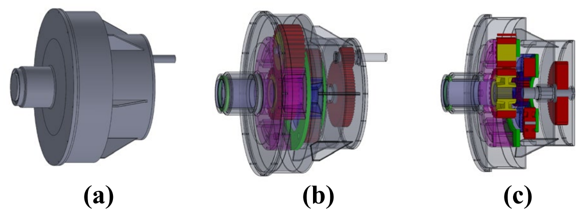


Fig. 1 Different views of a 3D assembly model with occluded components

Fig. 1c). In case of multiple planes, the section view should be defined by the intersection of the selected planes.

Additional important informative data normally not visualized can be either explicitly stored in the models, as annotations for instance, or implicitly deducible by model analysis. As an example, some assemblies could have parts with oversimplified shapes because they are supplied from external vendors. In this case, the functionality of such a simplified part can be deduced by analysing the context of use of the part in the assembly or by accessing other information, e.g. bill of material, encoded in external systems, such as in product data management (PDM) or product lifecycle management (PLM) systems. While such information can be available to engineers and designers, it might be more difficult to obtain for researchers who might not have access to these additional sources or might not own the necessary engineering knowledge. Therefore, it would be of utmost importance the availability of tools allowing a better inspection of the components and characteristics of the assembly to facilitate the comprehension of their suitability for the specific needs.

In this perspective, to ease the inspection of CAD (computer-aided design) models when users cannot access to their CAD desktop application, this paper presents a Web-based application specifically designed to cope with assembly models. The application, named CAD3A (CAD assembly analysis application), allows the visualization of standard CAD models as well as the computation and the visualization of their salient characteristics implicitly embedded in the model from any kind of devices, including mobile devices, having a Web browser and an internet connection without browser plugins. Among the assembly characteristics, CAD3A enables the computation of the Degree Of Freedom (DOF) among pairs of parts in contact, the identification of regular patterns of parts and the parts' functional classification. Moreover it offers the possibility to export the CAD model converted in a Web-format (encoded as X3D) as well as the results of the computed characteristics using the JSON format for the exchange of data between Web clients. The present paper extends the work described in [23] by providing insight on the content of CAD assembly models, which is the specific 3D content type supported by our framework, and on the improved capabilities of the system. The novel contribution of this paper is manifold: (i) an improved Web-based visualization of CAD assembly model structure that provides not only the geometry of the constituent parts but also their hierarchical organisation within the assembly as defined by the designer; (ii) the remote identification of repeated parts and their (if any) distribution according to regular patterns, which represent information characterizing assembly models particularly useful for manufacturing considerations on the product, such as assembly planning, spare parts storage.

The rest of paper is organized as follows. Section "Related Works" provides an overview on the existing applications to deal with 3D models on the Web. Section "3D CAD Assemblies" describes the content of CAD assembly models by introducing terminology and relevant elements for their characterization. Section "The Web Architecture Framework" illustrates the general Web framework and focuses on the description of the developed services. Section "The Prototype" presents the developed CAD3A application with its UI functionalities and illustrates examples on medium-complex CAD assembly models. Finally, Section "Conclusion" ends the paper providing conclusions and future steps.

Related Works

Nowadays Web has become the main channel for the sharing and dissemination of knowledge and content. Through the years, 3D data became a consolidated type of content in addition to text, video and images. As a side effect, dealing with 3D content though Web applications has opened a relevant number of issues, mainly related to real time visualization. Due to the fact that depending on the specific application, 3D content may strongly vary in terms of stored data and format, different tools have been developed to deal with it over the Web. For instance, various tools have been developed in biochemistry and biology for the molecular visualization [37] and in medicine for the remote access to medical images, as it allows multiple experts from distinct locations to view patient data and discuss their findings [12, 29, 31]. Targeted to the cultural heritage domain [35], presents 3DHOP, a tool well suited for the visualization of high-resolution single objects (especially dense models coming from 3D scanning), which is conversely not appropriate to manage complex scenes made of several low-poly objects as when working with CAD, procedural or hand-modelled geometries. For a more detailed overview of existing methods for Web-based scientific data visualisation [17], provides an outline categorized by application domain.

Among the general-purpose tools, MeshLabJS [10] provides a Web-based graphical interface that allows to interactively edit 3D models represented as triangle meshes by exploiting a modern Web browser that is able to run C++ code compiled into a javascript at a pretty acceptable speed. Similarly, HexaLab [5] is a WebGL application for real time visualization, exploration and assessment of hexahedral meshes.

In addition to Web applications providing a graphical interface, it is worth mentioning that some online services have been published to enable the possibility to remotely process 3D contents As an example, Campen and colleagues published an online service called WebBSP [7] which is able to remotely run a few specific geometric operations. The user is required to upload an input mesh from a standard Web browser and select a single geometric algorithm from a set of available operations. The algorithm is actually run on the server and a link to download its output is returned, and the Web interface does not allow visualizing the 3D model. Similarly, Attene et al. [2] extended the capabilities of the visualization virtual service (VVS) infrastructure [46] allowing the possibility to remotely run complex customizable shape processing workflows. Users are allowed building and running their own workflows by selecting among available operations, each of them published as a single Web service. Again, the output is sent back to the user as a downloadable file and not 3D viewer is provided by the infrastructure.

The aforementioned Web applications focus on specific 3D content representations (either triangle or hexahedral meshes) which are typically used in many application fields, but not for the design of industrial product in CAD systems.

The type of 3D content influences not only the visualization but also the interaction with the information associated with 3D data. As an example of a Web application in the product design context, Nyamsuren et al. [33] proposed a tool for model revision control, based on WebGL technology, that calculates and represents differences between CAD models. In this domain, due to the need of companies to communicate with different suppliers, several applications have been developed for the conversion and/or visualisation of CAD files. In [44] a list of available viewers is presented. Most of them are desktop applications, possibly providing a Web version as a plugin. The common provided functionalities are the import of a variety of legacy and standard CAD formats, the visualisation of the associated assembly tree and metadata as well as the geometry in various modalities (e.g solid, wireframe, transparency setting). In some cases, as SpinFire [43], Clari3d [11] or 3DViewStation [1], they provide capability to perform some dimensional and distance evaluation over the 3D geometry, such as wall thickness, bounding box, volumes and surface area and to add annotations. In addition to the commercial systems, also some free tools are available. Some of them provided by CAD suppliers, e.g. [3], allow to freely visualise their native CAD formats. Purely Web-based applications, like ShareCAD [41], Creators3D [13], eMachine Shop [15], have strong limitations in dealing with assemblies since, in general, they only show the whole assembly as a single object with few capabilities to see the internal composing elements. Anyhow, none of the aforementioned applications allows the possibility of evaluating and visualising characteristics of assemblies which are significant when looking for models for design reuse or for testing analysis applications. This includes the inspection of the type and arrangement of the composing parts and the mating conditions between two parts (i.e. how two parts are connected together) and the functionality of the designed components.

These aspects are important topics in research and they have been faced in automation context. For instance, Park and Oh [34] developed an automatic method to extract kinematic information from assembly models, Yiu and Regli [47] presented a classifier for CAD models integrating machine learning techniques, while Vilmart et al. [45] studied the pattern, in particular symmetries and alignments, of repeated parts to extract possible assembly structures. Anyhow, most of the solutions developed to analyze CAD models are designed to be integrated as an application in a local workstation and not available as Web applications, as the one proposed in this paper. To deal with these aspects, recently Simões et al. [42] present a system to prepare and import CAD data for visualisation and interaction on the Web. Their X3D application allows user to interactively add information on the assembly behaviour and appearance, allowing to specify the material and kinematics information to the assembly parts. As CAD3A, their system imports CAD models in STEP format and translate them in X3D format for their inclusion in the Web browser, but differently from CAD3A no services for the automatic extraction of assembly characteristics are provided.

3D CAD Assemblies

The term "CAD assembly" refers to 3D models, generally employed in industrial engineering applications, whose representation scheme is usually the boundary representation (B-Rep). This representation is a de-facto standard in commercial CAD systems and provides an exact representation of the surfaces outlining the shape of 3D models. Generally, CAD assembly models are made up of several parts that are

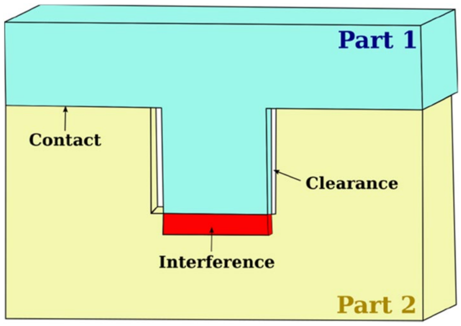


Fig. 2 Possible interfaces between parts of an assembly model

related together by hierarchical relationships, where the root node indicates the complete CAD assembly, intermediate nodes indicate the organisation in sub-assemblies and the leaf nodes refer to the atomic composing parts. The resulting hierarchical structure is not unique and can reveals some designer intents. For example, assemblies can be organized in a way that the forthcoming assembly simulation steps are eased, or they can be organized with respect to visualization or they can be decomposed according to criteria based on the constitutive materials. Such decomposition defines sub-assemblies and corresponds to the way designers may focus on a product. Note that hierarchical structure information is not necessary for visualization purpose of the 3D model, but its rendering can improve the general understanding of CAD model. For this reason, CAD3A aims at preserving hierarchical structure information during the conversion process from CAD models to models suitable for Web visualization. Details on this aspect are reported in section "X3D generation".

CAD assembly models are also characterized by *interfaces*, which describe the relationships between the parts of a model once those parts are arranged in the 3D space. These relationships may be grouped into *interferences*, *contacts* and *clearances* as shown in Fig. 2.

Interferences represent unrealistic and unrealizable configurations that can be generated by some mistakes or designed on purpose to convey a certain meaning [39, 40], such as the intersections among screw and nut threads, or when considering flexible parts, as springs, seals and insulating parts, or when designing parts assembled by shrink-fitting. The contacts between two parts contribute in the definition of a fundamental concept to analyze mechanisms, i.e. the *mobility*. The mobility of an object is defined by the connections between parts, called *joints* or *kinematic links*, that are made up of the contacts of each rigid geometric part [34] and that determinate the allowed DOF. Kinematic links are divided in two groups: upper kinematic pairs and lower kinematic pairs. A kinematic pair is said to be a lower pair if the involved parts have surface area contact between them. Different lower kinematic pairs can be identified according to the types of surfaces involved in the contact and the number of allowed DOF [32]. The common kinematic pair classification is depicted in Fig. 3.

An upper kinematic pair arises when two surfaces are constrained to remain in contact along a common line or at a common point. An example of this kind of joints is represented by ball bearings, where the balls are in contact by points to the inner and outer rings. Finally, the clearances occur when the presence of a distance between two or more surfaces of two parts convey a functional meaning. Understanding the mobility of CAD assemblies is a key feature to be analyzed when evaluating the general behaviour of a product, anyhow the general motion as well as the single DOF between two parts can be deduced by running complex simulations or analysing the geometry of contact surfaces (operations that require engineering expertise and specific CAD tools). With the aim of improving the general understanding of CAD models and the sharing of meaningful information, in CAD3A there is the possibility to compute and visualize in a simple and clear manner this kind of information.

These characterizing information can be shared using different file formats. On the one hand, proprietary formats can

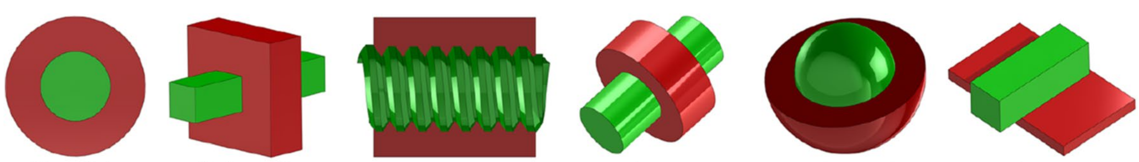


Fig. 3 Lower kinematic pairs [28]

enrich model information depending on the adopted CAD system producing files whose content can have slight differences even if they represent the same model. On the other hand, neutral file formats allow sharing standard information with no content customization. For these reason, CAD3A supports the STEP standard format (ISO 10303-203 and ISO 10303-214) as an input file format, which encodes both the geometry of the assembly models and associated information.

Beside the above-mentioned information, there exist other data related to CAD models necessary in manufacturing process of the designed products. For instance, annotations are used to explicitly express some geometric properties (such as tolerances and thread pitches) or component material. These additional information is usually stored in separated systems that can be linked to proprietary CAD tools; therefore, a generic 3D visualization application is not able to display such an information. To support researchers' activities CAD3A aims at computing and visualizing other relevant information that generally cannot be achieved without proper tools or specific engineering competences. In particular, the ability of identifying fasteners (e.g. screws, bolts, nuts and c-clips), parts that characterize specific functional sets and their arrangement in the 3D space following specific patterns allows to deduce information on assembly operations. For these reasons, CAD3A focuses also on the computation and visualization of *part classification* and of *regular pattern*. Details on these aspects are reported in Sections Part Classification and Pattern Detection.

The Web Architecture Framework

CAD3A has been designed as a Web application providing different functionalities as separated services that can be performed regardless the platform and the language adopted by the experts in their development. The distribution of CAD3A as a Software as a Service (SaaS) represents a benefit for the companies that decide to exploit it and for researchers who desire to visualize and analyse specific information related to the 3D models. This choice has an high impact on the quality and management of the developed system [14] due to the fact that new functionalities can be easily added even using different programming languages, databases, hardware and software environment, depending on what fits best [8]. In addition, the devices adopted by the final users do not require high computing performances since they are used as browsers and the computing capacity is demanded at the server side.

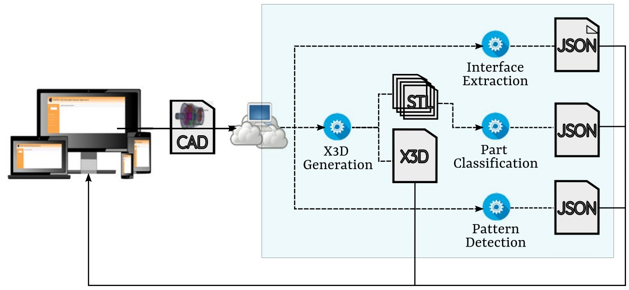


Fig. 4 Software as a service (SaaS) scheme. The central icon represents the Web application and the other ones describe the adopted file format to transfer information

In the following, Sect. provides details on the entire Web architecture, while Sects. from "X3D generation" to Pattern Detection describe the single modules employed as services in the CAD3A Web architecture.

Web Architecture

The scheme of the provided Web application is illustrated in Fig. 4. Users can upload on CAD3A's Web page a CAD assembly model encoded as a STEP file (both AP203 and AP214 are supported). The Web page is accessible through different possible devices (laptop, phone, tablet or desktop). The uploaded model is then processed by the **X3D generation** service that creates an optimized representation of the model itself according to the X3D language, which is an open standard for 3D graphics representation over the Web [6]. Such a representation enables the possibility to display the model in a 3D view where the various 3D components of the assembly model can be selected, rotated or zoomed. The 3D visualization shows the relative position of the single parts in the entire assembly model. To support the visualization of the CAD model organisation, the proposed system detects the hierarchical structure of the model by analyzing the part names in the generated X3D representation (see section "X3D generation").

In addition to the model visualization, the system allows the possibility to run several analyses on the 3D model, each of them extracting specific features and conveying their semantic meaning to the user through a graphical user interface. According to the CAD assembly characterization illustrated in Section 3D CAD Assemblies, the proposed services aim at detecting and visualizing the kinematic relationships as well as regular patterns and a functional classification of parts. Kinematic relationships are extracted by the

Interfaces extraction service (described in Section Interfaces Extraction), which uses the methodology described in [25] to identify the relationships between adjacent parts of the assembly model and DOFs between those having faces in contact. Other features are extracted by the services **Part classification** and **Pattern detection**, which provide respectively a functional classification of the parts according to their shape adopting the learning process described in [38] and an identification of regular arrangement in the 3D space of repeated parts (i.e. different instances of the same geometric part). Details on these services are reported in Sections Part Classification and Pattern Detection. All the above-mentioned services work on server side and transfer the results of their analysis back to the client Web page by exploiting JSON data-interchange standard. Finally, users have the possibility of exporting the information computed by the different services and reloading such an information for further analysis or use in their own applications.

X3D Generation

Managing 3D data over the internet is not an easy task [36]; moreover, the current Web viewers do not directly handle CAD data. Considering this lack, the goal of the proposed service is to provide an online converter from STEP to X3D file format, such that CAD assemblies can be examined and manipulated into a Web page. Among the different formats suitable for 3D content visualization on the Web (e.g. VRML, X3D, XML3D, O3D), the X3D format has been chosen since it is standard-based and is suitable for DOM integration [4], this means that the front-end elements can be manipulated through the Web interface (the DOM) and the modifications can be synchronized to the backend (the X3D).

Algorithm 1 The X3D model generation algorithm

procedure GenerateX3D(StepModel)

    STL_Files = {}
    for each $part_i$ in StepModel
        stl_i = Tesselate($part_i$)
        STL_Files ← stl_i
    end for

    X3D_Files = {}
    for each stl_i in STL_Files
        $x3d_i$ = Convert(stl_i)
        X3D_Files ← $x3d_i$
    end for

    X3D_model = Unify(X3D_Files)

    Return x3D_model
end procedure

Listing 1: Snippet of the structure of a generated X3D file.

```
<?xml version="1.0" encoding="UTF-8"?>
<scene>
  <shape>
    <indexedFaceSet coordIndex="...." >
      <coordinate point="...." />
    </indexedFaceSet>
    <appearance>
      <material DEF="part_name"
                diffuseColor="1.00␣0.10␣0.10"
                transparency="0.4"/>
    </appearance>
  </shape>
  <shape> ... </shape>
  <shape> ... </shape>
</scene>
```

As shown in Algorithm 1, given a STEP file encoding a CAD assembly model, the service iterates over the parts of the assembly and performs a triangulation of each of them. Each shape is labelled with the corresponding name (provided by the STEP file) allowing an 1-to-1 map between the STEP and the X3D for further interactions. To allow a tree-visualization representing the hierarchical structure of the CAD model, the label of each part is obtained by concatenating the names of the father components of the part in the assembly tree separated by a special character. Such a label is used as a filename to save each part as a STL file (standard triangulation language, i.e. a format file to represent the surface geometry of a three-dimensional object by the unit normal and vertices of the triangles approximating the original surface). Once all the STL files have been generated, they are unified into a single X3D file. X3D files are written in XML format, then the unification procedure works on the identification of relevant tags.

List 1 shows an example of a generated X3D file. A set of "shape" nodes is used to represent single assembly parts. Such a representation includes both geometry information (i.e. the "indexedFaceSet" node) and some visualization settings (i.e. the "apperance" node). The "indexedFaceSet" node defines both geometric and topological information by the "coordinate" node and the "coordIndex" attribute, respectively. Visualization settings are instead encoded by the "material" node inside the "apperance" node. This last node encodes also part names representing hierarchical data using the attribute "DEF" of its "material" tag.

Algorithm 2 The hierarchy structure algorithm

```
procedure HierarchyVisualization(x3D_model)
    HierarchyModelList = {}
    HierarchyPartList = {}
    for each shape in X3D_model
        name = GetX3DName(shape)
        hyerarchicalComponents = GetHyerarchicalComponents(name)
    end for
    HierarchyPartList ← hyerarchicalComponents

    for each Q_i in HierarchyPartList
        currentParent = StepModel
        while Q_i ≠ ∅ do
            currentComponent = GetFirstElement(Q_i)
            currentComponentName = GetName(currentComponent)
            parentName = GetName(StepModel)
            parentToSeek = BuildChildrenName(parentName, currentName)
            existingParent = Find(parentToSeek)
            if existingParent ≠ null then
                new_rel = CreateHierarchy(existingParent, currentComponent)
                HierarchyModelList ← new_rel
            end if
            currentParent = existingParent
            Remove currentComponent from Q_i
        end while
    end for

    Return HierarchyModelList
end procedure
```

Once the X3D has been generated, the Web client processes the information stored in each "apperance" node. This procedure is illustrated in Algorithm 2. Iterating over the shape tags in the X3D file, it is possible accessing each single part of the assembly model. Thus for each part, the main procedure reads the part name that has been generated preserving in the name its complete hierarchy (*GetX3DName()* procedure) and slits it according to the special character used in its definition (*GetHyerarchicalComponents()* procedure). The output is a list Q_i describing the family tree of the specific part, where the first element corresponds to the entire assembly model and possible other elements $q_i (i > 0)$ encode the name of the child of element $q_{i-1} \in Q_i$.

As a result, the hierarchical structure is visualized as a tree graph in which leaves represent the parts of the assembly model, the root the entire assembly model, while intermediate nodes represent sub-assemblies of the original model (see Fig. 8, right side).

Interfaces Extraction

Despite engineers spend a lot of time designing and enriching CAD models, most of CAD models are exchanged in standard format, e.g. STEP that is the most used in exchanging CAD data between companies, not incorporating the interface information. For this reason, this service aims at extracting the joints risen by contacts and the volumetric interfaces present in CAD assembly models.

The joint identification problem can be translated into a DOF identification problem, Table 1 shows the correspondences between joint types and DOFs. Thus, the **interface extraction** service computes the DOFs between pairs of touching parts by identifying the set of allowed translations and rotations between them. The general idea is that each contact surface inhibits a movement along a certain direction, the combination of all the constrained movement provides the resulting DOF expressed as the set of possible translation and rotation. Here, both sets are expressed by normalized vectors characterizing either a translation direction or a rotation axis according to the global reference frame. Concerning volumetric interfaces, the **interface extraction** service simply returns a tag representing the intersection with no additional information. Details on the DOFs and volumetric interfaces identification procedure are described in [25, 27].

Table 1 DOF values according to the joint type (see Fig. 3), where *R* indicates a rotation, *T* a translation, the subscripts *u*, *v* and *n* the vectors along which the rotations/translations are allowed. Note that *u*, *v* and *n* are generic vectors and thus they may not coincide with principal coordinate axes

Type	DOF
Revolute	R_u
Prismatic	T_u
Screw	R_u
Cylindrical	R_u, T_v
Spherical	R_u, R_v, R_n
Planar	R_n, T_u, T_v

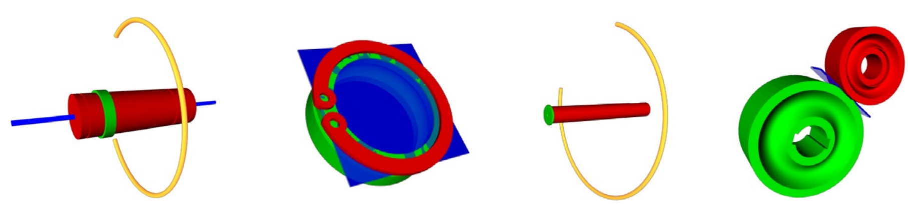
Once the joints and the volumetric interfaces are extracted, we graphically visualize them. Since the maximum DOF of a rigid body in space is defined by three translations and three rotations, distinct shapes and colors have been adopted to represent a specific transformation; specifically blue lines and yellow circles are used to represent translations and rotations respectively. An example is displayed in Fig. 5. Since translations and rotations need to communicate an axis, both of them could be represented with lines, but at the expense of intuitiveness. The choice of using two separated elements is led by the purpose of improving the information perception at first glance. In case of volumetric interfaces, a transparent blue box is displayed to highlight the intersection region. Figure 5a represents a cylindrical joint whose DOF is a translation and a rotation both along the cylinder axis, then the line identifies the translation direction while the rotation axis is represented by the normal to the plane underlying the circle line. Figure 5b shows a planar joint. In this case, the allowed translations are all the ones belonging to the mating plane and represented as blue lines. Since there is not an unique combination of vectors describing the directions orthogonal to a certain normal, a transparent blue plane has been added to describe that every combinations of the depicted lines are possible. Figure 5c illustrates a revolute joint where only a rotation is allowed and it is represented by the yellow circle. Finally, Fig. 5d shows two intersecting parts and the resulting box to communicate the region where the volumentric intersection occurs.

Part Classification

The **part classification** service aims at labeling the constituent parts of an assembly model according to their shape under a functional point of view. Despite the use of rules and standards, the functionality of a component may be not directly available in the CAD model but rather specified in some metadata or description in the PDM systems [38]. Thus, the specification of the type of the components in a CAD model strongly depends on the information manually specified by designers. This is a big limitation, also considering that to access this information the user is forced to rely on the PDM system. To provide a part classification that eases the assembly inspection, this service aims to automatically identify the functionalities of the parts by considering different shape characteristics. Anyhow, the part shapes are sometimes oversimplified making parts with different functions very similar in their shape [28]. These situations frequently occur when standardized components are acquired from suppliers making arduous an automatic recognition of component types. To deal with these oversimplified situations, this service provides a geometric-oriented classification (i.e. cylinder-like, cube-like, sphere-like and torus-like) other than canonical categories (including bearing, gear, c-clip, nut, shaft, screw and bolt, spacer, key, linkage arm and part of bearing).

The classification is based on a learning process applied on a collection of descriptors of the parts [38]. According to the analysis of Jayanti et al. [19], 3D spherical harmonics, shape distribution (with D2 measure), inner distance, size values (i.e. surface area and volume), and proportions among the minimum bounding box dimensions are the most suitable descriptors to discriminate the classes above cited. In the proposed framework, the 3D spherical harmonics, the D2 shape distribution and the inner distance are computed respectively by the procedure defined in [20] (with its default parameters), [21] and [22] on a mesh representation of the part; while the surface area, volume and proportions among the minimum bounding box dimensions are obtained using the API provided by the CAD kernel of SolidWorks®.

Once the parts are classified, the result is displayed in the Web page by coloring each class in a different way, as depicted in Fig. 6. Considering the high number of classes, colors has been chosen trying to maximize color differences for parts that are normally in contact. Note that this visualization does not cover the initial 3D visualization, i.e. the user can choose to display the classification results or



(a) Cylindrical joint **(b)** Planar joint **(c)** Revolute joint **(d)** Volumetric intersection

Fig. 5 Examples of DOF representations adopted in CAD3A. Yellow circles and blue lines represent allowed rotations and translations, respectively, (in accordance with the DOF described in Fig. 3); while blue transparent boxes indicate volumetric intersections

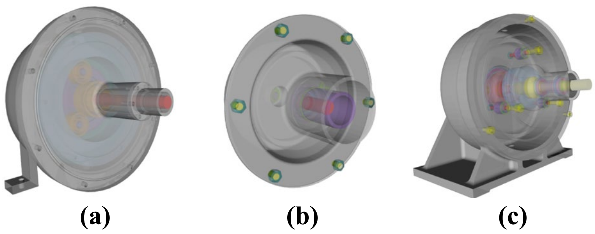


Fig. 6 Example of part classification visualization

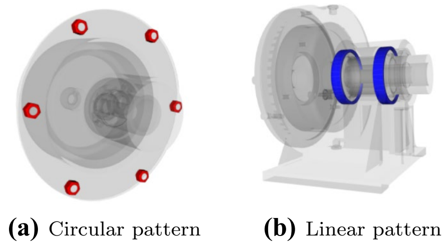


Fig. 7 Examples of pattern visualisation adopted in CAD3A, where the entire model is transparent grey and the components belonging to a pattern are colored in red and blue if the detected pattern is circular or linear respectively

simply visualize the 3D model using the UI described in section Development Environment.

Pattern Detection

Complex industrial assemblies normally include multiple occurrences of the same component. These repeated parts are often positioned within the assembly according to some regular patterns. The most common example is given by circular or linear patterns of screws and nuts frequently used to block together parts, such as cages and covers. Knowing the presence and position of such repeated elements might be of benefit for several applications. For instance, it offers suggestions for the standardisation of parts or for the spare parts storage. At the production level, it provides information useful for the evaluation of the assembling operations and costs. To support to a wider extent a better understanding on how the assembly is organised, CAD3A includes a service to detect the presence of repeated parts and identify their possible distribution according to regular patterns. This service is based on the algorithm described in [24]. Parts are considered as repeated whether they are explicitly encoded in the STEP file as occurrences of the same part, or they have the same shape. This last condition is detected by analysing few shape and topological information of the B-Rep of the parts. Two parts are then considered to be of equal shape when they have the same volume and surface, and equal number of faces of a specific geometric type (e.g. planar, cylindrical) and percentage of the surface area of a specific type over the whole part's surface area. Once the repeated elements are detected, the service verifies if they are arranged according to some predefined patterns. The service returns linear and circular patterns where elements are equally distributed. The algorithm works as a series of grouping and filters. First repeated parts whose barycenters are co-planar are grouped in subsets having the same distance. Groups are then processed starting from those with the largest number of elements and with lower distances between elements, considering the proximity a valid indicator for the pattern existence. Then it verifies if the elements in the group are lying over a circle or a line, in case elements of the same group can belong to more than one pattern, the one involving the highest number of parts is chosen.

Here, the result visualization exploits again the usage of colors. Since multiple patterns may exist in a model, distinguishing them in a possible multi-pattern view may be hard even if using different colors. Then we decided to visualize just one pattern per time (selected by the user) and to display them with a predefined color per type (blue for linear patterns and red for circular ones). A visualization example is illustrated in Fig. 7.

The Prototype

CAD3A is designed and implemented as a Web application, where a back-end module manages input models producing suitable outputs for the a front-end module that deals the rendering and the GUI. The current prototype is accessible online via HTTP.[1] As a matter of example, CAD3A Web page provides the possibility to play with a set of precomputed 3D assemblies, selected from the 3D Assembly Repository [26].

In the following, we describe some technical details on the services implementation (section Development Environment) and the developed graphical user interface (Section Data Visualization and User Interaction).

Development Environment

CAD3A has been deployed on an Windows Apache server endowed with a PHP interpreter.

The X3D generation service has been implemented as a pair of two pieces of code. The former is a python script exploiting SolidWorks® API to convert the input STEP assembly file into a set of STL files, each of them representing a single part of the assembly. The latter is written in C# and using Meshlabserver [9] converts each STL file into a

[1] http://cad3a.ge.imati.cnr.it/webapp.

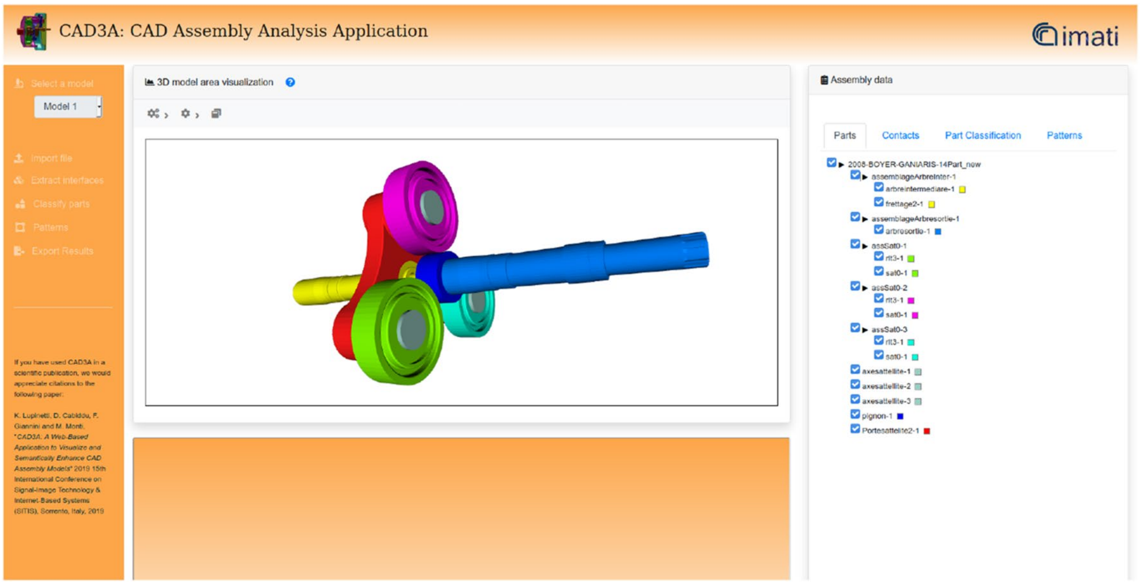


Fig. 8 CAD3A Web page. The left sidebar hosts buttons to both import/export data and run available services. The right sidebar provides extracted information about the imported model. The central area hosts both the visualization canvas and a log report area to show run-time information

X3D file, and finally merges them into a single X3D file encoding the whole model, while keeping the single parts separated.

The interface extraction service is a C# software. It takes the uploaded STEP file as an input, and generates a JSON file where the relationships among the assembly parts are encoded. List 2 shows an example of the structure adopted to describe the joint, where the involved parts are identified as "Source" and "Target" and the "DOF" tag lists the set of allowed and not allowed translations and the allowed rotations.

Listing 2: JSON example returned by the "Extract interfaces" service.

```
{
    ''Source'':''FLANGE_HUB-1-1''
    ''Target'':''BOLT_1-1-1''
    ''ContactType'':0
    ''DOF'':
        ''Translation'':[(1.0, 0.0, 0.0)]
        ''TranslationNotAllowed'':[]
        ''Rotation'':[(1.0, 0.0, 0.0)]
}
```

The part classification service is a C# software. It takes the whole set of the STL files generated by the aforementioned service X3D generation and returns a JSON file where, for each assembly part, its functional classification is encoded.

The pattern detection service is a C# software. It takes the uploaded STEP file as input and returns a JSON file where the detected patterns are encoded. Each pattern is described as the list of its constituting parts and its characterising parameters, such as the type of pattern, its radius and center.

Data Visualization and User Interaction

The graphical user interface is a dynamic HTML page exploiting Javascript to be responsive with respect to user interaction. As illustrated in Fig. 8, the content of the Web page is divided in three main parts: (i) a side menu on the left; (ii) a 3D viewer in the middle of the page (called "3D model area visualization"); and (iii) an information box on the right side (called "Assembly data") showing the result of processing services organized into different tabs.

The left side menu allows loading an input as a STEP file (AP203 or AP214) or as "Result" previously computed and exported; moreover, it allows running the desired processing services and exporting the results as a zip file. Each button of the menu is connected to a specialized PHP script, which is responsible to run the server side operations described in Sections X3D Generation, Interfaces Extraction, Part Classification and Pattern Detection, and to return their output information to the Web page so that it can be properly displayed. Computed data can be exported by selecting desired data (the X3D file, the contacts, the classification and/or the pattern data) and clicking the "Export results" button, which downloads a folder in a ZIP format with the selected data. Such a result can be re-imported for further analysis using the "Result" import functionality. Once a service has been executed, the color transparency of the relative button changes, assuming a grey shadow to indicate the button has been disabled. In addition to the above mentioned buttons, at the top of the menu it is possible to select from a drop-down menu pre-computed models.

The "3D model area visualization" is implemented as a X3DOM scene [4]. It exploits the WebGL support integrated by default in modern browsers to embed the 3D content into a browser context by the definition of an XML namespace that enables a $< x3d >$ tag to be used in a standard HTML [16]. At the top-left corner of the 3D view there are three buttons that allow to (i) show/hide all the parts, (ii) change the visualization mode selecting between opaque and transparent options, (iii) reset the view centering the model in its original position. In the 3D view, the imported assembly can be rotated, translated and zoomed. As default, the components are visualized in transparency mode to allow a partial visualization also of the internal ones. In both the visualization modes, through "Part" tab in the right box, it is possible (i) to hide/show a single part or a set of parts grouped in the hierarchy structure by unchecking/checking the corresponding part flag; and (ii) to change the color of each part by clicking on the color button next to the name part and choosing the desired color in the pop-up window. In addition, to help the identification of correspondence between the 3D part and its name, if a user clicks over a part in the 3D view, the part color changes in opaque mode and its name in the hierarchy-tree view becomes bold.

Pressing the "Extract interfaces" button, the service detecting the relationships among the assembly parts is called and the pairs of parts sharing a contact or a volumetric intersection are listed in the "Contacts" tab of the Assembly data" box. To easy the interpretation of the performed analysis, clicking on one record only the parts involved in the selected interface are rendered (one in green and the other in red) together with additional elements communicating the relative degree of freedom, as described in section Interfaces Extraction and illustrated in Fig. 5.

If the user press the "Classify parts" button, once the classification is terminated, in the "Assembly data" box a new tab appears named "Part classification" showing the list of detected classes. Parts belonging to different classes are colored with different colors as described in Sect. Part Classification and illustrated in Fig. 6, anyhow the user can set a desired nuance by clicking on the button next to the classification name. To allow a functional visualization, it is also possible to hide/show a class unchecking/checking the corresponding flag among the element in the "Part classification" list.

The result of the last service is displayed in the "Patterns" tab of the "Assembly data" right box. As for the contacts, a user can visualize one pattern per time. Once a pattern is selected, the 3D view will shows it in the opaque mode

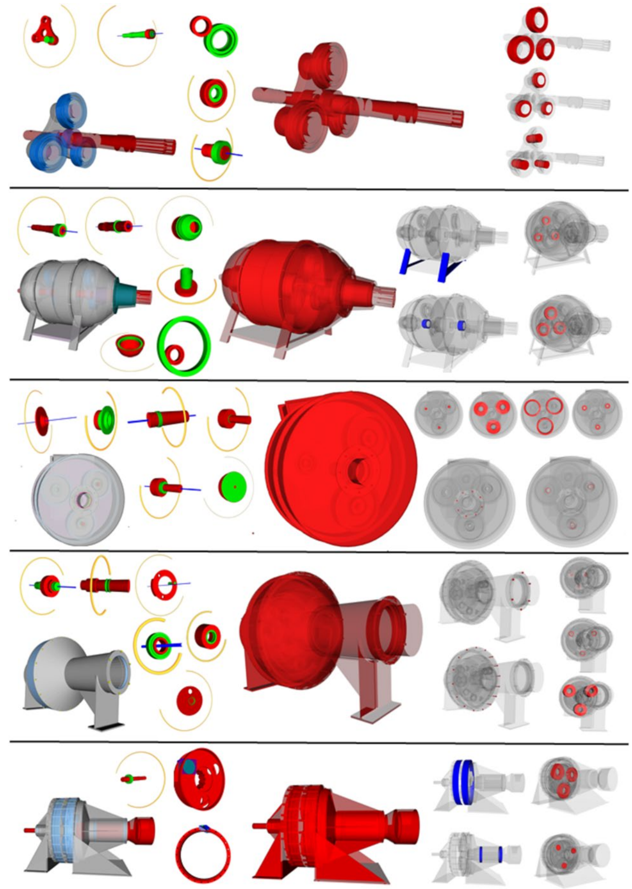


Fig. 9 Visualisation of sample models available on CAD3A (from the top, model 1, 2, 3, 4 and 10), and of the obtained results from the developed services: the part visualisation (in the center), examples of classified parts and extracted interfaces (left side), examples of extracted patterns, linear (in blue) and circular (in red) (right side)

within the assembly, as illustrated in Fig. 7, while in the "Patterns" tab-related information (e.g. the name and the number of parts involved and the distance between them) will be given.

Figure 9 shows a collections of models rendered in the 3D view according to the different user choices. For any model, in the center, left and right sides of the picture respectively, the figure shows (i) the default visualization of the "Part" tab, (ii) the highlight of selected results of the classification and interface computation, (iii) some patterns. These examples are available in the application by choosing the desired model in the predefined list of the "Select a model" item above "Import file" in CAD3A.

Conclusions

In this paper, a Web application to visualize and analyze 3D CAD assembly models has been introduced. The provided analysis capabilities allow the identification of the DOFs existing between two parts in contact, the classification of the parts according to the functional point of view and the detection of some regular arrangements (linear or circular) of repeated parts in the 3D space. Once the requested analysis has been performed, results are structured and stored in JSON format, which allows the exchange of data between client and server applications, and the visualization in the proposed Web application. This solution allows reviewing 3D CAD content on the Web just having a browser and an internet connection.

The proposed system has been tested over several assembly models (from simple models up to complex models made up of hundred pieces) to guarantee its reliability. From our tests, the system results robust and the provided rendering features help to verify the correctness of the proposed services with simplicity and immediacy. Indeed, through such CAD Web-application different services can be easily integrated no matter how they are developed.

In the future, there are several improvements we aim to achieve. First of all, the computational complexity of the single services should be faced to short the actual latency time mainly due to the current file read/write operations for module communication. In this perspective, to reduce the dimension of the saved files and to improve the performances when loading previously analysed models or simply converted in X3D format, we also plan to better exploit the information on the repeated parts, by simply storing one representative occurrence and the positions of the other occurrences.

Additional services are foreseen for the analysis of the assemblies, including the visualisation of the repeated parts independently of their belonging to regular patters. Moreover, we plan the integration of CAD3A as inspection tool of our public 3D CAD assembly dataset[2] conceived to support the evaluation and comparison of retrieval methods on assembly models. This would be the first step to achieve an online model and information retrieval on CAD assembly models designing new methods suitable for the Web.

Our preliminary usage tests with researchers demonstrate that the provided graphical user interface is of utmost value to support research activities. Anyhow, we aim at performing more structured user studies (involving different types of target users, e.g. researchers in different fields, mechanical engineers or designers) to test the GUI experience and validate its effectiveness.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

[2] http://3dassemblyrepository.ge.imati.cnr.it/index.html

References

1. 3dviewstation. https://www.3dviewstation.com/products/3dviewstation-webviewer-version.html
2. Attene M, Cabiddu D, Gagliardo S, Giannini F, Monti M. A Web repository to describe and execute shape oriented workflows. Computer-Aided Design and Applications. 2016;13(5):637–46. https://doi.org/10.1080/16864360.2016.1150710.
3. autodesk. https://viewer.autodesk.com/
4. Behr J, Eschler P, Jung Y, Zöllner M. ACM. X3DOM: a DOM-based HTML5/X3D integration model. In: Proceedings of the 14th International Conference on 3D Web Technology 2009. p. 127–35.
5. Bracci M, Tarini M, Pietroni N, Livesu M, Cignoni P. Hexalab.net: an online viewer for hexahedral meshes. Comput Aided Des. 2019;110:24–36.
6. Brutzman D, Daly L. X3D: extensible 3D graphics for Web authors. Elsevier; 2010.
7. Campen M. WebBSP 0.3 beta, 2010. http://www.graphics.rwth-aachen.de/webbsp
8. Chen L. In: Microservices: architecting for continuous delivery and devops. In: 2018 IEEE International Conference on Software Architecture (ICSA) 2018;39–397.
9. Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F, Ranzuglia G. MeshLab: an Open-Source Mesh Processing Tool. In: V. Scarano, R.D. Chiara, U. Erra (eds.) Eurographics Italian Chapter Conference. The Eurographics Association, 2008. https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136
10. Cignoni P, Idini M, Gabriele S, Giammori S, Delogu G.M, Busato D, Maggiordomo A, Nicoletti A, Loddo M, Marini G. MeshLabJS 2015. https://www.meshlabjs.net/
11. clari3d. www.clari3d.com/
12. Congote J, Segura A, Kabongo L, Moreno A, Posada J, Ruiz O. In: Interactive visualization of volumetric data with webgl in real-time, In: Proceedings of the 16th International Conference on 3D Web Technology, Web3D '11. New York, NY, USA: Association for Computing Machinery; 2011. p. 137–46. https://doi.org/10.1145/2010425.2010449.
13. creators3d. https://www.creators3d.com/online-viewer
14. Dragoni N, Giallorenzo S, Lafuente A.L, Mazzara M, Montesi F, Mustafin R, Safina L. Microservices: yesterday, today, and tomorrow. In: Present and ulterior software engineering, 2017;195–216. Springer
15. emachineshop. https://www.emachineshop.com/free-online-step-file-viewer/
16. Evans A, Romeo M, Bahrehmand A, Agenjo J, Blat J. 3D graphics on the web: A survey. Computers & Graphics. 2014;41:43–61.
17. Franke L, Haehn D. Modern scientific visualizations on the web. In: Informatics, vol. 7, p. 37. Multidisciplinary Digital Publishing Institute, 2020.
18. Ivson P, Nascimento D, Celes W, Barbosa SD. CasCADe: A novel 4D visualization system for virtual construction planning. IEEE transactions on visualization and computer graphics. 2017;24(1):687–97.
19. Jayanti S, Kalyanaraman Y, Iyer N, Ramani K. Developing an engineering shape benchmark for CAD models. Computer-Aided Design. 2006;38(9):939–53.
20. Kazhdan M. Rotation invariant shape descriptors, 2019. http://htmlpreview.github.io/?https://github.com/mkazhdan/ShapeSPH/blob/master/descriptors.html
21. Kazhdan M, Funkhouser T, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3D shape descriptors. Symposium on geometry processing. 2003;6:156–64.

22. Liu YS, Fang Y, Ramani K. Idss: deformation invariant signatures for molecular shape comparison. BMC bioinformatics. 2009;10(1):157.
23. Lupinetti K, Cabiddu D, Giannini F, Monti M. CAD3A: A Web-based application to visualize and semantically enhance CAD assembly models. In: 2019 15th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), 2019;462–469. https://doi.org/10.1109/SITIS.2019.00080
24. Lupinetti K, Chiang L, Giannini F, Monti M, Pernot J. Regular patterns of repeated elements in CAD assembly model retrieval. Computer-Aided Design and Applications. 2017;14(4):516–25. https://doi.org/10.1016/j.jcde.2015.02.001. https://doi.org/10.1080/16864360.2016.1257193
25. Lupinetti K, Giannini F, Monti M, Pernot JP. Automatic extraction of assembly component relationships for assembly model retrieval. Procedia CIRP. 2016;50:472–7.
26. Lupinetti K, Giannini F, Monti M, Pernot J.P. A 3D CAD Assembly Benchmark. In: S. Biasotti, G. Lavoué, R. Veltkamp (eds.) Eurographics Workshop on 3D Object Retrieval. The Eurographics Association, 2019. https://doi.org/10.2312/3dor.20191065
27. Lupinetti K, Giannini F, Monti M, Pernot JP. Content-based multicriteria similarity assessment of CAD assembly models. Computers in Industry. 2019;112:103111.
28. Lupinetti K, Pernot JP, Monti M, Giannini F. Content-based cad assembly model retrieval: Survey and future challenges. Computer-Aided Design. 2019;113:62–81.
29. Mahmoudi SE, Akhondi-Asl A, Rahmani R, Faghih-Roohi S, Taimouri V, Sabouri A, Soltanian-Zadeh H. Web-based interactive 2D/3D medical image processing and visualization software. Computer Methods and Programs in Biomedicine. 2010;98(2):172–82. https://doi.org/10.1016/j.cmpb.2009.11.012. https://www.sciencedirect.com/science/article/pii/S0169260709003022.
30. Mouton C, Sons K, Grimstead, I.:ACM. Collaborative visualization: current systems and future trends. In: Proceedings of the 16th International Conference on 3DWeb Technology, 2011;101–10.
31. Mwalongo F, Krone M, Reina G, Ertl T. State-of-the-art report in web-based visualization. Computer Graphics Forum. 2016;35(3):553–75. https://doi.org/10.1111/cgf.12929. https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12929.
32. Norton RL. Design of Machinery, vol. no. 3. McGraw-Hill Higher Education; 2003.
33. Nyamsuren P, Lee SH, Kim S. A Web-based revision control framework for 3D CAD model data. International Journal of Precision Engineering and Manufacturing. 2013;14(10):1797–803.
34. Park SC, Oh JW. Kinetic model extraction from a geometric model. Computer-Aided Design and Applications. 2015;12(3):338–43.
35. Potenziani M, Callieri M, Dellepiane M, Corsini M, Ponchio F, Scopigno R. 3DHOP: 3D heritage online presenter. Computers & Graphics. 2015;52:129–41.
36. Potenziani M, Callieri M, Dellepiane M, Scopigno R, et al. Publishing and consuming 3D content on the web: a survey. Foundations and Trends® in Computer Graphics and Vision. 2018;10(4):244–333.
37. Rose AS, Hildebrand PW. NGL viewer: a Web application for molecular visualization. Nucleic acids research. 2015;43(W1):W576–9.
38. Rucco M, Giannini F, Lupinetti K, Monti M. A methodology for part classification with supervised machine learning. AI EDAM. 2019;33(1):100–13.
39. Shahwan A, Foucault G, Léon JC, Fine L. Towards automated identification of functional designations of components based on geometric analysis of a DMU. In: GTMG2011- 12émes Journées du Groupe de Travail en Modélisation Géométrique, 2011.
40. Shahwan A, Foucault G, Léon J.C, Fine L. Deriving Functional Properties of Components from the Analysis of Digital Mock-ups. Engineering Computations 2014;16 . https://hal.inria.fr/hal-00922958
41. sharecad. https://beta.sharecad.org/
42. Simões B, Carretero, M.d.P., Santiago, J.M. . Photorealism and kinematics for web-based cad data. In: The 25th International Conference on 3D Web Technology. Web3D '20. New York, NY, USA: Association for Computing Machinery; 2020. https://doi.org/10.1145/3424616.3424710.
43. spinfire. https://www.actify.com/solutions/spinfire/
44. tenlinks. http://www.tenlinks.com/cad/products/viewers.htm
45. Vilmart H, Léon JC, Ulliana F. From cad assemblies toward knowledge-based assemblies using an intrinsic knowledge-based assembly model. Computer-Aided Design and Applications. 2018;15(3):300–17.
46. Virtual Visualisation Service. http://visionair.ge.imati.cnr.it/
47. Yiu Ip C, Regli WC. Content-based classification of CAD models with supervised learning. Computer-aided Design and Applications. 2005;2(5):609–17.