Identification de caractéristiques de forme et de structure d'assemblages de pièces CAO pour la recherche dans des bases de données

Katia Lupinetti

2018-ENAM-0003

École doctorale n° 432 : Science des Métiers de l'ingénieur

Doctorate ParisTech

THESIS

to obtain the degree of doctor issued by

l'École Nationale Supérieure d'Arts et Métiers

Spécialité : "Informatique - Traitement du signal"

and

l'Università degli Studi di Genova

Ingegneria meccanica, energetica e gestionale

Curriculum maccanica, misure e robotica (ciclo XXX)

Presented and defended publicly by

Katia LUPINETTI

January 24th, 2018

Identification of shape and structural characteristics in assembly models for retrieval applications

Directors of thesis: **Jean-Philippe PERNOT** and **Franca GIANNINI**

Co-director of thesis: **Marina MONTI**

Jury

M. Jean-Luc MARI, Professor, LIS, Aix-Marseille University	President
M. Nabil ANWER, Professor, LURPA, Paris-Sud University	Reviewer
M. Giorgio COLOMBO, Professor, Dipartimento di Meccanica, Politecnico di Milano	Reviewer
Mme. Rezia MOLFINO, Professor, DIME, Università degli studi di Genova	Examiner
M. Jean-Philippe PERNOT, Professor, Arts et Métiers ParisTech,	Examiner
Mme. Franca GIANNINI, Research director, IMATI-CNR Genova	Examiner
Mme. Marina MONTI, Senior researcher, IMATI-CNR Genova	Examiner

T H E S I S

Arts et Métiers ParisTech - Centre d'Aix-en-Provencce, LSIS, France
Università degli Studi di Genova, CNR-IMATI. Ge, Italia

Success is the ability to go from one failure
to another with no loss of enthusiasm.

Sir Winston Churchill

Acknowledgments

The present thesis was developed in the framework of a co-tutoring agreement between the Università degli Studi di Genova and the Arts et Métiers ParisTech (campus of Aix-en-Provence).
The financial support of the Institute of Applied Mathematics and Information Technology (IMATI) of the National Council of Research (CNR) in Genoa is gratefully acknowledged.

I would like to thank the members of the jury for accepting to review my PhD thesis. In particular, I express my gratitude to **Nabil ANWER**, **Jean-Luc MARI**, **Giorgio COLOMBO** and **Rezia MOLFINO** for taking the time and effort during the Christmas period to examine and evaluate this work and for the fruitful discussions on the perspectives of this thesis.

The accomplishment of this work would not be possible without the support of all my advisors, thus I would like to say "thank you" to **Franca GIANNINI**, **Marina MONTI** and **Jean-Philippe PERNOT**. In particular, Franca and Marina, who supervised also my master degree and, despite that experience, they didn't desist to follow up my PhD. And Jean-Philippe, who was my supervisor in France and opened my mind to the amusing word of mechanical engineering.
Every one of them has passed down to me different aspects and advices on both research career as well as on personal life. I would like to express them how much I feel lucky for the stunning persons they have been for me and I'm honored for having had the chance to learn from them during the time spent together. Their encouragement, trust, and guidance have been priceless for me to conclude this adventure and to continue my journey.
Thank you for having been much more than just PhD advisors.

Thanks are also extended to every member of the **IMATI** institute over the years I worked there, not only for sharing scientific ideas during my work but also for the kindness and friendly time we have spent together. Among them I have found not only colleges but friends to share the life with. I'm sure we will keep in touch for a long while.
I extend my gratitude to all the members of the **LSIS** laboratory who welcomed me when I arrived in France. Parmi tous, Je voudrais remercier le directeur du laboratoire LSIS **Lionel ROUCOULES**, qui m'a encouragée tout le temps à parler en française ; **Grazyna CAUQUIL** pour son grand aide et patience à m'apprendre le française et pour tous nos cafés du matin plein de conseils ; **Romain PINQUIE** et **Yòsbel GALVIS** pour nos pauses de l'après-midi et pour notre dînes ensemble. Vous êtes ma famille française.

I would like to thank my parents, my sisters, my brother and my friends, for their endless love, patience, support all along those years and for their effort trying to understand what I was doing.

And finally to the person without whom nothing of this had could be possible: my husband. Also this time, the biggest THANK YOU is for him, because he, more that everyone else, has shared every single moment of this journey.
Because for every success, every cry (and there were many), every race against the time, every computer issue to be solved, you were there for me. I know I can succeed in everything I do because of you.
Thank you for having supported me, advised me, spurred me and putting up with me all the times even when was very difficult staying close to me. For this and much more (I don't want bother who reads) thank you!

Sincerely,
Katia

Contents

Abbreviation

ABS Anti-lock Braking System

AP Application Protocol

API Application Programming Interface

B-rep Boundary Representation

CAD Computer-Aided Design

CAM Computer-Aided Manufacturing

CGS Constructive Solid Geometry

CRM Customer Relationship Management

DMU Digital Mock-Up

DOF Degree Of Freedom

EAM Enriched Assembly Model

EDB Electronic Brake-force Distribution

ERP Enterprise Resource Planning

ESP Electronic Stability Control

FAG Face Adjacency Graph

GFAG General Face Adjacency Graph

JSON JavaScript Object Notation

MCS Maximum Common Subgraph

MC Maximum Clique

MPM Manufacturing Process Management

PDM Product Data Management

PDP Product Development Process

PLM Product Lifecycle Management

PPM Project Portfolio Management

SCM Supply Chain Management

STEP Standard for the Exchange of Product model data

UI User Interface

Glossary

$\boldsymbol{\mathcal{A}_C}$ The set of arcs that represent the contacts between parts

$\boldsymbol{\mathcal{A}_J}$ The set of arcs that represent the joints between parts

$\boldsymbol{\mathcal{A}_S}$ The set of arcs that represent the assembly hierarchical structure

$\boldsymbol{a_k^{hl}}$ The *arc* of $\boldsymbol{G_k}$ between the nodes $\boldsymbol{(n_k^h, n_k^l)}$

$\boldsymbol{a_q^{ij}}$ The *arc* of $\boldsymbol{G_q}$ between the nodes $\boldsymbol{(n_q^i, n_q^j)}$

$\boldsymbol{c_A}$ Vector with the criteria of similarity of the relationships

$\boldsymbol{c_N}$ Vector with the criteria of similarity of the parts

$\boldsymbol{(C_{q,k,c_N,c_A})_h}$ The generic *h-th* clique in the graph $\boldsymbol{G_{q,k,c_N,c_A}}$

$\boldsymbol{G}$ The attributed multi-graph representation of the EAM descriptor of a generic model

$\boldsymbol{G_k}$ The attributed multi-graph representation of the EAM descriptor of the *k-th target* model in the dataset

$\boldsymbol{G_q}$ The attributed multi-graph representation of the EAM descriptor of the *query* model

$\boldsymbol{G_{q,k,c_N,c_A}}$ The association graph between $\boldsymbol{G_q}$ and $\boldsymbol{G_k}$

$\boldsymbol{\mathcal{N}_A}$ The set of nodes associated with sub-assemblies

$\boldsymbol{\mathcal{N}_P}$ The set of nodes associated with parts

$\boldsymbol{n_k^i}$ The *i-th node* of $\boldsymbol{G_k}$

$\boldsymbol{n_q^i}$ The *i-th node* of $\boldsymbol{G_q}$

$\boldsymbol{T_{\mathcal{A}_C}}$ The set of attributes for arcs that represent the contacts between parts

$\boldsymbol{T_{\mathcal{A}_J}}$ The set of attributes for arcs that represent the joints between parts

$\boldsymbol{T_{\mathcal{A}_S}}$ The set of attributes for arcs that represent the assembly hierarchical structure

$\boldsymbol{T_{\mathcal{N}_A}}$ The set of attributes for nodes associated with sub-assemblies

$\boldsymbol{T_{\mathcal{N}_P}}$ The set of attributes for nodes associated with parts

$\boldsymbol{\alpha_C}$ *Contact* similarity criteria

$\boldsymbol{\alpha_{C_{num}}}$ *Allowed DOF for contact* similarity criteria

$\boldsymbol{\alpha_{CT}}$ *Component type* similarity criteria

$\boldsymbol{\alpha_{J_{num}}}$ *Allowed DOF for joint* similarity criteria

$\boldsymbol{\alpha_{PT}}$ *Pattern type* similarity criteria

$\boldsymbol{\alpha_{Sh}}$ *Shape* similarity criteria

$\boldsymbol{\alpha_{Si}}$ *Size* similarity criteria

$\boldsymbol{\eta}$ Local similarity measure

$\boldsymbol{\mu^{joint}}$ Joint similarity measure

$\boldsymbol{\mu^{position}}$ Position similarity measure

$\boldsymbol{\mu^{shape}}$ Shape similarity measure

$\boldsymbol{\mu^{structure}}$ Structure similarity measure

$\boldsymbol{\varphi}$ Partial similarity measure

$\boldsymbol{\phi}$ Global similarity measure

$\boldsymbol{\Phi_{\mathcal{A}_C}}$ The function that pairs an arc that represents the contacts between parts with a value of attributes in $\boldsymbol{T_{\mathcal{A}_C}}$

$\boldsymbol{\Phi_{\mathcal{A}_J}}$ The function that pairs an arc that represents the joint between parts with a value of attributes in $\boldsymbol{T_{\mathcal{A}_J}}$

$\boldsymbol{\Phi_{\mathcal{N}_A}}$ The function that pairs a node associated with sub-assemblies with a value of attributes in $\boldsymbol{T_{\mathcal{N}_A}}$

$\boldsymbol{\Phi_{\mathcal{N}_P}}$ The function that pairs a node associated with parts with a value of attributes in $\boldsymbol{T_{\mathcal{N}_P}}$

Résumé long en Française

Contents

1 Introduction général

La capacité de récupérer un élément spécifique parmi beaucoup objets est très significant. De nos jours les systèmes de récupération de texte peuvent extraire des informations à partir d'un texte et fournir des résultats significatifs (e.g. Google™, Bing™ or Yahoo™), par contre les systèmes de récupération de contenu 3D ont encore une grande marge d'amélioration [113]. La récupération de modelés similaires représente un moyen de faciliter la conception de nouveaux produits permettant la réutilisation des modèles CAO existants et l'accès aux connaissances intégrées [9]. Il a été estimé qu'une grande partie de l'activité de conception est basée sur la réutilisation de solutions précédentes pour résoudre de nouveaux problèmes de conception [56, 124]. En effet, plus de 75% de l'activité de conception comprend la réutilisation de connaissances déjà existantes [50]. Par conséquent, les recherches de modèles 3D sont utiles pour éviter de passer du temps à réinventer ou à re-concevoir des solutions existantes.

La récupération de modèles CAO 3D ne concerne pas uniquement la réutilisation de modèles existants ou de leurs informations associées. En fait, elle peut fournir des avantages dans diverses activités d'ingénierie [18]. Par exemple, dans la normalisation et la rationalisation des produits, la récupération de modèles similaires peut aider à identifier des pièces interchangeables à partir de projets distincts afin de réduire les coûts de gestion et de fabrication. Une autre application pour la récupération de modèles CAO 3D est de supporter la gestion des opérations de maintenance. En effet, connaissant le taux d'usure d'un composant C, l'identification dans les modèles CAO de composants similaires à C est utile pour planifier les opérations de maintenance en vue d'organiser les stocks dans un entrepôt.

Depuis un certain temps, les outils de Conception Assistée par Ordinateur (CAO) sont intégrés aux systèmes PDP (Product Data Management), qui supportent le processus de conception en stockant et en indexant les données du projet pour leur récupération rapide. Le suivi des données implique généralement les spécifications techniques du produit, les dispositions pour sa fabrication et son assemblage, les types de matériaux qui seront nécessaires pour le produire et d'autres informations. Cependant, les informations de conception sont pour la plupart contextuelles, c'est-à-dire que de nombreuses informations sont contenues dans le modèle CAO lui-même, et de tels systèmes fournissent un support limité dans les recherches géométriques [56].

Pour surmonter ces limitations, pour la récupération de modèles 3D, des méthodes basées sur le contenu sont développées en fonction de descriptions géométriques. La récupération de modèles 3D utilisant des caractéristiques géométriques a été étudiée en profondeur [8, 25, 41, 57, 117], où la pratique courante voit l'utilisation de descripteurs de formes pré-calculées ou des signatures qui facilitant la récupération de forme similaire. Cependant, dans le cas de produits complexes constitués de plusieurs pièces, une méthode basée uniquement sur la forme n'est pas suffisante pour récupérer le modèle d'assemblage. En fait, les modèles 3D avec des formes similaires peuvent être assemblés de différentes manières, impliquant des caractéristiques cinématiques différentes et ensuite des relations différentes entre leurs pièces.

1.1 Récupération d'assemblage : problèmes

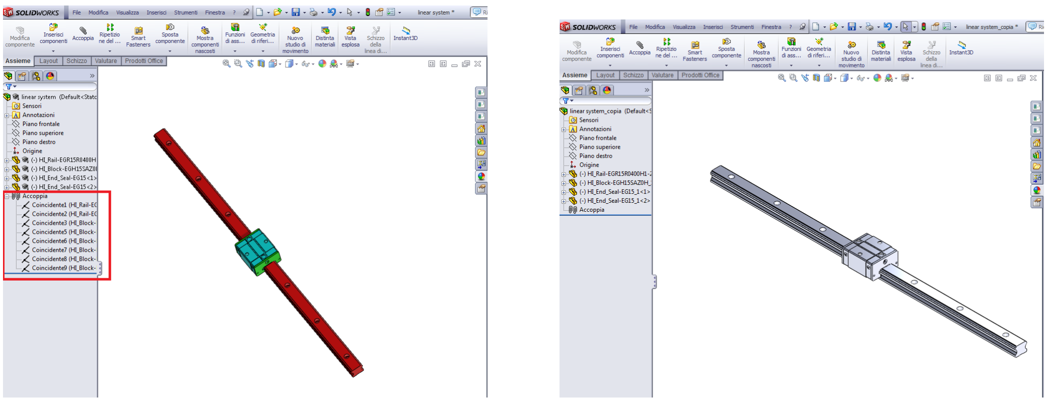
Dans cette section, nous discutons des principaux problèmes dans les systèmes de récupération d'assemblage.

- **Problèmes de taille**
  La popularité des systèmes numériques dans les industries a donné naissance à un grand nombre de modèles de CAO et la taille d'une modèle peut comprendre plus de 1 million de pièces représentant plusieurs de tera-octets de données [52]. La taille et le nombre de modèles présents dans les bases de données rendent difficile la mise en place d'un système de récupération efficace qui répond aux besoins des utilisateurs.

- **Problèmes de nommage**
  Les systèmes de récupération de texte reposent strictement sur l'intégration manuelle des annotations. En utilisant ce type de interrogation, des modèles avec la même forme pourraient ne pas être récupérés, car ils n'ont pas le même texte dans leurs annotations.

- **Problèmes de stockage**
  De nombreuses informations importantes (comme les liens cinématiques, les contraintes ou les attributs) ne sont pas toujours explicitement stockées dans les modèles CAO. De plus, certains problèmes peuvent survenir en utilisant des formats de fichiers standard pour échanger des modèles CAO.

  Un exemple est rapporté dans Figure 1, où un modèle a été créé avec le logiciel CAO SolidWorks®en positionnant les pièces par contraintes d'accouplement (voir Figure 1(a)). Une fois le modèle a été stocké en STEP 214 et rouvert (toujours dans SolidWorks®), les informations des contraintes ne sont plus disponible (voir Figure 1(b)).



(a) Original model　　(b) STEP file

Figure 1: Exemple de perte d'information stockant un modèle au format STEP 214

- **Problèmes de similarité**
Deux modèles peuvent être similaire selon différentes caractéristiques. Par exemple, à des fins de visualisation, les modèles d'assemblage peuvent être considérés comme similaires s'ils ont la même forme globale, mais du point de vue des opérations d'assemblage, les conditions d'accouplement déterminent les relations spatiales entre les pièces et la dimension de l'assemblage. De plus, leur peuvent avoir différents *type de similitudes*. Par exemple en Figure 2, le modèle M_1 est *globalement similaire* au modèle M_2, car ils ont pièces analogues. Ensuite, les deux modèles M_1 et M_2 sont considérés *partiellement similaires* à M_3 et M_4, puisque les deux premiers sont complètement inclus dans la seconde deux. En fin, les modèles M_3 et M_4 sont *localement similaires*, puisqu'ils partagent des pièces similaires.

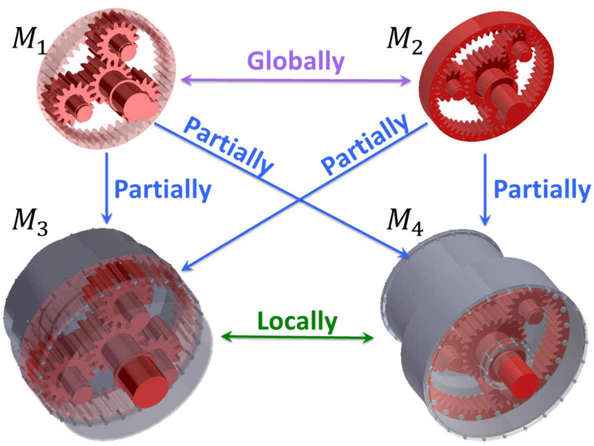
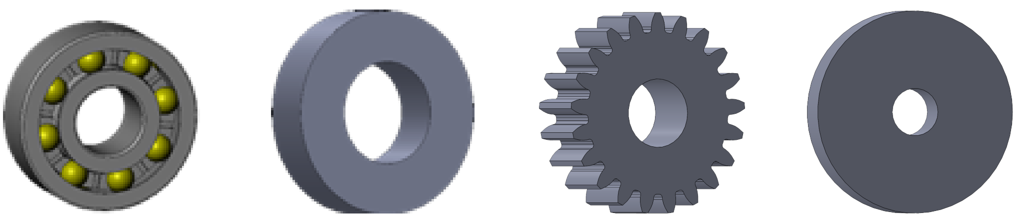


Figure 2: Exemple de différents types de similitude

- **Problèmes de représentation**
Les éléments ayant la même forme peuvent avoir une fonctionnalité complète différente et vice versa. Par exemple, les roulements décrits dans Figure 3(a) et Figure 3(b), ainsi que les engrenages Figure 3(c) et Figure 3(d) ont des formes très différentes même s'ils identifient les mêmes composants. Inversement, Figure 3(b) Figure 3(d) ont une forme très similaire mais une fonction complètement différente.



(a) Detailed bearing (b) Simplified bearing (c) Detailed gear (d) Simplified gear

Figure 3: Exemples de composants avec différentes formes et fonctions

Proposition

Pour relever les défis de la récupération de modèles d'assemblage, nous proposons un nouveau descripteur de modèle d'assemblage, à travers lequel nous pouvons représenter des différents types d'informations d'un modèle. Le descripteur est organisé en quatre couches différentes, qui incluent les différents types d'informations présentes dans les modèles d'assemblage. Les informations présentes dans le descripteur sont automatiquement extraites en analysant les modèles et en gérant les éventuels problèmes découlant de leur conception.

La représentation choisie est basée sur un graphe, car elle convient pour des opérations de correspondance partielle et locale. En particulier, nous utilisons une structure multigraphes attribuée, où les nœuds correspondent et pièces aux composants de l'assemblage et les arcs codent différents types de relations entre les composants. Les attributs permettent d'inclure et d'organiser les informations de l'objet représenté. En utilisant cette représentation, la comparaison de deux modèles d'assemblage peut être basée sur des techniques d'appariement de graphes. Dans cette thèse, l'appariement de graphe a été formulé comme un problème de recherche du Maximal Sous-graphe Commun (MCS) entre deux multi-graphes attribués. Ensuite, la détection du MCS entre deux multi-graphes attribués a été réduite à la détermination de la Clique Maximale (MC) dans un graphe d'association, qui est défini sur la base des critères de similarité choisis par l'utilisateur.

Puisque deux modèles d'assemblage peuvent être similaires selon plusieurs critères, nous cherchons à évaluer différents types de similarité entre les modèles d'assemblage en définissant un ensemble de mesures appropriées. Ensuite, une mesure de similarité unique est définie combinant l'ensemble précédent de mesures de similarité. En fin, dans le but de faciliter la compréhension de l'utilisateur, nous visons à définir également une visualisation efficace des résultats récupérés.

Considérant les principaux problèmes de recherche de modèles d'assemblage, nous avons pour objectif de définir un système de recherche fortement basé sur les informations présentes dans le modèle géométrique lui-même et capable d'extraire automatiquement les données nécessaires pour éviter que l'utilisateur doit ajouter aux modèles les informations manquantes. De plus, étant donné qu'un modèle d'assemblage peut inclure des composants décrits de manière simplifiée et, par conséquent, difficiles à identifier, nous allons exploiter le contexte d'utilisation pour mieux caractériser les composants d'assemblage.

2 Revue de littérature

Le problème de la récupération de forme 3D a été largement étudié ces dernières années et un grand nombre de travaux existent sur ce sujet, traitant à la fois du modèle représenté par des maillages 3D et des modèles représentés par B-Rep [13, 25, 24, 30, 36, 37, 43, 47, 49, 53, 57, 117, 132]. Bien que ces techniques soient capables de récupérer des parties uniques de modèles d'assemblage, elles ne prennent pas en compte les relations entre les pièces et donc ne sont pas vraiment utiles pour la description et la récupération des assemblages.

Pour surmonter ces limites, plus récemment, des efforts ont été consacrés à la récupération des assemblages. Pour approfondir l'analyse des techniques traitant directement ou indirectement de l'identification des similarités des modèles d'assemblage, nous utilisons des critères qui peuvent être regroupés dans les cinq macro-catégories suivantes:

- contexte,
- caractérisation de modèle d'assemblage
- caractérisation de descripteur
- modèle de requête adopté
- type de similitude abordé

Table 1 résume les travaux qui traitent de la similarité du modèle d'assemblage avec différentes fins. Pour faciliter la lecture, les symboles suivants sont utilisés dans le tableau:

- C: calculé,
- R: lu,
- PC: partiellement calculé,
- ●: complet,
- ◐: incomplet.

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
Retrieval of solid models based on assembly similarity [100]	Product information reuse	Search for models with similar assembly work instructions	Curvature	-	-	Structure	-	-	-	Assembly	-	Yes	Global	Part model	●	✓	-	-
			Dihedral angle	-	-	Kinematic link	-	-		Part	✓							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	✓	C													
			Set of 2D projections	-	-													
			Angle distance	-	-													
A matching method for 3D CAD models with different assembly structures using projections of weighted components [61]	Design model reuse	Search for globally similar assembly models with different hierarchical structure	Curvature	-	-	Structure	-	-	-	Assembly	✓	Yes	Global	Assembly model	●	✓	-	-
			Dihedral angle	-	-	Kinematic link	-	-		Part	-							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	-	-													
			Set of 2D projections	✓	C													
			Angle distance	-	-													

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
An assembly retrieval approach based on shape distributions and Earth mover's distance [125]	Design model reuse	Search for globally similar assembly models	Curvature	-	-	Structure	-	-	-	Assembly	-	No	Global	Assembly model	◐	✓	-	-
			Dihedral angle	-	-	Kinematic link	-	-		Part	✓							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	✓	C													
			Set of 2D projections	-	-													
			Angle distance	-	-													
Assembly model retrieval based on optimal matching [118]	Design model reuse	Search for globally similar assembly models	Curvature	-	-	Structure	-	-	Yes	Assembly	✓	Yes	Global and local	Assembly model	●	✓	-	-
			Dihedral angle	-	-	Kinematic link	✓	R		Part	✓							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	✓	R	Part arrangement	-	-										
			Shape distribution	-	-													
			Set of 2D projections	-	-													
			Angle distance	-	-													

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
3D shape retrieval considering assembly structure [83]	Design model reuse	Search for globally similar assembly models	Curvature	-	-	Structure	-	-	No	Assembly	✓	Yes	Global and local	Assembly model	●	✓	-	-
			Dihedral angle	-	-	Kinematic link	✓	R		Part	✓							
			Part statistics	-	-	Geometric constraints	✓	R		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	-	-													
			Set of 2D projections	-	-													
			Angle distance	✓	C													
Content-based assembly search: A step towards assembly reuse [32]	Product information reuse	Search for globally and partially similar assembly models	Curvature	-	-	Structure	-	-	Yes	Assembly	✓	Yes	Global and local	Mating graph	●/◐	✓	✓	-
			Dihedral angle	-	-	Kinematic link	✓	R		Part	✓							
			Part statistics	-	-	Geometric constraints	✓	R		Feature	-							
			Geometric info	✓	R	Part arrangement	-	-										
			Shape distribution	-	-													
			Set of 2D projections	-	-													
			Angle distance	-	-													

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
Relaxed lightweight assembly retrieval using vector space model [51]	Design model reuse	Search for globally and partially similar assembly models	Curvature	-	-	Structure	✓	R	No	Assembly	✓	Yes	Global	List of parts	◐	✓	✓	-
			Dihedral angle	-	-	Kinematic link	-	-		Part	-							
			Part statistics	✓	C	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	-	-													
			Set of 2D projections	-	-													
			Angle distance	-	-													
A geometric reasoning approach to hierarchical representation for B-rep model retrieval [70]	Design model reuse	Search for globally similar assembly models with different hierarchical structure	Curvature	-	-	Structure	-	-	-	Assembly	✓	No	Global and local	Part model	●	✓	✓	-
			Dihedral angle	-	-	Kinematic link	-	-		Part	✓							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	✓	C													
			Set of 2D projections	-	-													
			Angle distance	-	-													

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
A similarity-based reuse system for injection mold design in automative interior industry [71]	Design model reuse	Search for similar assemblies to design reuse or mold planning reference	Curvature	-	-	Structure	✓	R	Yes	Assembly	✓	Yes	Global and local	Part or assembly model	●	✓	✓	-
			Dihedral angle	-	-	Kinematic link	✓	R		Part	✓							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	✓	C													
			Set of 2D projections	-	-													
			Angle distance	-	-													
Generic face adjacency graph for automatic common design structure discovery in assembly models [134]	Design model reuse	Search for frequent similar sub-assembly models	Curvature	✓	C	Structure	✓	R	No	Assembly	✓	No	Global and local	Assembly model set	●	✓	✓	✓
			Dihedral angle	✓	C	Kinematic link	✓	R		Part	✓							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	✓	C	Part arrangement	-	-										
			Shape distribution	-	-													
			Set of 2D projections	-	-													
			Angle distance	-	-													

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
Reuse-oriented common structure discovery in assembly models [126]	Design model reuse	Search for frequent similar sub-assembly models	Curvature	-	-	Structure	-	-	-	Assembly	✓	Yes	Local	Assembly model set	●	✓	✓	✓
			Dihedral angle	-	-	Kinematic link	✓	C		Part	✓							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	✓	C													
			Set of 2D projections	-	-													
			Angle distance	-	-													
A flexible assembly retrieval approach for model reuse [28]	Design model reuse	Search for globally and partially similar assembly models	Curvature	-	-	Structure	✓	C	Yes	Assembly	✓	Yes	Global and local	Assembly model	●/◐	✓	✓	-
			Dihedral angle	-	-	Kinematic link	✓	PC		Part	✓							
			Part statistics	-	-	Geometric constraints	✓	PC		Feature	-							
			Geometric info	-	-	Part arrangement	✓	C										
			Shape distribution	✓	C													
			Set of 2D projections	-	-													
			Angle distance	-	-													

Table 1: Summary of the assembly retrieval methods

De l'analyse de ces travaux, nous pouvons observer que la quasi-totalité d'entre eux font face à la réutilisation du modèle de conception et supposent la pleine disponibilité de l'information nécessaire pour dériver leurs descripteurs de modèle d'assemblage. Ceci peut être une grosse limitation, car toutes les données nécessaires ne sont pas présentes dans le modèle CAO. Supposer qu'un utilisateur ajoute toutes les données manquantes n'est pas raisonnable, car cette pratique est ennuyeuse pour l'utilisateur.
De plus, certains travaux pourraient être étendus pour caractériser les modèles d'assemblage au niveau local mais ils n'exploitent pas cette fonctionnalité.

En général, nous observons que la recherche concernant la caractérisation et la récupération des modèles d'assemblage CAO est toujours en cours et de nombreux objectifs doivent encore être atteints. Peu de problèmes présentés dans la section 1.1 sont résolus. La plupart des travaux traitent le problème de la récupération des modèles partiellement similaires; qui de toute façon ont des solutions qui ne prennent en compte que la structure hiérarchique. Toutefois, la limitation est que la similarité de structure devient une contrainte et que l'utilisateur ne peut pas récupérer un modèle de assemblage inclus dans un de plus grand si la requête n'est pas représentée comme un sous-assemblage dans le modèle cible. Cette hypothèse peut affecter des scénarios dont le but est la maintenance des composants d'assemblage. En effet, dans cette situation, un composant inclus dans un modèle d'assemblage doit être identifié malgré sa structure conçue. De plus, tous les travaux supposent que le modèle de requête a le même nombre de composants ou moins que le modèle cible, à l'exclusion du cas où le modèle de requête est plus grand que la cible.

Peu de travaux traitent de différents types de similitudes entre les modèles d'assemblage et généralement la géométrie, la taille des pièces d'assemblage et le type différent de leurs relations sont utilisés. Cependant, l'extraction de ces données n'est pas confrontée et l'information est supposée être disponible ou ajoutée par l'utilisateur. De plus, la pratique consistant à caractériser des pièces par leur forme ne permet pas de traiter les éventuelles descriptions simplifiées des composants dans les modèles d'assemblage.

Considérant les principales limites rencontrées dans l'état de l'art de la récupération d'assemblages, nous visons d'abord à surmonter le problème de l'extraction automatique des algorithmes de raisonnement par l'information sur la représentation géométrique des modèles.

3 Cadre de récupération d'assemblage

L'évaluation de la similarité est une procédure en deux étapes. Au début, nous définissons un descripteur approprié des modèles que nous voulons évaluer, puis nous utilisons cette signature pour la comparaison et l'évaluation de la similarité.

Notre cadre général est illustré à la Figure 4 et il est basé sur le descripteur EAM (Enriched Assembly Model) [12]. Pour chaque modèle CAO dans la base de données et le modèle requête, un descripteur EAM est calculé. Ensuite, nous comparons deux EAM pour évaluer leur similarité selon plusieurs critères. Les résultats sont stockés et fournis à l'utilisateur, qui peut les classer en fonction du niveau de similitude qui l'intéresse. Le cadre comprend des processus exécutent à la fois en avance et en temps réel. Le processus exécuté en avance crée un descripteur complet pour chaque assemblage dans la base de donnée, tandis que les processus en temps réel calculent un descripteur EAM partiel pour le modèle requête et effectuent la comparaison.

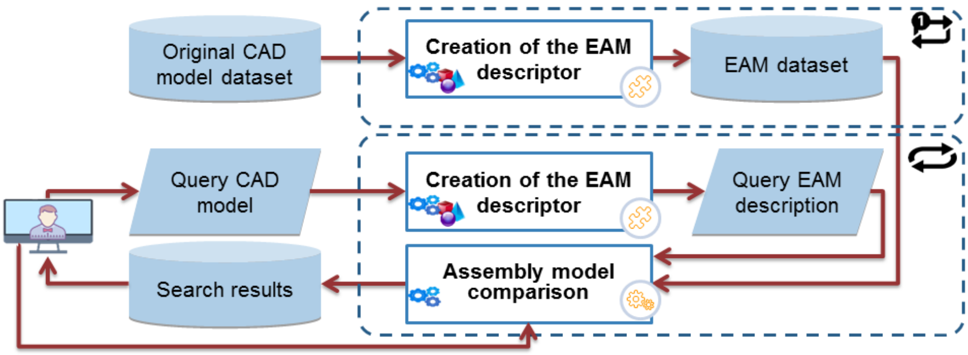


Figure 4: Cadre de récupération d'assemblage

Le descripteur multi-couche appelé Entiched Assembly Model (EAM) été composé par quatre couches conceptuelles différentes: structure, interface, forme et statistique. Chaque couche caractérise les assemblages à différents niveaux de détail, les données de chaque couche sont illustrées en Figure 5.

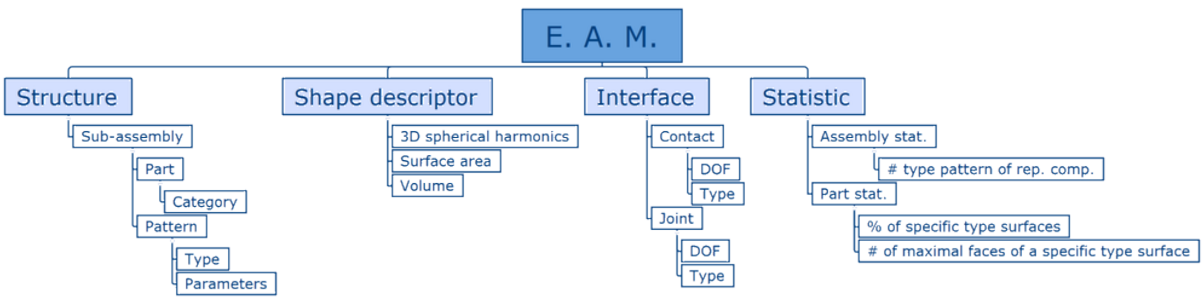


Figure 5: The multi-layered Enriched Assembly Model

Toutes les données dans le EAM sont extraits automatiquement en analysant la géométrie et la structure d'un modèle CAO. Ces données sont ensuite organisées dans une structure multi-graphe attribuée. L'EAM permet l'utilisation de plusieurs arcs, c'est-à-dire qu'il est possible que deux arcs ou plus soient incidents à la même paire de nœuds [119]. Cette particularité est générée par la nature différente des informations stockées, telles que la structure et les informations d'interface, qui définissent différents types de relations entre les composants d'assemblage. En utilisant la représentation graphe, le problème de récupérer modèles similaires est transposé en trouver les modèles qui ont descripteurs EAM similaires à le modèle donné comme requête.

Les sections suivantes décrivent avec plus de détails les différentes couches EAM avec leurs données.

3.1 Couche de structure

La couche de structure de l'EAM code la *structure du produit* (c'est-à-dire comment les pièces sont rassemblées dans le modèle comme spécifié par le concepteur) et deux attributs (*Pattern_List* and *Component_Type*), qui sont utilisés pour caractériser les composants de l'assemblage.

Structure du produit

Definition 3.1. La *structure du produit* est une décomposition hiérarchique d'un produit (un modèle d'assemblage) en termes de sous-ensembles du modèle CAO jusqu'à ses pièces constituantes.

La structure du produit définit la relation "fait-de" entre des sous-assemblages et des pièces, qui est représentée dans l'EAM par des arcs dirigés entre les nœuds avec la signification suivante:

- le nœud racine correspond à l'ensemble du modèle d'assemblage;
- les nœuds intermédiaires représentent des sous-assemblages;
- les feuilles sont associées aux pièces constituant de le modèle d'assemblage.

Figure 6 montre un exemple de la couche de structure d'un EAM. L'objet est un moteur dont le premier niveau est divisé en trois sous-ensembles: un piston (S1), un vilebrequin (S2), une masse (S3) et deux pièces de liaison (P3, P10). Au second niveau, il y ont les composants des sous-ensembles S1, S2 et S3.

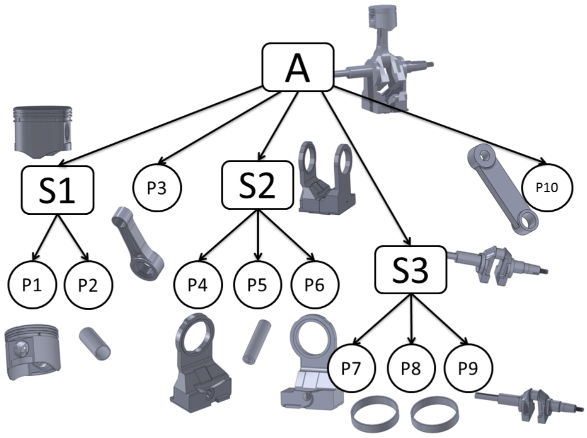


Figure 6: Exemple de couche de structure d'un EAM

Dans cette couche, les informations sur les composants d'un modèle d'assemblage sont ajoutées par l'utilisation d'attributs. Les attributs considérés spécifient la disposition (*Pattern_List*) des pièces répétées dans le modèle d'assemblage et le type d'éléments représentés par les nœuds de l'EAM (*Component_Type*):

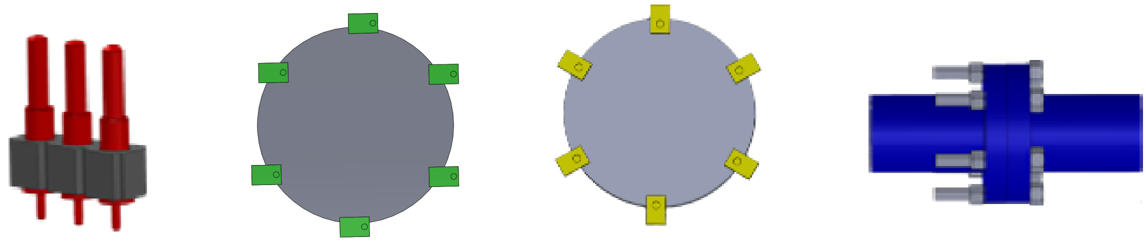
- Un *pattern* est un arrangement régulier de pièces répétées dont les centres de gravité sont équidistants. L'attribut *Pattern_List* est associé au nœud racine, pour gérer le caractère non unique de la décomposition structurelle, puisqu'il fournit des informations sur la position mutuelle des pièces.
  Ces configurations affectent la production, y compris les opérations d'assemblage. Par conséquent, leur présence peut également être utilisée dans l'évaluation de la similarité entre les assemblages.

  L'attribut *Pattern_List* est caractérisé comme suit:

$$Pattern\_List = \{Pattern \times RP\} \quad (1)$$
$$Pattern = Pattern\_Type \times Parameters \quad (2)$$

  où RP représente la liste des pièces répétées.

  Les types de *pattern* considérés et détectés (*Pattern_Type*) sont: **translation linéaire** (pins rouges en Figure 7(a)), **translation circulaire** (plaques vertes en Figure 7(b)), **rotation circulaire** (plaques jaunes en Figure 7(c)) et **réflexion** (bride bleue en Figure 7(d)) [29, 76]. Dans le cas de deux entités répétées, nous supposons que les configurations possibles sont une translation linéaires et une réflexion, puisque pour définir un *pattern* circulaire, l'algorithme de [76] requiert au moins trois entités répetées. L'algorithme pour leur détection est basé sur le travail de Chiang et al. [29] et sa généralisation sur les assemblages [76].



(a) Translation linéaire (b) Translation circulaire (c) Rotation circulaire (d) Réflexion

Figure 7: Exemples de modèles avec des pièces répétées

Afin de caractériser la position des pièces répétées, chaque type de *pattern* a un ensemble de paramètres qui fournissent également une indication sur la taille du *pattern*. Plus précisément, pour chaque type de pattern, on lui attribue le *step*, qui indique la distance entre chaque élément répété, et le *numéro de pièces répétées*, tandis que *pattern center*, *radius* et *angle* sont assignés pour les *pattern* de translation et de rotation circulaires. L'ensemble des paramètres est résumé dans Table 2.

Table 2: Paramètre défini en fonction du type de pattern

	Step	Number of repeated parts	Center, radius, angle
Linear translation	✓	✓	
Circular translation	✓	✓	✓
Circular rotation	✓	✓	✓
Reflection	✓	(=2)	

Figure 8 montre un *pattern* circulaire de vis (cercle bleu) avec ses paramètres : le *step* en violet, le *radius* en vert, le *center* en rouge et le *angle* en jaune.

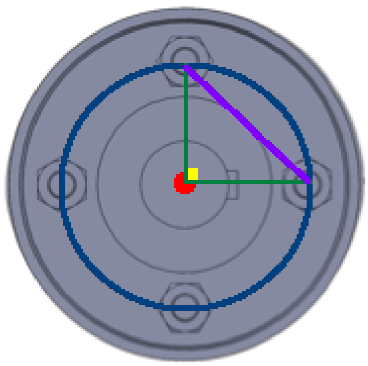


Figure 8: Exemple d'attributs d'un motif circulaire

- Les produits peuvent être classés en catégorie en fonction de leur fonction [58]. Afin de prendre en charge la récupération des modèles d'assemblage en fonction de la fonction de leurs composants, nous considérons l'attribut *Component_Type*, où il est défini comme suit:

$$\begin{aligned} Component\_Type \;=\; & \{\text{bearing, c-clip, cylinder like, cube like, gear, key,} \\ & \text{linkage arm, nut, parts of bearing, screw and bolt,} \\ & \text{shaft, spacer, sphere like, torus like, miscellaneous}\} \end{aligned} \tag{3}$$

  Ces catégories ont été sélectionnées en étendant la "classification fonctionnelle" du National Design Repository [99], qui consiste de 70 modèles organisés en 7 classes. Donc, la classification proposée a 2354 éléments (obtenus à partir de référentiels en ligne [2, 4] et des projets de les étudiants de l'école d'ingénierie) dans les 15 classes indiquées par l'ensemble *Component_Type*. La validation de l'appartenance à une classe a été réalisée au moyen d'entretiens avec des experts du domaine, c'est-à-dire des ingénieurs et des concepteurs mécaniques.

  Quoi qu'il en soit, la flexibilité de la structure permettra d'améliorer la classification avec d'autres classes, par ex. couvercle, boîtier, plaque, rivet, support, ressort ou piston.
  De plus, cette classification permet de discerner des éléments correspondant éventuellement à des fixations (par exemple des vis, des boulons, des écrous et des c-clips) avec des éléments correspondant à des parties importantes caractérisant des ensembles fonctionnels spécifiques. Cette distinction permet de comparer deux assemblages par leurs pièces principales dans un premier temps puis d'affiner la comparaison en incluant également des composants mineurs.

  Ces classes ne sont pas au même niveau de spécification, étant plus orientées géométriquement (par exemple cylindrique ou en forme de tore) et d'autres se référant à l'artefact technologique spécifique permettant une fonction spécifique (par exemple la fonction de transmission de puissance pour des engrenages et des arbres, la fonction de fixation pour des clés et des entretoises), comme illustré en Figure 9.

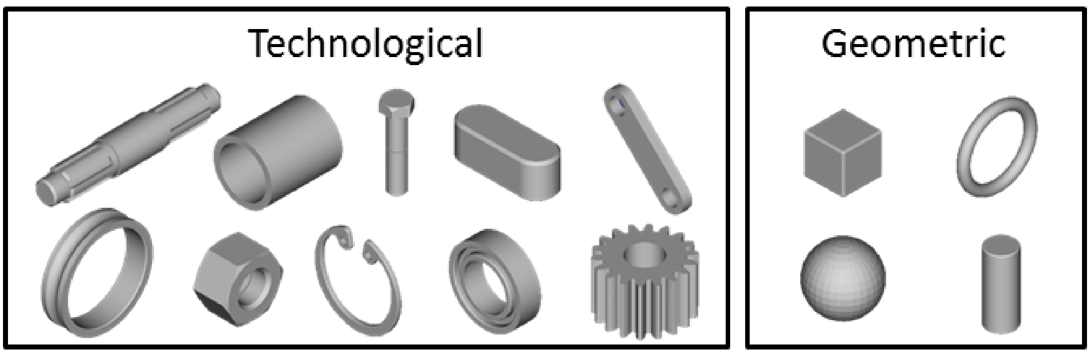


Figure 9: Attributs associés aux nœuds

Un exemple d'information présente dans la couche de structure de l'EAM pour un modèle d'assemblage est illustré en Figure 10, où les nœuds, les arcs et les attributs sont représentés. En particulier, l'étiquette sur le nœud racine représente la liste des pattern dans l'entier modèle, c'est-à-dire deux patterns de rotation circulaires composés de quatre vis et de quatre écrous respectivement; les étiquettes sur les nœuds correspondent à l'attribut *Component_Type*; et la décomposition de l'assemblage entre ses composants est représentée par les arcs, c'est-à-dire que chaque arc exprime la relation "faite-de" entre les composants de l'assemblage.

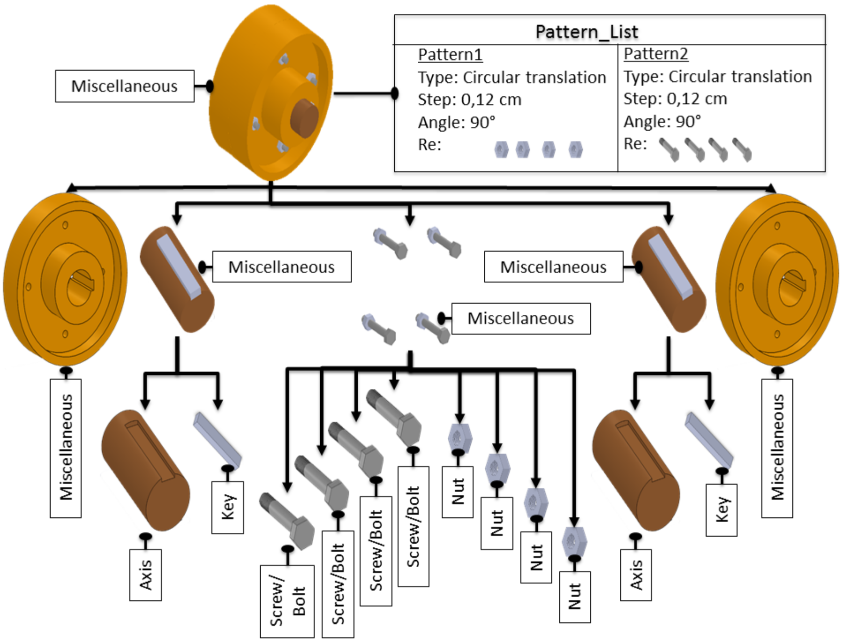


Figure 10: Exemple de le couche structure de l'EAM d'un modèle avec ses attributs

3.2 Interface assembly layer

La couche d'interface vise à décrire les relations entre les différentes pièces du modèle d'assemblage quelle que soit sa structure. Il est exprimé à deux niveaux de détail différents: les informations des contacts et l'information de joint.

Informations des contacts

Une fois que les pièces sont été positionnées, elles peuvent identifier des interfaces, c'est-à-dire des **contacts**, des **interférences** or des **espacements** conformément aux règles suivantes [110]:

- Un **contact** entre deux pièces C_i et C_j définit une ou plusieurs surfaces (ou courbes) partagées sans aucun volume partagé.
- Une **interférence** entre C_i et C_j définit un volume partagé entre eux.
- Un **espacement** se produit lorsqu'une distance entre deux surfaces de pièces C_i et C_j donne une signification fonctionnelle.

Les interfaces et espacements n'existent pas entre deux objets réels. Cependant, de telles configurations irréelles peuvent apparaître pendant la phase de modélisation, par ex. lorsque l'on considère des parties idéalisées et simplifiées à des fins de simulation, ou lors de la vérification des interférences à un stade de conception intermédiaire où les dimensions ne sont pas entièrement ajustées. Parfois, de telles configurations résultent simplement d'erreurs de modélisation.

Nous cherchons à décrire des modèles d'assemblage comme dans des configurations réelles, c'est-à-dire sans interférences ni espacements, de sorte que les modèles décrivant le même objet puissent être reconnus comme similaires même si des erreurs se produisent. Pour cette raison, nous considérons uniquement les contacts comme des relations entre les parties, et nous essayons de résoudre certaines erreurs récurrentes d'interférences par le biais du raisonnement géométrique.
De plus, chaque élément en contact définit une contrainte qui autorise un certain degré de liberté entre les pièces, à travers lequel un produit mécanique peut être classé [98]. Ensuite, le *informations des contacts* fournit une description détaillée des relations de partie, c'est-à-dire les types de contacts entre deux parties et leur DOF.

Ainsi, nous pouvons donner la définition suivante.

Information de contact

Definition 3.2. Un *information de contact* entre deux pièces P_i et P_j définit le degré de liberté généré par deux entités (face, arête ou sommet) entre P_i et P_j et il est caractérisé par des attributs ayant les domaines suivants:

$$DOF_{contact} = T \cup R \tag{4}$$
$$Contact\_Type = \{\text{Surface, Curve, Point, UnSolved}\} \tag{5}$$
$$Face\_Contact\_Type = \{\text{NoFace, Plane, Cylinder, Cone, Sphere, Torus, FreeForm}\} \tag{6}$$

Dans la définition 3.2, le DOF est formé par un ensemble de possibles translations T et un ensemble de possibles rotations R, les deux sont des vecteurs exprimés $v = (x, y, z)$ avec une norme égale à 1 ($||(x, y, z)|| = 1$), c'est-à-dire que v est un verseur.

Deux entités peuvent être partiellement en contact, par exemple, le contact entre les pièces (partie grise dans Figure 11) est seulement une partie des faces dans les B-reps des deux pièces, alors, les contacts ne peuvent pas être associé aux entités de B-reps.
Ainsi, l'attribut *Contact_Type* identifie le type de géométrie associé au contact entre deux pièces P_i et P_j, c'est-à-dire si le contact peut être représenté comme une surface, une courbe ou simplement un point. Si une interférence volumétrique existe entre deux pièces et nous ne sommes pas capables de le résoudre, nous supposons que P_i et P_j doivent être en contact en leur configuration réaliste, puis nous fixons l'attribut *Contact_Type* comme "Non résolu".

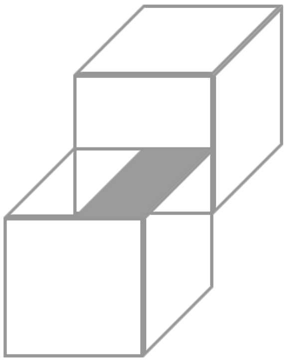


Figure 11: Deux pièces avec un contact de type surface

On peut remarquer qu'une paire de pièces (P_i, P_j) peut avoir plusieurs *informations de contact* en fonction du nombre de contacts entre elles. Par exemple, les parties de Figure 12 sont en contact par cinq différentes parties de faces, alors elles ont cinq *informations de contact.*

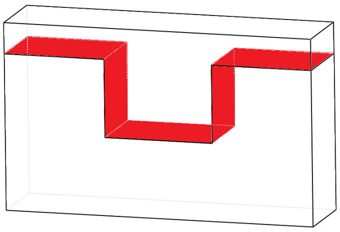


Figure 12: Deux pièces avec plusieurs contacts

Enfin, nous voulons observer que quantifier les translations et les rotations autorisées ne dépend pas seulement du type de contact mais inclut également la prise en compte de la dimension, comme l'évaluation de la portion de contact sur les surfaces des faces en contact. Par exemple, les deux boîtes en Figure 13 peuvent tourner le long des axes **n**, mais le modèle en Figure 13(a) peut effectuer une rotation complète de 360°, tandis que celui en Figure 13(b) a juste une rotation partielle. La reconnaissance de ces différences nécessite une analyse plus approfondie que nous ne verrons pas dans cette thèse, limitant notre détection sur les directions des mouvements autorisés par chaque contact.

(a) Rotation complète (b) Rotation partielle

Figure 13: Modèles avec mêmes contacts et DOF mais différents mouvements autorisés

Information de joint

La mobilité des composants est un concept fondamental pour analyser les mécanismes et elle est définie par les connexions entre les pièces. Ces connexions, appelées joint, sont réalisées par des contacts de chaque partie géométrique rigide [89] et peuvent être classées par le nombre de degrés de liberté autorisés [86].

Ensuite, nous pouvons donner la définition suivante:

Information de joint

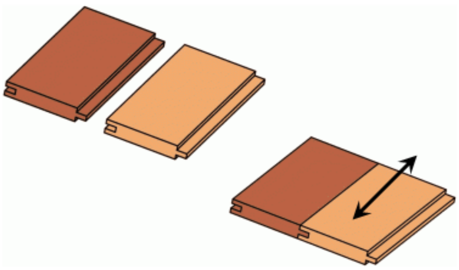
Definition 3.3. Une *information de joint* entre deux pièces P_i et P_j définit le degré de liberté induit par toutes les contraintes imposées par les contacts entre P_i et P_j et il est caractérisé par des attributs ayant les domaines suivants:

$$DOF_{joint} = T \cup R \tag{7}$$

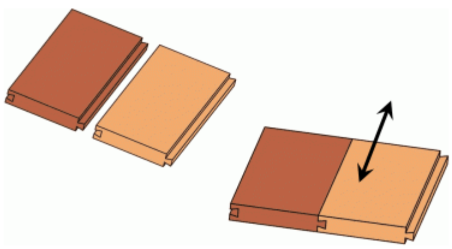
$$Joint\_Type = \{\text{Surface, Curve, Point, UnSolved, Mixed}\} \tag{8}$$

Dans la définition 3.3, le DOF de le joint entre deux parties est calculée selon la théorie des mécanismes [92], c'est-à-dire en composant les tenseurs cinématiques de tous les contacts partagés par les pièces.

Notez que les degrés de liberté autorisés au niveau du joint identifient une situation différente par rapport à celle au niveau des contacts. Par exemple, en Figure 14, les deux pièces sont en contact par des faces planes permettant une translation d'une partie par rapport à l'autre. Au niveau des joint ces deux exemples sont reconnus comme similaires, puisqu'ils peuvent seulement effectuer une translation sans perdre aucun visage en contact. Au contraire, au niveau du contact, ils sont discernés par la normale des faces appartenant à des coupes diagonales, montrant les différentes technologies employées.



(a) Tongue-groove joint



(b) Dovetailed joint

Figure 14: Même joint généré par différents contacts

3.3 Couche de forme

La forme du modèle est une information riche et fiable, importante pour distinguer les assemblées et cet information est codifié par le couche de forme par l'use des descripteurs de forme.

Un bonne descripteur de forme doit prendre en compte à la fois les fonctionnalités locales et les fonctionnalités globales. De plus, cela devrait être invariant par translations et rotations. De nombreux descripteurs de forme ont été définis pour comparer des modèles géométriques, dont la plupart sont calculés sur une représentation des objets 3D par maillage [25, 57, 30, 117]. Iyer et al. a souligné qu'il n'y a pas de descripteur de forme unique qui convient à toutes les formes possibles et à toutes les fins de comparaison [57]. Au contraire, selon le type d'objet, un descripteur de forme spécifique peut mieux fonctionner que d'autres. En effet, à partir de l'analyse du travail de Iyer et al. [57] et Jayanti et al. [58] on s'avère que les harmoniques sphériques fonctionnent bien avec des formes de révolution, la distribution de forme avec des parties prismatiques et la distance interne décrit bien les différences d'épaisseur des composants. En outre, dans le domaine de l'ingénierie mécanique, la surface et le volume d'un composant affectent sérieusement la fabrication d'un objet. En effet, ces valeurs de taille sont utile pour le remplacement de pièces et directement calculable à partir des données B-rep.

Nous avons décidé de ne stocker que des harmoniques sphériques 3D et des valeurs de taille. Cette décision est soutenue par l'ambition de ne pas affecter négativement le système de récupération introduisant des informations redondantes à traiter à chaque fois lors d'une comparaison. Les harmoniques sphériques sont représentent par un vecteur de 544 éléments selon la définition de Kazhdan et al.[62].

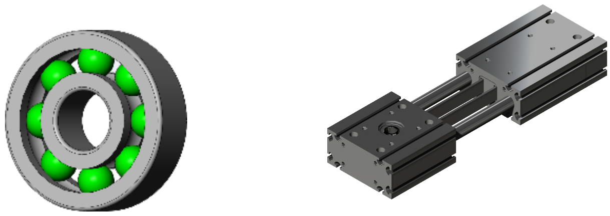
Ensuite, nous pouvons dire que la couche de forme est définie par deux attributs, le *Shape* et le *Size*, où

$$Shape = \mathbb{R}^{544} \tag{9}$$

$$Size = \mathbb{R}^{+} \times \mathbb{R}^{+} \tag{10}$$

3.4 Couche statistique

La couche statistique a été conçue pour faciliter le filtrage des grands base de données et réduire le nombre de modèles à comparer.
Les statistiques de un assemblage représentent le nombre de pattern de pièces répétées d'un type spécifique (par exemple des pattern linéaires, des pattern circulaires). Cette information est utile pour réduire le nombre de comparaisons entre les modèles. Supposons qu'un utilisateur recherche un roulement (Figure 15(a)) dans un base de données. Ce type de modèle est caractérisé par la présence de billes répétées disposées selon un motif circulaire. En utilisant la statistique de l'assemblage, nous sommes capables de rejeter facilement et rapidement les modèles comme celui illustré en Figure 15(b), qui ne possède pas cette caractéristique.



(a) (b)

Figure 15: Exemples d'assemblages avec (a) et sans (b) pattern de type circulaire

Les statistiques de un pièce comprennent: 1) le pourcentage de surfaces de un type spécifique (planaire, cylindrique, sphérique, forme libre, toroïdale) par rapport à la surface totale, 2) le nombre de faces maximales (faces adjacentes qui partageant la même surface au-dessous sont considérés comme une seule face) associé à une surface de un type spécifique (c'est-à-dire plane, cylindrique, sphérique, de forme libre, toroïdale). L'utilisation de tels pourcentages permet de rejeter directement certaines formes, réduisant ainsi le nombre de candidats pour le processus de correspondance.

Nous représentons le pourcentage de surfaces de type spécifique avec une valeur comprise entre 0 et 1 et le nombre de faces maximales avec un nombre naturel, alors nous avons une paire de values ($[0,1] \times \mathbb{N}$) pour chaque type de surface (planaire, cylindrique, sphérique, forme libre, toroïdale), cinq couples.
Bien que les statistiques d'assemblage représentent le nombre de modèles d'un type spécifique et que nous ayons quatre types de modèles différents, nous pouvons dire que la couche statistique est définie par deux attributs, *Part_Statistics* et *Assembly_Statistics*, où

$$Part\_Statistics = ([0,1] \times \mathbb{N})^5 \tag{11}$$

$$Assembly\_Statistics = \mathbb{N}^4 \tag{12}$$

Bien sûr, à partir de cette première liste de statistiques, des descripteurs supplémentaires peuvent être utilisés pour enrichir la caractérisation du modèle d'assemblage. Par exemple, le nombre de composants d'un type spécifique peut être inclus dans la couche statistique.

3.5 Représentation graphique

L'EAM présenté est représenté par un *multi-graphe attribué*, où les nœuds sont associés aux composants (pièces ou sous-assemblages) du modèle CAO et les arcs codent relations spécifiques entre eux. Afin d'illustrer la procédure de comparaison, cette section vise à formaliser la représentation de un EAM avec un graphe et son espace des attributs.

Soit G une multi-graphe attribuée qui représente un descripteur EAM

$$G = \mathcal{G}(\mathcal{N}, \mathcal{A}, \Phi_{\mathcal{N}}, \Phi_{\mathcal{A}}), \tag{13}$$

où $\mathcal{N}$ est l'ensemble des nœuds, $\mathcal{A}$ est l'ensemble des arcs et $\Phi_{\mathcal{N}}$ et $\Phi_{\mathcal{A}}$ sont respectivement les fonctions d'attributs des nœuds et des arcs.

Différents types de nœuds et d'arcs sont définis, en fonction du type différent d'informations extraites du modèle d'assemblage ou déduits par des processus de raisonnement. En particulier, nous avons $\mathcal{N} = \mathcal{N}_P \cup \mathcal{N}_A$ et $\mathcal{A} = \mathcal{A}_S \cup \mathcal{A}_C \cup \mathcal{A}_J$, où

- $\mathcal{N}_P$ est l'ensemble des nœuds associés aux pièces
- $\mathcal{N}_A$ est l'ensemble des nœuds associés aux sous-assemblages
- $\mathcal{A}_S$ est l'ensemble des arcs qui représentent la structure hiérarchique de l'assemblage
- $\mathcal{A}_C$ est l'ensemble des arcs qui représentent les contacts entre les pièces
- $\mathcal{A}_J$ est l'ensemble des arcs qui représentent les joints entre les pièces

Suite à cette notation, la fonction d'attribut des nœuds est définie comme

$$\Phi_{\mathcal{N}}(n) = \begin{cases} \Phi_{\mathcal{N}_P}(n) \text{ if } n \in \mathcal{N}_P \\ \Phi_{\mathcal{N}_A}(n) \text{ if } n \in \mathcal{N}_A \end{cases} \tag{14}$$

tandis que la fonction d'attribut des arcs est définie comme

$$\Phi_{\mathcal{A}}(a) = \begin{cases} \Phi_{\mathcal{A}_C}(a) \text{ if } a \in \mathcal{A}_C \\ \Phi_{\mathcal{A}_J}(a) \text{ if } a \in \mathcal{A}_J \end{cases} \tag{15}$$

où chaque fonction est définie comme suit:

$$\Phi_{\mathcal{N}_P} : \mathcal{N}_P \longrightarrow T_{\mathcal{N}_P} \tag{16}$$
$$\Phi_{\mathcal{N}_A} : \mathcal{N}_A \longrightarrow T_{\mathcal{N}_A} \tag{17}$$
$$\Phi_{\mathcal{A}_C} : \mathcal{A}_C \longrightarrow T_{\mathcal{A}_C} \tag{18}$$
$$\Phi_{\mathcal{A}_J} : \mathcal{A}_J \longrightarrow T_{\mathcal{A}_J} \tag{19}$$

La portée de ces fonctions dépend par les attributs associés à chaque nœud et arc. Pour chaque pièce, les attributs considérés sont le descripteur de forme, la taille, le type de composant, le pattern et les statistiques pour les pièces, donc

$$T_{\mathcal{N}_P} = \mathit{Shape} \times \mathit{Size} \times \mathit{Component\_Type} \times \mathit{Pattern} \times \mathit{Part\_Statistics} \tag{20}$$

Comme décrit dans la section 3.3, les harmoniques sphériques sont le descripteur de forme adopté représenté avec un histogramme de 544 cases; les valeurs de taille indiquent le volume et la surface d'une pièce; le type de composant peut être l'un de ceux illustrés dans Figure 9; les patterns sont décrits par le type (translation linéaire, translation circulaire, rotation circulaire et réflexion) et les paramètres (voir Table 2); les statistiques de la pièce se réfèrent aux pourcentages de types spécifiques de surfaces exprimés en valeurs comprises entre 0 et 1 et au nombre de faces maximales exprimées en valeurs naturelles.

Dans le cas d'un nœud associé à un sous-assemblage, les attributs considérés codent les patterns présents dans le sous-assemblage comme une liste de pièces répetées qui forment le pattern ($RP \subseteq \mathcal{N}_P$) et la caractérisation du pattern. De plus, les attributs codent des statistiques d'assemblage, qui sont les nombres de pattern de pièces répétées d'un type spécifique.

Il s'ensuit que:

$$T_{\mathcal{N}_A} = Pattern\_List \times Component\_Type \times Assembly\_Statistics \tag{21}$$

En ce qui concerne les arcs, les arcs représentant la structure d'assemblage n'ont aucun attribut (c'est à dire $T_{\mathcal{A}_S} = \emptyset$) tandis que les arcs représentant les contacts et les joints entre pièces ont des attributs spécifiant le mouvement autorisé par leur DOF.
En outre, dans le cas d'arcs représentant des contacts, nous associons l'attribut $Component\_Type$, qui indique le type de contact et, si le type est un surface, nous indiquons aussi le type de surface par l'attribut $Face\_Contact\_Type$. Dans le cas des arcs représentant des jointes, en plus du DOF, nous associons l'attribut $Joint\_Type$, qui indique le type de contacts qui ont donné naissance aux joints.
Ainsi, pour les arcs, les attributs sont définis comme suit:

$$T_{\mathcal{A}_C} = DOF_{contact} \times Contact\_Type \times Face\_Contact\_Type \tag{22}$$

$$T_{\mathcal{A}_J} = DOF_{joint} \times Joint\_Type \tag{23}$$

3.6 Procédure de matching

En utilisant des descripteurs basés sur des graphes, le problème de trouver la similarité locale entre deux modèles équivaut à trouver le sous-graphe commun maximal (MCS) entre deux graphes: G_q et G_k qui représentant le modèle requête et le modèle de comparaison respectivement. Parmi les différentes techniques pour résoudre le problème MCS [31], nous traduisons notre problème en un problème de Clique Maximum (MC) [16], c'est-à-dire que nous visons à détecter une clique (un sous-graphe où pour chaque paire de nœuds il y a un arc qui les relie) dans un *graphe d'association* qui est approprié dérivé de G_1 et G_2.

Dans un graphe d'association, les nœuds représentent des paires de *nœuds similaires* entre G_q et G_k et les arcs identifient *relations similaires*. Nous nous référons aux critères de similarité des nœuds et des arcs comme c_N et c_A respectivement et ils sont décrits dans la section 4. Le graphe d'association n'est pas unique et sa définition dépend des critères de similarité. Une fois sélectionnés c_N et c_A, il existe un graphe d'association *unique* (selon ces critères) et nous l'indiquons avec G_{q,k,c_N,c_A}. Les cliques maximales dans G_{q,k,c_N,c_A} représentent les sous-graphes communs entre G_q et G_k selon le critère c_N et c_A. Le générique r^{th} clique dans le graphe G_{q,k,c_N,c_A} est exprimé par $(C_{q,k,c_N,c_A})_r \subseteq G_{q,k,c_N,c_A}$ et l'ensemble de toutes les cliques pour le graphe d'association G_{q,k,c_N,c_A} est noté $\mathcal{D}_{q,k} = \{(C_{q,k,c_N,c_A})_r, r \in \{1, \cdots Num_{qk}\}\}$, où Num_{qk} est le nombre de cliques maximum dans le graphe d'association G_{q,k,c_N,c_A}.

Dans ce cadre, les cliques sont détectées en utilisant une méthode d'appariement de graphes exacte basée sur l'approche de Bron et Kerbosh bien connue [19].

4 Critères de similitude

Les conditions de similarité qui peuvent être définies par rapport aux nœuds, c'est-à-dire c_N, concernent: la forme, la taille, le type de composant et le type de pattern. Chacun de ces critères est spécifié par les valeurs de sa fonction d'attribut. Dans la suite, nous indiquons avec $\Phi_{\mathcal{N}_*}(n)_{\restriction NodeAttributeSet}$ la projection de la fonction d'attribut $\Phi_{\mathcal{N}_*}(n)$ sur un certain $NodeAttributeSet$.

4.1 Similitude de nœuds

La similarité de forme est basée sur le descripteur de forme de chaque nœud, c'est-à-dire les harmoniques sphériques 3D. Deux harmoniques sphériques 3D peuvent être comparées de différentes manières. Kazhdan et al. [62] ont illustré certaines propriétés des harmoniques sphériques, en particulier, ils ont démontré que la distance L_2 est la plus appropriée pour l'évaluation de la similarité. Ensuite, nous donnons la définition suivante:

Similaires selon le critère de forme

Definition 4.1. Deux noeuds n_q^i et n_k^j sont considérés *similaires selon le critère de forme* si $\chi_{Sh}(n_q^i, n_k^j) = 1$, où

$$\chi_{Sh}(n_q^i, n_k^j) = \begin{cases} 1 & \text{if} \quad ||\Phi_{\mathcal{N}_P}(n_q^i)_{\restriction Shape} - \Phi_{\mathcal{N}_P}(n_k^j)_{\restriction Shape}||_2 < \epsilon_{Shape} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

où ϵ_{Shape} représente le seuil défini dans la requête.

Pour établir si deux parties ont une taille similaire, nous comparons leurs valeurs de volume et de surface présentes dans l'EAM, puis:

Similaires selon le critère de taille

Definition 4.2. Deux noeuds n_q^i et n_k^j sont considérés *similaires selon le critère de taille* si $\chi_{Si}(n_q^i, n_k^j) = 1$, où

$$\chi_{Si}(n_q^i, n_k^j) = \begin{cases} 1 & \text{if} \quad abs(|\Phi_{\mathcal{N}_P}(n_q^i)_{\restriction Size} - \Phi_{\mathcal{N}_P}(n_k^j)_{\restriction Size}) < \epsilon_{Size} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

où ϵ_{Size} représente le seuil défini dans la requête.

Les critères de similarité de type de composant et de modèle nécessitent de vérifier si les valeurs des attributs $CompType$ et $PatternType$ sont les mêmes pour les deux nœuds, alors nous avons les définitions suivantes:

Similaires selon le critère de type de composant

Definition 4.3. Deux nœuds n_q^i et n_k^j sont considérés *similaires selon le critère de type de composant* si $\chi_{CT}(n_q^i, n_k^j) = 1$, où

$$\chi_{CT}(n_q^i, n_k^j) = \begin{cases} 1 & \text{if} \quad \Phi_{\mathcal{N}_P}(n_q^i)_{\restriction CompType} = \Phi_{\mathcal{N}_P}(n_k^j)_{\restriction CompType} \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Similaires selon le critère du type de motif

Definition 4.4. Deux nœuds n_q^i et n_k^j sont considérés *similaires selon le critère du type de motif* si $\chi_{PT}(n_q^i, n_k^j) = 1$, où

$$\chi_{PT}(n_q^i, n_k^j) = \begin{cases} 1 & \text{if} \quad \Phi_{\mathcal{N}_P}(n_q^i)_{\restriction PatternType} = \Phi_{\mathcal{N}_P}(n_k^j)_{\restriction PatternType} \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

Soit $c_N = (\alpha_{Sh}, \alpha_{Si}, \alpha_{CT}, \alpha_{PT})$ un vecteur qui représente les critères qui peuvent être sélectionnés, où α_{Sh}, α_{Si}, α_{CT} et α_{PT} représentent respectivement l'activation de la forme, la taille, le type de composant et le critère de pattern. Plus précisément, le critère générique $\alpha_* \in \{0, 1\}$ et $\alpha_* = 1$ signifie que le critère * est activé.

Afin de réduire l'association de nœuds sans signification, deux nœuds peuvent être associés selon le critère de taille seulement si les critères de forme ou de type de composant sont sélectionnés, c'est-à-dire,

$$\alpha_{Si} = 1 \Longleftrightarrow (\alpha_{Sh} = 1) \text{ or } (\alpha_{CT} = 1) \tag{28}$$

Avec cette notation, nous pouvons définir l'ensemble des nœuds compatibles comme :

$$\mathcal{N}_{q,k,c_N} = \left\{ \gamma_N(n_q^i, n_k^j, c_N), \quad n_q^i \in \mathcal{N}_q \text{ and } n_k^j \in \mathcal{N}_k \right\} \tag{29}$$

où

$$\begin{aligned} &\gamma_N : \mathcal{N}_q \times \mathcal{N}_k \times c_N \to \mathcal{N}_q \times \mathcal{N}_k \\ &\gamma_N(n_q^i, n_k^j, c_N) = \begin{cases} (n_q^i, n_k^j) & \text{if} \quad \chi_{Node}(n_q^i, n_k^j, c_N) = 1 \\ \emptyset & \text{if} \quad \chi_{Node}(n_q^i, n_k^j, c_N) = 0 \end{cases} \end{aligned} \tag{30}$$

et

$$\chi_{Node}(n_q^i, n_k^j, c_N) = \left\lfloor \frac{\alpha_{Sh}\chi_{Sh}(n_q^i, n_k^j) + \alpha_{Si}\chi_{Si}(n_q^i, n_k^j) + \alpha_{CT}\chi_{CT}(n_q^i, n_k^j) + \alpha_{PT}\chi_{PT}(n_q^i, n_k^j)}{\alpha_{Sh} + \alpha_{Si} + \alpha_{CT} + \alpha_{PT}} \right\rfloor \tag{31}$$

Le symbole $\left\lfloor * \right\rfloor$ indiquer la partie entière de *.

La fonction χ_{Node} a une valeur dans $\{0, 1\}$ et décrit la compatibilité de deux nœuds selon les critères qui peuvent être choisis pendant la requête, c'est-à-dire que n_q^i et n_k^j sont compatibles selon c_N si $\chi_{Node}(n_q^i, n_k^j, c_N) = 1$.

Généralement, l'image de la fonction γ_N est un sous-ensemble de $\mathcal{N}_q \times \mathcal{N}_k$ et sa cardinalité dépend strictement des critères spécifiés pour l'association des nœuds. Le nombre de nœuds d'association diminue à mesure que les critères spécifient que plus de conditions de similarité doivent être satisfaites.

4.2 Similitude de arcs

Une fois que des paires de nœuds similaires ont été générées dans le graphe d'association, on doit définir s'il existe un arc entre la paire générique de nœuds d'association (n_q^i, n_k^l) et (n_q^j, n_k^h) et cela dépend du critère de relation similaire c_A.

Dans la suite, les différentes compatibilités pour les arcs sont introduites, où l'arc générique dans G_q entre la paire de nœuds (n_q^i, n_q^j) est indiqué comme $a_q^{ij} \in \mathcal{A}_q$ et l'arc générique dans G_k entre la paire de nœuds (n_k^h, n_k^l) est indiqué comme $a_k^{hl} \in \mathcal{A}_k$.

Compatibilité par rapport aux contacts

Definition 4.5. Deux arcs a_q^{ij} et a_k^{hl} sont considérés *compatible selon le critère de contact* si $\chi_C(a_q^{ij}, a_k^{hl}) = 1$, où

$$\chi_C(a_q^{ij}, a_k^{hl}) = \begin{cases} 1 & \text{if} \quad (a_q^{ij} \in \mathcal{A}_{C_q}) \text{ and } (a_k^{hl} \in \mathcal{A}_{C_k}) \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

Deux pièces peuvent avoir plusieurs *informations de contact*, cela signifie que plusieurs arcs de contact peuvent exister entre les nœuds qui correspondent aux deux pièces. Ensuite, la compatibilité selon le critère DOF autorisé des contacts doit considérer l'ensemble des arcs de contact présents entre les deux nœuds. En particulier, nous indiquons avec $\mathcal{A}_{C_q}^{ij}$ l'ensemble des arcs de contacts entre les nœuds n_q^i et n_q^j et avec $\mathcal{A}_{C_k}^{hl}$ l'ensemble des arcs de contacts entre les nœuds n_k^h et n_k^l.

Compatibilité par rapport aux DOF autorisés pour les contacts

Definition 4.6. Défini

- $\mathcal{A}_{C_q}^{ij}$ et $\mathcal{A}_{C_k}^{hl}$
- $(C_q^{ij})_s = \Phi_{\mathcal{A}_C}(c_s)_{\restriction Contact\_Type} \in Contact\_Type$ for $c_s \in \mathcal{A}_{C_q}^{ij}$,
- $(T_q^{ij})_s \cup (R_q^{ij})_s = \Phi_{\mathcal{A}_C}(c_s)_{\restriction DOF_{contact}}$ pour $c_s \in \mathcal{A}_{C_q}^{ij}$,
- $(C_k^{hl})_t = \Phi_{\mathcal{A}_C}(c_t)_{\restriction Contact\_Type} \in Contact\_Type$ for $c_t \in \mathcal{A}_{C_k}^{hl}$,
- $(T_k^{hl})_t \cup (R_k^{hl})_t = \Phi_{\mathcal{A}_C}(c_t)_{\restriction DOF_{contact}}$ for $c_t \in \mathcal{A}_{C_k}^{hl}$,

l'ensemble des arcs $\mathcal{A}_{C_q}^{ij}$ et $\mathcal{A}_{C_k}^{hl}$ sont considérés *compatible selon le DOF autorisé pour le critère de contacts* si $\chi_{C_{num}}(\mathcal{A}_{C_q}^{ij}, \mathcal{A}_{C_k}^{hl}) = 1$, où

$$\chi_{C_{num}}(\mathcal{A}_{C_q}^{ij}, \mathcal{A}_{C_k}^{hl}) = \begin{cases} 1 & \text{if} \quad \Big[\Big(|(T_q^{ij})_s| = |(T_k^{hl})_t|\Big) \text{ and } \Big(|(R_q^{ij})_s| = |(R_k^{hl})_t|\Big)\Big] \\ & \qquad \forall c_s \in \mathcal{A}_{C_q}^{ij} \quad c_t \in \mathcal{A}_{C_k}^{hl} \quad \text{or} \\ & \quad \Big[(C_q^{ij})_s = "UnSolved"\Big] \quad \text{or } \Big[(C_k^{ij})_t = "UnSolved"\Big] \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

Compatibilité par rapport au DOF autorisé pour les joints

Definition 4.7. Défini

- $J_q^{ij} = \Phi_{\mathcal{A}_C}(a_q^{ij})_{\restriction Joint\_Type} \in Joint\_Type,$
- $T_q^{ij} \cup R_q^{ij} = \Phi_{\mathcal{A}_J}(a_q^{ij})_{\restriction DOF_{joint}},$
- $J_k^{hl} = \Phi_{\mathcal{A}_C}(a_q^{hl})_{\restriction Joint\_Type} \in Joint\_Type,$
- $T_k^{hl} \cup R_k^{hl} = \Phi_{\mathcal{A}_J}(a_k^{hl})_{\restriction DOF_{joint}},$

Deux arcs a_q^{ij} et a_k^{hl} sont considérés *compatible selon le critère du DOF pour les joints* if $\chi_{J_{num}}(a_q^{ij}, a_k^{hl}) = 1$, où

$$\chi_{J_{num}}(a_q^{ij}, a_k^{hl}) = \begin{cases} 1 & \text{if } \left[\left(|T_q^{ij}| = |T_k^{hl}|\right) \text{ and } \left(|R_q^{ij}| = |R_k^{hl}|\right)\right] \quad \text{or} \\ & \left[J_q^{ij} = "UnSolved"\right] \quad \text{or } \left[J_k^{hl} = "UnSolved"\right] \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

Concernant les arcs, nous avons un vecteur $c_A = (\alpha_C, \alpha_{C_{num}}, \alpha_{J_{num}})$ qui représente les critères qui peuvent être sélectionnés, où α_C, $\alpha_{C_{num}}$ et $\alpha_{J_{num}}$ représentent respectivement le contact, le DOF autorisé pour les contacts et le DOF autorisé pour les critères joints. Aussi pour les arcs, le filtre générique $\alpha_* \, in\{0,1\}$ et $\alpha_* = 1$ signifie que le filtre * est activé.

L'ensemble des arcs compatibles est défini comme :

$$\mathcal{A}_{q,k,c_A} = \left\{\gamma_A(a_q^{ij}, a_k^{hl}, c_A), \qquad a_q^{ij} \in \mathcal{A}_q \text{ and } a_k^{hl} \in \mathcal{A}_k\right\} \tag{35}$$

où

$$\begin{gathered} \gamma_A : \mathcal{A}_q \times \mathcal{A}_k \times c_A \rightarrow \mathcal{N}_{q,k,c_N} \times \mathcal{N}_{q,k,c_N} \\ \gamma_A(a_q^{ij}, a_k^{hl}, c_A) = \begin{cases} ((n_q^i, n_k^h), (n_q^j, n_k^l)) & \text{if } \chi_{Arc}(a_q^{ij}, a_k^{hl}, c_A) = 1 \\ \emptyset & \text{if } \chi_{Arc}(a_q^{ij}, a_k^{hl}, c_A) = 0 \end{cases} \end{gathered} \tag{36}$$

et

$$(n_q^i, n_k^h) \qquad \text{and} \quad (n_q^j, n_k^l) \in \mathcal{N}_{q,k,c_N} \tag{37}$$

$$a_q^{ij} = (n_q^i, n_q^j) \in \mathcal{A}_{C_q} \cup \mathcal{A}_{J_q} \tag{38}$$

$$a_k^{hl} = (n_k^h, n_k^l) \in \mathcal{A}_{C_k} \cup \mathcal{A}_{J_k} \tag{39}$$

$$\chi_{Arc}(a_q^{ij}, a_k^{hl}, c_A) = \left\lfloor \frac{\alpha_C \chi_C(a_q^{ij}, a_k^{hl}) + \alpha_{C_{num}} \chi_{C_{num}}(\mathcal{A}_{C_q}^{ij}, \mathcal{A}_{C_k}^{hl}) + \alpha_{J_{num}} \chi_{J_{num}}(a_q^{ij}, a_k^{hl})}{\alpha_C + \alpha_{C_{num}} + \alpha_{J_{num}}} \right\rfloor \tag{40}$$

Le symbole $\lfloor * \rfloor$ indique la partie entière du plancher de *.

Puisque la similitude de contact entraîne le DOF autorisé pour la similarité des contacts, ce qui nécessite la similitude du DOF autorisé pour les joints, pour éviter les vérifications inutiles, lorsqu'un critère de compatibilité sur les arcs est sélectionné, les deux autres sont fixés à 0.

Une fois sélectionnés les critères de comparaison (c_N pour les nœuds et c_A pour les arcs), il existe un **an unique** graphe d'association (selon ces critères) et nous nous y référons comme suit :

$$G_{q,k,c_N,c_A} = \mathcal{G}(\mathcal{N}_{q,k,c_N}, \mathcal{A}_{q,k,c_A}) \tag{41}$$

où $\mathcal{N}_{q,k,c_N} = \left\{\gamma_N(n_q^i, n_k^j, c_N)\right\}$ et $\mathcal{A}_{q,k,c_A} = \left\{\gamma_A(a_q^{ij}, a_k^{hl}, c_A)\right\}$.

Les cliques maximum dans le graphe d'association G_{q,k,c_N,c_A} représentent les sous-graphes communs entre G_q et G_k selon les critères c_N et c_A.

Le générique r^{th} clique dans le graphe G_{q,k,c_N,c_A} est exprimé comme :

$$(C_{q,k,c_N,c_A})_r \subseteq G_{q,k,c_N,c_A} \tag{42}$$

et l'ensemble de toutes les cliques du graphe d'association G_{q,k,c_N,c_A} est noté :

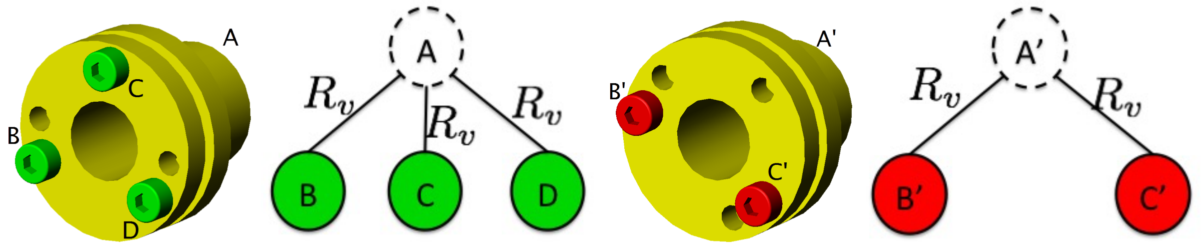
$$\mathcal{D}_{q,k} = \{(C_{q,k,c_N,c_A})_r\}_{h=1}^{Num_{qk}} \tag{43}$$

où Num_{qk} est le nombre de cliques maximum dans le graphe d'association G_{q,k,c_N,c_A}.

4.3 Exemple

En Figure 16(a) nous avons un modèle d'assemblage représentant une bride avec trois vis et une partie de son multi-graphe attribué est représentée en Figure 16(b). Par souci de lisibilité, le nœud racine correspondant à l'ensemble du modèle d'assemblage est omis et seuls les arcs $\mathcal{A}_J$ sont représentés. Nous comparons ce modèle avec celui en Figure 16(c), dont la portion de multi-graphe attribuée est représentée en Figure 16(d). Les nœuds des deux multi-graphes attribués représentent la partie des modèles CAO, où le même type de ligne indique la même valeur du descripteur de forme harmonique sphérique (c'est à dire des pièces de formes similaires) et des pièces de même couleur appartiennent à des patterns d'un type spécifique (c'est-à-dire vert pour la translation circulaire et rouge pour la translation linéaire). Les arcs représentent les contacts de type joint où les étiquettes indiquent le DOF autorisé entre deux pièces liées.

Dans l'exemple, on suppose que l'utilisateur cherche des assemblages dans lesquels des pièces de forme similaire sont connectées par les mêmes relations. Donc, deux nœuds créent un nœud d'association si leurs pièces correspondantes ont une forme similaire, tandis que les arcs d'association sont ajoutés seulement si les arcs joints (entre les paires de nœuds dans les multi-graphes attribués) ont le même nombre de rotations et le même nombre des translations. Le graphe d'association, résultant de ces critères, est illustré en



(a) Bride avec trois vis

(b) G_1

(c) Bride avec deux vis

(d) G_2

Figure 16: Exemple de deux modèles d'assemblage et leurs multi-graphes EAM G_1 et G_2

Figure 17, où il y a six cliques maximales possibles, c'est-à-dire:

$$
\begin{aligned}
(C_{1,2,c_N,c_A})_1 &= \{AA', BB', CC'\},\\
(C_{1,2,c_N,c_A})_2 &= \{AA', BC', DB'\},\\
(C_{1,2,c_N,c_A})_3 &= \{AA', CB', DC'\},\\
(C_{1,2,c_N,c_A})_4 &= \{AA', CC', DB'\},\\
(C_{1,2,c_N,c_A})_5 &= \{AA', BB', DC'\},\\
(C_{1,2,c_N,c_A})_6 &= \{AA', BC', CB'\}.
\end{aligned} \tag{44}
$$

Chaque clique représente un sous-graphe possible entre les deux multi-graphes attribués G_1 et G_2.

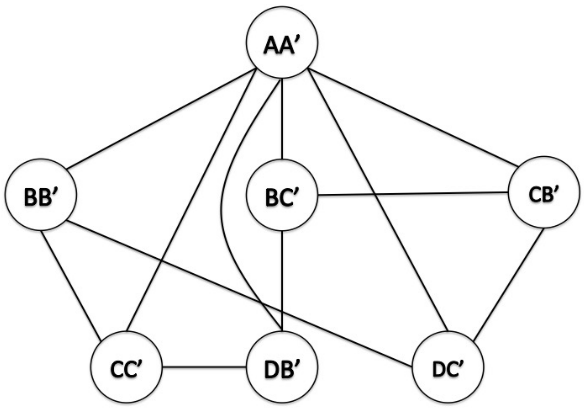


Figure 17: Graphe d'association pour les objets dans Figure 16

5 Évaluation de la similarité

Deux modèles peuvent être similaires selon différents critères, donc nous définissons un ensemble de mesures $\mathcal{S} = [\mu^{shape}, \mu^{joint}, \mu^{position}, \mu^{structure}]$, où chacun d'eux caractérise un seul aspect de la similarité entre deux modèles d'assemblage. Les éléments similaires entre deux assemblages correspondent aux sous-graphes communs entre les deux multi-graphes attribués associés à la requête et au modèle comparé. Ainsi, la similarité est calculée sur les cliques détectées $(C_{q,k,c_N,c_A})_r$. Dans la suite nous indiquons la clique générique comme C et l'ensemble de ses nœuds comme $\mathcal{N}_C$.

5.1 Mesure de similarité de forme : μ^{shape}

Cette mesure est basée sur le descripteur de forme de chaque nœud dans la clique. Le couche de forme de EAM code les harmoniques sphériques 3D et les valeurs de taille. Puisque deux objets peuvent avoir exactement la même forme avec des dimensions différentes, nous avons décidé d'utiliser uniquement les harmoniques sphériques 3D pour l'évaluation de la similarité de forme de pièce, alors que les valeurs de taille doivent être considérées comme une mesure séparée, qui pourrait être utilisé pour affiner les résultats. Comme expliqué par Kazhdan et al. [62], la distance L_2 est une norme appropriée pour évaluer les harmoniques sphériques 3D, alors nous définissons la similitude de forme d'une clique C comme la moyenne de la similarité de forme de chaque nœud dans la clique.

μ^{shape}

Definition 5.1. $\mu^{shape}(C) =$

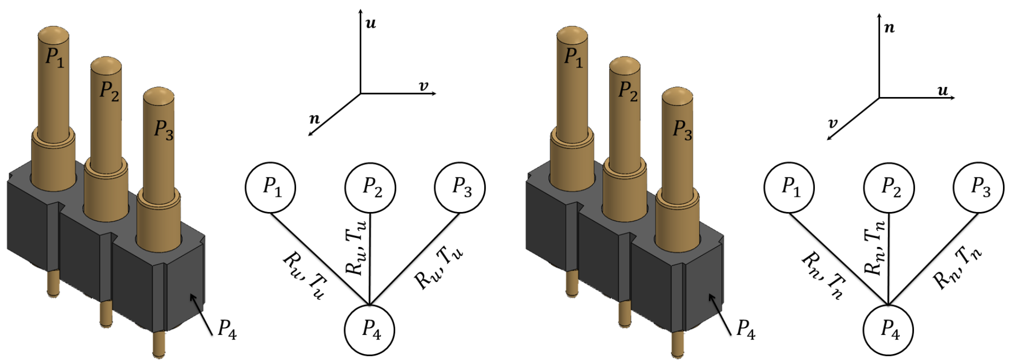
$$\frac{1}{|\mathcal{N}_C|} \sum_{(n_q^i, n_k^j) \in C} 1 - \left\| \frac{\Phi_{\mathcal{N}_P}^{Shape}(n_q^i)}{\|\Phi_{\mathcal{N}_P}^{Shape}(n_q^i)\|} - \frac{\Phi_{\mathcal{N}_P}^{Shape}(n_k^j)}{\|\Phi_{\mathcal{N}_P}^{Shape}(n_k^j)\|} \right\|_2 \tag{45}$$

5.2 Mesure de similarité de joint : μ^{joint}

Pour évaluer combien deux assemblages sont similaires en termes de DOF relatif parmi leurs parties, nous définissons une mesure sur les joints, $\mu^{joint}(C)$. Un joint peut provenir de contacts de différents types (Surface, Courbe, Point ou Non résolu). Dans le cas de joints provenant de contacts de type "Surface", nous savons calculer le DOF des deux pièces liées, sinon nous n'avons que l'information du type de joint. Puisque pour les arcs conjoints nous avons deux différents attributs, nous définissons $\mu^{joint}(C)$ comme une combinaison de deux autres mesures, l'une basée sur la similarité donnée par l'articulation de type "Surface" $\mu^{joint}_{surface}(C)$ et l'autre qui considère le type "Courbe" et "Point" $\mu^{joint}_{couve,point}(C)$.

Le $\mu^{joint}_{surface}(C)$ ne doit pas être affecté par des cadres de référence différents, c'est-à-dire si deux modèles d'assemblage, identifiant le même objet, sont décrits dans différents cadres de référence ou simplement tournés ou déplacé, leur mesure de similarité devrait être la même. Quoi qu'il en soit, les informations sur le DOF dans l'EAM dépendent du cadre de référence du modèle d'assemblage. En conséquence, une simple comparaison entre les DOF des éléments correspondants n'est pas appropriée.

Par exemple, si on évalue la distance entre les axes définis par la DOF entre les pièces P_1 et P_4 en Figure 18(a) avec celle entre les mêmes pièces en Figure 18(b), c'est-à-dire l'angle défini par l'axe u et n, nous aurons une variation de 90 degrés même si les objets sont les mêmes.



(a) Modèle de requête (b) Modèle de comparaison

Figure 18: Exemple d'objet incorporé dans deux cadres de référence différents

Au contraire, les variations de chaque paire d'axes définie par le DOF entre les pièces (P_1, P_4), (P_2 P_4) et (P_3 P_4) sont les mêmes dans les deux configurations. Ensuite, nous considérons la variation des axes de rotation et de translation définis par la DOF d'une pièce avec toutes les parties en contact. Par conséquent, nous calculons $\mu^{joint}_{surface}$ sur les nœuds des cliques au lieu d'utiliser les arcs. Nous gérons les configurations dans lesquelles nous avons plusieurs angles définis par les axes du DOF en utilisant des matrices pour représenter toutes les variations possibles des angles de rotation/translation impliqués par une pièce. Chaque élément de ces matrices identifie le produit scalaire d'une paire d'axes, comme spécifié en la définition 5.2 et 5.3.

Matrice des variations de translations

Definition 5.2. $Var_{Tra}(n)$ est la matrice des variations de translations de nœud n, où son élément générique est défini comme : $(Var_{Tra}(n))_{i,j} = t_i \cdot t_j$, où $t_i, t_j \in Tra(n)$ et $Tra(n) = \{\bigcup_h Tra^h \quad \forall n^h : (n, n^h) \in \mathcal{A}_J\}$ est l'ensemble de toutes les translations par rapport à un nœud n.

Matrice des variations de rotations

Definition 5.3. $Var_{Rot}(n)$ est la matrice des variations de rotations de nœud n, où son élément générique est défini comme: $(Var_{Rot}(n))_{i,j} = r_i \cdot r_j$, où $r_i, r_j \in Rot(n)$ et $Rot(n) = \{\bigcup_h Rot^h \quad \forall n^h : (n, n^h) \in \mathcal{A}_J\}$ est l'ensemble de toutes les rotations par rapport à un nœud n.

Le choix d'utiliser le produit scalaire est conduit par le fait que les axes sont normalisés, alors il correspond au cosinus de l'angle entre les axes considérés.

La variation finale pour un nœud dans une clique est calculée par la moyenne de ces éléments de matrice, où avec *moyenne* d'une matrice, nous entendons la moyenne arithmétique des éléments dans la matrice divisée par le nombre d'éléments, c'est à dire:

$$\sigma(M) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} x_{i,j} \qquad \forall x_{i,j} \in M \tag{46}$$

où N est la dimension de la matrice M.

En utilisant cette approche, lorsqu'un seul arc est incident sur un nœud, sa variation est 1. De plus, la mesure de joint doit également prendre en compte les cas où les joints sont de type "non résolus". Lorsque nous avons un contact ou une joint de type "Unsolved", nous n'avons pas d'informations sur son DOF, et la seule différence entre les deux modèles est représente par le nombre d'axes générés par le DOF qui concourent à définir la matrice de variation. Donc, nous utilisons cette information pour distinguer ces cas, en divisant la moyenne de chaque matrice de variation par le nombre de lignes de la matrice. Ensuite, nous définissons une mesure conjointe $\mu^{joint}_{courbe,point}(C)$ qui affecte la valeur maximale 1 si les joints sont du même type (courbe ou point) et une valeur inférieure si le les articulations sont de type différent. La valeur la plus basse est fixée à 0,8 et elle a été choisie empiriquement afin de diminuer légèrement la mesure.

Ensuite, nous définissons les mesures suivantes :

μ^{joint}

Definition 5.4. $\mu^{joint}(C) =$

$$\begin{cases} \mu^{joint}_{surface}(C) & \text{si } \mu^{joint}_{curve,point}(C) = 0 \\ \mu^{joint}_{curve,point}(C) & \text{si } \mu^{joint}_{surface}(C) = 0 \\ \frac{\mu^{joint}_{surface}(C)+\mu^{joint}_{curve,point}(C)}{2} & \text{autrement} \end{cases} \tag{47}$$

$\mu^{joint}_{surface}$

Definition 5.5. $\mu^{joint}_{surface}(C) =$

$$\frac{1}{|\mathcal{N}_C|}\sum_{(n^i_q,n^j_k)\in C}\left[1 - \frac{d_{Rot}((n^i_q,n^j_k)) + d_{Tra}((n^i_q,n^j_k))}{2}\right] \tag{48}$$

où,

- $d_{Rot}((n^i_q,n^j_k)) = abs\left(\frac{\sigma(Var_{Rot}(n^i_q))}{|Rot(n^i_q)|} - \frac{\sigma(Var_{Rot}(n^j_k))}{|Rot(n^j_k)|}\right)$
- $d_{Tra}((n^i_q,n^j_k)) = abs\left(\frac{\sigma(Var_{Tra}(n^i_q))}{|Tra(n^i_q)|} - \frac{\sigma(Var_{Tra}(n^j_k))}{|Tra(n^j_k)|}\right)$
- $Var_{Rot}(n^i_*)$ est une matrice de la variation des rotations du nœud n^i_* du modèle représenté par le graphe G_*. Si $Rot(n^i_*) = \emptyset$ donc $\frac{\sigma(Var_{Rot}(n^i_*))}{|Rot(n^i_*)|} = 1$
- $Var_{Tra}(n^i_*)$ est une matrice de la variation des rotations du nœud n^i_* du modèle représenté par le graphe G_*. Si $Tra(n^i_*) = \emptyset$ donc $\frac{\sigma(Var_{Tra}(n^i_*))}{|Tra(n^i_*)|} = 1$

$\mu^{joint}_{curve,point}$

Definition 5.6. $\mu^{joint}_{curve,point}(C) =$

$$\frac{1}{|\mathcal{N}_C|} \sum_{((n^i_q,n^j_k),(n^l_q,n^h_k))\in C\times C} \left[1 - d_{edge}((a^{ij}_q, a^{lh}_k))\right] \tag{49}$$

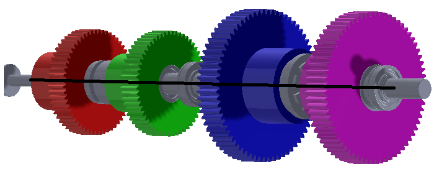
où, $a^{ij}_q = (n^i_q, n^j_q)$ et $a^{lh}_k = (n^l_k, n^h_k)$ et

$$d_{edge}((a^{ij}_q, a^{lh}_k)) = \begin{cases} 1 & \text{si } \Phi_{\mathcal{A}_J}(a^{ij}_q)_{\restriction JointType} \neq \Phi_{\mathcal{A}_J}(a^{hl}_k)_{\restriction JointType} \\ 0.8 & \text{autrement} \end{cases}$$

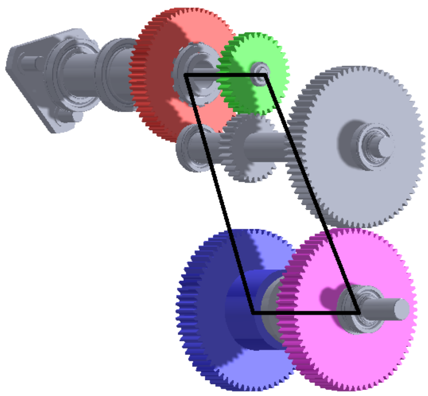
5.3 Mesure de similarité de position : $\mu^{position}$

Une autre caractéristique saillante qui augmente ou diminue le niveau de similarité entre deux modèles d'assemblage est l'agencement relatif des composants d'assemblage. Par exemple, si nous considérons les assemblées en Figure 19, ils ont des pièces de forme et de relations du type joint qui sont similaires (c'est-à-dire que les parties colorées ne sont pas en contact entre elles) mais la disposition de ces pièces discerne les deux modèles.

Ensuite, nous visons à définir une mesure qui caractérise la position des pièces similaires.



(a)



(b)

Figure 19: Exemple de modèles d'assemblage avec des pièces similaires disposées en différentes configurations

Pour reconnaître ces configurations, nous considérons le *verseur directionnel* entre le centre de gravité d'une pièce P et le centre de gravité de chaque pièce P_i dans la clique qui n'est pas en contact avec P L'utilisation de verseurs rend la taille de la mesure indépendante. Quoi qu'il en soit, même en utilisant des verseurs, leur comparaison est affectée par le cadre de référence. Pour surmonter ce problème, la similarité de position relative est calculée en suivant la même approche que celle adoptée pour le calcul de la similarité de joint dans le cas d'un type de surface.

$\mu^{position}$

Definition 5.7. $\mu^{position}(C) =$

$$\frac{1}{|\mathcal{N}_C|} \sum_{(n_q^i, n_k^j) \in C} 1 - d_{Dir}((n_q^i, n_k^j)) \tag{50}$$

où,

- $d_{Dir}((n_q^i, n_k^j)) = abs(\sigma(Var_{Dir}(n_q^i)) - \sigma(Var_{Dir}(n_k^j)))$
- $Var_{Dir}(n_q^i)$ est une matrice de la variation des versets directionnels entre les centres de gravité des pièces correspondant au nœud n_q^i et le noeud n_q^l tel que $(n_q^i, n_q^l) \notin \mathcal{A}_C$,
- $Var_{Dir}(n_k^j)$ est une matrice de la variation des versets directionnels entre les centres de gravité des pièces correspondant au nœud n_k^j et le noeud n_k^l tel que $(n_q^i, n_q^l) \notin \mathcal{A}_C$.

5.4 Mesure de similarité de structure : $\mu^{structure}$

Un modèle d'assemblage a différentes structures de produit en fonction de l'intention de conception de l'utilisateur, donc nous visons à définir une mesure de structure $\mu^{structure}(C)$.

L'EAM code la structure hiérarchique d'un modèle d'assemblage par les arcs $\mathcal{A}_S$ qui représentent la relation "fait de" parmi les composants d'un modèle. Dans ce cas, pour chaque paire de nœuds (n_q^i, n_k^j) et (n_q^l, n_k^h) dans la clique, nous comparons les relations structurelles des paires de nœuds (n_q^i, n_q^l) et (n_k^j, n_k^h). Cela signifie que nous devons vérifier si les nœuds n_q^i et n_q^l appartiennent (ou n'appartiennent pas) au même sous-assemblage dans G_q et si les nœuds n_k^j et n_k^h appartiennent (ou n'appartiennent pas) au même sous-assemblage dans G_k.

En utilisant une distance, qui affecte 1 si la paire de nœuds $(n_q^i,\ n_q^l)$ a une relation différente de la paire $(n_k^j,\ n_k^h)$ ou 0 sinon, nous définissons la mesure de similarité de structure d'une clique C comme suit:

$\mu^{structure}$

Definition 5.8. $\mu^{structure}(C) =$

$$\frac{1}{|\mathcal{N}_C|^2} \sum_{((n_q^i, n_k^j),(n_q^l, n_k^h)) \in C \times C} \left[1 - d_{Str}\Big((n_q^i, n_k^j), (n_q^l, n_k^h)\Big)\right] \tag{51}$$

où,

- $d_{Str}(((n_q^i, n_k^j), (n_q^l, n_k^h)) =$
$$\begin{cases} 1 & \text{if } \Big[(\exists n_q^* \in \mathcal{N}_q) \text{ and } (\nexists n_k^* \in \mathcal{N}_k)\Big] \text{ or} \\ & \Big[(\nexists n_q^* \in \mathcal{N}_q) \text{ and } (\exists n_k^* \in \mathcal{N}_k)\Big] \\ & \qquad \text{such that} \\ & \Big[\Big((n_q^i, n_q^*), (n_q^l, n_q^*) \in \mathcal{A}_{S_q}\Big) \text{ and} \\ & \Big((n_k^j, n_k^*), (n_k^h, n_k^*) \in \mathcal{A}_{S_k}\Big)\Big] \\ 0 & \text{autrement} \end{cases}$$

5.5 Combinaison de mesures de similarité pour différents niveaux de similarité

Jusqu'à présent, nous avons défini des mesures de similarité pour les différents aspects caractérisant un assemblage. Nous devons les combiner afin de fournir une mesure unique pour classer les modèles dans le cadre de recherche. La composition des mesures de similarité devrait considérer la possibilité de pondérer chaque valeur de similarité pour un certain facteur de pertinence choisi pendant la requête ou pendant la consultation des résultats. Une autre étude intéressante concerne la capacité d'attribuer des poids appropriés en fonction d'un scénario d'utilisation particulier. Ensuite, la mesure de similarité locale de l'assemblage proposée (γ_l) est définie comme la moyenne pondérée des mesures de similarité uniques appartenant à S.

Similitude locale

Definition 5.9. La *similitude locale* entre deux modèles est évaluée par la fonction γ_l:

$$\begin{aligned} \gamma_\ell &: \mathcal{D}_{q,k} \times \mathcal{W} \longrightarrow [0,1] \\ \gamma_\ell(C, w) &= \frac{w^{sh}\mu^{shape}(C) + +w^{jo}\mu^{joint}(C) + w^{po}\mu^{position}(C) + w^{st}\mu^{structure}(C)}{w^{sh} + w^{jo} + w^{po} + w^{st}} \end{aligned} \tag{52}$$

où $\mathcal{W} = [0,1]^4$

La mesure locale exprime combien la partie correspondante des deux modèles d'assemblage est similaire, pour fournir des informations sur la façon dont les parties sont globalement ou partiellement similaires, la mesure locale γ_l est pondérée en fonction d'une certaine *facteur de couverture*, où avec le terme *facteur de couverture* nous nous référons au pourcentage d'éléments considérés comme similaires par rapport à tous les éléments du modèle de requête et des modèles de comparaison. Plus précisément, nous définissons deux facteurs de couverture comme suit:

Couverture

Definition 5.10.

$$PCF = \text{Facteur de couverture partielle} = \frac{|\mathcal{N}_C|}{|\mathcal{N}_q|} \quad (53)$$

$$GCF = \text{Facteur de couverture global} = \frac{2|\mathcal{N}_C|}{|\mathcal{N}_q| + |\mathcal{N}_k|} \quad (54)$$

où, $\mathcal{N}_C$, $\mathcal{N}_q$ et $\mathcal{N}_k$ représente les nœuds de la clique C, les nœuds du modèle de requête et les nœuds du modèle de comparaison.

Les similarités globales et partielles sont définies en équilibrant la similarité locale sur ces *facteurs de couverture.* Un poids plus précis pourrait tenir compte de la «pertinence» des pièces, par exemple, en termes de volume ou d'appartenance de classe (c'est-à-dire qu'un rivet pourrait être négligeable par rapport à un engin).

Similitude partielle

Definition 5.11. La *similitude partielle* entre deux modèles est évaluée par la fonction γ_p:

$$\begin{aligned} \gamma_p &: \mathcal{D}_{q,k} \times \mathcal{W} \longrightarrow [0,1] \\ \gamma_p(C,w) &= PCF \cdot \eta(C,w) \end{aligned} \quad (55)$$

Similarité globale

Definition 5.12. La *similarité globale* entre deux modèles est évaluée par la fonction γ_g:

$$\begin{aligned} \gamma_g &: \mathcal{D}_{q,k} \times \mathcal{W} \longrightarrow [0,1] \\ \gamma_g(C,w) &= GCF \cdot \eta(C,w) \end{aligned} \quad (56)$$

Notez que la similarité globale implique les similitudes partielles et locales. Alors que la similarité partielle implique la similarité locale, l'inverse ne tient pas. En fin, nous avons les résultats suivants:

- Deux modèles sont *100% globalement similaires* si et seulement s'ils sont *100% localement similaires* et $GCF = 1$
- Deux modèles sont *100% partiellement similaires* si et seulement s'ils sont *100% localement similaires* et $PCF = 1$

6 Expérimentes et discussions

Pour tester l'efficacité du cadre proposé, un ensemble de modèle d'assemblage est requis. Malheureusement, les benchmarks de formes mécaniques les plus connus dans la littérature, comme le Princeton Shape Benchmark (PSB) [112], le National Design Repository (NDR) [99] et l'Engineering Shape Benchmark (ESB) [58], ne sont pas appropriés pour notre but, puisqu'ils classent juste des pièces et ne considèrent pas les modèles d'assemblage, et, jusqu'ici, aucune base de données publique n'existe pour évaluer et comparer les systèmes de récupération d'assemblages [51, 96]. Les auteurs d'autres méthodes de récupération d'assemblages ont développé leur propre ensemble de données, de toute façon, ils ne sont pas publics. Ensuite, nous avons construit un ensemble de modèle d'assemblage qui contient 140 modèles disposés en 12 classes (comme indiqué dans Table 3) avec 15057 parties au total dont 5343 sont uniques.

Table 3: Classification des assemblages CAO dans notre ensemble de test

Catégorie	Numéro de modèle
Actionneur linéaire	10
Bride d'accouplement	5
Charnière	4
Éolienne double rotors	13
Mélangeur à hélice	18
Mill max	8
Multiplicateur éolienne	22
Piston	5
Réducteur hydrolienne	6
Rotor hydraulique	6
Roulements	36
Trains d'atterrissage	7
Total	**140**

Pour afficher les résultats de manière intuitive et conviviale, nous avons développé plusieurs pages Web dynamiques basées sur HTML5, jQuery, Ajax et PHP, où une librairie X3D est utilisée pour la visualisation du modèle. De cette façon, l'utilisateur ne verra pas une liste de noms, mais une vue d'ensemble 3D des modèles comparés avec les composants reconnues comme similaires mises en évidence comme illustré en Figure 20. Dans la vue 3D, les composants correspondants sont colorés en bleu tandis que le reste du modèle est en rouge. Les barres de l'histogramme indiquent les valeurs des mesures de similarité locales (barre orange), globale (vert) et partielle (violet).

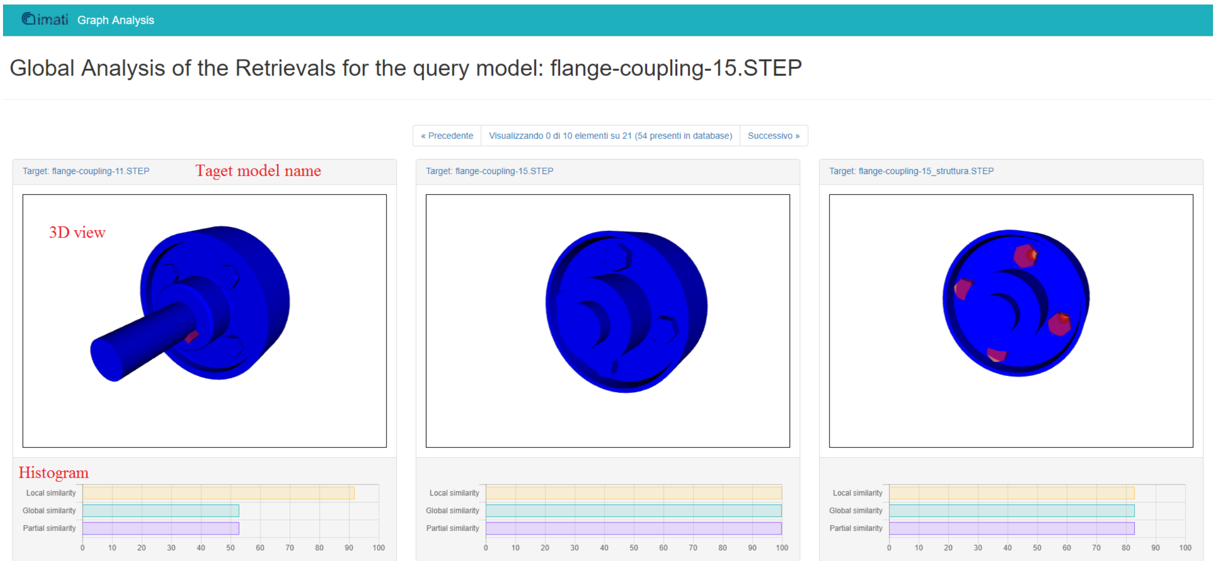


Figure 20: Illustration des modèles de comparaison pour une requête donnée

Chaque modèle peut être analysé dans une autre page (voir Figure 21), où modèle de requête et modèles de comparaison sont affichés. Dans cette vue 3D, chaque paire de parties correspondantes est mise en évidence par une couleur différente et les valeurs des mesures individuelles sont indiquées dans un graphique radar. Dans le graphique, chaque axe identifie un critère de similarité différent ($\mu^{forme}, \mu^{joint}, \mu^{position},\ mu^{structure}$). En outre, nous rapportons deux valeurs qui indiquent le nombre d'éléments similaires sur le nombre d'élément du modèle de requête ($\frac{|\mathcal{N}_C|}{|\mathcal{N}_q|}$) et le nombre d'éléments similaires sur le nombre d'élément du modèle de comparaison ($\frac{|\mathcal{N}_C|}{|\mathcal{N}_k|}$). Nous avons ajouté cette information dans le tableau de similarité pour améliorer le contenu informatif.

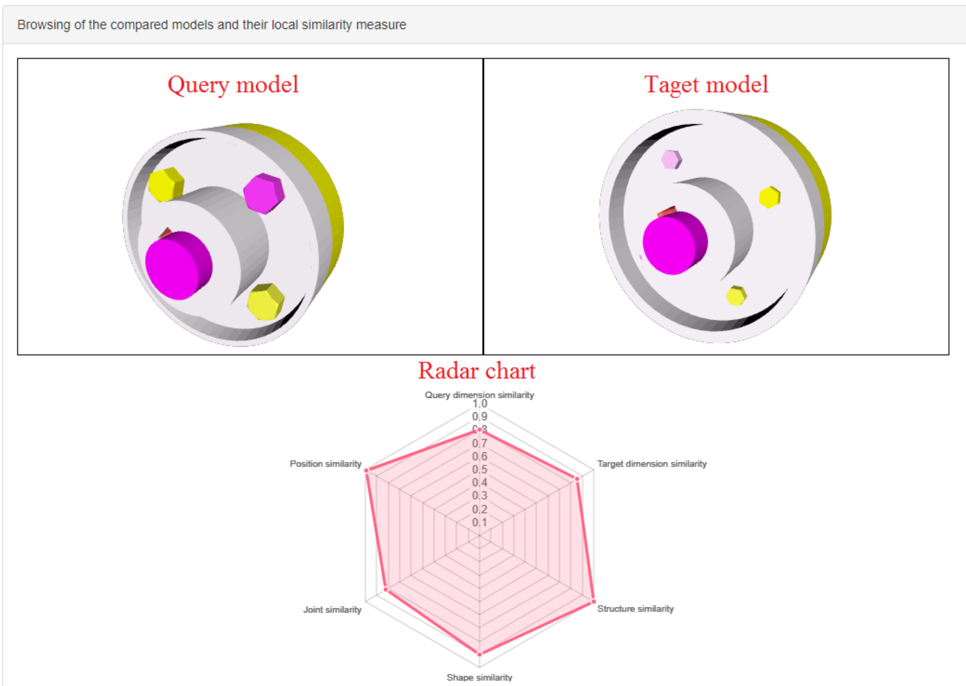


Figure 21: Analyse de la similitude entre deux modèles

6.1 Bride

Le premier exemple est basé sur une bride mécanique, c'est-à-dire un composant utilisé pour connecter deux objets. Dans cet exemple, le modèle de requête est illustré en Figure 22. Le modèle a quatre vis et quatre écrous arrangés dans un pattern de translation circulaire, deux brides principales, deux axes et deux clefs. Toutes les pièces sont organisées dans une structure plate, c'est-à-dire sans aucun sous-assemblage. Le modèle ne présente aucune intersection volumétrique et chaque vis est en contact avec l'écrou correspondant à travers une face cylindrique idéalisée (c'est-à-dire qu'aucun fil n'est conçu).

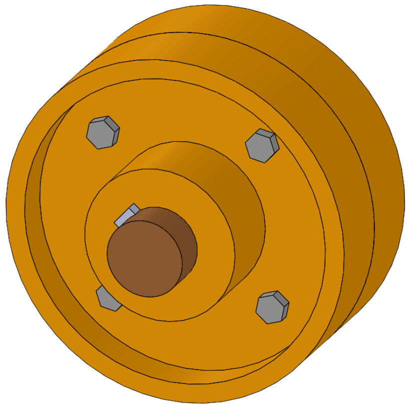


Figure 22: Bride utilisé comme modèle de requête

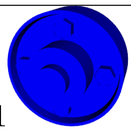
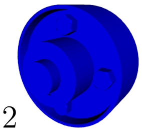
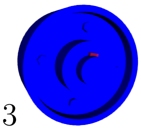
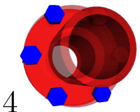
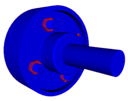
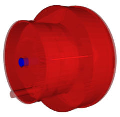
Les critères sélectionnés pour la recherche exigent que les nœuds soient similaires en fonction de la forme, du type de composant et du type de pattern. Dans ce cas, le seuil ε_{Shape} pour la forme est mis à 0.20, donc deux composants doivent avoir des formes similaires à 80% en fonction des valeurs de leurs harmoniques sphériques 3D. Au final, le critère de similarité pour les arcs nécessite que deux paires de nœuds compatibles aient le même nombre de rotations et de translations permises.

Les valeurs simples des mesures $\mu^{forme}, \mu^{joint}, \mu^{position}$ et $\mu^{structure}$ sont signalées dans Table 4. Les poids utilisés pour calculer γ_l, γ_p et γ_g sont $\mathcal{W}_{bride} = \{w^{sh}, w^{jo}, w^{po}, w^{st}\} = \{1, 1, 1, 1\}$. Cela signifie que les mesures $mu^{shape}, \mu^{joint}, \mu^{position}$ et $\mu^{structure}$ sont utilisées avec la même importance pour calculer les valeurs finales de similarité (global, partiel et local).

Le premier modèle correspond au modèle de requête, il est donc évident que toutes ses mesures ont des valeurs égal 1 car ils correspondent parfaitement.

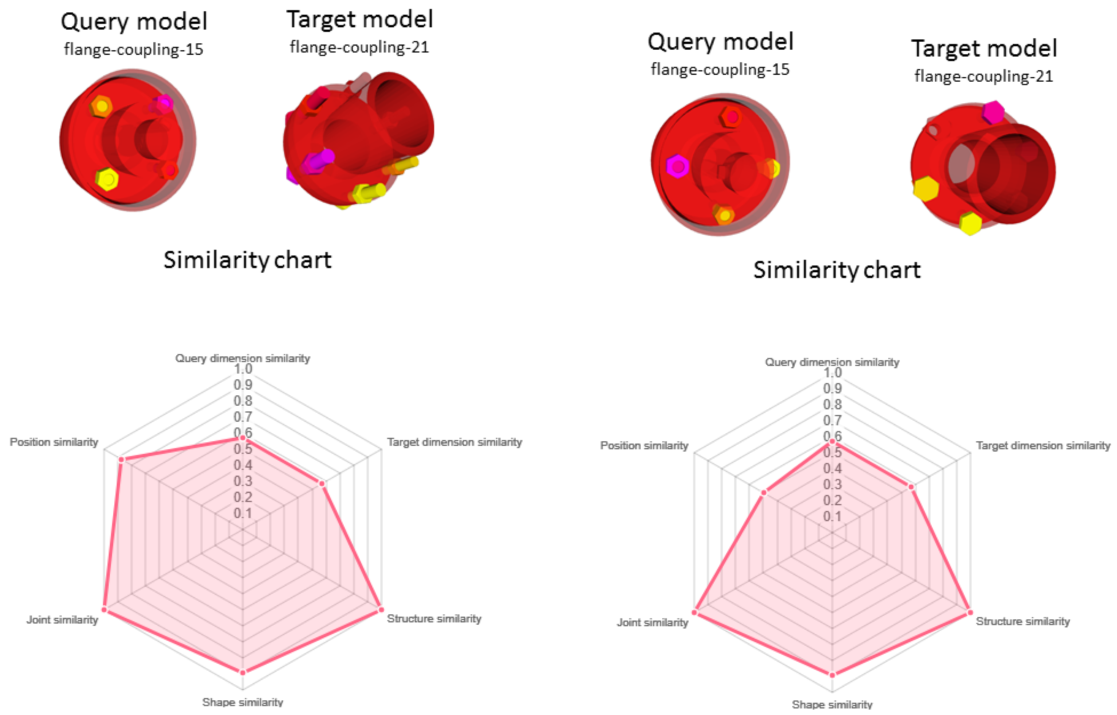
Le deuxième modèle a les mêmes composants du modèle de requête, c'est-à-dire le même nombre de pièces, la même forme et les mêmes contacts mais organisés de manière différente. Dans le modèle de requête, la structure est plate, tandis que dans ce modèle, l'ensemble des vis et l'ensemble des écrous sont rassemblés en formant deux sous-assemblages. Ainsi, $\mu^{structure}$ est inférieur à 1 et ce facteur diminue la valeur finale de la mesure de similarité locale. Puisque toutes les composantes de la requête et du modèle du comparaison sont appariées, les valeurs des similarités partielles et globales correspondent à celles locales, soit $PCF = 1$ dans la définition 5.11 et $GCF = 1$ dans la définition 5.12.

Table 4: Évaluation de similarité pour le modèle de requête en Figure 22

Modèle	μ^{shape}	μ^{joint}	$\mu^{position}$	$\mu^{structure}$	γ_l	γ_p	γ_g
1 	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2 	1.00	1.00	1.00	0.35	0.83	0.83	0.83
3 	0.77	0.68	0.99	1.00	0.86	0.74	0.74
4 	0.89	1.00	0.87	1.00	0.94	0.54	0.54
5 	0.74	1.00	0.95	1.00	0.92	0.52	0.52
6 	0.77	0.00	0.82	1.00	0.65	0.19	0.05

Le troisième modèle de Table 4 est une bride dont les vis et les écrous présentent une intersection volumétrique, puis ils sont appariés grâce à l'utilisation de l'attribut "UnSolved". Différent des arcs du modèle de requête, dans ce modèle les arcs de type "UnSolved" n'ont pas le nombre de rotations et de translations autorisées, alors selon les définitions 5.5, cette différence affecte leur similarité au niveau du joint. Dans cet exemple, le nombre de composants similaires est de douze et le nombre de composants dans les modèles de requête et de comparaison est de quatorze, alors les mesures partielles et globales sont inférieures à la locale selon le même facteur, soit $GCF = \frac{2 \cdot 12}{14+14} = 0.86$ dans la définition 5.11 et $PCF = \frac{12}{14} = 0.86$ dans la définition 5.12.

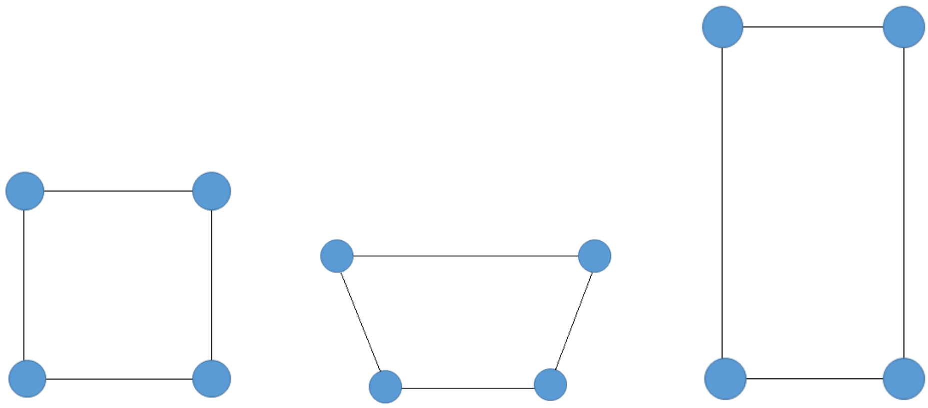
Les quatrième et cinquième modèles de Table 4 ont des mesures très similaires, mais en un coup d'œil, la couverture des deux modèles de comparaison est différente. En effet, la couverture des modèles est mesurée en fonction du nombre d'éléments assortis et les deux modèles ont le même nombre d'éléments assortis: quatre vis et quatre écrous pour le quatrième modèle contre quatre vis, deux flasques principaux, un arbre et une clé dans le modèle cinquième. Une évaluation utilisant le volume peut améliorer la perception visuelle de la similarité, mais nous pensons qu'en général, il est plus significatif de considérer la pertinence des parties correspondantes, c'est-à-dire que les éléments de fixation devraient être moins importants qu'un arbre. Bien sûr, ce genre de considération nécessite une étude sur l'importance pour chaque catégorie de composants des différents types dans les modèles d'assemblage mécanique.



(a) (b)

Figure 23: Two different results for the fourth model in Table 4 where the arrangement of the parts is different and detected by different values of the position measure

Pour le quatrième modèle, nous reportons en Figure 23 également deux cliques différentes, qui correspondent à deux ensembles différents de pièces similaires. Les modèles de requête et de comparaison ont un pattern circulaire de vis et d'écrous, mais avec un nombre différent d'éléments répétés (c'est-à-dire quatre dans le modèle de requête et six dans le modèle de comparaison). Ainsi, dans le modèle de comparaison, il n'est pas possible de trouver quatre vis et écrous équidistants qui couvrent toute la circonférence. La solution la plus similaire qui peut être trouvée a une distance constante entre les éléments du modèle (Figure 24(b)). Ce type d'information est mis en évidence dans l'évaluation de similarité par la mesure de position. En effet, dans le premier cas, où la distance entre l'ensemble des vis et des écrous est constante comme dans le modèle de requête, la valeur de position est supérieure à la seconde où les vis et les écrous sont disposés comme dans Figure 24(c), perdant l'arrangement qui caractérise le modèle de requête. Ensuite, la mesure de similarité de position aide le classement des objets récupérés proposant avant ceux avec un arrangement régulier comme celui présent dans le modèle de requête.



(a) Baricenters des vis du modèle de requête dans Figure 23

(b) Baricenters des vis assorties dans le modèle du comparaison en Figure 23(a)

(c) Baricenters des vis assorties dans le modèle du comparaison en Figure 23(b)

Figure 24: Représentation des baricenters à vis pour les modèles en figure Figure 23

Le cinquième modèle de Table 4 est très similaire au modèle de requête, la forme de ses parties diffère seulement pour les arbres et les brides principaux qui ont une frontière plus épaisse que celle de la requête. Cependant, dans ce modèle les vis et les écrous présentent des dégagements, alors ces composants ne sont pas en contact. Le fait que tous les composants ne correspondent pas est reflété par les mesures partielles et globales qui sont inférieures à la locale. Cette différence indique que les parties correspondantes sont très similaires, mais ne couvrent pas tout le modèle de requête ni le modèle de comparaison. Ceci est confirmé en analysant les valeurs des mesures uniques rapportées dans Table 4. Comme prévu, les valeurs de joint, de la position et de la structure sont très élevées, tandis que la valeur de la similarité de forme met en évidence de petites différences dans les composants appariés. La variation la plus significative concerne le nombre de composants appariés. Dans cet exemple, les modèles de requête et de comparaison ont quatorze pièces et huit d'entre eux sont appariés, alors $PCF = GCF = 0.57$ et ces valeurs affectent négativement les mesures partielle et globale. En outre, ce qui empêche une correspondance complète est le type de contact. En particulier, les quatre écrous du modèle cible ne sont pas en contact avec les vis et la clé et l'arbre présente des contacts différents. En effet, comme illustré dans Figure 25, la clé et l'arbre dans le modèle de requête sont en contact par trois faces planes, donc une translation est autorisée, alors que dans le modèle de comparaison les deux parties sont en contact par quatre visages planaires et aucun mouvement est possible.

Enfin, à partir des valeurs des différents niveaux de mesures de similarité, l'utilisateur peut facilement comprendre que le sixième modèle n'est pas adapté à un objectif de réutilisation de conception. En effet, la mesure de similarité partielle est très faible, ce qui indique que peu d'éléments du modèle de requête ont pas été appariés.

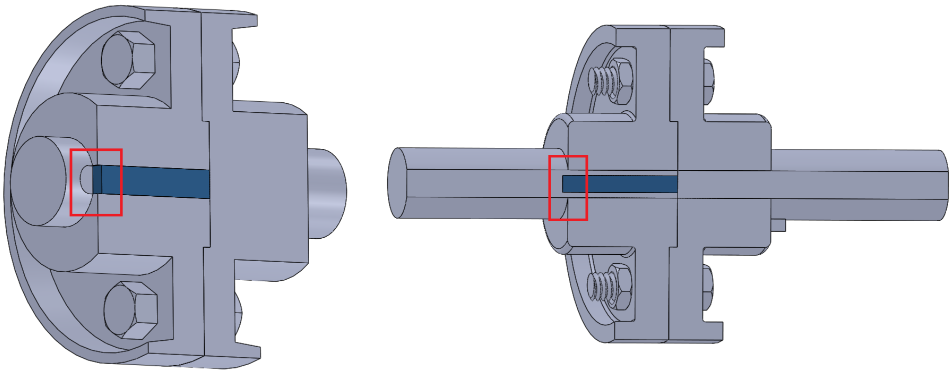


Figure 25: Différents contacts entre la clé et l'arbre des modèles d'assemblage de requête (à gauche) et de comparaison (à droite)

6.2 Réducteurs planétaires

Comme deuxième exemple, nous rapportons une boîte de vitesse planétaire. Cet ensemble fonctionnel a plusieurs paires d'engrenages planétaire comme le modèle représenté en Figure 26, qui a été utilisé comme modèle de requête. Dans cet exemple, nous assouplissons les critères de similarité en considérant le type de composant et le type de modèle, tandis que le critère de similarité pour les arcs exige que deux paires de parties similaires aient le même nombre de rotations et de translations permises.

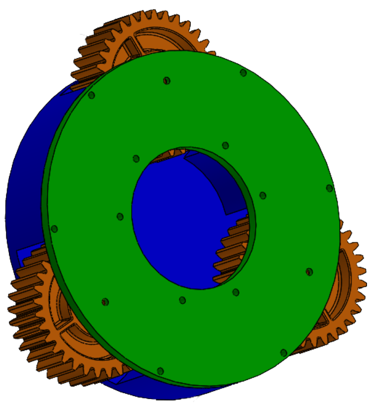
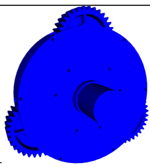
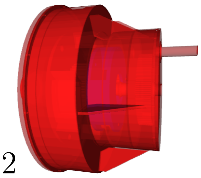
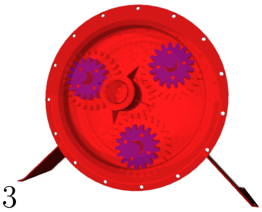
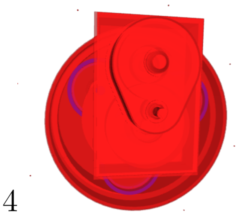
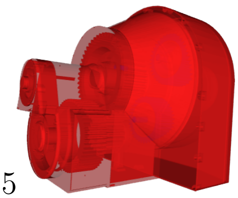
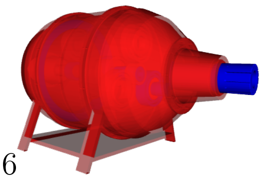


Figure 26: Réducteurs planétaires utilisés comme modèle de requête

Les résultats de cette recherche sont illustrés dans la Table 5, où les poids utilisés pour calculer γ_l, γ_p et γ_g sont $\mathcal{W}_{engrenage} = \{w^{sh}, w^{jo}, w^{po}, w^{st}\} = \{1, 1, 1, 0\}$. Cela signifie que $\mu^{structure}$ n'est pas considéré pour calculer les valeurs finales de similarité (globale, partielle et locale). Notez que ne pas tenir compte du critère de similarité de forme n'implique pas de ne pas l'évaluer. En effet, les critères de similarité sont utilisés pour construire le graphe d'association, tandis que l'évaluation de similarité classe les solutions récupérées.

Table 5: Évaluation de similarité pour le modèle de requête en Figure 26

Modèle	μ^{shape}	μ^{joint}	$\mu^{position}$	γ_l	γ_p	γ_g
1 	1.00	1.00	1.00	1.00	1.00	1.00
2 	1.00	1.00	1.00	1.00	1.00	0.18
3 	0.78	0.00	1.00	0.59	0.44	0.06
4 	0.66	0.00	1.00	0.55	0.42	0.04
5 	0.76	0.00	0.80	0.52	0.39	0.06
6 	0.74	0.00	0.61	0.45	0.34	0.10

En général, pour tous les modèles récupérés, on peut observer que la mesure globale est beaucoup plus faible que les autres. Cela suggère que le modèle de requête est inclus dans les modèles cibles.
Le premier modèle correspond au modèle de requête, alors toutes ses mesures ont des valeurs maximales car elles correspondent parfaitement.

Le deuxième modèle a des valeurs élevées de similarités locales et partielles, ce qui suggère que les valeurs de similarité individuelles sont élevées et que la totalité de la requête est incluse dans le modèle de comparaison. En effet, ses mesures de similarité ont des valeurs presque maximales et seule la mesure globale est faible, du fait que le modèle de requête est entièrement inclus dans un modèle plus grand.

Dans le troisième modèle récupéré, les composants appariés sont les trois engrenages et les trois axes, dont les formes sont similaires à celles du modèle de requête, alors μ^{shape} a une valeur élevée. μ^{joint} est égal à zéro car aucun contact n'est présent entre les composants correspondants. Comme tous les composants sont déconnectés, il n'y a pas de variation de la rotation/translation de l'incident à comparer. Cette définition affecte négativement la valeur finale de la mesure de similarité locale lorsque le poids w^{joint} n'est pas nul. Quoi qu'il en soit, nous pensons qu'assigner $\mu^{joint} = 1$ quand aucun contact n'est présent peut être trompeur pour l'utilisateur pendant son analyse de modèles.

Toujours dans le quatrième modèle de Table 5, les composants correspondants sont les trois engrenages et les trois axes, mais différemment d'avant, dans ce modèle les engrenages sont modélisés avec une forme simplifiée et ils sont reconnus grâce à l'attribut $CompType$ de l'EAM, qui identifie trois simples anneaux comme des engrenages exploitant le contexte des composants. Notez qu'en incluant la compatibilité de forme, cette configuration ne serait probablement pas récupérée (c'est-à-dire qui dépend par le seuil de similarité de forme choisi), puisque les formes des engrenages planaires sont très différentes de celles proposées dans le modèle de requête.

Pour les deux derniers modèles de Table 5, les mêmes considérations que pour le troisième modèle tiennent, c'est-à-dire que trois engrenages et trois arbres sont récupérés dont la forme est similaire à celle du modèle de requête. Encore une fois le μ^{joint} est zéro puisqu'il n'y a aucun contact entre les parties récupérées.

7 Conclusion et perspectives

La conception de nouveaux produits peut bénéficier de la réutilisation des solutions existantes et des leur informations associées, il est donc utile d'avoir de bons instruments pour la récupération des modèles CAO. De nos jours, les systèmes CAO sont intégrés aux systèmes de gestion des données de produits (PDM), qui permettent de suivre et de contrôler les données relatives aux produits existants. Ces systèmes sont très appropriés pour gérer des recherches textuelles (par exemple du matériel ou des spécifications), mais offrent des capacités limitées pour des recherches basées sur la géométrie (par exemple des recherches basées sur formes ou articulations).

Pour surmonter ces limites, l'objectif général de cette thèse était la proposition d'un système de recherche d'assemblage capable de comparer des modèles d'assemblage selon différents critères de similarité. Les informations sont organisées en une structure multi-couche, appelée modèle EAM (Enriched Assembly Model), à travers laquelle un modèle d'assemblage est caractérisé selon quatre couches de données: la couche de structure, la couche d'interface, la couche de forme et la couche statistique. La couche de structure représente la structure d'assemblage hiérarchique du modèle CAO que été spécifié par le concepteur. En outre, il comprend des informations concernant le type de composants et les dispositions des pièces répétées. La couche d'interface spécifie les contacts existant entre deux pièces et leur joint équivalent. La couche de forme vise à caractériser la forme des composants d'assemblage en utilisant des descripteurs d'harmoniques sphériques et leur taille en utilisant des informations de volume et de surface. En fin, la couche statistique a pour but de filtrer de grands bases de données et de réduire le nombre de modèles à comparer en utilisant plusieurs valeurs numériques, telles que le nombre de modèles de composants répétés d'un type spécifique.

Ce descripteur a été conçu à l'aide d'une structure basée sur un graphe et la comparaison de deux modèles d'assemblage a été traduite dans un problème de correspondance de graphe. En utilisant le MCS (Sous-graphe Commun Maximum) correspondance et en exploitant le pourcentage de composants correspondants entre le modèle de requête et le modèle de comparaison, nous pouvons évaluer différents types de similarité: global (le modèle de requête est complètement similaire au modèle de comparaison), partial (le modèle de requête est inclus dans le modèle de comparaison) et local (le modèle de requête et le modèle de comparaison partagent des composants similaires). En fin, puisque les modèles d'assemblage peuvent être similaires selon différents critères, nous avons défini un ensemble de mesures pour évaluer les différents similitudes.

Des extensions et ses perspectives possibles sont en organisant la discussion dans les trois points principaux suivants:

(i) l'intégration dans les systèmes PDM/PLM,

(ii) la définition d'un descripteur de assemblage,

(iii) la comparaison de deux EAM,

(iv) l'amélioration de l'expérience utilisateur.

Intégration dans les systèmes PDM/PLM

Le cadre de récupération proposé a été conçu pour être un système autonome, de sorte qu'il peut également traiter un base de données non organisé de modèles 3D. De toute façon, si ce système est intégré dans des systèmes PDP ou PLM, d'autres données utiles peuvent être incluses dans le processus de raisonnement pour analyser des situations complexes difficilement reconnaissables en utilisant uniquement des informations géométriques. Par exemple, sachant que pour un produit certains composants sont acquis auprès d'un fournisseur peut être utilisé pour sélectionner un ensemble approprié de modèles pour l'identification de ses composants.

La définition d'un descripteur de assemblage

L'un des principaux objectifs pour la définition du descripteur d'assemblage était la capacité de capturer les caractéristiques d'un assemblage et la possibilité d'extraire automatiquement les données requises du modèle CAO. Cet objectif a été difficile en raison de la présence de simplifications et de configurations irréalistes souvent présentes lors de la conception des modèles d'assemblage. Dans cette thèse, nous avons extrait des informations significatives, y compris mais sans s'y limiter, la classification des composants de l'assemblage, la disposition des pièces répétées et les contacts entre les pièces.
À l'avenir, l'EAM peut être amélioré avec des autres informations, tels que en ajoutant un description de forme de l'entière modèle d'assemblage.

La comparaison de deux EAM

Jusqu'à présent, nous comparons chaque modèle de la base de données avec le modèle de requête. Certains d'entre sont immédiatement rejetés par l'utilisation de certains filtres, tels que le nombre de modèles d'un certain type. À l'avenir, nous visons à exploiter ces caractéristiques de filtrage pour développer un système d'indexation de la base de données afin de faciliter la sélection des modèles, qui sont des candidats significatifs pour une certaine requête, évitant ainsi un certain nombre de vérifications inutiles.

En ce qui concerne la procédure de comparaison, puisque un EAM a été représenté comme un multi-graphe attribué, la comparaison de deux modèles d'assemblage a été traduite en un problème d'appariement de graphe. Afin d'évaluer différents types de similarité (globale, partiale et locale), il est nécessaire de détecter les sous-graphes communs entre deux modèles d'assemblage et de résoudre un problème de MCS. Considérant les techniques actuelles pour résoudre le problème *Sous-graphe Commun Maximum* (MCS), il a été réduit en un problème de Clique Maximum (MC). Une manière intéressante de réduire la complexité du problème MC serait de réduire la taille du graphe d'association sur lequel le problème MC est basé. Pour réduire sa taille, à l'avenir, nous visons à explorer deux possibilités.

- **Réduire le nombre de nœuds d'association**
  La première façon de réduire le nombre de nœuds d'association est de ne pas considérer les composants moins important, tels que les boulons et les écrous. Cette possibilité doit être soigneusement étudiée pour identifier quels sont les composants les moins importants et comment traiter les relations entre les composants restants une fois que ceux qui n'ont pas de sens ne sont pas pris en compte.
  Une autre option pour réduire le nombre de nœuds dans le graphe d'association est de envisager comme un seul nœud d'association des ensembles significatifs de nœuds, tels que les nœuds associés aux pièces répétées ou qui sont solidaires, cet-à-dire que les distances mutuelles entre les pièces restent les mêmes pendant le mouvement de l'objet. De cette façon, le graphe d'association sera plus petit, et, si une possible solution sera récupère, alors elle doit être analysé en profondeur. Pour atteindre cet objectif, outre la gestion des macro-nœuds dans la représentation des descripteurs et les procédures relatives à la mise à jour de ses relations, nous devons également comprendre comment identifier des ensembles significatifs de composants et comment les gérer dans les définitions de la mesure de similarité.

- **Réduire le nombre d'arcs d'association**
  Dans ce but, nous cherchons à extraire d'autres informations utiles du modèle d'assemblage pour mieux caractériser l'arrangement des pièces. En particulier, nous voudrions utiliser des relations supplémentaires dans l'EAM, qui encodent la position mutuelle des composants dans les modèles d'assemblage, par exemple si deux arbres différents sont coaxiaux ou parallèles dans l'espace 3D du modèle d'assemblage.

L'amélioration de l'expérience utilisateur

Pour cette tâche, nous avons proposé la définition d'une requête partielle, une visualisation appropriée pour la navigation des résultats et enfin la définition des plusieurs mesures, qui évaluent la similarité d'assemblage selon différents critères de similarité.
Des efforts supplémentaires devraient aborder les perspectives suivantes.

- **Évaluer les combinaisons de poids de mesure**
  Jusqu'à présent, les poids utilisés pour combiner l'ensemble des mesures pour calculer la similarité ont été fixés par l'utilisateur. Pour déterminer quels poids répondent le mieux aux besoins de l'utilisateur, il est nécessaire d'étudier comment les combinaisons de poids affectent le score final et, surtout, d'inclure les commentaires des utilisateurs pour indiquer quels résultats sont considérés pertinents pour la requête spécifique afin de collecte de données pour la spécification par défaut de poids.

- **La définition sémantique d'un modèle de requête**
  Un sujet de recherche intéressant, lié à la définition de la requête, est représenté par l'interprétation d'une requête textuelle telle qu'elle peut être traduite en un graphe à utiliser comme requête abstraite dans notre cadre de recherche.

- **Inspection des résultats**
  Même si, en général, les capacités de visualisation fournies offrent la possibilité d'identifier les composants dans les modèles d'assemblage récupérés qui sont similaires à la requête, dans le cas d'assemblages complexes, cette visualisation n'est pas toujours suffisante à la compréhension. Donc d'autres études sont nécessaires pour améliorer la compréhension dans de tels cas.

- **La navigation dans l'ensemble de données**
  L'utilisation de la comparaison réciproque de tous les modèles dans la base de données peut faciliter la navigation des résultats en visualisant des modèles similaires à celui sélectionné en fonction de toutes les mesures définies.

Introduction

The ability of retrieving a specific item among crowd of items is of utmost importance. Whilst nowadays text-based retrieval systems are able to extract information from huge text and provide significant results (e.g. Google™, Bing™ or Yahoo™), 3D content retrieval systems have still extensive room for improvement [113]. This type of retrieval can provide an important way to facilitate new designs allowing the reuse of existing CAD models and the access to the embedded knowledge [9]. It has been estimated that a large part of the design activity is based on the reuse of previous solutions to address new design problems [56, 124]. Indeed more than 75% of the design activity comprises reuse of previously existing knowledge [50]. Therefore, 3D engineering model searches are useful to avoid to spend time re-inventing, or re-designing, existing solutions.

Retrieving 3D CAD models does not regard only the reuse of existing models or of their associated information. Actually, its employment can provide benefits in various engineering activities [18]. For instance, in product standardization and rationalization, the retrieval of similar models can help in identifying interchangeable parts from separate projects to reduce the management and manufacturing costs. Another application for 3D CAD model retrieval is for supporting the management of maintenance operations. Indeed, knowing the rate of wear of a component C, the identification in CAD models of components similar to C is useful to plan maintenance operations organizing stocks in the warehouse.

To achieve this goal it is essential to have instruments to archive and retrieve models according to several suitable criteria. In fact, depending on the objectives of the search, the aspects defining what is similar and what is not may change. Since a while, Computer-Aided Design (CAD) tools are integrated with Product Data Management (PDP) systems, that support the design process storing and indexing project data for a fast retrieval. The data tracking usually involves the technical specifications of the product, provisions for its manufacturing and assembling, the types of materials that will be required to produce it and other information. However, design information is mostly contextual, i.e. many information is contained in the CAD model itself, and such systems provide a limited support in geometric searches [56].

To overcome these limitations, content-based methods for 3D model retrieval are being developed based on geometric descriptions. The retrieval of 3D models using geometric characteristic has been deeply investigated [8, 25, 41, 57, 117], where the common practice sees the use of pre-computed shape descriptors or signatures facilitating similar shape retrieval. However, 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 characteristics and then different relationships between their parts.

In this thesis, we intend to address the problem of assembly retrieval identifying and extracting meaningful characteristics to compare assembly models. This objective carries with it some challenges making our intent riveting from the scientific point of view.

Challenges

- **Assembly model description**
  The ability of retrieving existing models requires to structure and to organize the data such that the information can be efficiently retrieved [26]. The organization of the data may be influenced by the type of data itself and also by the interlinks between different information.

  However, when information is managed, it is essential to keep in mind the point of view of different users. Because the importance and the correctness of the retrieved results is determined also by the user background and needs. Hence, the same information can be relevant for an user and unrelevant for another. These divergences could be recognized by the retrieval system, for instance classifying a model according to different keys. Thus, it is necessary to define an assembly descriptor, which is able to characterize models using data of different nature.

- **Assembly model analysis**
  The characterization of assembly models based on its CAD description makes use of geometric reasoning. However, the analysis of an assembly model presents some problems. A notable difficulty is due to the large size of the databases and of the assembly models themselves. Indeed, a DMU can represent more than 1 million parts corresponding to several tera-bytes of data [52]. Other issues may derive from the way assemblies are modeled, including, but not limited to:

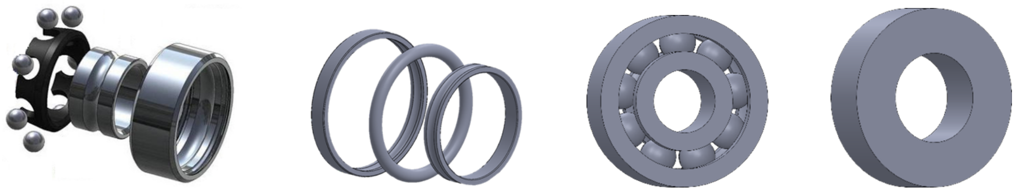
  i Different organization in sub-assemblies.
  Same objects can be organized according to different sub-assembly structures, depending on the objectives for which the model is created.

  ii Relations between parts not available.
  Sometimes relations between the parts of an assembly are not explicitly stored (e.g. constraints). Among the possible causes, this lack may derive from the file formats used by CAD systems to exchange assembly models, which do not allow storing this information.

  iii Position errors
  Additional problems, due to position errors, may be present causing unwished volumetric intersections or clearances and making challenging the effective analysis of assemblies.

  Thus, the procedures for the creation of an assembly descriptor have to take into account and manage all these issues to allow the extraction of the necessary information and an effective assembly comparison.

- **Similarity perception**
  Despite the use of conventions and rules in product definition, same components can be modeled in different ways. An example of this practice is illustrated in Figure 27, where a bearing is designed in four different manners: both as assembly models and as parts, in addition, each representation can be exhaustive allowing to recognize the bearing or simplified with some idealized shapes. These situations frequently occur when components are acquired from suppliers.



(a) Detailed bearing at assembly level

(b) Simplified bearing at assembly level

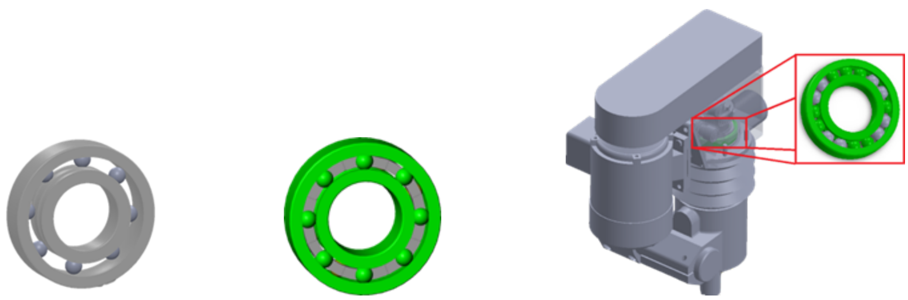
(c) Detailed bearing at part level

(d) Simplified bearing at part level

Figure 27: Examples of bearing components with different representations and resolutions

  This common practice makes arduous a comprehensive definition of assembly similarity. For instance, using only shape information to compare models in Figure 27 can be misleading since components with so different shapes have the same functionality. Hence, it is crucial to understand what has to be considered as similar in order to provide a solution that best fulfills user requirements.

- **Multiple matching**
  The similarity between two assembly models should be assessed not only in terms of global match. Supposing to have a query model as depicted in Figure 28(a), beside models entirely similar to the query, as the model in Figure 28(b), it is reasonable to retrieve also models that include the query, as the one depicted in Figure 28(c). Then, for engineering applications partial matching is definitely desirable. Anyhow, this aspect is not simple to be achieved in complex assembly models.



(a) Example of query model

(b) A possible retrieved model globally similar to the query

(c) A possible retrieved model which includes a component similar to the query model

Figure 28: Examples of global and partial matching

- **Evaluation of the assembly similarity**
  Due to the multiple similarity perception and matchings between two assembly models, it is arduous to rank the retrieved models in a proper manner for the user. For instance, on the one hand, we have two models with the same shape (e.g. shape similarity measure at 100%) and similar mating conditions (e.g. mating similarity measure at 80%). On the other hand, we have two models with the similar shape (e.g. shape similarity measure at 80%) and same mating conditions (e.g. mating similarity measure at 100%). If we consider a simple average between the two measures, we have that both the pairs of assemblies are similar at 90%. In this case, two different pairs of models can achieve the same values of similarity measure even if they are similar according to different criteria. Thus, it is challenging differencing these types of results to the user who performs a particular query.

Proposition

To face the challenges involved in assembly model retrieval, we propose a novel assembly model descriptor, through which we can represent different types of information of a model. The descriptor is organized in four different layers, which include the different type of information present in assembly models. The information present in the descriptor is automatically extracted analyzing assembly models and managing possible problems deriving from their design.

With the aim of providing a solution that best fulfills user requirements, in this thesis, we propose several application scenarios, which highlight what designers' necessities are and help us in the identification of interesting similarity criteria through which querying databases.

The chosen representation is graph-based, since it is suitable for partial matching operations and thus to address multiple matching requirements. In particular, we use an attributed multi-graph structure, where nodes correspond to assembly components and arcs encode different types of relationships between components. The attributes allow to include and organize information of the represented object. Using this representation, the comparison of two assembly models may be based on graph matching techniques. In this thesis, with the aim of permitting multiple types of matching, the graph matching has been formulated as a problem of finding the maximum common subgraph (MCS) between two attributed multi-graphs. Then, the detection of the MCS between two attributed multi-graphs has been reduced to determining the maximum clique (MC) in an association graph, which is defined based on the similarity criteria chosen by the user.

Since two assembly models can be similar according to several criteria, we aim to evaluate different types of similarity among assembly models by defining a set of proper measures. Then, an unique similarity measure is defined combining the previous set of similarity measures. In the end, with the intent of easing the user comprehension, we aim to define also an efficient visualization of the retrieved results.

Manuscript organization

The proposition of this thesis and how it has been achieved is described in six chapters as follows:

- **Chapter 1** has two main goals. First it aims at presenting general notions on the digital representation of assembly models, which are useful to understand the main problematics in assembly retrieval, and at investigating the scenarios that can benefit from the use of retrieval techniques. Then, our goal is depicted in its proper context, the main issues and challenges are highlighted.

- **Chapter 2** provides an overview of techniques for similarity assessment and model retrieval based on the intrinsic description of an assembly model. For a comprehensive analysis, we compare works in the state of the art according to several criteria proposed at the beginning of this chapter.

- Our proposition is illustrated in **Chapter 3**: a framework for a flexible assembly retrieval capable of extracting all the required data by reasoning on the information included in the digital model. The proposed system aims to allow the search of assembly models according to different selectable similarity criteria.

- **Chapter 4** illustrates the assembly model descriptor, the so-called Enriched Assembly Model (EAM), used to encode the information necessary for the retrieval. The EAM is represented as a multi-graph. The various modules for the creation of the EAM are described as well. Each module aims at extracting different information from the CAD model possibly coupling geometric and topological characteristics with engineering knowledge specific of the considered context.

- Using an attributed multi-graph representation, the problem of retrieving similar assembly models is performed by graph matching. **Chapter 5** illustrates the adopted techniques and the developed measures for the similarity assessment.

- The proposed concepts have been implemented in a prototype software fully developed by the author. The system has been tested using CAD models either downloaded from online repositories and produced by students in university courses. The discussion of the prototype and examples of obtained results are illustrated in **Chapter 6**.

- We discuss limits, possible evolutions and perspectives in the **Conclusions and perspectives** section.

- Finally, an **Appendix** section is included to provide background information on graph theory.

1

Scientific positioning of the research

The purpose of the chapter is to provide common notions about the Product Development Process (PDP), in order to understand the engineer requirements during the design phases.
Since querying any databases relies on the available and/or inferable information, we summarize the content of a CAD assembly model and more in general of a Digital Mock-up (DMU).
At last, this chapter proposes a comprehensive analysis of user intents for assembly model retrieval to understand what are the interesting similarity keys according to different scenarios, and what are the open issues.

Contents

1.1 Product Lifecycle

1.1.1 Product Development Process

Companies and enterprises have to be more and more competitive and the current market pushes them to release innovative products at a price appropriate for retailers and at the same time advantageous for customers. This target is increasingly difficult to achieve, as underlined by the well-known triptych cost-quality-delay [40]. In addition, there are always new competitors to deal with, more demanding customers to satisfy, suppliers to bargain and team to manage at global level. It follows that, business companies have to manage an incredible amount of data and people. Moreover, for making as much profit as possible, it is essential competing better against larger competitors looking for a way to speed up the Product Development Process (PDP), where PDP regards the entire set of activities required to move from the concept of a new product to a product ready for the market.

There is no a unique PDP, but it is adapted according to the product that companies aim at realizing [39]. For instance, the development of an airplane requires a set of tests under different physical phenomena, that are not necessary for the development of a coffee machine. Anyhow, a generic PDP structure, which can be adapted to companies and products, is illustrated in Figure 1.1. The common stages through which a new product passes during its life are: concept, research, development, prototyping, launch, usage and disposal. Depending on the company and on the type of product to be developed, the order of some phases can be rearranged.

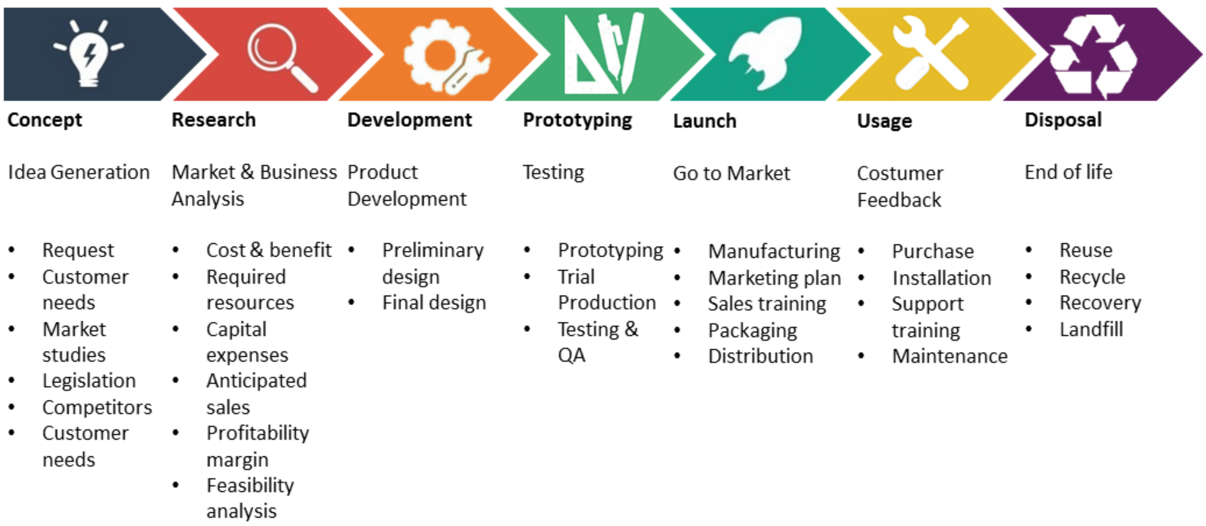


Figure 1.1: Traditional Product Lifecycle

In the concept phase, product requirements are defined based on a certain request, customer needs and market opportunities or simply to compete with competitors. Research phase addresses target market and determines deadlines, costs and pricing. Then, from the specifications, some aesthetic characteristics can be identified and some examples may be drawn, without defining any precise design but just sketching ideas.
After this theoretical phase the detailed design comes in the development phase. This phase refers to a set of activities that aim to translate the idea (i.e. the concept) of a product into a fully specified shape to be produced.

Here, CAD systems are commonly used to create a detailed design. This process usually includes virtual simulation to verify the behavior of the proposal with respect to specific scenarios. If something does not work properly, development team goes back into previous phases modifying some parameters (as shape modifications and/or dimension adjustments) fixing the right mode of operation. Thus, product design is a cyclic and iterative process.
For most of the products, this development phase progresses to prototyping in order to verify the quality of the developed product. Often, only a subset has to be tested for durability under realistic conditions and must withstand the toughest conditions. Also at this stage, if some tests are not satisfied, product development comes back at previous phases.

Generally, product development processes are not executed in a linear sequence but rather in a cyclic way (as in Figure 1.2), where the modifications and adjustments sometimes can cause some compromise in the various phases of the PDP [39]. Only when the product is in line with all the analysis done, the PDP is concluded and it passes to the following phases of its lifecycle. The number of iterations necessary to reach the end of the PDP affects the time and thus the cost of the production, then it is important for the companies trying to reduce the number of iterations capitalizing the knowledge of the existing products.

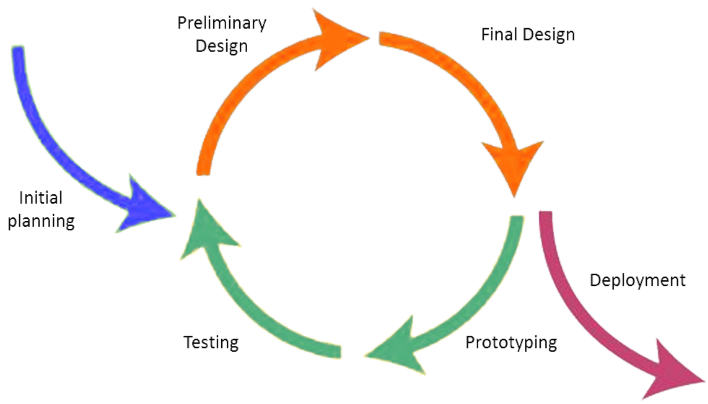


Figure 1.2: Iterative Product Development Process

Once the prototype satisfies testing stage, the product is ready for production and the organization of the distribution. These activities are part of the launch phase.
Now the new product is available for the usage by customers. This stage must manage the setting up of the operations to install the product and to support the future maintenance. The life of a product ends with its disposal, which should consider a possible reuse of the product or its landfill.

In order to improve the PDP, we aim to provide a retrieval system to access existing information and data such as components, sub-systems, materials, planning process, manufacturing strategies, production costs, as well as the geometry of the 3D models [132]. Moreover, 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. The different application scenarios are presented in section 1.3.

1.1.2 PLM Systems

Due to the complexity of modern products, whose development involves several disciplines, day by day PDP has become more and more collaborative. As an example, we can think to the development of a car. Most of its components were mechanical from the lock to the clutch, while now electronic components are widely used, e.g. for the ignition system, for the brake control by Anti-lock Braking System (ABS) and Electronic Brake-force Distribution (EBD), and for the control of the stability and reduction of vibrations at high speed by Electronic Stability Control (ESP). Moreover, many cars have integrated GPS systems, so during the development of a car also electronic systems have to be supported.
It is not reasonable that a car company can manage all these aspects by itself. It is more convenient to entrust and delegate some tasks to other companies which are expert in a particular field. In this way, a potentially beginner car company can achieve final product quicker boasting the best technology. Thus, PDP relies on cross-activities involving groups of experts and multiple partners. These groups face different problems of the product development and it is very common that they use different systems and information structures that might be incompatible among them.
Moreover, to reduce the time of development of complex products, it often involves the simultaneous development of their components [120]. This presumes an appropriate collaboration among all the specialists involved in the product development and production. To this aim, Product Lifecycle Management (PLM) systems allow merging information in a central environment. This product-centric information system may be accessible from several locations and allow simultaneous development process in order to reduce developing time.

The adoption of PLM systems positively impacts on the PDP and it has been embraced by leading business strategies and supporting technologies [52]. As suggested by the name, PLM is supposed to be the set of systems apt to manage the whole life of a product, from its conception and realization to its distribution, maintenance and recycle. These systems usually can be connected, as shown in Figure 1.3, with business systems to manage other important agents, as to communicate with customers by Customers Relationship Management (CRM) systems, manage suppliers (Supply Chain Management - SCM) and organize resources within enterprises (Enterprise Resource Planning - ERP).

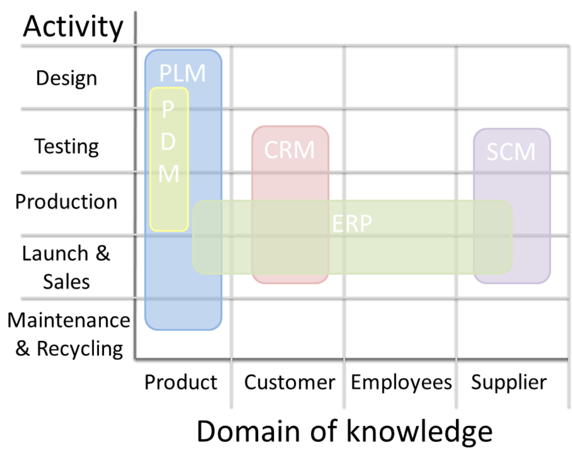


Figure 1.3: Interdependence among business systems

To support industries with the management, usage and reuse of data associated with CAD production, PLM systems communicate with different tools used during the development of a product. First of all, to manage the lifecycle of a product, it is essential having the data associated with the product. This information is present in PDM systems. In addition, PLM systems can include Project Portfolio Management (PPM) that describes methods to analyze and manage ongoing projects, and Manufacturing Process Management (MPM) that describes manufacture plan.
In order to cover collaboration and communication among the technologies involved in PLM systems, Digital Mock-Up (DMU) solutions are adopted to speed up the product development [103]. For instance, DMU solutions enhance the efficiency of reviewers starting in a very early stage verification process [81].

Considering the importance of the DMU in the PLM, our goal in this thesis is to provide a retrieval system which considers the information present in the DMU. The elements of a DMU are presented in section 1.2.

1.1.3 Digital Mock-Up in the Product Lifecycle

DMU is a digital representation of a product, a set of 3D polymorphic representations, complemented with some product data such as the description of what the model represents, its status and other useful information. DMU plays a central role when defining complex systems. It allows to have a global vision of the current status of a product and to evaluate progresses of projects in a simple manner such that involved participants are able to understand despite different backgrounds and experiences.

Currently DMUs are widespread employed in several industrial fields such as automotive, aerospace and also naval industries [109].
The goal of DMU is not restricted to represent a model, indeed, the mainstream use of DMU in industry is promoted by the need of replacing physical tests by virtual ones. It is worth adding that the complete design of a car can require up to 40 physical prototypes, each costing more than one million dollars [52]. Jean-Claude Hironde, researcher and deputy senior vice president at Dassault Aviation has claimed that using digital mock-up they have eliminated the need for physical prototype of the Falcon 7X. While, Harald Okruch, CAD/CAM management at Bombardier Rotax estimated a saving of around $360,000 per year thanks to digital prototypes in the first part of design phase [52].

The geometry of a product is the main information present in the DMU [108]. Figure 1.4 shows an example of DMU of a wind turbine speed reducer using its 3D representation, where different colors correspond to components. For a better understanding the DMU is sectioned by a vertical plane. The information present in the DMU is not limited to the geometry of a model and we will discuss the other data in more details in section 1.2.

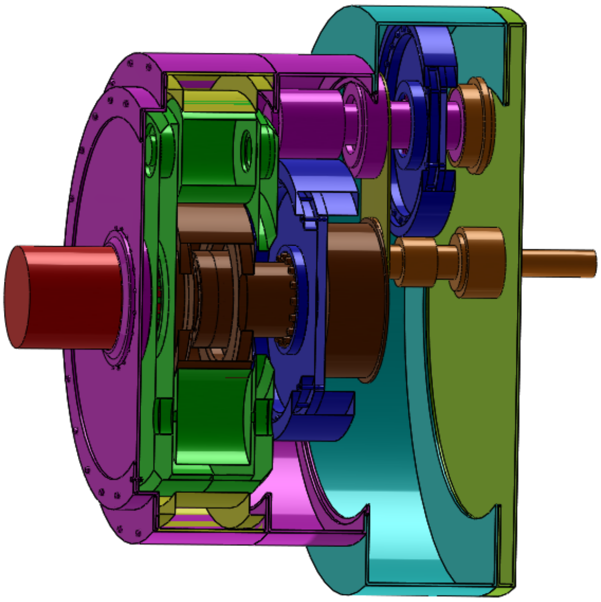


Figure 1.4: A DMU geometry of a wind turbine speed reducer

1.2 Elements of a Digital Mock-Up

We explore the content of the DMU in order to understand what information we can rely on to satisfy some user requirements during a retrieval process.

A broader definition of DMU was provided by Dollner et al. [34] declaring that a DMU represents a clearly defined set of data in the product model, whereas the term "product model" includes all of the information gathered during the product development process. In general, a DMU consists of three types of data: geometrical data, product structure and meta data [103].

1.2.1 Geometrical data

Geometrical data refer to the shape of components and are generated by CAD systems, which originate 3D representations in terms of assembly models. In general, the term *component* refers to both parts and sub-assemblies of an assembly model. The main representation scheme adopted in CAD systems to represent solid objects is the boundary representation (B-rep). This is a de-facto standard in commercial CAD systems since it provides an evaluated representation of the elements describing the shape of objects.
Generally, the elements used to generate a boundary representation are object, shell, face, loop, edge, vertex and the geometric information attached to them. Examples of geometric information are face surface and edge curve equations and vertex point coordinates. In addition, a boundary representation must represent how the entities are related to each other, i.e. the topology.

Properties of a boundary representation

The boundary representation is one of the most used representation scheme in Computer-Aided Design since it satisfies some important properties [102], as described in the following.

Domain Using a boundary representation, many complex objects can be described. The modeling space of boundary models depends on the selection of surfaces that can be used, then all the limited and regular objects can be represented and this representation has a high level of expressive power.

Validity If a B-rep corresponds at least to one object in the domain, then it is a valid representation. The validity of a boundary model is quite difficult to establish. Validity criteria split into topological constraints and geometric constraints. While it is possible to manage topological validity, it is hard to enforce geometric correctness without penalizing the speed in interactive design.

Completeness or non ambiguity A boundary representation is not ambiguous if and only if the representation of its faces is unambiguous.

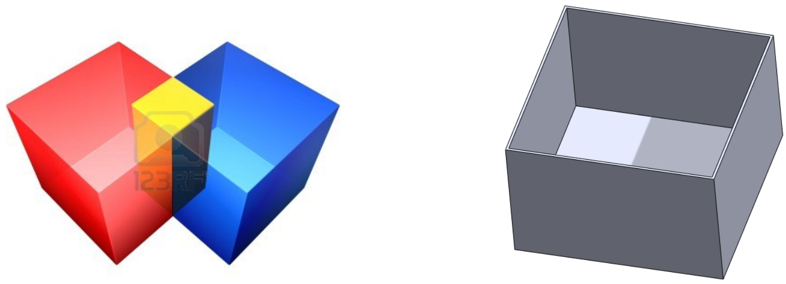
Uniqueness In general, a solid can have different boundary representations. However, it is possible to construct a unique representation considering only maximal faces, i.e. all adjacent faces having different underling surfaces.

Closure of operations The set of boundary models is usually not closed under set operations, e.g. union, intersection. The natural closed operations for boundary models are the *Euler operators* [80], which are frequently used in implementation of the set operations.

The boundary representation benefits of a lot of properties while the validity is one of the few lack. A boundary model is *valid* if it defines the boundary of an object. In general, common objects represented with boundary models are closed and 2-manifold, i.e. each point of the surface has a 2D neighborhood homeomorphic to an open disc. Thus, the validity criteria of a boundary model includes the following conditions: [80]

- The set of faces of the boundary model closes, i.e. forms the complete skin of the solid with no missing parts.
- The faces of the model do not intersect each other (inducing self intersections) except at common vertices or edges.
- The boundaries of faces are simple curves that do not intersect themselves.
- The faces must be oriented in the same way.

The first and the second conditions guarantee that self-intersections are not present in the object, therefore cases as in Figure 1.5(a) are not allowed. The first condition forbids objects such as the open box in Figure 1.5(b). Sometimes, these rules can be broken allowing non-manifold configurations.



(a) Self intersection

(b) Bow without the top face

Figure 1.5: Configurations not allowed

Anyway, non-manifold configurations are not common in product models as in real objects. Then, in the context of this thesis, we assume the following hypothesis.

Hypothesis 1

The representation of each part is valid, i.e. its B-rep defines only manifold configurations.

Part descriptions available in DMU can use analytic or parametric representations [108]. In this work, we will present some analysis to deduce implicit information from the geometric representation of an assembly model. This process, which will be illustrated in chapter 4, is based on analytic representation and, so far, on spherical parts modeled as free-form surfaces. Therefore, we add the following hypothesis.

Hypothesis 2

The faces involved in contacts are modeled using either analytic surfaces or free-form surfaces if and only if they represent a complete spherical part.

In the end, the level of detail of the representation depends on the lifecycle stage as the objectives and requirements of the ongoing activity. For instance, standard components (e.g. screws, nuts, bearings, gears or circlips) can be not modeled by the designers using the CAD software functionalities but imported from supplier catalogs and/or 3D databases. Thus, for a given component, depending on the supplier, multiple geometric representations and with different levels of detail may exist, for instance, a bearing may be represented as an assembly model or as a part, and in both cases, its shape can be detailed or simply sketched out. However, the details included in the simplified representation are sufficient for further analysis, as simulation [103].

In addition, other useful information present in CAD systems is left out in DMU representation, as tolerances, tapers, threads and product structure. Then, in our work we consider the following hypothesis.

Hypothesis 3
The DMU may include simplified representations of standard parts imported from database.

1.2.2 Product structure

Designing an assembly model is a complex process, which aims to create objects conforming predefined requirements by a combination of functions. An assembly model is made up of parts that are related together by some relationships, as shown in Figure 1.6.

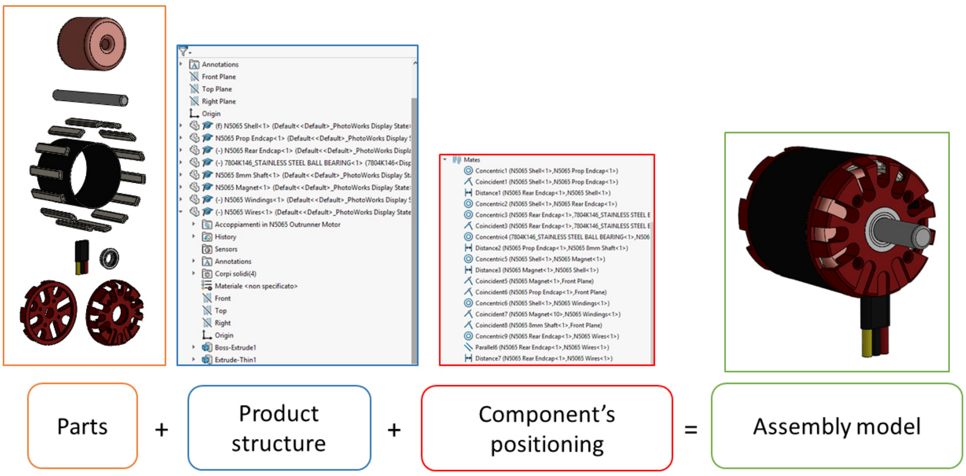


Figure 1.6: Assembly model of an electrical engine

The components of an assembly model can be gathered using a hierarchical and logical structure of dependence among developed parts. Such a structure is not unique and is used to organize product data in a manner appropriate to the domain field of experts.

The more common adopted structures are as-designed or as-planned [120]:

as-designed aggregates elements by their function. In this case, sub-assemblies represent a unit that satisfies a specific function.

as-planned reflects how parts have to be manufactured or assembled from a manufacturing or a process planning perspective.

Figure 1.7 shows a snapshot of a commercial CAD software (SolidWorks®), which depicts an assembly model with its product structure. In this example, the components are organized according to their functional properties.

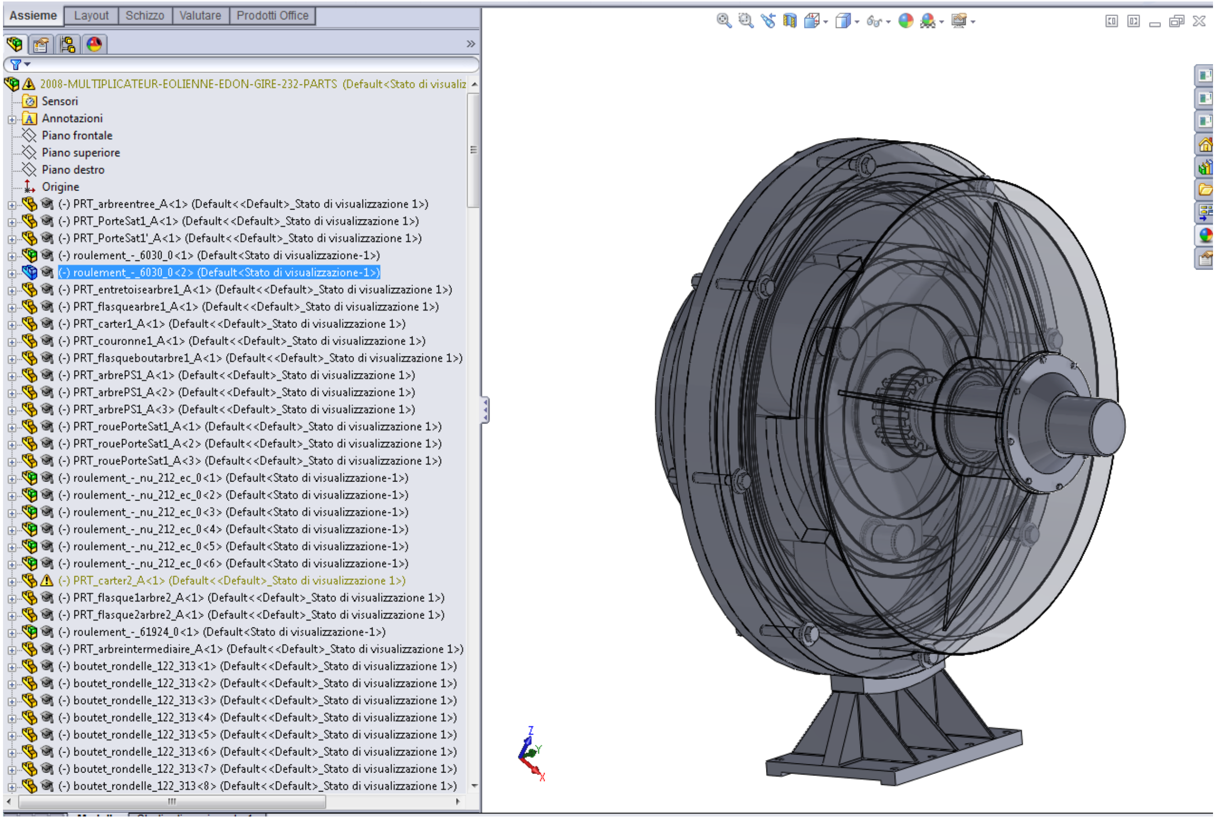


Figure 1.7: Assembly model of a speed reducer

Depending on the company approach, other structures are allowed, such as maintenance structure or quality structure. Sometimes, parts can be organized with respect to their relative positioning for visualization issues. They can also be grouped with respect to their material for simulations.

The product structure information is not necessary for visualization purpose of the DMU, it is rather connected with design intent grouping product parts according to some criteria. Because of the meaning of the model structure, during DMU creation, it is stored separately from the geometry even if modern CAD systems allow to include it in the CAD models.

Considering our purpose of retrieving assembly models in order to improve the PDP, the product structure can represent an important similarity factor to distinguish similar models.

1.2.3 Components' positioning

In 2004 Whitney [128] explains that designing an assembly model requires giving to parts their location in the 3D space. In DMU components are positioned relatively to each others through contacts. These contacts can correspond in the real object to the connection that two parts have to be welded or glued forming a single component. Hence, the pure geometric information present in the DMU to assess the contact between two components can be ambiguous [108]. Further, the engineers will interpret the DMU and derive the correct contact information. In this thesis, we will focus on the geometric information about the contacts, which can be represented in different ways [108]:

Kinematic links Kinematic links (or joints) create relationships between parts that determinate the positions of components and allowable movements, i.e. the allowed degrees of freedom (DOF).

The 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. The possible lower kinematic pairs are depicted in Figure 1.8.

An upper kinematic pair arises when two surfaces are constrained to remain in contact along a common line or at a common point [122]. An example of this kind of joints is represented by ball bearings, where the balls are in contact by vertices to the inner and outer rings.

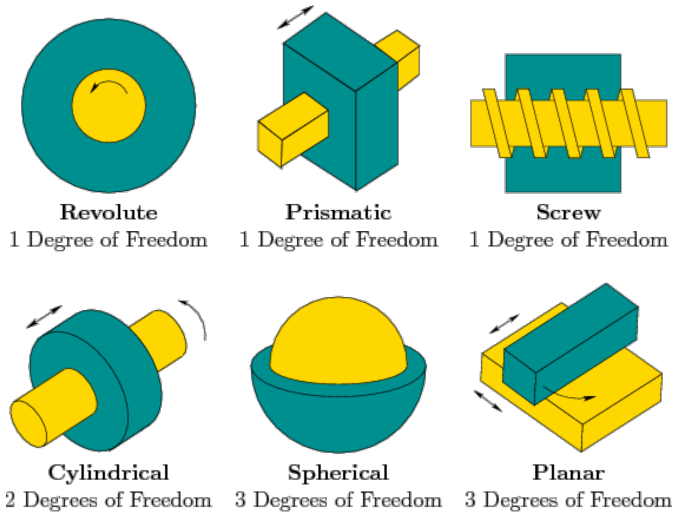


Figure 1.8: Lower kinematic pairs [6]

Geometric constraints Assembly constraints fix the position of entities (faces, edges or vertices) of parts of an assembly model. In particular, geometric constraints define the alignment or the distance of two parts by their entities. Figure 1.8 represents some of the possible geometric constraints. Typically, constraints between parts include: parallel, perpendicular, coincident, tangent and concentric.

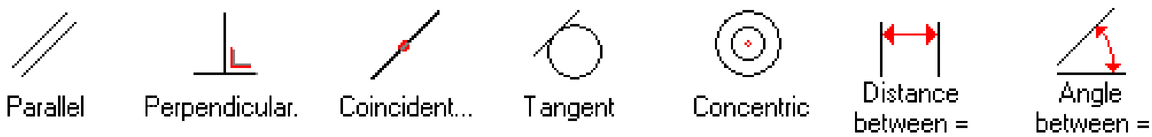


Figure 1.9: Example of some geometric constraints

Absolute position In this case, parts are placed in the 3D space simply working on the frame of reference, using an affine transformation matrix per object.

Despite CAD systems provide capabilities to easily specify and store the relative positions and relations between parts, these data may become invalid during modifications. For this reason, in the initial design stage, it is common, that in DMU parts are just gathered in hierarchies of sub-assemblies but without information about what parts are connected or how. This practice sets another hypothesis on which this thesis is based.

Hypothesis 4

Only absolute positions are available in CAD files, while the kinematic links and the constraints should be deduced in an ad-hoc process.

1.2.4 Assembling CAD model

Once parts are arranged in the 3D space, unrealistic or unrealizable configurations may be present in an assembly model [108]. An example of this situation is represented by the volumetric interferences (i.e. self intersections) between two models. These configurations belong to the *interface* that may exist between two parts of an assembly model. In particular, interfaces can be grouped into interferences, contacts and clearances as shown in Figure 1.10.

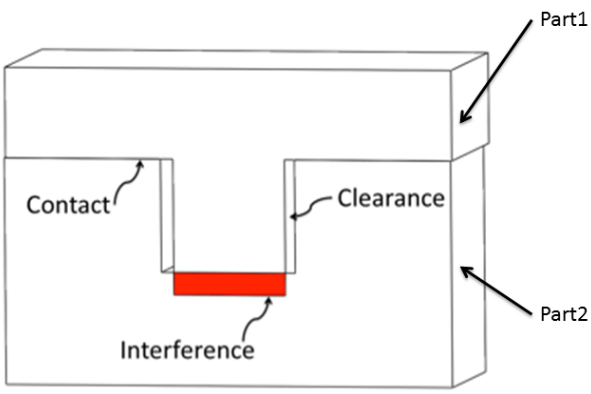


Figure 1.10: Interface relations between parts in an assembly model

They can be defined as follow [109, 110], applying topological concepts [64]; $cl(*)$ and $int(*)$ represent respectively the solid closure (i.e. the union of the solid interior and its boundaries) and the solid interior (i.e. the set of interior points of S in the Euclidean space $\mathbb{R}^3$).

Interference An interference gives rise to a shared volume between two components. More precisely, two solids C_1 and C_2 are said to have an *interference* if and only if

$$Z_i(C_1, C_2) = cl(int(C_1) \cap int(C_2)) \neq \emptyset \tag{1.1}$$

where $Z_i(C_1, C_2)$ represents the *interference zone* between the solid models C_1 and C_2.

Contact A contact between two components defines one or more shared surfaces or curves, without any shared volume.
More precisely, two solids C_1 and C_2 are said to be in *contact* if and only if

$$Z_c(C_1, C_2) = (cl(C_1) \cap cl(C_2)) - Z_i(C_1, C_2) \neq \emptyset \tag{1.2}$$

where $Z_c(C_1, C_2)$ represents the *contact zone* between the solid models C_1 and C_2.

Clearance A clearance occurs when a distance between two or more surfaces of two components conveys a functional meaning.
More precisely, two solids C_1 and C_2 are said to be at *clearance* with respect to a distance ρ if and only if

$$Z_j(C_1, C_2) = (C_1 \cap (C_2 \bigoplus S)) - (Z_i(C_1, C_2) \cup Z_c(C_1, C_2)) \neq \emptyset \tag{1.3}$$

where $C_2 \bigoplus S$ represents the dilation of the solid C_2 with respect to a structural element S [44], S is a closed sphere of radius ρ and $Z_j(C_1, C_2)$ represents the *clearance zone* between the solid models C_1 and C_2.

Some of these configurations should be solved during the product design, because they are not possible in a real environment and they are generated by some mistakes. For instance, an interference is a non-realistic representation in the sense that it implies overlapping volumes of two components in a product, which is not acceptable since this leads to non-physical configurations.
Nevertheless, not all interfaces have to be resolved, because some of them may be created on purpose [103]. For instance, the intersections among thread screws and nuts, or flexible parts, as springs, seals and insulating parts, or designing parts that will be assembled by shrink-fitting can be done on purpose. Thus, some of these configurations can be interpreted as an imprecise design while other as a deliberate artifact to reflect some conventional meaning [108]. Considering this dual nature, for this thesis, we made the following assumption.

Hypothesis 5

The representation of the entire assembly model can have unrealistic arrangements due to the wrong positioning of its components or made on purpose by designers.

1.2.5 Attributes

Beside component geometry and product structure, there exist other information that can be used to build a DMU. Annotations are used to express explicitly some geometric properties as major/minor diameter, pitch and number of threads [108]. As explained before, a DMU can be simplified and details may be not described by geometric representation, then, additional attributes can be used to characterize parts.
For instance, component material and physical properties are represented as annotations. They are necessary to enable the manufacturing of a product [115] or for some simulations. In the end, other attributes are used to identify name, number and version of a product, to distinguish its status in the PDP and to provide details about description, material and product manufacturing information.

The mentioned information may be present in the DMU as attributes but it is not mandatory. The absence of conventions among designers and the change of industrial context over the time has made challenging to exploit them. Annotations were studied also by Iyer et al. [57], who assert that this type of information is not robust and of little use for CAD retrieval. Thus, we set the following hypothesis for this thesis.

Hypothesis 6

The proposed approach will not rely on attributes represented as additional textual information in the DMU.

1.3 Assembly retrieval: application scenarios

Maximizing the reuse of existing models and the associated information is an effective way to improve the PDP avoiding the proliferation of similar models, reducing the time for reinventing existing solutions and focusing on real innovative aspects. This goal can be achieved by using retrieval systems, which allow to find models in a database that best match a given query. The match usually is represented by a score, which identifies the similarity between the query and the target model, i.e the compared model.

The ability of retrieving assemblies in not-structured databases satisfies many scenarios with different purposes. In this section, we first illustrate different types of similarity among assembly models (section 1.3.1) and then we present five macro purposes where retrieving assemblies according to those similarities brings benefits: digital model reuse (section 1.3.2), product information reuse (section 1.3.3), component standardization & rationalization (section 1.3.4), design update & maintenance (section 1.3.5) and reverse engineering (section 1.3.6).

For each macro purpose, we present some possible scenarios of assembly retrieval focusing on the associated suitable query specifications and similarity types. The proposed set of scenarios does not aim to be exhaustive but just to provide an overview of some interesting applications. It is worth to note that some scenarios can be applied to more than one macro purpose.

1.3.1 Types of similarities

The formalization of the concept of similarity is one the most important abstract concept in human perception [20]. Its complexity is ascribable to multiple elements, such as who evaluates the similarity [7], the purpose for which the similarity is assessed [25, 49, 24, 100, 12] and also the types of objects that are considered [22]. Moreover, in many situations it happens that two objects are not entirely similar but they present just partial similarity [20].

Actually it is necessary to have the possibility to detect if an entire assembly is completely similar to another entire one, if it is contained into another (not necessary as sub-assembly) or if two assemblies contain similar sub-assemblies. We refer to those similarities as global, partial and local. Partial and local similarities were identified by Liu et al. [135] as *part-in-whole* and *whole-to-whole*. In their definition, part-in-whole similarity determines whether an input shape is inside a whole shape, e.g. a query wheel is in a target car. A whole-to-whole similarity aims to measure partial similarity between two global shapes, e.g. a woman and a mermaid share similar parts, then they can be considered similar.

Specializing on assembly models, these types of similarities can be described using the set theory. Being $\cong$ the symbol indicating the similarity according to given criteria and $A = (P_A, R_A)$ and $B = (P_B, R_B)$ two assemblies, where P_* and R_* represent the set of parts and relationships respectively, then A and B are considered similar with the highest score, i.e. similarity measure equal to 1, according to the following definitions:

Global similarity

Definition 1.3.1. *A is 100% globally similar to B iff $\forall \quad a_i \in P_A$, $\exists \quad b_h \in P_B$ and $\forall \quad b_h \in P_B$, $\exists \quad a_i \in$ such that $a_i \cong b_h$ and for each relation $(a_i, a_j) \in R_A$, $\exists \quad (b_h, b_k) \in R_B$ such that $(a_i, a_j) \cong (b_h, b_k)$ where $a_i \cong b_h$ and $a_j \cong b_k$*

Partial similarity

Definition 1.3.2. *A is 100% partially similar to B iff it exists $B' \subseteq B$, such that $\forall \quad a_i \in P_A$, $\exists \quad b_h \in P_{B'}$ and $\forall \quad b_h \in P_{B'}$, $\exists \quad a_i \in A$ such that $a_i \cong b_h$ and for each relation $(a_i, a_j) \in R_A$, $\exists \quad (b_h, b_k) \in R_{B'}$ such that $(a_i, a_j) \cong (b_h, b_k)$ where $a_i \cong b_h$ and $a_j \cong b_k$*

Local similarity

Definition 1.3.3. *A is 100% locally similar to B iff it exists $A' \subset A$ and $B' \subset B$ such that $\forall \quad a_i \in P_{A'}$, $\exists \quad b_h \in P_{B'}$ and $\forall \quad b_h \in P_{B'}$, $\exists \quad a_i \in P_{A'}$ such that $a_i \cong b_h$, for each relation $(a_i, a_j) \in R_{A'}$, $\exists \quad (b_h, b_k) \in R_{B'}$ such that $(a_i, a_j) \cong (b_h, b_k)$ where $a_i \cong b_h$ and $a_j \cong b_k$*

Figure 1.11 depicts an example of the described types of similarities. Let the model M_1 of Figure 1.11 be the query model. It is *globally similar* to the model M_2, since M_2 has analogous components with limited differences. While if the query is completely included in the target model (as the model M_3 and M_4) we speak about *partial similarity*. In addition to these two types of similarities, also models M_3 and M_4 can be considered similar. Since they share similar components, we refer to them as *locally similar*.

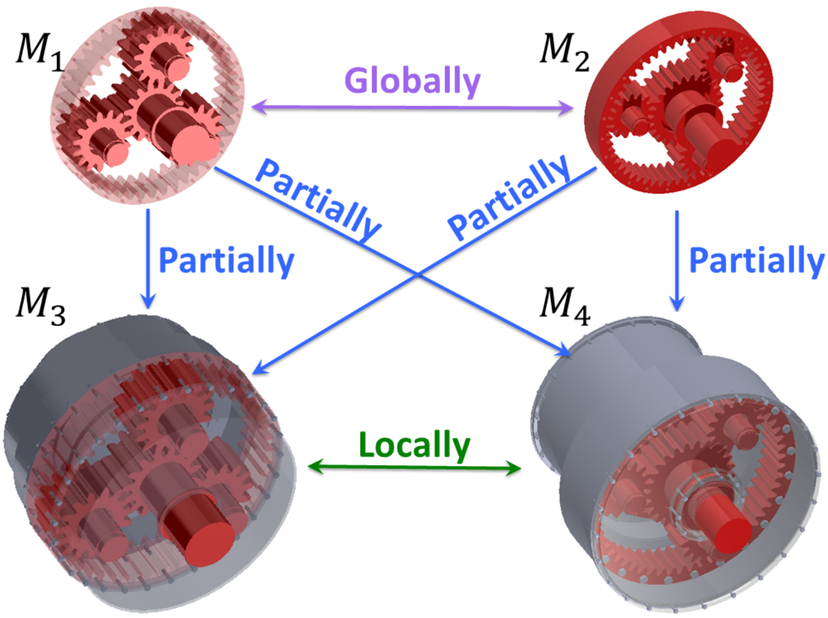


Figure 1.11: Example of different types of similarities

Note that, global similarity implies partial and local similarities. While partial similarity implies local similarity, the vice versa does not hold. In addition, the global and local similarities are symmetric, while it is not true for the partial similarity, because the query model has to be included in the target model.
Considering the definitions 1.3.1, 1.3.2 and 1.3.3, we can observe that global and partial similarities can be expressed in terms of the local one. Indeed, the global definition 1.3.1 corresponds to the local one 1.3.3, when $A' = A$ and $B' = B$; while the partial definition 1.3.2 corresponds to the local one 1.3.3, when $B' = B$.

Thus, these type of similarities can be defined in terms of the local similarity and of a *coverage factor*. With the term *coverage factor*, we refer to the percentage of elements of the query and target models that are considered similar with respect to all the elements in the two models.
More precisely, we define two coverage factors as follow:

Coverage factors

Definition 1.3.4.

$$\text{Partial coverage factor} = \frac{|S|}{|Q|}. \tag{1.4}$$

$$\text{Global coverage factor} = \frac{2|S|}{|Q|+|T|}. \tag{1.5}$$

where,

- $|S|$ represents the number of similar elements
- $|Q|$ represents the number of elements in the query model
- $|T|$ represents the number of elements in the target model

Using these factors, then we can re-define global and partial similarity as follow.

Global similarity using coverage

Definition 1.3.5. *A is 100% globally similar to B iff A is 100% locally similar to B and* $\frac{2|S|}{|Q|+|T|} = 1$

Partial similarity using coverage

Definition 1.3.6. *A is 100% partially similar to B iff A is 100% locally similar to B and* $\frac{|S|}{|Q|} = 1$

1.3.2 Digital model reuse

Digital model reuse

Design model reuse is the inclusion of previously designed components in new products as they are or making minor modifications on them.

Retrieval technologies have shown their value in design reuse [28, 45, 88]. Indeed, having an efficient tool for retrieving and then reusing existing models avoids the creation of duplicates, saving costs and time. To be effective retrieval systems should allow designers searching a model to modify just by providing a rough query. The rough query can be a simple description of model functions, a sketch or a simplified CAD model. This last situation is normal at the early stage of new product design, when user can just sketch some parts and specify their relationships and then search for similar existing solutions to reuse. In addition, the model used for the query can be also detailed. This is the case when designing product variations, where the new product is obtained with simple modifications of a previous model.

Thus, several scenarios can be imagined:

- The first scenario considers the retrieval of CAD models starting from their 2D sketches. At the beginning of the design, the creation of a simplified CAD model just to query the database may be a time consuming activity. In this case, it is easier for a designer to represent the noteworthy characteristics of the required model in a 2D hand-made sketch. Indeed, for an engineer, this representation is able to express intuitively shape characteristics, without any training on the adopted retrieval system.
  Some works adopt this approach for specifying the query but consider only single part models. For instance Pu et al. [93] proposed a 2D sketch-based user interface for 3D CAD model retrieval, where users can express their intent by sketching the 2D shapes corresponding to the three views of 3D models (i.e. front, top, side). More recently, Liu and al. [74] relaxed the query specification, requiring as input a simple free-form sketch that naturally expresses the main characteristics of the component instead of drawing the three orthogonal views.
  Similar studies connected to assembly models are still in an embryonic stage [74, 35]. In this situation, it is reasonable that retrieved models are either globally or locally similar to the proposed query, depending on the complexity of the sketched model, since the sketch generally represents in a stylized way the salient characteristics of the overall shape without providing details of the interior (e.g. the parts constituting an assembly).

- When the model is complex, it is not simple to provide a sketch just to perform the search and it is worth using a simplified CAD model. As a second scenario, let us suppose that a senior designer leaves the company where he/she have worked and a new designer needs to design a specific product, e.g. a torsen differential unit, modifying an existing one. He/she doesn't know the file name of the components, thus he/she has to examine all the existing solutions. In this case, it may be useful to start designing the new component approximately and operating a search over the entire database, in order to identify a restricted collection of assemblies to be inspected, which have parts with shape similar and comparable relationships. Since the component used as query is just a draft, it is reasonable that each part is considered similar despite the presence of neglectful details such as some holes and fillets missing in the query model. The expected relationships among parts should be specified in form of constraints or joints, in such a way that the motion between two parts is encoded. Since parts inside a differential unit can be gathered as subassemblies in different ways, structural similarity is not important in this scenario. Moreover, the models to be retrieved can be not only globally similar to the query but also include it. Thus, in this case for instance, power transmission chains which incorporate torsen differentials should be retrieved likewise.

- In the scenario proposed by Chen et al. [28], let us image a user has a CAD assembly model of a retired colleague and he/she wants to figure out how to modify it. When design intent is easy to understand it is also simple forecasting how a modification on a component will affect the entire assembly model. Otherwise, if design target is not obvious, then project costs and development time may have a negative impact. Chen et al. explain how the hierarchical assembly structure, interface relationships and constraints can be helpful for understanding implicit design intent [28] and what is not so evident in a model can be clear in a similar one. Then, to retrieve design intent of a model, the user can search assembly models having similar structure, interface relationships and constraints. In this assumption, it is possible that he/she is interested in retrieving both globally and partially similar models.

- For the fourth scenario, we imagine a designer who is interested in speed reducer units to be placed inside wind turbine. Then he/she defines a query in a virtual way by a graph where he/she can choose predefined standard parts with specific functionalities and connect them specifying their joints and/or their mutual arrangement.
In this case, the query may be not completely included in the target model, and it is reasonable that only a set of components of the query is matched with another set of the target model. Suppose a designer aims to retrieve a component as depicted in Figure 1.12 with a shaft (red part), two bearings aligned (blue components) separated by a spacer (purple part), and c-clip (green part). In this case, a retrieval tool should not prevent from retrieving models without the c-clip or other fastener parts, thus the level of similarity required is partial or even local.

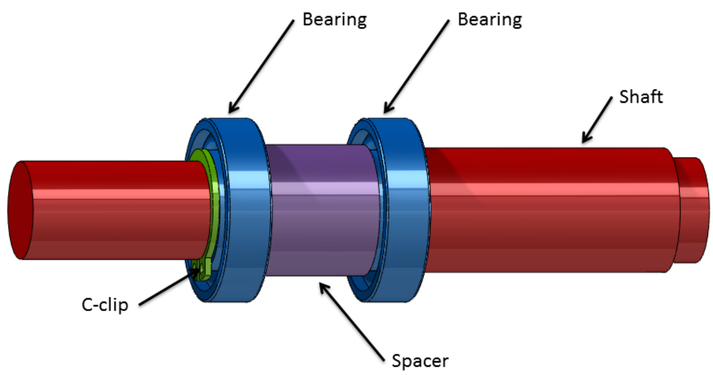


Figure 1.12: Example of query model

1.3.3 Product information reuse

Product information reuse

It identifies the process of mining a database for retrieving design information and documentation associated to a given product.

Generally, this process allows to obtain knowledge for the design of a new product by getting technical information and costs related to manufacturing associated with previously developed similar products [101].

- Deshmukh et al. [32] give an example of using assembly retrieval for accessing cost and reliability reports of specific or similar products. They describe the retrieval criteria useful for a user seeking a rocket motor that contains a specific component (i.e. a Beryllium liner) of a specific size. Among the set of motor models, the designer would then search for an assembly by specifying the shape, size and material of parts present in the assembly. To perform this search, the primary condition is to know part characteristic such as material information. For this scenario, Deshmukh et al. give more importance to part type, size and material attributes, while how the parts are connected together is less pertinent, and then, not considered as similarity criterion.

- Supposing a designer wants to retrieve assembly work instruction of some products, e.g. a table fan. An important criteria of similarity is represented by the mating conditions, as they often are directly related to the assembly process [32]. So the user, who is trying to retrieve similar assembly models according to this criterion, is expected to be able to express how parts are connected to each other. For this scenario, Deshmukh et al. [32] propose that user specifies mating conditions between parts building a mating graph, i.e. a graph where nodes are associated with single parts and arcs represent mating conditions. Each node will have several attributes that allow to characterize the parts, such as the category (if the part is standard or custom), the geometry and the type (i.e. the sub-category for standard parts).
  In this scenario, the user usually is not interested in creating large query graph, but small graph with highly distinguishing characteristics, thus retrieved models can also be partially similar to the query model including it.

1.3.4 Product standardization & rationalization

Product standardization & rationalization

Standardization is the process of defining common characteristics among a set of components such that they are compatible each other. This process allows the rationalization of products.
Rationalization is the process of reducing the number of products to develop (eliminating or outsourcing products and product variations) in order to invest more in the products that generate the most profit.

On the one hand, standardizing and rationalizing components allow to reduce the number of necessary parts to be designed to create new objects. On the other hand, before product manufacture, engineers can search for similar equivalent components already present in the warehouse, reducing also manufacturing operations. With this practice, it is possible getting a considerable time saving, especially in case of complex devices with many parts which may also require a complex design and/or production process, as well as reducing the proliferation of very similar components.

In the goal of standardization, Zhang et al. [134] illustrate how finding common components of different 3D CAD models is useful for exploiting the commonality of products that could effectively inspire designers to adopt frequently used solutions for new products and speed up the design process while reducing component variation. In this scenario, the user provides a set of CAD assembly models and the system proposes a collection of elements that are in common among them. In this context, similarity criteria include both the shape of the parts and their joint relationships, in order to consider the relative position among parts. In this scenario query and target models share similar components and no one is included in the other, thus they are locally similar.

1.3.5 Design update & maintenance

Design update & maintenance

Design update is the practice of revising models as a result of changing needs or conditions no longer valid.
Maintenance refers to those activities necessary to preserve the status of an object preventing its damage due to deterioration of components.

- For the design update scenario, we suppose that engineers have discovered that a developed transmission mechanism has to support more load than expected. One solution is the replacement of the adopted bearing with a more appropriate one; the user knows the model to be replaced and he/she seeks where this CAD model is included in other assembly models to estimate procedures and costs for replacing the component, thus in this case the required similarity is partial.

- In case of maintenance, knowing the rate of the wear and tear of components in assemblies, engineers what to know how many of those components are present in a set of products in order to manage the stocks in the warehouse. Also in this case, the user knows the model to be replaced and he/she seeks where it is included in other assembly models, thus the required similarity is partial.

1.3.6 Reverse engineering

Reverse engineering

In mechanical field, reverse-engineering is the process that create a 3D digital model starting from a physical object.

The reconstruction of a digital model starts capturing data from real objects. This acquisition may be done through different devices, as a camera, a laser scanner or a 3D computed tomography (CT), which provide different kind of data to be analyzed and interpreted.

If we have a real object and we know that its digital representation is present somewhere in a database, then, in an indirect way, we can say that retrieving existing similar CAD assembly models ease the operation of reverse engineering, where the input of the query is dependent by the device used in the acquisition phase. For instance, we may have a set of pictures using a camera, a point cloud by the laser scanner and a set of 3D images of the internal structure of the object using the CT. It is important to observe that the first two techniques provide just geometric information about the visible shape of the considered object, while with CT, it is possible reasoning on the interfaces of the parts that compose an assembly model, getting information also about mating conditions.
It follows that the similarity condition will be restricted to the information provided during the measurement phase. Thus, overall shape will be considered in case of a camera or laser scanner, while from CT data we can include shape similarity on the single parts and mating information.

The digital representation of an object acquired by a scanner process does not present functional information, that is necessary for redesign purposes. To overcome this limit, Lin et al. [73] aim to recover functional mechanical assemblies starting with a raw point cloud, noisy and affected by missing regions due to undercut portion hard to be scanned.

1.3.7 Synthesis on the scenarios

The proposed scenarios are summarized in Table 1.1, where for each of them, their use cases are listed with an indication of suitable description of the query, the criteria according to which models must be considered similar and the type of similarity that models are expected to have.

From these examples, we have deduced some of the criteria and types of similarity useful in the comparison of two assemblies. In particular, the valuable characteristics are:

- the shape of the entire assembly model,
- the shape of the constituent parts,
- the hierarchical structure of an assembly model,
- the mating conditions,
- the allowed motion,
- the functionality of its components.

Moreover, these scenarios underline the utility of querying databases using different representations of the assembly model that the user aims to retrieve. Then, to provide a flexible tool for the retrieval of assemblies, which can be tailored to the user needs, retrieving methods should allow partial and multi-modal queries including the specification of a CAD model as input.

In this thesis, our goal is to define a retrieval system, which considers multiple criteria related to the part shapes, the interlinks between parts and functionality aspects that are implicitly deduced from the 3D data. We aim to support the three types of similarities (global, partial and local) defined in section 1.3.1 and the input for the proposed system can be a CAD model or a subset of its attributes. In this way, we aim to satisfy scenarios as the (ii), (iii), (iv), (vi), (vii), (viii) and (ix) reported in Table 1.1.

Hence, the challenge is to find an assembly representation able to support user requests at different levels of detail. In addition, associated data should be automatically extracted to avoid tedious manual instantiations. The difficulties associated to this objective are discussed in the next section.

Table 1.1: Summary of the scenarios and associated criteria of the query

Purpose		Scenario	Query				
			Model description	Similarity conditions	Target model similarity		
					Global	Partial	Local
Digital model reuse	i)	Starting from the 2D design sketch, designer searches for existing solutions	2D sketches of part/assembly	Geometric shape [93]	✓		✓
	ii)	A new designer wants to design a model modifying an existing one. He doesn't know the file name of the model	Simplified 3D CAD assembly model	Similar shape of constituent elements and same mating conditions	✓	✓	
	iii)	Search for assembly components with similar design intent	3D CAD assembly model	Hierarchical assembly structure, constraints, mating relationships [28]	✓	✓	
	iv)	Searching configurations of speed reducer containing specific types of components	Virtual representation of an assembly model	Part functionality and arrangement		✓	✓
Product information reuse	v)	Retrieve a rocket motor assembly that contains a Beryllium liner of a specific size to have access to associated data such as cost and reliability reports	3D CAD assembly model and text	Shape, material and size of constituent elements [32]	✓	✓	
	vi)	Idealize a table fan by its mating conditions and looking for solutions similar or that include the query	Mating graph	Mating conditions [32]	✓	✓	
Product standardization & rationalization	vii)	Identify local structures shared by multiple models	Set of 3D CAD assembly models	Shape of parts and joint relationships [134]			✓
Design update & maintenance	viii)	A transmission component has to support more load than expected, thus adopted bearings have to be substituted with more suitable	3D CAD assembly model	Functionality of constituent elements and same joint relationships		✓	
	ix)	Organize the planning of the maintenance	3D CAD assembly model	Functionality and shape of constituent elements with the same joint relationships		✓	
Reverse engineering	x)	Recover the CAD model of an acquired assembly using a laser scanner	Point cloud	Overall shape [73]	✓	✓	✓

1.4 Assembly retrieval: issues

We have seen in section 1.3 examples of scenarios where the objective is the retrieval of CAD models presenting characteristics similar to those specified in the query. In this section, we discuss the main issues in assembly retrieval systems.

- **Size issues**
  The process of digitalization started in the late 70s, when for the fist time computer-aided drawing programs have been used [48] and the concept of Virtual Product Development was introduced. These tools have eased 2D drawing by using functions which sketch lines and circles. Lately in the 80s, the design process was completely revolutionized with the launch of 3D CAD programs. The success of this kind of tools came both from making it possible the visualization of realistic models, the integration of product-related knowledge and the customization of production activities. In the 90s, standardized neutral data exchange formats consent also data exchanging between design and simulation, allowing widespread diffusion of Virtual Product Development. The development of the Boeing 777 was one of the first documented examples entirely designed and pre-assembled in a digital 3D environment [116].

  Since then, the popularity of digital systems in industries gave rise to a huge number of digital 3D CAD models and the size of a DMU can include more than 1 millions parts representing several tera-bytes of data [52]. The size and the number of DMUs present in large databases make challenging providing an efficient retrieval system that properly satisfy user needs. Then, one of the purpose of this thesis is to extract some characteristics of an assembly model, which allow to reduce comparison.

- **Naming issues**
  Text-based retrieval systems strictly rely on the manual integration of annotations. Using this kind of search, some models may be not retrieved, because they have not the same text in their annotations even if they are semantically related to the query. For instance, parts represented in Figure 1.13 are named as *nut* and despite their different shape and different purpose they are not distinguished by text search. On the other hand, assemblies that have some differences in their names and annotations may be considered comparable according to their characteristics. 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.

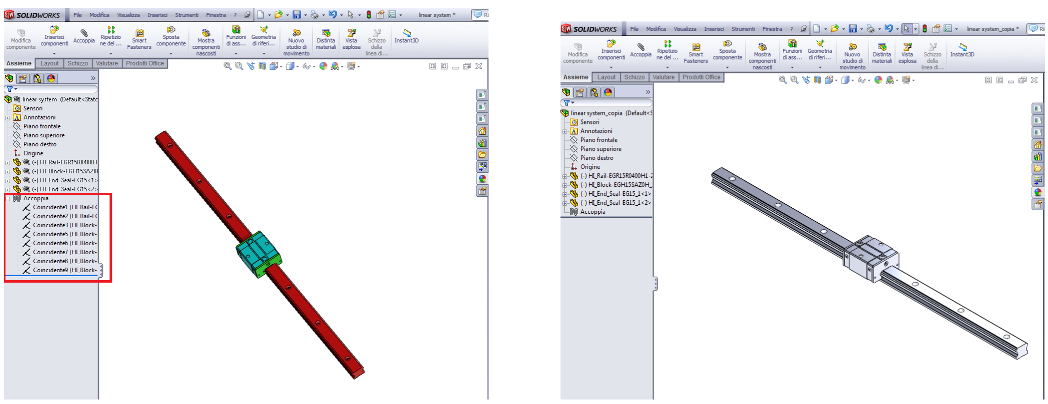
(a) $nut\_01m025aco011x\_1\_15$ (b) $nut\_3\_ajals\_103$

Figure 1.13: Part with same annotation but different shape

For this reason, in this thesis, we exploit the intrinsic description of a model (e.g. shape description) as a key to perform searches over model databases. Moreover, a name-independent search is able to increase the possibilities of the user requests.

- **Storage issues**
  Many important information (as the kinematic links, constraints or attributes) is not always explicitly stored in DMU. Moreover, some problems may arise by using standard file formats to exchange CAD models.

  An example is reported in Figure 1.14, where a model has been created with the CAD software SolidWorks®positioning the parts by mating constraints (see Figure 1.14(a)). Once the model has been stored as STEP 214 and re-opened (always in SolidWorks®), the information of the constraints is no longer available (see Figure 1.14(b)).



(a) Original model (b) STEP file

Figure 1.14: Example of information loss storing a model in STEP 214 format

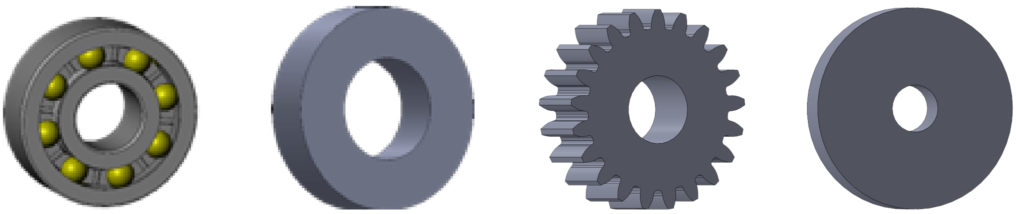
Considering the precarious nature of some information, in this thesis, we base our analysis on data surely available in the DMU, while other useful information will be extracted by reasoning on the geometry data.

- **Similarity issues**
  As illustrated in section 1.3.1, two models may present different type of similarities. The presented scenarios highlighted that users may be interested in retrieving assemblies that are similar according to different characteristics.

  For example, for visualization purpose, assembly models can be considered similar if they have the same global shape, but from the point of view of the assembly operations what is important are the mating conditions, which specify spatial relationships between parts and the dimension of the components. Differently, in assembly modeling, individual parts might be grouped, even in a single component, such as for parts acquired by third parties; in this case, two different assembly representations of the same real object organized with different hierarchical relationships may not be recognized as similar.

  For these reasons, in this thesis, we aim to define a retrieval system, which incorporate similarity criteria at different levels (e.g. geometry, structure, kinematic, annotation).

- **Representation issues**
  In section 1.2.1 we have discussed that in a DMU the representation of the shape of the parts can be idealized with simplified outlines. With this simplification, it is arduous to characterize assembly components. In addition, elements with the same shape may have a complete different functionality and vice versa. For example, the bearings depicted in Figure 1.15(a) and Figure 1.15(b), as well as the gears in Figure 1.15(c) and Figure 1.15(d) have very different shapes even if they identify the same components. Conversely, Figure 1.15(b) Figure 1.15(d) have very similar shape but a completely different function.



(a) Detailed bearing (b) Simplified bearing (c) Detailed gear (d) Simplified gear

Figure 1.15: Examples of components with different shapes and functions

  To overcome this issue, in this thesis, we aim to characterize assembly components exploiting their context of use. Indeed, if we focus on orange and blue parts of the assembly model in Figure 1.16, it is easy to understand what corresponds to a gear (orange part) and what to a bearing (blue part in the zoom area), despite the idealized shapes.

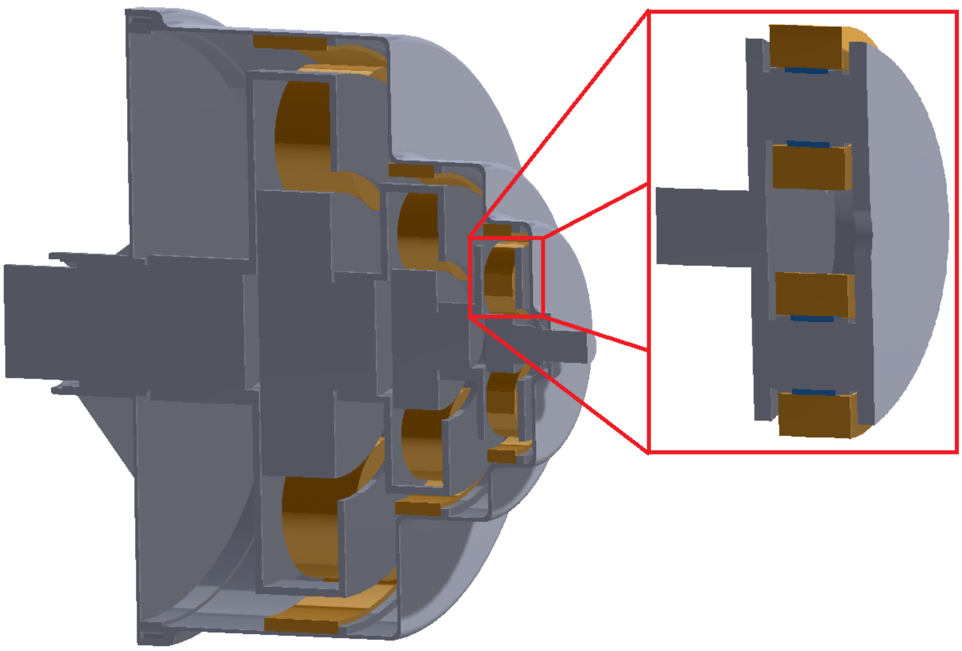


Figure 1.16: Gears and bearings embedded in their context of use

- **Query issues**
  The different types of similarity between two assembly models rise another difficulty in creating an effective retrieval system. Indeed, to cover as much application scenarios as possible, it would be desirable querying databases using different descriptions of the query model.
  This possibility is particularly important since, as seen, at the early design stage, the designer can be interested in expressing abstract or incomplete queries, e.g. simply by specifying some attributes of the assembly model, just to take inspiration from or to reuse the available models. On the contrary, in case the user is interested in finding assemblies constituted by geometrically similar parts with almost the same configuration, a complete CAD model is more suitable for expressing the query. The ability of the system to allow to define query describing assembly model by different representation and/or at different levels of details is strictly dependent on the information included in the assembly descriptor of the retrieval system.

  Moreover, since the similarity criteria are many and of different nature, it is important to study how they should be proposed in an intuitive way to the user.

- **Visualization issues**
  An additional issue in assembly retrieval systems is represented by the visualization of the results, more precisely concerning how to propose the results and highlight the different similarities to the user.

  Providing the user with a list of names of the retrieved assembly models may be a low informative descriptions, since the name can be associated to standard codes of a company that not always are of ease comprehension.
  In addition, working with multiple similarity criteria, by the visualization of the results, the user should understand quickly according to what similarity criteria the models have been retrieved, as well as what type of similarity (global, partial or local) has been evaluated.
  In the end, a further complication arises from the fact that similar components can be inside assembly models, then highlight the retrieved components may be difficult in complex assembly models.

1.5 Conclusions

In this chapter, we have illustrated the benefits that can be achieved during the PDP by the use of an assembly retrieval system, that is able to capture the similarities between two models according to different criteria.
The similarity criteria depend on the purpose of the search, then it is possible that two models are considered similar according to a defined set of criteria and dissimilar according to another one.

The identified criteria are mainly based on the shape of the entire assembly model and of its constituent parts, its hierarchical structure, its joints and mating conditions or the functionality of its components. Moreover, these criteria can be satisfied for the entire query model either for a portion of it or for a portion of a target model. These possibilities define three types of similarities (global, partial and local).
Beside the criteria and the type of similarity, an assembly retrieval system should manage multiple types of queries to allow a search adaptable to the designer requirements.

Considering the main issues in assembly model retrieval, in this thesis, we aim to define a retrieval system strongly based on the information present in geometric model itself and able to automatically extract the required data to avoid tedious manual instantiations. Moreover, since an assembly model may include components described in a simplified manner and, therefore difficult to identify, we will exploit the context of use to better characterize assembly components.

In the next chapter, we will investigate the literature in order to identify which types of scenario are addressed and what information can be deduced in an assembly model while reasoning on its geometric data.

2

State of the art

This chapter intends to examine the current state of the art to identify the addressed scenarios of assembly model retrieval and determinate the characteristics of an assembly model that can be deduced from their geometrical description. To allow a fair and comprehensive analysis of all the works, some useful criteria are identified and described at the beginning of the chapter.
Using the proposed criteria, we illustrate the works that treat assembly models focusing on those that are strictly related to the assembly retrieval and others that describe assembly characteristics.

Contents

2.1 Criteria for evaluating literature in assembly model retrieval

The problem of 3D shape retrieval was widely investigated in the last years and a huge number of works exist on this topic, dealing both with model represented as 3D meshes and models represented as B-Rep [13, 25, 24, 30, 36, 37, 43, 47, 49, 53, 57, 117, 132]. Although these techniques are able to retrieve single parts of assembly models, they do not take into account the relationships between parts and thus they are not really useful for assembly description and retrieval.

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 of assembly models, in this section, we identify and describe several criteria useful in the analysis of the state of the art proposed in section 2.2. The criteria we present can be grouped into the following five macro-categories:

- Context (section 2.1.1),
- Assembly characterization (section 2.1.2)
- Assembly descriptor (section 2.1.3)
- Query model (section 2.1.4).
- Type of similarity (section 2.1.5).

2.1.1 Context

To characterize the existing state of the art works, we classify them according to the context criterion, which includes the **scenario** of the work and the **objective** of the retrieval method.
The considered **purposes** are those illustrated in section 1.3, i.e. digital model reuse (section 1.3.2), product information reuse (section 1.3.3), product rationalization & standardization (section 1.3.4), design update & maintenance (section 1.3.5) and reverse engineering (section 1.3.6). The **scenario** indicates the considered application scenario. For example, if a work aims to retrieve assembly designs according to the similarity of their implicit kinematic links then, the extraction of the kinematic links represents the scenario while the model design reuse is the purpose.

2.1.2 Assembly characterization

In order to understand which criteria of similarity can be assessed by the works present in the current state of the art, it is important to know the type of information used to describe the assembly models in the retrieval system. Thus, the assembly characterization macro-criterion refers both to the type of data and knowledge the authors use to typify the assembly model and the way the assembly information is obtained. In particular, this criterion expresses the parameters used to characterize the geometry of the components of assembly models (**Part information**), the used assembly topological characteristic (**Topological information**) and if the attributes are present in the description of the models (**Attributes**), as described in the following.

Part information
This criterion consists of all the measurable quantities which the considered methods use. The study of different papers revealed the use of some categories of measures:

- *Curvature*
  The normal curvature at a point P on a surface varies around the normal direction of a surface. The maximum and the minimum values of the normal curvature are named as principal curvatures and the difference of their signs can characterize the point on a surface. Then, the works, that use normal curvature to characterize the shape of the parts of assembly models, sample points on part faces and then evaluate the average of the different type of points.
- *Dihedral angle*
  A dihedral angle is the internal angle defined by two adjacent faces on an edge. According to the normals of the faces and the direction of the edge, a dihedral angle can be concave, convex or smooth.
- *Part Statistics*
  Two assembly models can be compared according to their constitutive parts. The works that adopt this description refer to the number of parts in the assembly models and how many times each part appears in the assembly model.
- *Geometric info*
  With geometric info we indicate if works characterize the parts of an assembly model using the faces of the B-rep of the parts. Generally, for each face it is specified the surface type, i.e. planar, cylindrical, conical, spherical, toroidal or other; the surface convexity, i.e. if convex, concave or planar; the number of loops and the number of outer edges.
- *Shape distribution*
  The shape distribution is a shape descriptor used to evaluate the similarity of two parts. It is described by Osada et al. [87], where the 3D shape of each part is characterized by the distances of random sampled points on the surface of the parts. Several distances can be used to compute the shape distribution and most of the time, in the considered works, the Euclidean distance was employed.

- *Set of 2D projections*
  It is a view-based method to characterize components of assembly models according to their shape regardless of their structure. Since view-based methods are not robust to translation and rotation in the space, a set of projections is collected.
- *Angle distance*
  The angle distance is a two-dimensional distribution, where the first dimension indicates the normalized distribution of distances between sampled points on the part, while the second dimension refers to the normalized distribution of inner products between the surface normal vectors.

in Table 2.1, for each of these described characteristic, it is specified if it is used in a specific work or not (column **Used**). In addition, we examine how this information is obtained, i.e if they are available in the assembly model or if they are somehow computed (column **Extracted**).

Topological information
The topological information criterion refers to the different relationships among the components of an assembly model, in particular we analyze if the current works make use of the following data:

- *Structure*
  The structure refers to the hierarchical decomposition of an assembly model, as described in section 1.2.2.
- *Kinematic link*
  It refers to the kinematic relationship defined by the contacts between two parts as described in section 1.2.3. In the analyzed works, this information sometimes is referred as mating conditions or joints between two parts. Then, with the kinematic link parameter, we aim to characterize the works that make use of these information.
- *Geometric constraints*
  The geometric constraints refer to the assignment of a particular contact between two parts a of assembly models, as described in section 1.2.3.
- *Part arrangement*
  Since different parts can be positioned in an assembly model in several ways, with this parameter, we aim to investigate if the analyzed works are able to recognize different arrangements in the 3D space of different elements of an assembly model.

For a comprehensive study of the state of the art, we examine which of these relationships are used to characterize an assembly model and how these information is acquired. Thus, we also characterize the criteria as **Used** and **Extracted**, that indicate respectively which relations are used and 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.

Attributes
We have seen, in section 1.2.5, that in the DMU some attributes can be associated with the assembly model to specify details on the shape of the parts or other useful data. Then, for the analysis of the state of the art, we introduce the field **Attributes** to indicate if the considered method uses annotations as structured metadata (e.g. from ontology, or thesaurus) or simple text annotations.

2.1.3 Assembly descriptor

The information used to characterize an assembly model can be organized in different ways to define an assembly model descriptor, which is functional for the matching process used in retrieval applications. Through this macro-criterion, we aim to characterize the descriptor used in the different methods. In particular, we use the following criteria.

Level of components
The criterion **Level of components** indicates at what level (i.e. assembly, part, or feature) the assembly is characterized, and the column **Present** indicates which level of components is described by the descriptor of the method. 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.

Scale sensitivity
This criterion specifies if the assembly descriptor is able to differently characterize assembly with the same number of parts assembled in the same order but with different size.

Level of descriptor
This criterion may take two values: **global** or **local**. It indicates if the assembly descriptor is able to capture local characteristics useful for the assessment of partial and local similarity described in section 1.3.1. Note that even if the descriptor is able to characterize the assembly model at local level, it is not guaranteed that the retrieval method exploit this ability to access local or partial similarity.

2.1.4 Query specification

In order to investigate how the literature answers to the query issue described in section 1.4, we use the **Query specification** macro-criterion, which specifies how the query is expressed, i.e. the **Type of query model** and its **Completeness**.

Type of query model
This criterion indicates the type of input data to express the query model. According to the studied works, it can assume the following values:

- A single assembly CAD model
- A set of assembly CAD models
- A part CAD model
- A set of part CAD models
- An assembly abstract descriptor

Completeness
Some works allow to left unspecified by the user some details of the query model. Since this ability answers to one of the query issues, we include this criterion in the evaluation. The completeness of the query model refers both to CAD assembly models where not all the parameters are specified, i.e. a model partially defined, and to abstract representations of assembly models, i.e. descriptors that somehow represent CAD models.

2.1.5 Type of similarity

As anticipated, even if an assembly descriptor is able to characterize a model at local level, it is not guaranteed that the method for the retrieval will exploit this characteristic. An example of this situation occurs when the representation of an assembly model is graph-based and the matching method adopts isomorphism of graph or subgraph. Indeed, a graph-based representation is able to capture local similarity between two assembly models, that is represented by a common subgraph between the two graph representations. If the retrieval method uses graph isomorphism matching applied to the entire query and the entire target models, then the similarity is evaluated at global level and the local similarity will not be captured. To capture local similarity the usage of subgraph isomorphism is preferable.

For this reason, we introduce **Type of similarity** criterion (as described in section 1.3.1), which specifies the similarity assessed by the retrieval method.

2.2 Assembly model retrieval literature

While the literature of shape retrieval is very vast in the last decades, the interest of the research community in assembly model retrieval is quite recent and few works directly deal with this topic. In this section, we discuss the main methods for assembly retrieval by using the criteria illustrated in section 2.1. The methods are gathered by the **type of similarity** criterion, i.e. if the identified similarity is global, partial or local and by the **level of descriptor** criterion, i.e. if assembly models are described at global or local level.
More precisely, the following sections describe methods for the retrieval of globally similar models based on global assembly descriptors (section 2.2.1), the retrieval of globally similar models based on local assembly descriptors (section 2.2.2), and the retrieval of globally and partially similar models (section 2.2.3). In the end, the main remarks of the proposed methods are discussed in section 2.2.4 and a summary is provided in Table 2.1.

2.2.1 Retrieval methods of globally similar models using global assembly descriptors

In this subsection, we describe some methods which address assembly model retrieval at global level for different purposes. In particular, we focus our attention on the methods proposed by Renu and Mocko [100], Katayama and Sato [61] and Wang et al. [125]. The common characteristic in this works is the assembly model characterization. They use an assembly model descriptor that characterize the models at global level; moreover, they do not use assembly relationships (as kinematic relationships or constraints) to describe the models.

✏ Retrieval of solid models based on assembly similarity [100]

In order to reuse decisions made during the automotive development process, this research explores the use of model similarity and text analysis approaches to develop a relationship between solid models and assembly work instructions. To reach this objective, the authors have fixed the follow three targets:

(i) Evaluate solid models for similarity in terms of their assembly processes.

(ii) Investigate the natural language processing approaches required to analyze assembly work instructions.

(iii) Use part geometry information to mine databases of assembly work instructions and retrieve relevant work instructions.

In this work, the authors have faced the first target, i.e determining solid model similarity. The process to determine solid model similarity is divided into the following four steps:

- **Compute histogram-based similarity scores**
  In this step, Osada's method [87] is used to generate shape descriptor for each part in the two assembly models.
- **Generate clusters of similar solid models based on histogram score**
  The adopted shape descriptor provides similarity of overall shapes of solid models and it is used to generate clusters of similar models in order to investigate the similarity at a finer level.
- **Compute surface area and tessellation area distribution differences**
  In this stage, to recognize local differences between CAD models, as the one illustrated in Figure 2.1,the tessellations of solid models within each cluster are analyzed for surface area difference and tessellation area distribution differences.

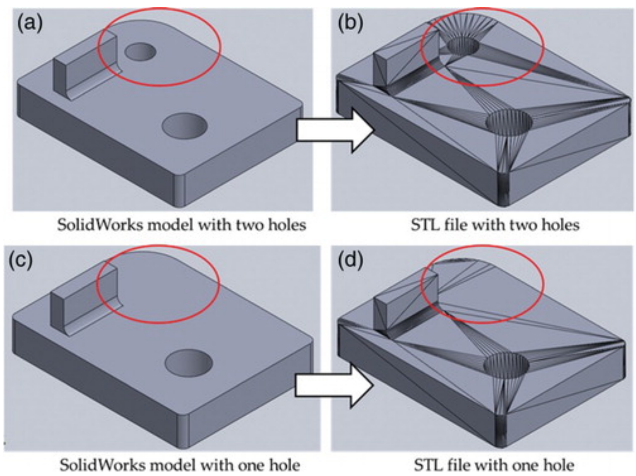


Figure 2.1: Local differences in CAD models and their tessellations (Figure 4 in [100])

- **Multi-index sort to generate ranked list of similar solid models**
  Finally, a multi-index sort is performed on difference parameters (e.g the global histogram similarity value, the value of the difference in surface area, the value of the difference in tessellation area) to rank similar models based on an assembly input model.

In this work, there is no evidence of the use of assembly relationships. The parts are described by a shape distribution, which is computed on the tessellations of the parts. Since the method involves area value in the evaluation of the similarity, the method is scale sensitive. Finally, this method, characterize assembly models just at the level of the parts and the query model has to be a complete assembly model.

✏ A matching method for 3D CAD models with different assembly structures using projections of weighted components [61]

Katayama and Sato [60, 61] evaluate the similarity of assembly models according to their hierarchical decomposition. Their idea is of defining a representation of assembly models, which convey the global shape of the assembly and the shape of the single assembly components.

Figure 2.1 illustrates the main steps of their method for the similarity evaluation of two assembly models, where different components are specified using different colors. Similarly to view-based method [117], they project different components of an assembly model into a 2D planes, where the different components are identified by their design name. The size of the 2D planes is proportional to the size of the CAD model.

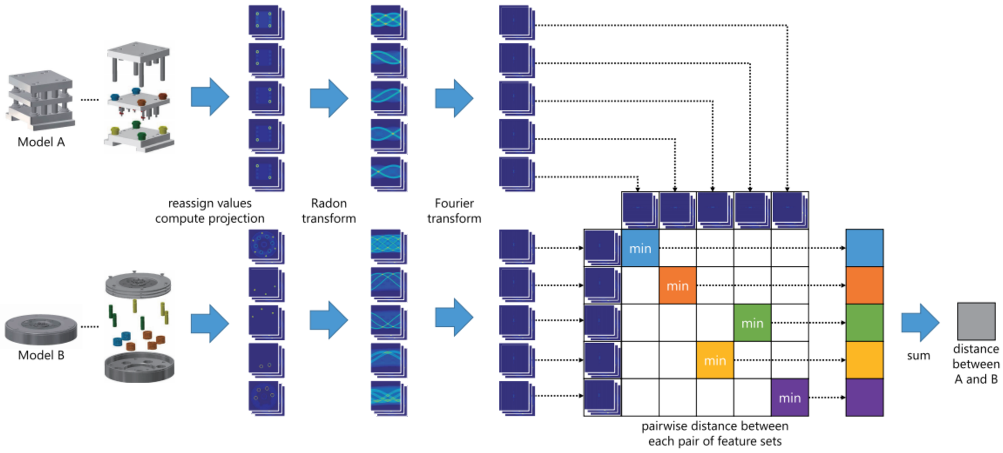


Figure 2.2: Example of the procedure to compute the distance in terms of shape and structure between two assembly models (Figure 5 in [61])

The obtained projection is rotation and translation dependent, then the 2D Radon transform and the Fourier transform are applied on the results of the projections. Then, for each pair of components, their distance is computed using the Euclidean distance. The final distance of two assembly models is represented by the sum of the distances between the corresponding components.

This method characterizes assembly models according to their shape and structure; anyhow, if two assembly models have different numbers components, then the two models are considered different.

An assembly retrieval approach based on shape distributions and Earth Mover's distance [125]

Authors suggest to decompose assembly models into a set parts and compare the shapes of all parts. In this approach, each part of the query assembly model has to be compared with each part of the target assembly model, moving from a one-to-one matching (e.g. comparing two assembly models by their overall shape) to a many-to-many matching.

An assembly model is described by a set of descriptors of each constituting component, as depicted in Figure 2.1. In particular, the canonical shape distribution of Osada et al. [87] has been used to characterize the shape of each part of an assembly model. Then, the histogram, which encodes the shape distribution, is transformed into a point, whose dimension is established by the number of bins in the histogram.

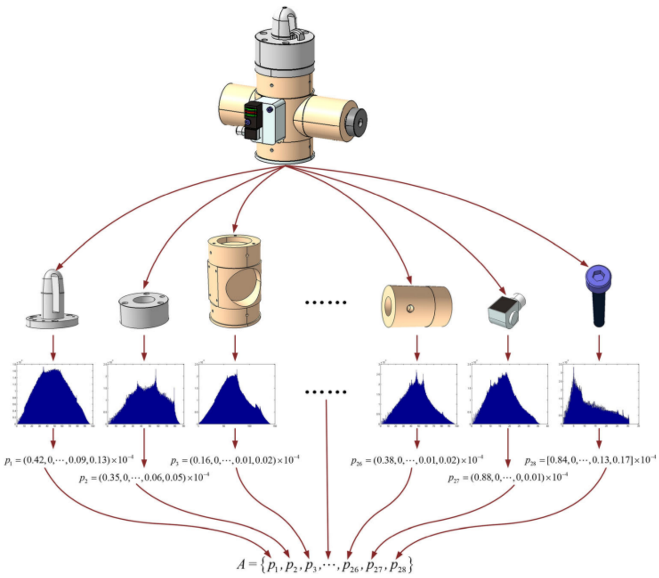


Figure 2.3: Example of the description of an assembly model (Figure 3 in [125])

Once the assembly descriptors are computed, the assembly retrieval is performed matching different point sets by using a well known strategy for image retrieval, the Earth Mover's Distance proposed by Rubner et al [105].

Using this strategy, assembly models are characterized at the level of the parts and partial query models are allowed. Indeed, since the relationships between parts are not considered in this work to describe an assembly model, then the query model can be represented just as a set of parts which have to be present in the target model.

These approaches are strongly based on the shape information and do not use assembly relationships, then no **topological information** described in section 2.1.2 is used. Concerning the importance of relationships in the definition of an assembly model, e.g. geometric constraints, kinematic links, or part arrangement, in our opinion, these methods restrict the application scenarios where an assembly retrieval system can be applied. Next section will present retrieval methods for globally similar models, that make use of assembly relationship information and whose assembly descriptor is able to characterize models at local level.

2.2.2 Retrieval methods for globally similar models using local assembly descriptors

The relationships of assembly models can be represented by the use of graph-based descriptors. Several authors define different graph structures, where the common practice represents individual assembly components as nodes, links between components as arcs of the graph and other information is represented in form of attributes of nodes and arcs. In this section, we describe the methods proposed by Tao and Huang [118], Miura and Kanai [83] and Deshmukn et al. [32]. These methods address assembly model retrieval at global level, but differently from the ones reported in section 2.2.1, these ones exploit assembly relationships to characterize assembly models.

✏ **Assembly model retrieval based on optimal matching [118]**

This work presents an assembly model retrieval method to find assembly model for design reuse and generation of manufacturing plans.
The *component attributed relational graph* (CARG) represents an assembly model as a direct graph where the nodes represent the components and the arcs correspond to connections between two components. Several attributes are associated with the nodes and the arcs for the description of the assembly model. In particular, each node encodes the surface type, the surface convexity, the loop number of a face and the edge number of its outer loop. While an arc represents the adjacency relationship between two components and encodes the *types of contact surface pair* and the *connection relations*, which can assume the following values: screw connection, pin joint, key joint, rivet joint, bearing, belt, chain and bonding or welding.

Even if in principle, local similarity could be detected, since the assembly descriptor is a graph, the matching procedure is based on a global evaluation of the similarity. The matching procedure computes the similarity between two components ($S(P^1, P^2)$) considering the surface properties, the surface area and volume components and the connection relations. To evaluate the similarity between two assemblies A^1 and A^2, a compatibility matrix ($SM(A^1, A^2)$) is build, where the element in the i-th row and j-th column is $S(P^1_i, P^2_j)$. Then the similarity between assembly A^1 and A^2 is computed as

$$S(A^1, A^2) = \frac{SM(A^1, A^2)_{max}}{\max m, n}, \tag{2.1}$$

where $SM(A^1, A^2)_{max}$ is the value of the optimal matching compatibility matrix $SM(A^1, A^2)$ which is evaluated using Kuhn-Munkres algorithm [68, 85] and m and n are the numbers of components in A^1 and A^2.

This method uses simple geometric information and kinematic links to characterize parts of assembly models and their relationships. Geometric information can be read from the B-rep of each parts, and we suppose the kinematic link are read likewise. The use of surface area makes the method sensible to dimension differences, while the values as "rivet joint", "bearing" and so on, used to characterize the connection relation, suggests to us that some attributes are supposed available in CAD models.

✏ **3D shape retrieval considering assembly structure [83]**

The goal of this woks is to provide a 3D shape retrieval method which satisfies the following four requirements:

(i) Evaluating assembly structure similarity, i.e. the method should evaluate similarity shape and structure of the assembly.

(ii) Maximum matching ability, i.e. the similarity measure should not change significantly if a minor component changes in the assembly.

(iii) Insensitivity to the movable components, i.e. the similarity measure should not consider relative positions of the components in an assembly model.

(iv) Flexible control of similarity evaluation, i.e the similarity should be easy controlled by the designer.

The assembly descriptor used is an assembly graph, where each node corresponds to a component in the assembly and each arc indicates a contact, an interference or if at least a geometric constraint is present between two components. To characterize the geometry of the model, several attributes are attached to the nodes and the arcs. In particular, the area, the volume and the angle distance are associated to nodes, while the type of constraints characterizes the arcs (if they identify a constraint).

The used graph representation guarantees a local characterization of the assembly model, anyhow the assembly retrieval is performed by a global graph matching, that it is treated as the *stable marriage problem*, solved by the Gale-Shapley algorithm [42]. The similarity is evaluated at the level of the component by using the difference between shape feature of two components and at the level of the structure by using an assembly graph matching.

The method is scale sensitive because of area and volume information used in the matching procedure. Geometry and geometric constraint information are read from assembly models, then this method supposes that the required information is available in CAD files. Finally, the "assembly structure" the authors write in the title can be misleading, since it refers to the kinematic link and constraints between two assembly parts and not to the hierarchical decomposition of an assembly model.

✏ **Content-based assembly search: A step towards assembly reuse [32]**

The method described in [32] exploits outcomes of previous research described in [33, 45]. In these works the authors describe a system for extracting information useful for searching and retrieving assemblies from databases. The collection of functional data is not identified from text-based annotation in the CAD model, but by studying and analyzing the content of the assembly models.

The authors take into account many aspects that play a meaningful role in the description of an assembly model. The data structure used in this work is the *mating graph*. For each assembly in the database, its mating graph is built. In this graph each node is associated with a part in the assembly model and each arc represents a mating condition between two parts.

In particular, for each part in the assembly model, they consider the following information.

- **Category**
  This information describes if the part is *standard* or *custom* and if it is set up by the user or left unspecified.
- **Geometry**
  The authors specify two types of geometric information depending on the category of the parts. If the part is standard, then its geometric information is referred by a library of standard parts, otherwise it is referred by the best approximation in the assembly database returned by the approach described in [23], which is based on face-based attributed applied vectors (FAAVs).
- **Type**
  This information is defined only for standard parts and specifies the subcategory of the standard part. It is selected by the user or left unspecified.
- **Degree**
  Given a part P, its degree represents the total number of parts that are in contact with P.

Considering part relations, the authors represent the type of contact between two parts and give to the user the possibility to select the type of relation or to leave them unspecified. This chance allows to define also incomplete query, i.e. a query model where not all its parameters are defined.

The query model in this method is represented as a mating graph, where parts are represented as nodes with four attributes (category, geometry, type and degree) and if two parts are in contact then an arc exist between the two corresponding nodes. Some attributes can be left unspecified.

Most of the information used by the authors to characterize parts are related to attributes (e.g. category and type), while the geometric information characterize parts by using their B-rep representation, area and curvature distribution. This data are available in the B-rep of each part, but the assembly relationships are not always present. One limit of this work is that this assembly descriptor requires the user to add manually many information, making tedious the use of this retrieval method.

The proposed algorithm for mating graph-based search is based on *graph compatibility problem*, which is very different from the graph matching (or isomorphism problem). It is based on the combination of several heuristic approaches to perform it. To improve the results, the search space is reduced assigning a priority score to each node in the query graph. Then the algorithm attempts to match each node from the query graph to a node from the database graph by recursive operations. Anyhow this procedure does not allow to retrieve partial or local similarities between two assembly models.

The three presented methods identify assembly similarity at global level, anyhow the assembly descriptors used are able to characterize assembly model at local level. This means that adopting other matching approaches, these methods could evaluate assembly similarity also at partial or local level. In the next section, we report methods that assess also partial similarity of assembly models.

2.2.3 Retrieval methods for globally and partially similar models

Assessing the partial similarity of two assembly models is a recent search field and, so far, few methods have been proposed and developed. Different assembly descriptors can be used to achieve this purpose and in this section, we describe three possible assembly representations. In particular, Hu et al. [51] represent an assembly model as a vector, where each part model corresponds to an element in the vector. This approach only uses the geometrical information of a part and the relationships are not included in the representation. The descriptor proposed both by Li. et al. [71] and Zhang et al. [126] is able to capture geometrical and topological information of an assembly model; while Chen et al. [28] represent an assembly model by using a multilevel semantical and geometrical structure.

✏ **Relaxed lightweight assembly retrieval using vector space model [51]**

As relationships between the different parts are not always available in the CAD model, to overcome this issue, this work characterizes assembly models using their parts.

The generic j^{th} assembly in the database is represented through a vector in the n-dimensional space, as $d_j = (w_{1j}, \cdots, w_{ij}, \cdots, w_{nj})$, where n is the number of parts in the j^{th} assembly and w_{ij} is the weight of the i-th part in the j-th assembly. The weight is computed taking into account two factors. The first relevant factor considers the number of occurrences of a part and the assembly complexity, i.e. the number of composing parts.

The second aspect for the weight definition regards the uniqueness of a part. Indeed a part that occurs in few assemblies is more discriminating for matching operations. Then, considering the **part information** criteria of section 2.1.2, this descriptor uses *part statistics*, while no **topological information** is used.

Using this *vector space model* (VSM) representation, the similarity between two assemblies is established as a function of the angle between their associate VSMs.
The authors choose a relaxed retrieval since with the VSM representation an exact approach would be too restrictive.
In the relaxed matching algorithm, the query can be seen as an incomplete assembly model, since it is represented by a vector of several parts and it must be compared with the vectors of the assemblies in the database. This matching problem is solved using the graph theory, in particularly employing a bipartite graph. The parts of the query and of the assemblies originate the graph nodes, while the graph arcs represent the similarity between two parts.
The bipartite graph matching problem can be solved using the Kuhn-Munkres algorithm [68, 85]. However, this technique is very expensive computationally, $O(\mathbf{n^3})$, then the authors propose an *approximate matching algorithm*. With their greedy approach the matching process complexity is reduced to $O(\mathbf{n})$.

The main limitation of this method is the assumption that two assembly models are similar if they mostly share some of the same parts. This can be a filter to reduce the models to be compared, but the same parts can be arranged in different ways and with different contacts, thus we believe that comparing just the constituent parts in assembly similarity is too simplistic.

✏ **A similarity-based reuse system for injection mold design in automotive interior industry [70, 71]**

A study on the reuse of previous design to avoid starting from scratch is proposed by Li et al. [70, 71]. They aim to define a geometric reasoning approach independent from any CAD systems or design history.
Their method has been conceived for CAD parts, but they include a generalization for assembly models as well.

In [70], they exploit hierarchical representation of CAD models, which is composed of a tree-like structure (TR) that describes the global similarity, and an adjacent graph (ADJ), which characterizes local similarity. In this way, the method can support the assessment of global and partial similarity. Using this scheme, the root of the TR represents the entire model, the intermediate nodes represent a set of partial features and the leaves are associated with detail features, e.g. a single face of the solid model or a most detailed partition of surfaces. While, the ADJ encodes relationships between non-leaf nodes, for the parts it defines if two features have a common edge.

This representation can be used also for assembly models, anyway the author do not suggest directly to use this method, because kinematic information are not extracted and described explicitly.
Then, to retrieve CAD models of products in order to reuse the related mold designs, which mainly rely on experience of designers, the authors extend their work in [71], where, in case of assembly model, the TR corresponds to the assembly hierarchical decomposition, while the ADJ captures kinematic pairs between parts.

The similarity is computed by a subgraph isomorphism on the ADJs, which is based on the VF2 algorithm [31]. Then, the similarity is evaluated for each level of the TR in term of their shapes and relationships. In particular, the shape similarity S_i in the i^{th} level is evaluated by the following equation:

$$S_i = \sum_j \omega_j \times D2_{sim} \tag{2.2}$$

where, ω_j is the ratio between area of the j^{th} matched pair in the i^{th} level over the area of all the matched pairs in the i^{th} level and $D2_{sim}$ is the similarity between D2 shape distribution of matched pairs.

To sum up, in this work, parts are evaluated according to their shape using the D2 shape distribution, while the relationships encode information about the hierarchical structure and the kinematic link. There is no reference to the evaluation of kinematic link, then we assume they are supposed to be available in the CAD models. The use of the area of matching components in the similarity evaluation makes this method scale sensitive. In the end, it allows global and partial similarity, but both are achieved exploiting hierarchical structure. This suggest to us, that if two assembly models represent the same object using a different organization, then this method is not able to recognize as similar the two models.

✏ **Reuse-oriented common structure discovery in assembly models [126, 134]**

In this work, the authors explain that a component which is present in multiple products can convey significant design knowledge and can be reused to improve design efficiency and accelerate new product development. They extend their previous work [134], where they propose a generic face adjacency graph (GFAG) to discover the common design structures in assembly models automatically. The authors emphasize that this structure can capture topological and geometrical information of an assembly model and is suitable for assembly characterization since they use the concept of mating face pair (MFP) to encode the relationships in assembly models. Hence, in [134] an assembly model is represented by a graph, whose nodes correspond to parts of the assembly model and edges correspond to the MFP between two parts.

The shape of each part is represented by a face adjacency graph (FAG) [77], where the nodes correspond to faces of the B-rep of the part and the edges correspond to adjacency edges between faces of the part model. The FAG uses sampling strategy to capture shape characteristics of the B-rep part model. More precisely, sampled points belong to a face or an edge of the B-rep of a part. If the points belong to a face, then they are classified as as plane, convex, concave or transition point by the sign of the principal curvatures. Otherwise, they describe the edge as concave, convex or smooth (i.e. the dihedral angle) by using the tangent planes of the adjacent faces of the edge where the point belongs.
Then, considering the **part information** criteria of section 2.1.2, this descriptor uses *curvature*, *dihedral angle* and *geometric info*.

The relationships between two parts are read from commercial solid modeling platforms and can assume the following values: coincident, contact, offset and angle. Then, in this method the **topological information** of section 2.1.2 is represented by the *kinematic links*. The authors calculate a shape vector descriptor for each part in an assembly model using the sampled points of the FAG of a part and the ones of its mating parts. In this way, the description of a part depends from its contacts, then the description of a part model will change if different part models surround it.

Since, this characteristic can decrease the portion of common structure (i.e. local similarity) detected, the work is extended in [126], where the authors provide a graph descriptor that describes independently parts and mating relationships of an assembly model.

In [126] the parts are represented by vectors of shape distributions and the contacts are quantified by the equation 2.3:

$$L_p = \frac{S_p + S_{p'}}{2} \tag{2.3}$$

where the vector L_p represents in a single relation the multiple contacts between two parts p and p', whose shape vectors are S_p and $S_{p'}$.
Then, considering the **part information** criteria of section 2.1.2, this descriptor uses *shape distribution* and *kinematic link* as **topological information**.
No attributes are encoded and, since no normalization of shape distributions is mentioned, the used information varies under scaling operations.

Beside the fact that the mating information has to be available in the assembly model, there is no information about the semantic of the relation between two parts, i.e. the type of their involved contact. This characterization seems to be not sufficient to well characterize assembly models and how the parts are connected, since two parts can define different contacts, which produce different type of motions.

✏ A flexible assembly retrieval approach for model reuse [28]

Chen et al. [28] propose an approach, which aims to overcome the limitations due to the problems of not considering the hierarchy in product structure, not exploring the semantics of assembly interfaces and the absence of an indexing mechanism. Analyzing that problems, this works proposes a flexible assembly retrieval, defining and describing the following characterization:

- a multilevel assembly descriptor,
- an assembly matching technique,
- an assembly indexing and filtering method.

The assembly descriptor presented in this work takes into account different information levels. It includes topological structure, relationships between assembly components, and geometric information. More precisely, they consider the following information:

- **Topological structure**
  It illustrates how the components (i.e. the assembly, the sub-assemblies and the parts) are connected together and then it is able to store the hierarchical assembly structure.
- **Assembly semantics**
  It describes the type of the relationships between the parts in an assembly model through the *semantic assembly interface*. It is defined as multilevel, in particular it is divided into *function layer*, *implementation layer* and *geometry layer*.
  The function layer considers the degrees of freedom (DOFs) between two connected components in the assembly. It counts the number of translational, rotational or composite (i.e the combination of multiple DOFs together as the screw joint) degrees of freedom between two components.
  The implementation layer defines and counts the types of kinematic relations between two components.
  In the end, the geometric layer contains information about the shape of two connected components, e.g. if they are concentric, perpendicular, parallel or distant.
- **Geometrical information**
  With the previous information, only the assembly structure is encoded. To consider also the assembly shape, geometric information is computed for each assembly components and stored in the corresponding nodes. In particular, if the component is a part, then shape distribution vector of the part is computed. If the component is a sub-assembly, then the shape distribution vector od the bounding-box of the component is computed.
- **Attributes**
  Beside the previous information, other data can be considered, such as the *functions*, i.e. the task that a system or a components is able to perform; the *loads*, i.e. the forces, deformations or accelerations applied to a structure or its components; the *environmental conditions*.

Then, considering the **part information** criteria of section 2.1.2, this descriptor uses *shape distribution*, while the adopted **topological information** is the *structure*, the *kinematic links* and the *constraints*. The geometrical description is computed from the B-rep of each component in the assembly model, while the structure is read from the CAD model. About the extraction of the kinematic links, as far as we know, this work is one of the first assembly retrieval approach which try to identify automatically kinematic-pairs. Anyhow, the authors state that some complex kinematic-pairs are labeled manually as the interface between the parts.

Since the assembly descriptor contains copious information, the matching procedure is divided into two main steps to simplify the retrieve procedure. The primary process takes into account the topology structure of the multi-level assembly descriptor. The hierarchical graph matching is carried out using the VF2 subgraph isomorphism algorithm [31]. To prune the matching algorithm, in addition to the topology other semantic information is used. For the algorithm, two nodes are equivalent if the query node has less children than the compared node; while two arcs are equivalent if they have the same DOF. The second step refines the matching previously obtained considering the assembly semantics and the geometric information. This last step evaluates also the arrangement of the assembly components in the 3D space using an "assembly bone" representation, i.e. a structure composed of line segments which connect the geometric-centers of two components.
Then, in an implicit manner, the **topological information** makes use also of *part arrangement* information. We say implicit, since this information is not explicitly stored in the assembly descriptors but deduced and used during the matching process.

The use of this type of matching supports partial retrieval if the two compared assembly models have the same structure, i.e. only if the query model is present in the target model as sub-assembly. Even if this work is an improvement considering the other works in the state of the art, this hypothesis can exclude models which differ only for the structure. This has to be overcome, leaving to the user the possibility of selecting how important is the structure for his purpose.

In general, these methods are able to assess partial similarity by exploiting hierarchical structure. Then for these works, we include the value *structure* among the used **topological information** even if the assembly descriptor does not use it expressly. This practice suggests us, that if two assembly models represent the same object using a different organization, then these methods are not able to recognize as similar the two models, not even with a lower similarity measure.

2.2.4 Assessment of the assembly model retrieval literature

Table 2.1 summarizes works which address assembly model similarity for different purposes. In this selection, the works are ordered as they were discussed. For ease of reading, the following symbols are used in the table:

- C: computed,
- R: read,
- PC: partially computed,
- ●: complete,
- ◐: incomplete.

From the analysis of these works, we can observe that almost all of them face design model reuse and assume the full availability of the information necessary to derive their assembly model descriptors. This may be a big limitation, since not all of the necessary data are present in CAD model as discussed in section 1.4. Supposing that a user adds all missing data is not reasonable, since this practice is boring for the user.
Moreover, some works could be extended to characterize assembly models at local level but they do not exploit this capability.

In general, we observe that the research regarding the characterization and the retrieval of CAD assembly models is still in progress and many goals are still to be achieved. Few of the issues presented in section 1.4 are solved. Most works address the problem of retrieving models partially similar; which anyhow have solutions only taking into account the hierarchical structure. However, the limitation is that the structure similarity becomes a constraint and the user cannot retrieve an assembly model included in a bigger assembly if the query is not represented as a sub-assembly in the target model. This hypothesis can affect scenarios whose purpose is the maintenance of assembly components. Indeed, in this situation, a component included in an assembly model should be identified despite its designed structure. In addition, all the works suppose that the query model has either the same number of components or less than the target model excluding the case when the query model is bigger than the target.

Few works address different types of similarities between assembly models and usually the geometry, the size of assembly parts and the different type of their relationships are used. However, the extraction of these data is not faced and the information is supposed to be available or added by the user. Moreover, the practice of characterizing parts by their shape does not allow to treat the possible simplified descriptions of components in assembly models as discussed in section 1.2.1.

Finally, to the best of our knowledge, no work addresses the visualization issues of the results.

Considering the main limitations found in the assembly retrieval state of the art, we first aim to overcome the problem of the automatic extraction of the information trough reasoning algorithms on the geometry representation of models. To this aim, next section describes some works useful to recover the necessary data.

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	Purpose	Scenario	Part information	Used	Extracted	Topological information	Used	Extracted	Attributes	Level of components	Present	Scale sensitivity	Level of descriptor	Type of query model	Completeness	Global	Partial	Local
Retrieval of solid models based on assembly similarity [100]	Product information reuse	Search for models with similar assembly work instructions	Curvature	-	-	Structure	-	-	-	Assembly	-	Yes	Global	Part model	●	✓	-	-
			Dihedral angle	-	-	Kinematic link	-	-		Part	✓							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	✓	C													
			Set of 2D projections	-	-													
			Angle distance	-	-													
A matching method for 3D CAD models with different assembly structures using projections of weighted components [61]	Design model reuse	Search for globally similar assembly models with different hierarchical structure	Curvature	-	-	Structure	-	-	-	Assembly	✓	Yes	Global	Assembly model	●	✓	-	-
			Dihedral angle	-	-	Kinematic link	-	-		Part	-							
			Part statistics	-	-	Geometric constraints	-	-		Feature	-							
			Geometric info	-	-	Part arrangement	-	-										
			Shape distribution	-	-													
			Set of 2D projections	✓	C													
			Angle distance	-	-													

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
			Curvature	-	-													
			Dihedral angle	-	-													
			Part statistics	-	-	Structure	-	-		Assembly	-							
An assembly retrieval approach based on shape distributions and Earth mover's distance [125]	Design model reuse	Search for globally similar assembly models	Geometric info	-	-	Kinematic link	-	-	-	Part	✓	No	Global	Assembly model	◐	✓	-	-
			Shape distribution	✓	C	Geometric constraints	-	-		Feature	-							
			Set of 2D projections	-	-	Part arrangement	-	-										
			Angle distance	-	-													
			Curvature	-	-													
			Dihedral angle	-	-													
			Part statistics	-	-	Structure	-	-		Assembly	✓							
Assembly model retrieval based on optimal matching [118]	Design model reuse	Search for globally similar assembly models	Geometric info	✓	R	Kinematic link	✓	R	Yes	Part	✓	Yes	Global and local	Assembly model	●	✓	-	-
			Shape distribution	-	-	Geometric constraints	-	-		Feature	-							
			Set of 2D projections	-	-	Part arrangement	-	-										
			Angle distance	-	-													

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	Purpose	Scenario	Part information	Used	Extracted	Topological information	Used	Extracted	Attributes	Level of components	Present	Scale sensitivity	Level of descriptor	Type of query model	Completeness	Global	Partial	Local
3D shape retrieval considering assembly structure [83]	Design model reuse	Search for globally similar assembly models	Curvature Dihedral angle Part statistics Geometric info Shape distribution Set of 2D projections Angle distance	- - - - - - ✓	- - - - - - C	Structure Kinematic link Geometric constraints Part arrangement	- ✓ ✓ -	- R R -	No	Assembly Part Feature	✓ ✓ -	Yes	Global and local	Assembly model	●	✓	-	-
Content-based assembly search: A step towards assembly reuse [32]	Product information reuse	Search for globally and partially similar assembly models	Curvature Dihedral angle Part statistics Geometric info Shape distribution Set of 2D projections Angle distance	- - - ✓ - - -	- - - R - - -	Structure Kinematic link Geometric constraints Part arrangement	- ✓ ✓ -	- R R -	Yes	Assembly Part Feature	✓ ✓ -	Yes	Global and local	Mating graph	●/◐	✓	✓	-

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
Relaxed lightweight assembly retrieval using vector space model [51]	Design model reuse	Search for globally and partially similar assembly models	Curvature Dihedral angle Part statistics Geometric info Shape distribution Set of 2D projections Angle distance	- - ✓ - - - -	- - C - - - -	Structure Kinematic link Geometric constraints Part arrangement	✓ - - -	R - - -	No	Assembly Part Feature	✓ - -	Yes	Global	List of parts	◐	✓	✓	-
A geometric reasoning approach to hierarchical representation for B-rep model retrieval [70]	Design model reuse	Search for globally similar assembly models with different hierarchical structure	Curvature Dihedral angle Part statistics Geometric info Shape distribution Set of 2D projections Angle distance	- - - - ✓ - -	- - - - C - -	Structure Kinematic link Geometric constraints Part arrangement	- - - -	- - - -	-	Assembly Part Feature	✓ ✓ -	No	Global and local	Part model	●	✓	✓	-

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
A similarity-based reuse system for injection mold design in automative interior industry [71]	Design model reuse	Search for similar assemblies to design reuse or mold planning reference	Curvature	-	-				Yes			Yes	Global and local	Part or assembly model	●	✓	✓	-
			Dihedral angle	-	-													
			Part statistics	-	-	Structure	✓	R										
			Geometric info	-	-	Kinematic link	✓	R		Assembly	✓							
			Shape distribution	✓	C	Geometric constraints	-	-		Part	✓							
			Set of 2D projections	-	-	Part arrangement	-	-		Feature	-							
			Angle distance	-	-													
Generic face adjacency graph for automatic common design structure discovery in assembly models [134]	Design model reuse	Search for frequent similar sub-assembly models	Curvature	✓	C				No			No	Global and local	Assembly model set	●	✓	✓	✓
			Dihedral angle	✓	C													
			Part statistics	-	-	Structure	✓	R										
			Geometric info	✓	C	Kinematic link	✓	R		Assembly	✓							
			Shape distribution	-	-	Geometric constraints	-	-		Part	✓							
			Set of 2D projections	-	-	Part arrangement	-	-		Feature	-							
			Angle distance	-	-													

ARTICLE	CONTEXT		ASSEMBLY CHARACTERIZATION							ASSEMBLY DESCRIPTOR				QUERY SPECIFICATION		TYPE OF SIMILARITY		
Title	**Purpose**	**Scenario**	**Part information**	**Used**	**Extracted**	**Topological information**	**Used**	**Extracted**	**Attributes**	**Level of components**	**Present**	**Scale sensitivity**	**Level of descriptor**	**Type of query model**	**Completeness**	**Global**	**Partial**	**Local**
			Curvature	-	-													
			Dihedral angle	-	-													
			Part statistics	-	-	Structure	-	-		Assembly	✓							
Reuse-oriented common structure discovery in assembly models [126]	Design model reuse	Search for frequent similar sub-assembly models	Geometric info	-	-	Kinematic link	✓	C	-	Part	✓	Yes	Local	Assembly model set	●	✓	✓	✓
			Shape distribution	✓	C	Geometric constraints	-	-		Feature	-							
			Set of 2D projections	-	-	Part arrangement	-	-										
			Angle distance	-	-													
			Curvature	-	-													
			Dihedral angle	-	-													
			Part statistics	-	-	Structure	✓	C		Assembly	✓							
A flexible assembly retrieval approach for model reuse [28]	Design model reuse	Search for globally and partially similar assembly models	Geometric info	-	-	Kinematic link	✓	PC	Yes	Part	✓	Yes	Global and local	Assembly model	●/◐	✓	✓	-
			Shape distribution	✓	C	Geometric constraints	✓	PC		Feature	-							
			Set of 2D projections	-	-	Part arrangement	✓	C										
			Angle distance	-	-													

Table 2.1: Summary of the assembly retrieval methods

2.3 Assembly information extraction

In order to address the simplification and storage issues discussed in section 1.4, in this section, we present some works that deal with the extraction of information from assembly models for the assembly retrieval.

The type of extracted data are various and of different nature. From the analysis of the state of the art, we think that the relationships are the first important information to be deduced among the assembly components. Indeed, almost all the works which face assembly retrieval rely on topological information assuming that the data are available. Thus, in section 2.3.1 we present some works for the detection of assembly relationships.

Then, in this thesis, we aim to characterize the components of assembly models not only by their shape but also according to their functionality. This because, components with the same functionality can have different shapes and the user should have the possibility to select the criterion for assembly similarity (e.g. shape or function) that best fits his/her needs. For this purpose, we present in section 2.3.2 works that deal with the automatic classification of 3D models.

2.3.1 Detection of assembly relationships

Several works exist for the extraction of assembly contacts in different application domains. Yang et al. [129] proposed a simulation system for assembly process based on the constraints recognition, confirmation and navigation. Constraints detection is handled in the recognition phase, while the other phases compute some transformation matrices to move the assembly components in virtual systems. The recognition phase detects six types of constraints considering geometric information of the involved elements and it is based on the equivalence between them and the DOF. However it requires user intervention to track the position and the orientation of the components.

An automatic method to extract kinematic information from assembly models has been developed by Park and Ji [89]. They assert that most of the assembly geometric models have just revolute and prismatic joints. Their procedure is divided in three main steps. First, it identifies the regions of contact using the collision detection of boxes and parallel computation to improve the speed of calculation. Focusing on revolute joints, the second step aims to identify the contact surface of the cylinder shape by using Gauss map. Then, the center axis of the cylinder shape is created by using the normal vectors of the triangles that form the contact surface. The last step decides the type of joint using collision detection. Even though their approach identifies joints automatically, the proposed scenario it too restricted.
Swain et al. [114] have defined an extended liaison (joints) to integrate the information between the product model and the assembly process. This structure allow the identification of the assembly process of riveting, welding, screw fastening, bolt fastening and gluing. The proposed procedure is able to extract the joint details automatically from assembly models, but its main limit is the complexity of the algorithm.

For finite element analysis, Shahwan et al. [111] have described a qualitative reasoning process to detect component interfaces from assembly models. Their method is based on the definition of conventional and functional interfaces. To functionally classify the interfaces in their geometric reasoning process , they exploit additional knowledge expressed in ad-hoc ontology. Kim et al. [65] presented ontology based reasoning techniques for representing and differentiating assembly joints that are similar from geometric and topological point of view. They gave definitions and theorems for characterizing assembly joints in mereotopological representation (a formal ontology that combines mereological and topological concepts) requiring various data as fastener, screw/nut head and body attributes. However, it is not mentioned how this information are obtained from the assembly model.

2.3.2 Classification of assembly components

The automatic classification of 3D models has been widely addressed in literature. In the mechanical field, Ip et al. [54] define a feature space where they apply decision tree learning and reinforcement learning to classify solid models. This first effort allows the automatic classification of wheels, sockets and housing models. In [130], the authors present an automatic model classifier for CAD models integrating machine learning techniques. Using a series of shape descriptors, their approach aims at learning multiple CAD classifications and is applied to the classification of prismatic machined parts and parts with finishing features machined after part casting. The classification proposed by Pernot et al. [91] also exploits a series of shape descriptors and classifies products in terms of characteristics that might affect the simplification process for the Finite Element Analysis (FEA) of parts. Hence, their categories are: thin parts, parts with thin portions and normal. Qin et al. [97] present an automatic 3D CAD model classification approach based on the deep learning technique. Their method considers 28 different functional classes and combine different training strategies to simulate engineering manual classification processes.

However, as highlighted in [69] and [58] the functional classification of 3D models requires information on the context of use of the part. Shahwan et al. [109, 111] analyze functional interferences from the geometric interferences of parts in an assembly and identify functional designations, as cap-screw, tubular rivet, gear. The main limitation of this method is the complete entrustment in the design methodologies. The extension of this work [110] uses mechanical equilibrium state analysis for assigning to geometric interfaces only one functional interface. The approach is semi-automatic and user has to identify the start and the end of the kinematic chain in the assembly model.

2.4 Conclusions: synthesis of the needs

To achieve a powerful retrieval of assembly models according to different similarity criteria much work has still to be done. Indeed, considering the issues presented in section 1.3, many aspects are still to be considered:

- **Naming issues**
  Text-based retrieval systems are limited, since two models are considered similar only if they have similar text in their annotations. Thus, we have focused our analysis on works that exploit the intrinsic description of assembly models (e.g. shape description or geometric constraints) as a key to perform searches over model databases. This restriction has emphasized that just few works directly deal with this subject, leaving space for considerable and significant developments.

- **Size issues**
  The explosion of digital data has made available 3D models to designers offering the possibility of reusing existing solutions. Anyhow, the size of the digital models (also in terms of the constituent part number) and of the databases make challenging providing an efficient retrieval system that properly satisfy users' needs. To this purpose, retrieval methods often split the comparison procedure in two steps: a primary similarity assessment extracts candidate models, while a second refining improves the retrieval.

- **Storage issues**
  The absence of information as the kinematic links, constraints or attributes derived by the practice of not explicitly modeling them or by using standard file formats makes challenging the development of an efficient retrieval system. Anyhow, many works assume the comprehensive availability of the required information or base the similarity on simple parameters that are always included or easily computed (e.g. the principal curvatures on part surfaces).

- **Similarity issues**
  Assembly models may be considered similar under various and different criteria (e.g. global shape, kinematic links, component dimensions), moreover different types of similarity can be fulfilled (i.e. global, partial and local). To the best of our knowledge, the majority of works, which allow to retrieve assembly models according to different criteria, combines shape criteria of similarity on the components with relationship criteria. Anyhow, few works allow to combine functional aspect of similarity (e.g. power transmitter) of the assembly components together with geometrical characteristics (e.g. round shape). In addition, few works address local similarity and those that face partial similarity strongly rely on the hierarchical assembly structure, i.e. two models can be considered partially similar only if one is present as sub-assembly in the other.

- **Representation issues**
  In section 1.2.1 we have discussed that in a DMU the representation of the shape of the components can be simplified collapsing sub-assemblies into single parts or idealizing part shape with simplified outlines. With this simplification, elements with the same shapes may have a complete different functionality and vice versa. Analyzing the existing works, we could not find evidence of the possibility of dealing with different representations of the same component.

- **Query issues**
  A useful feature of retrieval systems is represented by the chance of querying databases using different descriptions for the query model.
  As already discussed, this possibility is important especially in the early design stage. Moreover, it is important to allow specifying similarity criteria in an intuitive way. To this purpose, we have found that some works address the query issue giving the possibility of using query model partially defined or abstract query model not originated from existing CAD model.

- **Visualization issues**
  Visualization issues regard the ability of retrieval systems of proposing results and of highlighting the different similarities to the user in an intuitive and clear manner. Differently from the other issues, this aspect becomes crucial dealing with partial and local similarity, while it is less essential in global retrieval. Since most of the works in literature define retrieval methods just for globally similar models, visualization issues are not treated.

Considering the main limitation still present in the current literature, in the next chapter, we propose a retrieval framework based on an assembly descriptor, which is able to characterize assembly models with different type of information automatically extracted and allowing different types of similarity searches.

3

Overall framework

the previous chapters have pointed out the requirements and the related challenges of developing a retrieval system able to manage intrinsic descriptions of assembly models.
This chapter presents our proposition for a framework for the retrieval of similar assembly models according to criteria that can be convenient for designers at different stages of the Product Lifecycle. It is based on an assembly descriptor, called Enriched Assembly Model (EAM), which encodes all the required data automatically extracted by analyzing the geometry and structure of CAD models. It allows abstract queries, which can be further refined and applied on search results.

Contents

3.1 Retrieval system requirements

In chapter 1, the importance of assembly model retrieval systems has been discussed as a mean to improve the product development process. As most of the works present in literature argue, a flexible and user-oriented retrieval system requires high-level information that is not always available in CAD assembly models. Thus, innovative solutions must aim at devising a search framework that includes methods able to provide the extraction of important information from assembly geometry and hierarchical structure, allowing the retrieval of assemblies similar according to shape, arrangement of parts and/or functional characteristics.

In this section, we focus on *how* such a framework could be, discussing which are the important requirements it should satisfy.

3.1.1 Search flexibility

Flexibility is a term that can cover several aspects of a system. Normally, referring to software, flexibility is *the ease with which a system or a component can be modified for use in applications or environments other than those for which it was specifically designed* [5]. When focusing on retrieval systems, we are interested in pursuing the following points.

- **To address different types of similarities**
  For retrieval systems, flexibility refers first of all to the capability of addressing the retrieval according to different similarity types. We already mentioned in section 1.3.1 the difference between global, partial and local similarities. There is no best type of similarity, but one may respond better than another to different search objectives. On the one hand, global similarity can be used to look for existing similar models avoiding redesigning and duplicating something that is already available within the company or on the market. An example is presented in Figure 3.1 where the left box indicates wheals retrieved using as query model another wheal. On the other hand, partial similarity may be used to know which products adopt that specific component, e.g. the chairs containing a given wheel as shown in the right box in Figure 3.1. In this case, retrieved models include the query model but are not globally similar to it. In the end, local similarity is suitable to identify shared similarity between two models.

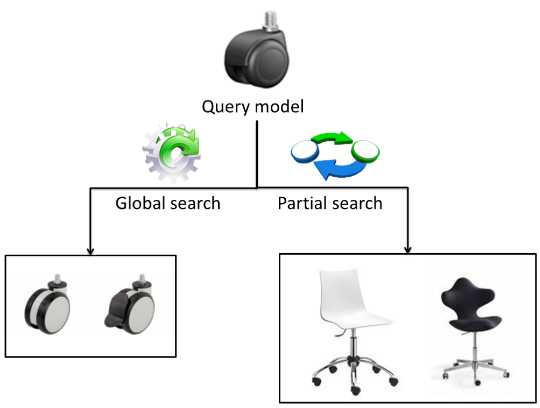


Figure 3.1: Example of global (left box) and partial (right box) search with the same query model.

For sake of precision we want to specify that, in all the reported cases, the target model may be not fully covered, i.e. some of its parts may not have a similar corresponding part in the query model. For instance the parts representing seats in the right box in Figure 3.1 are not matched with any part of the query model. The same consideration is true also for the query model, i.e. some of its parts might not have a similar part in the target model. For instance, the query model in Figure 3.1 has a wheel cover which is not present in the first wheel in the left box. We refer to this concept as the *coverage* (i.e. the percentage of matched components) of the query model and of the target model and we believe that a flexible retrieval system should also be able to allow the setting of coverage threshold.

- **To allow combining different search criteria dynamically evolving**
  Flexibility is characterized also by the capability of performing searches combining different similarity criteria and allowing a dynamic evolution of the search criteria. It means that a user should have the possibility of searching for something very specific or of starting his/her search with general criteria and then skimming results adding more filters according to the proposed results, i.e. the criteria can be combined in different ways according to different objectives.

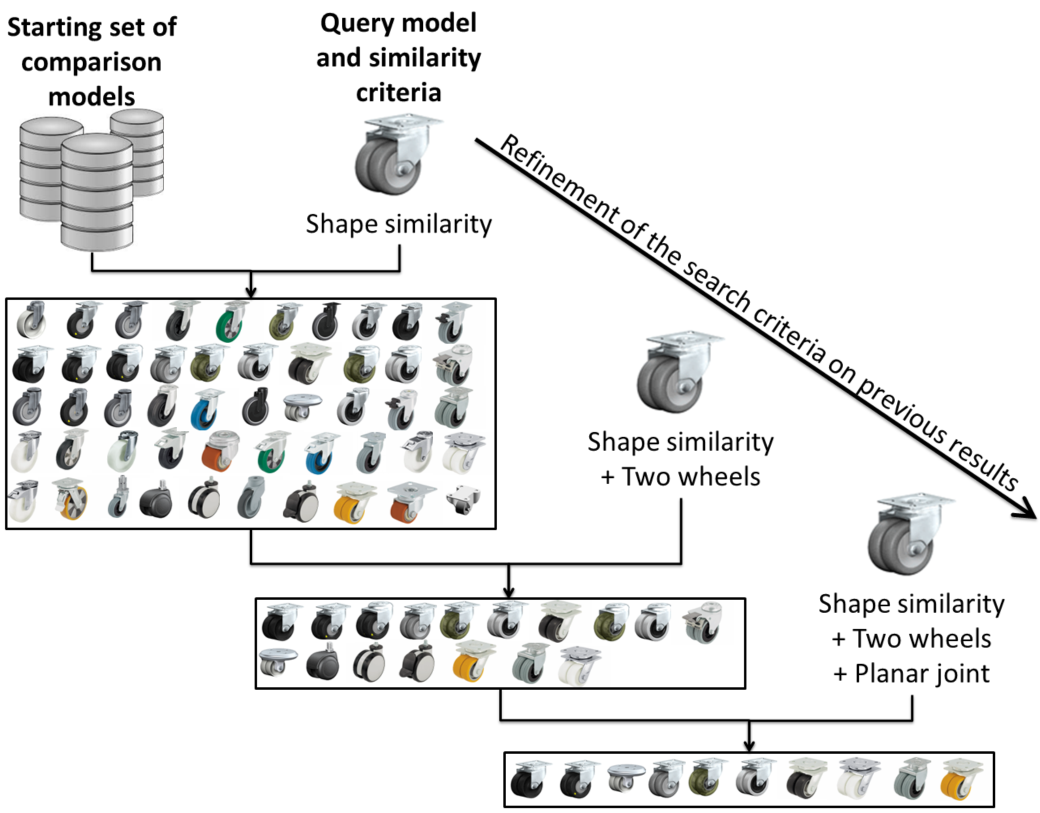


Figure 3.2: Search refinement process

Let us suppose a designer is looking for a trolley wheel as the one in the example in Figure 3.2. The first common idea is of filtering the DB according to the shape of the query model. Analyzing the proposed results, he/she realizes that there are results with different number of wheels, thus the user may decide to filter them adding the number of wheels as additional criteria. Among the filtered models, the user may decide to consider only the models having the specific upper joint necessary to fasten the wheel with the chair. In this case, only the models with a planar joint are finally retrieved. This simple example, representative of common situations, suggests that a retrieval system should support search combining different criteria and allow an iterative search refinement process.

- **To allow cut-off values**
  Considering the example of the search of wheels, user may be interested in the dimension of the model. Depending on the type of modification, he/she may be interested in retrieving models with exactly the same dimension or models with the same shape and different size. Thus a good retrieval system should support both situations, and this can be achieved, as represented in Figure 3.3, allowing to set also threshold values for the same criterion.

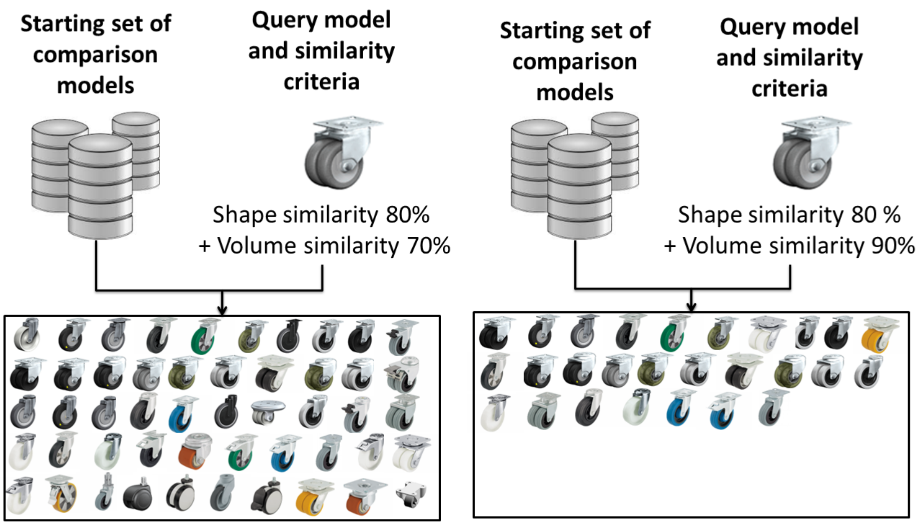


Figure 3.3: Search with threshold setting

Through these simple examples, it is clear that with the same query model and different search criteria, a database can be filtered in different ways, not only specifying some similarity conditions, as shape or joint, but also through a more general setting, such as the specification of a threshold for a specific similarity criterion. More features (as different similarity criteria, possibilities of setting thresholds or different types of similarity) are supported by the retrieval system, more the system is flexible.

3.1.2 Search effectiveness

A fundamental characteristic of a retrieval system is the appropriateness of the retrieved models that allows reducing the time required to verify the results and/or to run other queries to get more appropriate results.
Usually, *precision* (the ability of a system to avoid unwanted retrievals) and *recall* (the ability of the system to retrieve wanted items) are the two parameters used to assess the effectiveness of a retrieval system (i.e. the quality of its search results) [78, 84].

The elements needed to evaluate the effectiveness of a system are [79]:

i A collection of models;

ii A set of queries, which express a user information need;

iii A set of relevant judgments (ground-truth), i.e. a binary assessment for each model to indicate if it is *relevant* or *non relevant* for each query.

Given these elements, we can define precision and recall as [79, 107]:

$$Precision = \frac{\text{number of relevant retreived items}}{\text{number of retrieved items}} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \quad (3.1)$$

$$Recall = \frac{\text{number of relevant retreived items}}{\text{number of relevant items}} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \quad (3.2)$$

where,

- true positives are the relevant models that are correctly retrieved;
- false positives are non relevant models that has been incorrectly retrieved;
- false negatives are examples of relevant models that have been incorrectly not retrieved;

From these definitions, we can hint that the final values of precision and recall strictly depend on the evaluation of relevance done previously on the dataset according to the specific query model. Generally speaking, the relevance of a model is assessed according to a user's need. This means that a model which satisfies constraints specified in the query could be not relevant for the final purpose of the user. This subtle difference represents a great hurdle in the evaluation of the effectiveness of a system, since generally it is very hard to provide a fulfilled ground-truth.
Several shape benchmarks exists (e.g. the Princeton Shape Benchmark (PSB) [112], the National Design Repository (NDR) [99] and the Engineering Shape Benchmark (ESB) [58]) for the classification of part models, and so far, no public database exists to evaluate and compare assembly retrieval [51, 96].
Then, even if the effectiveness of a retrieval system is unquestioned important, its evaluation is a challenging task.

3.1.3 Ease of use

Another fundamental requirement is the ease of use of the system. It is pointless having a system able to retrieve all desirable models according to the most varied criteria, if the user is not able to formulate the query in a simple way. Guaranteeing an intuitive user interface allows saving time, reducing costs and preventing expensive and long training of the designers to learn how to use the system.

There exist many ways to consider the *ease of use* of a system. For instance, it may refer simply to the question *"how difficult or easy did you find this task?"* as proposed by Wetzlinger [127] or it may be linked to a more complete concept of *usability*. In the guideline for measuring the quality of a software ISO/IEC 9126 [59], usability includes several characteristics described as follows [11],

Understandability: the capability of the system to enable the user to understand how to achieve particular tasks. For example, interface should be easy to understand.

Learnability: the capability of the system to enable the user to learn the application.

Operability: the capability of the system to enable the user to operate and control it.

Attractiveness: the capability of the system to be attractive and pleasant to the user.

Usability compliance: the capability of the software to adhere to standards, conventions, style guides or regulations relating to usability.

In our case, we aim to have a more complex characterization of the *ease of use* than Wetzlinger but lighter than the concept of usability. In particular, our goal is to define a system which addresses the understandability and operability characteristics. We do not include the other characteristics, since they involve other areas of competence, which are not fully pertinent with the general purpose of this thesis. Indeed, learnability should involve psychologist aspects, attractiveness involves graphic design, while the usability compliance requires detailed study of the software standards.
Then, we describe how understandability and operability can be addressed in a retrieval system.

- **Understandability**
  To satisfy the understandability of a retrieval system, its user interface should be intuitive in the query specification including the similarity criteria. For instance, Deshmukh et al. [32] discuss the usability in their retrieval system and suggest that similarity criteria must be organized in a navigation taxonomy that groups together similar search criteria according to predefined categories reflecting the assembly design process.

- **Operatibility**
  The capability of a user of using a system depends also on his experience, then the operatibility of a system should meet both the ability of neophytes and that of expert designers. To this aim, the system should give to the expert user the possibility to customize the different criteria of similarity without constraints, and at the same time, it should offer a set of predefined queries ready to be used by less experienced users.

3.1.4 Informativeness of the results

If finding similar models is the first task for a retrieval system, then the second one is to support the user in the analysis of the results.
The complexity of information, which concurs to define the similarity among assemblies, makes difficult to assess the effective similarity and, as consequence, the selection of a model that best fits user's expectations can be challenging. Then, it is natural that between the retrieved models, a user will want to perform some further analysis. In this sense, the system should provide a visualization helping him/her in their inspection also gathering results according to the various similarity criteria. This is even more important in the case of large assemblies, where detecting the parts considered similar to a given query model might be particularly difficult.

We think that to support designers in this post-retrieval phase, the following actions should be eased:

- **Identification of the target model**
  Understanding the content of an assembly model by its name can be challenging, especially if the naming uses non familiar codes. Thus, a 3D visualization that allows rotating and zooming a retrieved assembly model can be more fruitful for the user to quickly understand which model has been retrieved.

- **Localization of the matched components in the target model**
  Working with partial similarities, it is important for the user to understand which similarity conditions of the query have been satisfied and from which components of the retrieved assemblies. Thus, it is important to highlight the elements recognized similar by the system and underline how much they fulfill the various similarity criteria.
  In addition, similar components can be internal to the assembly models and then hardly visible. To ease the comprehension, it is recommended having a tool able, for example to operate transparencies to outer components to allow the user to identify the similar elements.

- **Evaluation of the level of similarity**
  If the system is able to support different types of similarities (i.e. global, partial or local), then an additional requirement of a retrieval system is the capability to show how similar to the query the retrieved assemblies are. In this sense, we think that the user should quickly get an idea of which similarity each retrieved model fulfills, for instance by presenting the percentage of matched elements of both the query and the target model.

3.1.5 System openness

The evaluation of a retrieval system depends on also the ability of integration with existing systems. In this case, we refer to the possibility of satisfying as much development environments as possible. The retrieval system should be integrated in CAD systems with limited customizations. A possible way to achieve a cross platform system is sketched in Figure 3.4, where intermediate layers are adopted to allow the communication between CAD and retrieval systems. In this way, the system owns also the ability of adaptation, since the kernel of the retrieval system is independent from the employed CAD system, and customizations are required only for the geometric data extraction from the CAD models.

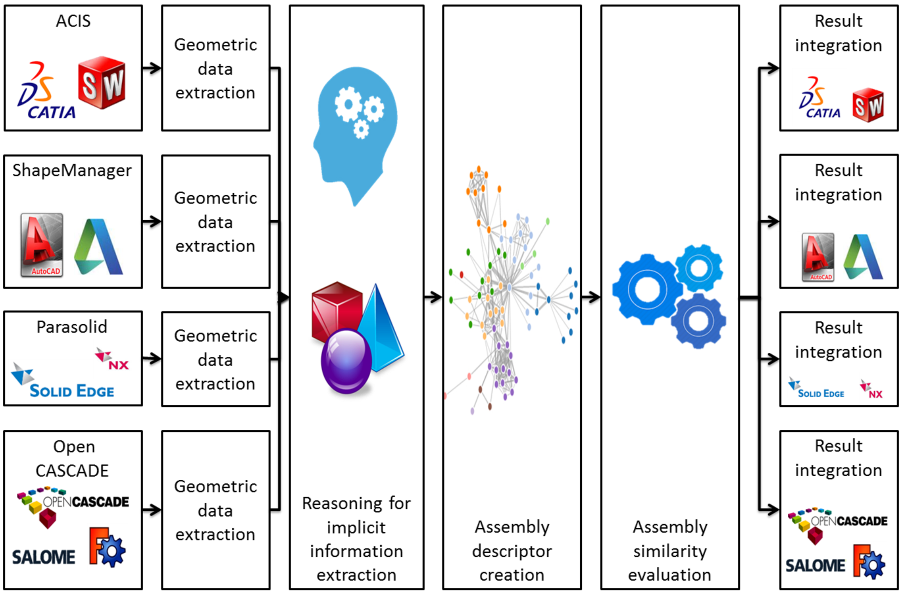


Figure 3.4: Intermediate layer for adaptation

3.2 Framework architecture

In Chapter 1 we have described the main issues arisen in the retrieval of assembly models and the analysis of the status of the art presented in Chapter 2 has shown that few of those issues have been faced so far.
We notice that a lot of works assume the availability of all the needed information to derive their assembly descriptor. In addition, to the best of our knowledge none of the system deal with the presence of possible simplifications in the description of the assembly components or of possible unrealistic configurations due to positioning errors. These assumptions might be a strong limitation in the applicability of the methods in real contexts.
Moreover, many works do not deal with the whole complexity of the assembly (e.g. considering just the shape of the constitutive parts of an assembly model without any information on their connection) or do not combine similarity criteria of different nature (e.g. considering functionality and shape similarity simultaneously). Few works offer the possibility of specifying partially a query or do not require the usage of an assembly CAD model, i.e. they admit other representations, as mating graphs or 2D sketches. Finally, visualization issues are not treated.

Considering these limitations, in this thesis, we aim to face the issues described in section 1.4, as described in the following.

- **Size issues**
  Size issues refer to the complexity of the assembly models in terms of the number of its constituting components and the large size of the databases to be queried. To provide an efficient retrieval system, we aim to identify and extract some distinctive features, which allow to reduce the number of models to be compared and speed up the retrieval process.

- **Naming issues**
  Many interesting information of assembly models are encoded in text format and managed by PDM and PLM systems. Anyhow, these systems offer limited capability of analysis on geometrical aspects that can be useful in many application scenarios. This issues is addressed in this thesis restricting to the use the CAD description of assembly models to perform searches over model database.

- **Storage issues**
  Since many important information (as the kinematic links, constraints or attributes) is not always explicitly stored in CAD models, an essential requirement of the proposed system is its ability to extract all the required data just analyzing the available data in a DMU model without user's help.

- **Similarity issues**
  Similarity issues refer to the fact that user may be interested in retrieving assemblies which are similar according to different characteristics. We address this difficulty providing a comprehensive assembly descriptor, which characterizes assembly models at local level under several aspects, included but not limited to shape of the parts and contact information. In addition, we aim to address different types of similarity (i.e. global, partial and local) adopting a matching method able to exploit the local characterization of the assembly descriptor.

- **Representation issues**
  To deal with the simplification often present in the representations of components, the proposed assembly descriptor will include the characterization of elements of the models with criteria that are beyond the shape, such as the type of the represented components.

- **Query issues**
  The possibility of using partial queries, i.e. queries where not all the information of the adopted assembly descriptor is specified, allows to satisfy more application scenarios. This permits also the usage of not only CAD models as query but of also different assembly representations, as a graph-based one that can provide indications on the type of components and on their relationships without details on their geometry. In addition, differently from most of those present in literature, in the proposed retrieval system the query model can be bigger (i.e. with a higher number of components) than the retrieved assemblies.

- **Visualization issues**
  To improve the informativeness of the results, in this thesis, we explore how to facilitate the interpretation of the proposed results and develop a proper user interface for the browsing of the results and their visualization.

The proposed system architecture is illustrated in Figure 3.5. It shows the different modules as well as the way they communicate at the different levels. There are three main levels: user interface, functional and data level. The framework considers both real-time processes and batch processes to be executed in advance. The batch processes converts models in the dataset into proper descriptor, called Enriched Assembly Model (EAM), while real-time processes compute EAM descriptor for the query and perform the comparison. It must be noted that the EAM, described in section 4.2, is a very rich model, including many information. Therefore, a complete EAM version is computed only for the models of the database. For the query model, only the layers containing information involved in the criteria used for the comparison are computed and exploited for the matching reducing the complexity of the system.
Once the dataset is entirely processed, the user can specify his/her query in the user interface level. If the model used for the query is already present in the database, then its EAM is already computed entirely, while if it is a new model, then its EAM is computed at run time. The retrieval module communicates with the user interface layer which displays the obtained results with their associated measures.

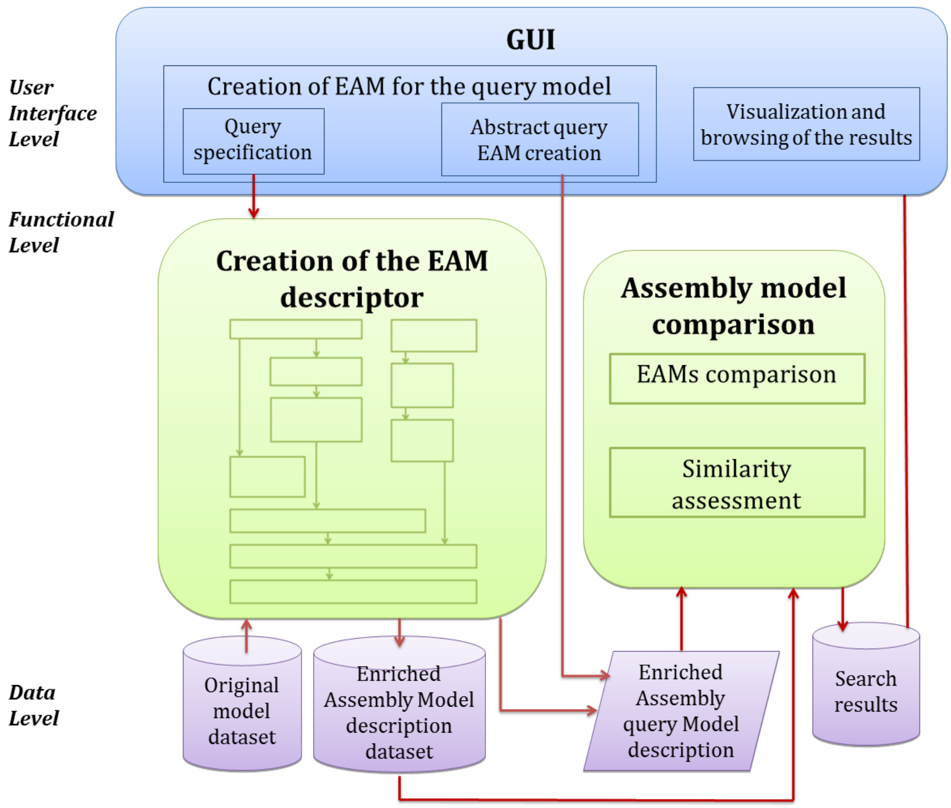


Figure 3.5: Framework diagram

3.2.1 User interface level

The user interface level includes the handling of the user interaction activities and the visualization aspects.
It provides a support for the specification of the search query and for the browsing of the search results. Thanks to a dedicated graphic user interface, the designer easily specifies his/her queries. The modules *Creation of EAM for the query model* is illustrated in section 4.4.13, while the description of the *Visualization and browsing of the results* is provided in section 6.2.2.

When we speak of *Query specification*, we refer to a set of information necessary to customize the search and thus required as input for the retrieval module. Generally, a query is composed by four elements, as illustrated in Figure 3.6.

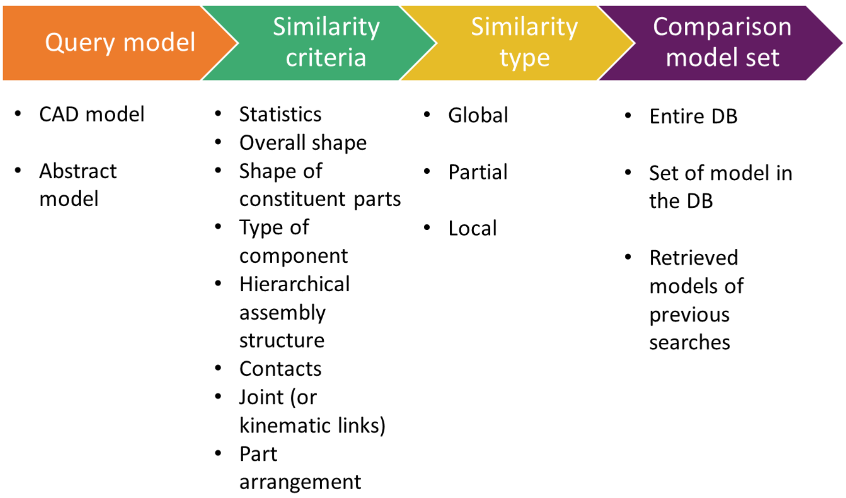


Figure 3.6: Elements to define a query

The first element of the query is represented by the model. If the user wishes to use an existing CAD model, it is sufficient to choose the desired model and automatically the module *Creation of the EAM description* generates its assembly descriptor as described in the section 3.2.2. As an alternative, with the *Abstract query EAM creation*, the query can be generated starting from an abstracted assembly model. This means that the user, through the interface, is able to build an EAM or part of it specifying the meaningful attributes.
Once selected the assembly models to be used for the performed search, i.e. the *Comparison model set* illustrated in Figure 3.6, it is necessary specifying which are the *Similarity criteria* (i.e. the elements and the attributes to be considered to evaluate the similarity) and the eligible *Similarity type*.

To exploit existing development environments, we choose of integrating the proposed framework in a commercial CAD system, offering a plug-in with a simple interface divided in sheets according to their functionalities. More precisely, we have four sheets, one for the creation of the assembly model descriptors dataset; two to choose the query model and to set the similarity criteria and the similarity type; and the last one to list the retrieved models and their measures. This last part will be discussed in section 6.2.2.

3.2.2 Functional level

It contains the main modules of the framework, which are detailed in the next chapters. In particular, chapter 4 will describe the *Creation of the EAM descriptor* and chapter 5 the *Assembly model comparison* module.

The EAM creation module oversees the computation of the assembly model descriptors of the CAD assemblies contained in the original database, as well as their storage within a dedicated database. Moreover, it deals with the creation of the EAM for the query model both in the case of a CAD model and of an abstract query model specified by the user through the user interface.
Except in case of abstract query specification, the construction of the EAM starts reading CAD model file and is composed of ten different modules, which extract explicit and implicit information of the CAD models meaningful for the similarity evaluation. Some modules can run in parallel. An overview of these modules and how they are combined together is provided in Figure 4.13 in section 4.4.2, while their complete description is reported in the sections: from 4.4.3 to 4.4.12. The creation of the EAM for the query model is described in section 4.4.13.

At the end of this process, an EAM is represented as an attributed multi-graph, i.e. a graph where nodes and arcs have associated attributes and multiple arcs are allowed between a pair of nodes. This structure allows encoding geometric and topological information of the assembly. Moreover, it owns invariant property for geometric data and it can be enriched thanks to the use of attributes. Details on this structure and its attributes will be provided in section 4.3.

The resulting descriptor is then used as an input of the *Assembly model comparison* module. It includes two modules: the *EAMs comparison* and the *Similarity assessment* modules. The first module is used to compare the descriptor of the query with the assembly descriptors in the pre-processed database (described in section 5.1). Adopting graph representation, if two models have a common feature, then their attributed graphs must have a common subgraph. Details on the similarity assessment using graph representation will be discussed in section 5.1.1. In our case, the similarity assessment between two EAMs is performed by matching their attributed multi-graphs and finding their maximum common subgraph (MCS), this approach will be described in section 5.1.2.
In the end, the *Similarity assessment* module computes similarity measures for each compared model as described in section 5.2.

3.2.3 Data level

It includes the input and the output data of the functional level.

The original models are stored in the *Original model dataset* and they are represented by CAD models present in the DMU. Once these models are processed in the functional level by the *Creation of EAM descriptor* module, their corresponding descriptors are created and stored into the *Enriched assembly model description dataset.* Analogously, the query model is stored in the *Enriched assembly query model description* for possible later reuse.

In the end, to allow an iterative search, this level contains also results of previous searches in the folder *Search results.*

3.3 Conclusions: remarks about the framework

In this chapter, we have described the requirements that a retrieval system should satisfy. In general, an optimal retrieval system should allow flexible and effective searches, ease of use and provides suitable informativeness of the results. Lastly, system openness guarantees the adaptation of the system to new specifications (e.g. supporting in input a new assembly representation) or features (e.g. supporting new search criteria).

The proposed system answers to the need of flexibility allowing different type of similarities, iterative search refinement and threshold setting. In addition, keeping the similarity type option separated from the similarity criteria leaves more autonomy to users without precluding ease of use. Finally, the separation in modules and sub-modules of the framework improves the ability of adaptation of the system.

The next chapters will deeper detail the proposed framework. In particular, chapter 4 will describe the creation of EAM and chapter 5 will illustrate the matching procedure and the evaluation of the similarity between assembly models.
In the end, chapter 6 will illustrate how the problem of the infomativeness of the results is addressed. It will also present and discuss some results obtained with a developed prototype of the proposed framework.

4

From CAD models to Enriched Assembly Models

An efficient and meaningful retrieval of CAD assembly models relies on a suitable description of assembly models, which encodes information that is already available in CAD models and other that has to be extracted.
This chapter presents and details the so-called Enriched Assembly Model (EAM) descriptor. It illustrates the main difficulties when extracting data, how they are overcome and how the EAM is generated.
The conclusion of this chapter reports results about the extraction of implicit information and illustrates some limitations and further works of the proposed techniques.

Contents

4.1 Assembly descriptor requirements

A fundamental problem dealing with assembly retrieval is the definition of a proper and effective descriptor for an assembly model [77]. An assembly descriptor can be a set of values, a graph or other abstract mathematical formulations that can be extracted from an assembly representation to capture its meaningful characteristics.

Below we list the requirements that an appropriate assembly descriptor must meet to effectively support the retrieval of assembly models.

Representing shape of parts and components. Basic condition for an assembly descriptor is the ability to capture the shape of its constituent elements, thus it is essential to be able to describe parts of a model.

Being scale sensitive. Dimensions of constituent parts are a crucial factor of similarity in many use cases, thus it is fundamental that the assembly descriptor encodes the size of the involved components.

Representing function of constituting components. To support designers in translating technical requests into draft, designs needs knowing the function associated to a specific component, where by "function" we mean an abstract formulation of a task that is independent of any particular solution. Thus, being able to capture the function is important.

Representing structural information of the assembly model. In an assembly model the constituent parts are frequently grouped into sub-assemblies, which are dependent on user intent. Thus, it is appropriate that an assembly descriptor is able to capture such sub-assembly structure.

Representing interaction of parts. For a more complete description of a model, it is important to include how parts interact determining the final allowed movements. Thus, joints between parts should be taken into account and captured by the descriptor.

Representing mutual position of parts. Models with the same set of parts may meet different requirements according to their arrangement, therefore a worthy assembly descriptor should represent how parts are located in 3D space in relation to each other. Thus, repetitions of parts should also be captured, symmetry as well.

Managing different levels of detail. An assembly descriptor should support the evaluation of different types of similarities, either focusing on detailed or macro characteristics of the assembly model. The macro characteristics ensure that small features, as fillet and chamfers, are not the major concern in similarity comparison, while a detailed description allows discriminating among local dissimilarity.

Being orientation invariant. A descriptor must be independent of the assembly location and orientation. This can be expressed more rigorously by requesting that the contained information must remain invariant under rigid transformations, i.e. translations and rotations of the assembly model.

4.2 The Enriched Assembly Model

Based on the analysis of the requirements and the review of the related works, attributed trees and attributed graphs result the two structures that better meet those requirements. Indeed, trees are able to capture the structure information of an assembly model and graphs allow encoding other relational information, as the mutual position or behavior of parts. In both structures, nodes represent components and parts of an assembly model, where additional information are associated as attributes to the various elements, which allow to have information about shape or size of parts and can be easily enhanced. Moreover, these structures provide an invariant property with respect to geometric transformations, and can uniquely represent models regardless if translations or rotations are applied.

We propose a multi-layered descriptor called Enriched Assembly Model (EAM), which has four different conceptual layers: structure, interface, shape and statistic. Each layer characterizes assembly models at different levels of detail, where the data of each layer are illustrated in Figure 4.1.

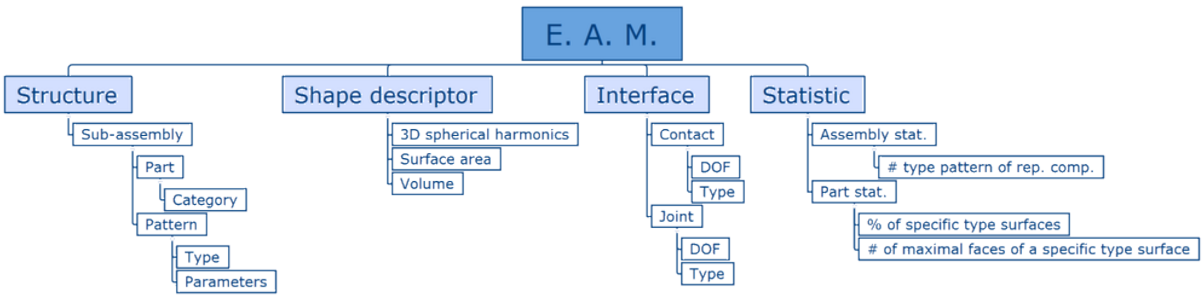


Figure 4.1: The multi-layered Enriched Assembly Model

An EAM encodes all the required data automatically extracted by analyzing the geometry and structure of a CAD model. These data are then organized in an attributed multi-graph structure that will be described in section 4.3. The EAM allows the use of multiple arcs, i.e.it is possible that two or more arcs are incident to the same pair of nodes [119]. This peculiarity is generated by the different nature of the information stored, such as the structure and the interface information, which define different type of relationships among assembly components (described in sections 4.2.1 and 4.2.2).

Using graph representation, the retrieval problem is transposed into finding in a database the CAD models, which have an EAM graph descriptor similar to the one of the model given as a query. This aspect will be discussed in section 5.1.

The following sections describe with more details the various EAM layers with their data and we will refer to these data as attributes of the EAM.

4.2.1 Structure layer

The structure layer of the EAM encodes the *product structure* (described in section 1.2.2) as specified by the designer and two attributes (*Pattern_List* and *Component_Type*), which are used to characterize the assembly components.
Let see in the following these elements.

Product structure

> Product structure
>
> **Definition 4.2.1.** The *product structure* is a hierarchical decomposition of a product (an assembly model) in terms of sub-assemblies of the CAD model up to its constituent parts.

The product structure defines the relation "made-of" between sub-assemblies and parts, which is represented in the EAM through directed arcs between nodes with the following meaning:

- the root node corresponds to the entire assembly model;
- intermediate nodes represent sub-assemblies;
- leaves are associated with the parts constituting the assembly model.

Figure 4.2 shows an example of the structure layer of an EAM. The object is an engine whose first level is divided into three sub-assemblies: a piston (S1), a crank shaft (S2), a mass (S3) and two linking parts (P3, P10). At the second level, there are the components of the sub-assemblies S1, S2 and S3.

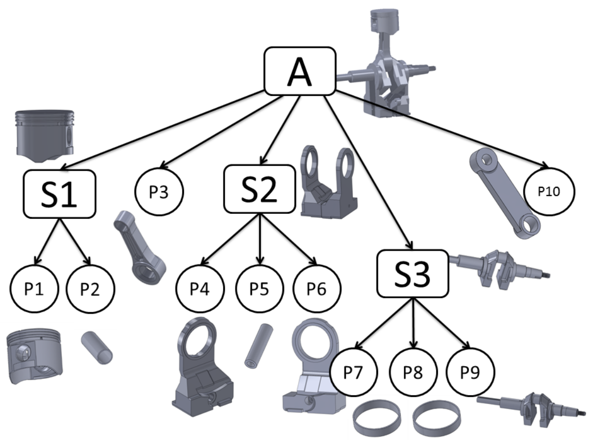


Figure 4.2: Example of structure layer of an EAM

As already discussed in section 1.2.2, this partition is not unique and reveals the designer intent, even if it is driven by some well accepted rules. For example, the assembly can be organized according to functional meaning, or in a way that the forthcoming assembly simulation steps are eased, or it can be decomposed according to the constitutive materials. Actually, designers can consider the product either as a whole object with its characteristics (e.g. volume, gravity center), or focusing on sub-assemblies (e.g. for kinematics purposes) or parts (e.g. for manufacturing issues). Anyway such decomposition into sub-assemblies is interesting from the point of view of similarity, since it corresponds to the way designers may focus on a product and then it reflects the user intent, thus it can represent a criteria of similarity for the user.

Attributes

In this layer, information on the components of an assembly model is added by the use of attributes. The considered attributes specify the patterns ($Pattern\_List$) of repeated parts present in the assembly model and the type of elements represented in the nodes of the EAM ($Component\_Type$):

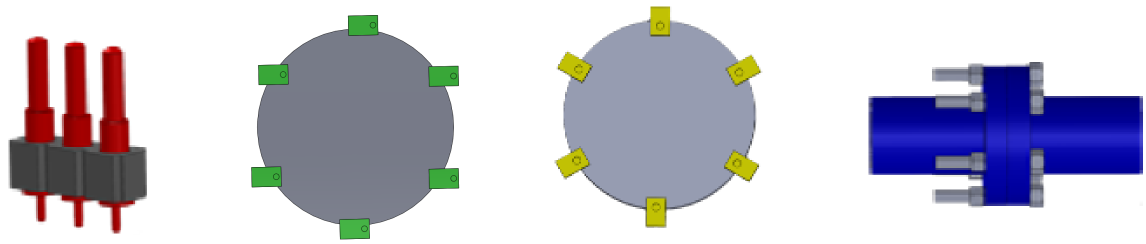
- A *pattern* is a regular arrangement of repeated components whose centers of gravity are equidistant. The attribute $Pattern\_List$ is associated to the root node, i.e. to the entire assembly model, to deal with the not uniqueness of the structural decomposition, since it provides information about mutual position of parts.
  These configurations affect production, including assembly operations. Therefore, their presence can also be used in the similarity assessment between assemblies.

  The $Pattern\_List$ attribute is characterized as:

$$\begin{aligned} Pattern\_List &= \{Pattern \times RP\} && (4.1) \\ Pattern &= Pattern\_Type \times Parameters && (4.2) \end{aligned}$$

  where RP represents the list of repeated parts.

  The considered and detected types of pattern ($Pattern\_Type$) are: **linear translation** (red pins in Figure 4.3(a)), **circular translation** (green plates in Figure 4.3(b)), **circular rotation** (yellow plates in Figure 4.3(c)) and **reflection** (blue flange in Figure 4.3(d)) [29, 76]. In case of two repeated entities, we assume that possible configurations are linear translation and reflection, since to define a circular pattern the algorithm in [76] requires at least three repeated entities. The algorithm for their detection is based on the work of Chiang et al. [29] and its generalization on assembly model [76] and the procedure for the setting of this attribute is described in section 4.4.6.



(a) Linear translation (b) Circular translation (c) Circular rotation (d) Reflection

Figure 4.3: Examples of patterns of repeated components

In order to characterize the position of repeated parts, each type of pattern has a set of parameters which also provide indication on the size of the pattern. More precisely, for each type of pattern, it is assigned the *step*, which indicates the distance between each repeated element, and the *number of repeated parts*, while the *pattern center*, *radius* and the *angle* are assigned for circular translation and rotation patterns.
Figure 4.4 shows a circular patter of screws (blue circle) with its parameters, i.e. the *step* in purple, the *radius* in green, the *center* in red and the *angle* in yellow.

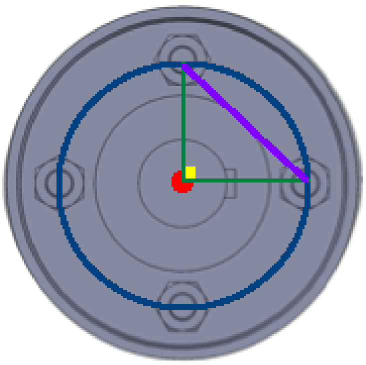


Figure 4.4: Example of attributes of a circular pattern

The set of parameters is summarized in Table 4.1.

Table 4.1: Parameter set according to the pattern type

	Step	Number of repeated parts	Center, radius, angle
Linear translation	✓	✓	
Circular translation	✓	✓	✓
Circular rotation	✓	✓	✓
Reflection	✓	(=2)	

- Products can be classified into category based on their function [58]. In order to support retrieval of assembly models according to the function of their components, we consider the *Component_Type* attribute, where it is defined as following:

$$\begin{aligned} Component\_Type \;=\; & \{\text{bearing, c-clip, cylinder like, cube like, gear, key,} \\ & \text{linkage arm, nut, parts of bearing, screw and bolt,} \\ & \text{shaft, spacer, sphere like, torus like, miscellaneous}\} \end{aligned} \tag{4.3}$$

  These categories have been selected because, to validate the envisaged approach, we initially focus on elements that are involved in speed and movement modification. Anyway, the flexibility of the structure will allow to enhance the classification with other classes, e.g. cover, housing, plate, rivet, bracket, spring or piston.
  Moreover, this classification allows to discern elements possibly corresponding to fasteners (e.g. screws, bolts, nuts and c-clips) with elements corresponding to important parts characterizing specific functional sets. This distinction enables the comparison of two assemblies by their main parts in a first step and then refining the comparison including also minor components.

  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 technological artifact which allow a specific function (e.g. power transmitter for gears and shafts, locator function for keys and spacers), as illustrated in Figure 4.5.

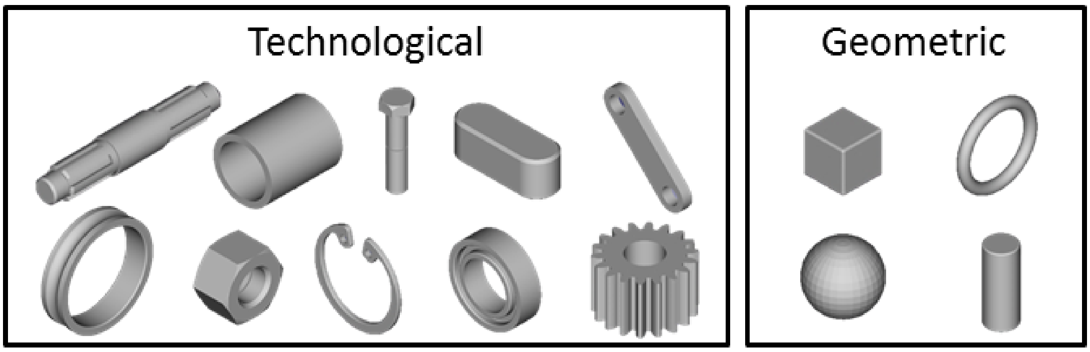


Figure 4.5: Attributes associated to nodes

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 through hole. Analogously, many mechanical parts, which are themselves assemblies and normally acquired by third parties, such as bearings, are frequently available from online catalogs and included in larger assemblies as a single component possibly with a simplified shape. This motivates the inclusion of the 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 classes.
As specified before, these attributes are not explicitly available in CAD models and thus their related values have to be derived through geometric reasoning detailed in section 4.4.10, 4.4.11 and 4.4.12.

An example of the information present in the structure layer of the EAM for an assembly model is illustrated in Figure 4.6, where the nodes, the arcs and the attributes are depicted. In particular, the label on the root node represents the list of patterns in the entire assembly model, i.e. two circular rotation patterns made of four screws and four nuts respectively; the labels on the nodes correspond to the *Component_Type* attribute; and the decomposition of the assembly in its components is represented by the arcs, i.e. each arc expresses the relationship "made-of" between the assembly components.

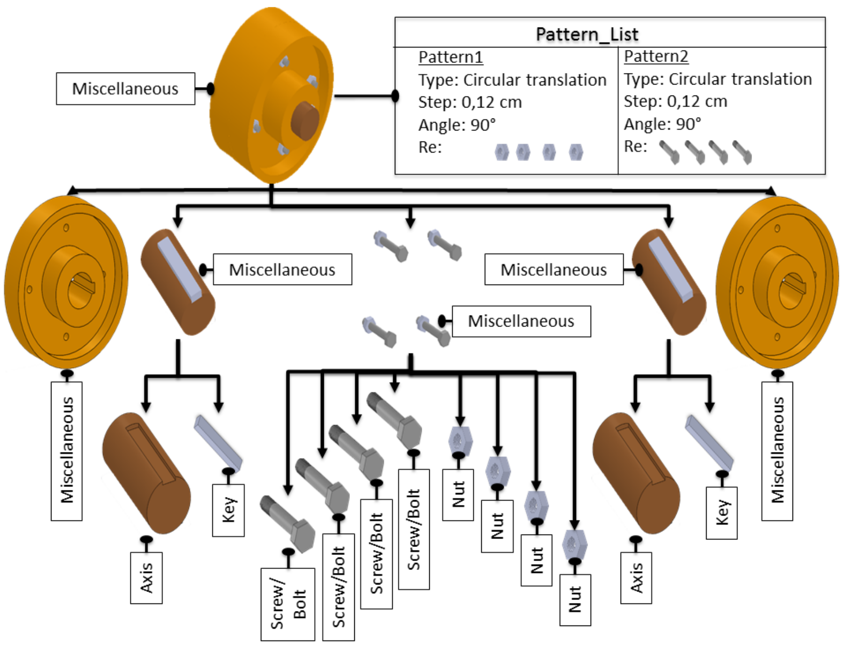


Figure 4.6: Example of structure layer of the EAM of an assembly model with its attributes

4.2.2 Interface assembly layer

The interface layer aims at describing the relationships between the different parts in assembly model regardless its structure. It is expressed at two different levels of details: the contact information and the joint information.

Contact information

Once parts are positioned, they can identify some interface, i.e. contacts, interferences and clearances, according to the following rules [110]:

- A contact between two components C_i and C_j defines one or more shared surfaces or shared curves, without any shared volume.
- A clearance occurs when a distance between two surfaces of components C_i and C_j conveys a functional meaning.
- An interference between C_i and C_j defines a shared volume between them.

As previously discussed in section 1.2.3, interfaces and some clearances do not exist between two real objects. However, such unreal configurations 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.

We aim to describe assembly models as in real configurations, i.e. without interferences and clearances, so that models, which describe the same object, can be recognized as similar even if errors occur. For this reason, we consider only contacts as relationships between parts, and we try to solve some recurring errors for interferences through geometric reasoning (see section 4.4.4 for details).
In addition, each element in contact defines a constraint which allows a certain degrees of freedom between the parts, through which a mechanical assembly can be classified [98]. Then, the *contact information* provides a detailed description of part relationships, i.e. the types of contacts between two parts and their DOF.

Thus, we can give the following definition.

Contact information

Definition 4.2.2. A *contact information* between two parts P_i and P_j is the degree of freedom generated by two entities (face, edge or vertex) between P_i and P_j and it is characterized by attributes having the following domains:

$$DOF = T \cup R \tag{4.4}$$

$$Contact\_Type = \{\text{Surface, Curve, Point, UnSolved}\} \tag{4.5}$$

$$Face\_Contact\_Type = \{\text{NoFace, Plane, Cylinder, Cone, Sphere, Torus, FreeForm}\} \tag{4.6}$$

In the definition 4.2.2, the DOF is formed by a set of possible translation T and a set of possible rotation R, both are expressed vectors $v = (x, y, z)$ with norm equals to 1 ($||(x, y, z)|| = 1$), i.e. v is a versor.

Two entities can be partially in contact, for instance, the contact between parts (gray part in Figure 4.7) is only a portion of the faces in the B-reps of the two parts, then, the contacts cannot be associated to the entities of B-reps. Hence, the *Contact_Type* attribute identifies the geometry type associated to the contact between two parts P_i and P_j, i.e. if the contact can be represented as a surface, a curve or just a point. If a volumetric interference occurs between two parts and we are not able to solve it, we assume that P_i and P_j should be in contact in their realistic configuration, then we set the attribute *Contact_Type* as "Unsolved". Details on the detection of this information are provided in section 4.4.4.

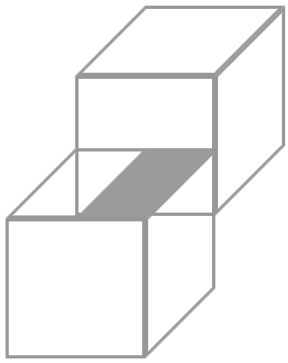


Figure 4.7: Two parts with a surface contact

One can notice that a pair of parts (P_i, P_j) may have multiple *contact information* depending on the number of contacts between them. For instance, the parts in Figure 4.8 are in contact by five different portions of faces, then they have five *contact information*.

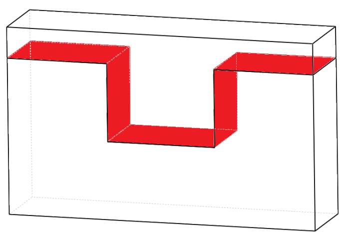


Figure 4.8: Two parts with multiple contacts

At last, we want to observe that quantifying allowed translations and rotations does not depend only on the type of contact but includes also dimension consideration, as the evaluation of the portion of contacts over the surfaces of faces in contact. For instance, both boxes in Figure 4.9 can rotate along the **n** axes, but the model in Figure 4.9(a) can perform a complete rotation of 360°, while the one in Figure 4.9(b) has just a partial rotation. Recognizing these differences requires a deeper analysis that we will not face in this thesis, limiting our detection on the directions of movements allowed by each single contacts.

(a) Complete rotation (b) Partial rotation

Figure 4.9: Models with same contacts and DOF but different allowed motions

Joint information

The mobility of components is a fundamental concept to analyze mechanisms and it is defined by the connections between parts. These connections, called joints, are made by contacts of each rigid geometric part [89] and can be classified by the number of degrees of freedom allowed [86].

Then we can give the following definition:

Joint information

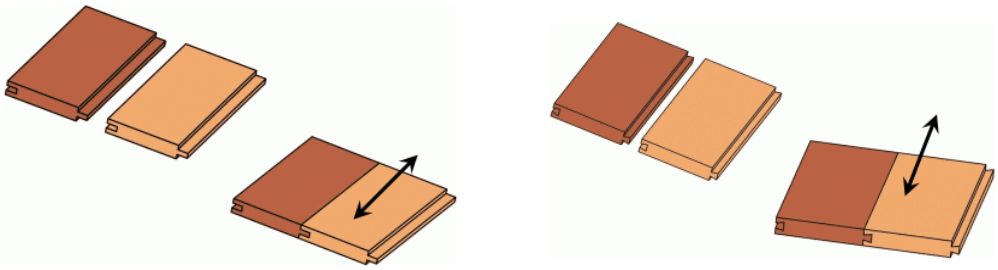
Definition 4.2.3. The *joint information* between two parts P_i and P_j is the degree of freedom induced by all the constraints imposed by the contacts between P_i and P_j and it is characterized by attributes having the following domains:

$$DOF = T \cup R \tag{4.7}$$

$$Joint\_Type = \{\text{Surface, Curve, Point, UnSolved, Mixed}\} \tag{4.8}$$

In the definition 4.2.3, the DOF of the joint between two parts is computed according to the mechanism theory [92], i.e. composing the kinematic tensors of all the contacts shared by the parts. This computation will be discussed in section 4.4.4.

Note that the allowed degrees of freedom at the joint level identifies a different situation respect the one at contact level. For instance, in Figure 4.10, two parts are connected by a tongue and groove joint as in Figure 4.10(a) and a dovetailed joint as in Figure 4.10(b). In both situations, the two parts are in contact by planar faces allowing a translation of one part respect the other. At the level of the joints these two examples are recognized as similar, since they can only translate without loosing any faces in contact. On the contrary, at the contact level, they are discerned by the normal of the faces belonging to diagonal cuts, showing the different employed technologies. Details on the detection of this information are provided in section 4.4.4.



(a) Tongue-groove joint (b) Dovetailed joint

Figure 4.10: Same joint generated by different contacts

4.2.3 Shape layer

High-level information, as kinematic pair or semantic knowledge (e.g. the type of components), is very efficient to describe assembly models. When they are not reliable or the search aims to be refined, then model shape is a rich and trustworthy information, important to discriminate assemblies.

A good shape descriptor should consider both local features and global features. Moreover, it should be transformation and representation invariant. Many shape descriptors have been defined to compare geometric models, where most of them are computed over a tessellated representation of the 3D objects [25, 57, 30, 117]. Iyer et al. highlighted that there is no unique shape descriptor which suits all the possible shapes and comparison purposes [57]. On the contrary, depending on the type of object, a specific shape descriptor can perform better than others. Indeed, from the analysis of the work of Iyer et al. [57] and Jayanti et al. [58] it turns out that spherical harmonics perform well with shapes of revolution, shape distribution with prismatic parts and inner distance well describes differences of components thickness. Considering that mechanical products are mainly formed by prismatic parts, thin-walled parts, solid of revolutions and motor bodies, in the following, we briefly illustrate the characteristic of the shape descriptors, which better characterize these parts, i.e. that have the best results in term of precision and recall in the comparison provided by Jayanti et al. [58].

- *Spherical harmonics* coefficients can be used to reconstruct an approximation of the underlying object at different levels. A spherical function can be decomposed as the sum of its harmonics as [47]:

$$f(\theta, \phi) = \sum_{l \geq 0} \sum_{|m| < l} a_{lm} Y_l^m(\theta, \phi) \tag{4.9}$$

where a_{lm} are the Fourier coefficients, i.e.

$$a_{lm} = \int_0^{2\pi} \int_0^{\pi} f(\theta, \phi) Y_l^{m*}(\theta, \phi) \sin(\theta) d\theta d\phi \tag{4.10}$$

and $Y_l^m(\theta, \phi)$ are the solutions to the normalized Laplace's equation in spherical coordinates:

$$Y_l^m(\theta, \phi) = k_{l,m} P_l^m(\cos\theta) \exp^{im\phi} \tag{4.11}$$

where $0 \leq \theta \leq \pi$, $0 \leq \phi < 2\pi$, $k_{l,m}$ is a constant, and P_l^m is the associate Legendre polynomial, and Y_l^{m*} is the complex conjugate.

Our evaluation on the adequacy of spherical harmonics as shape descriptors is based on the method proposed by Kazhdan et al. [62]. Their work decomposes a function into harmonics, sums the harmonics with their frequency and then computes the norm of each frequency component. The final result is represented by a normalized histogram, which reports the values of the sums of the harmonics for the given frequencies. In particular, in their work there are 544 bins in the histogram.

- *Shape distribution* represents the shape signature as a probability distribution sampled from a shape function measuring some geometric properties of a 3D model [87]. Typically, main measures used to compute this shape descriptor are listed below [57]:

  A3 measures the angle among three random points on the surface of a 3D model,

  D1 measures the distance between a fixed point and one random point on the surface,

  D2 measures the distance between two random points on the surface,

  D3 measures the square root of the area of the triangle formed by three random points on the surface,

  D4 measures the cubic root of the volume of the tetrahedron formed by four random points on the surface.

- The *inner distance* is defined as the length of the shortest path between landmark points within the part. In the work of Liu et al. [75] inner distances are stored as a histogram which represents a probability distribution of inner distances between all sampled point pairs on the surface.

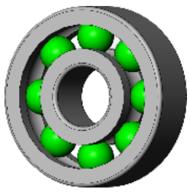
These shape descriptors are independent of translation, rotation. However, in the mechanical engineering domain, the surface area and the volume of a component seriously impact on the manufacturability of an object, then we include these numerical data under the name of *size values*, which are size dependent and are useful for part replacement and directly computable from the B-rep data.

Each presented shape descriptor is able to characterize a peculiarity of mechanical parts and for this reason they are used to classify parts according to the shape (see section 4.4.10) and the computation of these descriptors will be illustrated in section 4.4.9. Then, they are used in the shape based classification, which will be presented in section 4.4.10. Concerning the shape layer in the EAM, we decided to store only 3D spherical harmonics and size values. This decision is supported by the ambition of not affecting negatively the retrieval system introducing redundant information to be processed every time during a comparison.
Then we can say that the shape layer is defined by two attributes, the *Shape* and the *Size*, where

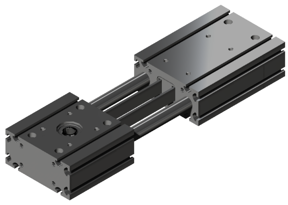
$$Shape = \mathbb{R}^{544} \tag{4.12}$$
$$Size = \mathbb{R} \times \mathbb{R} \tag{4.13}$$

4.2.4 Statistic layer

The statistic layer has been designed to ease the filtering of large datasets and reduce the number of models to be compared.
Statistics referring to the overall assembly represent the number of 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 models having similar manufacturing or assembly processes. Let us suppose a user is looking for a rolling bearing (Figure 4.11(a)) 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 Figure 4.11(b), which does not own this characteristic.



(a)



(b)

Figure 4.11: Examples of assemblies with (a) and without (b) circular pattern

Statistics for single parts include: 1) percentage of specific type surfaces (i.e. planar, cylindrical, spherical, free form, toroidal) compared to the overall area, 2) number of maximal faces (i.e. adjacent faces sharing the same underlying surface that are considered as a single face) of a specific type surface (i.e. planar, cylindrical, spherical, free form, toroidal). The use of such values allows discarding directly some shapes, thus reducing the number of candidates for the matching process.

We represent the percentage of specific type surfaces with a value in the range between 0 and 1 (i.e. $[0,1]$) and the number of maximal faces with a natural number, then we have a pair of values ($[0,1] \times \mathbb{N}$) for each type of surfaces (planar, cylindrical, spherical, free form, toroidal), i.e. five couples.
While, the assembly statistics represents the number of pattern of a specific type and we have four different pattern types, then, we can say that the statistic layer is defined by two attributes, the *Part_Statistics* and the *Assembly_Statistics*, where

$$Part\_Statistics = ([0,1] \times \mathbb{N})^5 \quad (4.14)$$

$$Assembly\_Statistics = (\mathbb{N})^4 \quad (4.15)$$

Of course, starting from this first list of statistics, additional descriptors can be used to enrich the characterization of the assembly model. For instance, the number of components of a specific type can be included in the statistic layer.

4.3 The EAM in attributed graph representation

The EAM presented in section 4.2 of a 3D CAD model is represented as an *attributed multi-graph*, where the nodes are associated with components (parts or sub-assemblies) of the CAD model and the arcs encode specific relationships between them. In order to illustrate its creation in section 4.4 and the comparison procedure by using attributed multi-graph representation in chapter 5, this section aims to formalize the graph representation and the space of the attributes of an EAM.

Let G be an attributed multi-graph representation of an EAM descriptor

$$G = \mathcal{G}(\mathcal{N}, \mathcal{A}, \Phi_{\mathcal{N}}, \Phi_{\mathcal{A}}), \tag{4.16}$$

where $\mathcal{N}$ is the set of nodes, $\mathcal{A}$ is the set of arcs and $\Phi_{\mathcal{N}}$ and $\Phi_{\mathcal{A}}$ are respectively the node and arc attribute functions.

Different type of nodes and arcs are defined, according to the different type of information extracted from the assembly model or deducted by reasoning processes. In particular, we have $\mathcal{N} = \mathcal{N}_P \cup \mathcal{N}_A$ and $\mathcal{A} = \mathcal{A}_S \cup \mathcal{A}_C \cup \mathcal{A}_J$, where

- $\mathcal{N}_P$ is the set of nodes associated with parts
- $\mathcal{N}_A$ is the set of nodes associated with sub-assemblies
- $\mathcal{A}_S$ is the set of arcs that represent the assembly hierarchical structure
- $\mathcal{A}_C$ is the set of arcs that represent the contacts between parts
- $\mathcal{A}_J$ is the set of arcs that represent the joints between parts

Following this notation, the node attribute function is defined as

$$\Phi_{\mathcal{N}}(n) = \begin{cases} \Phi_{\mathcal{N}_P}(n) \text{ if } n \in \mathcal{N}_P \\ \Phi_{\mathcal{N}_A}(n) \text{ if } n \in \mathcal{N}_A \end{cases} \tag{4.17}$$

while the arc attribute function is defined as

$$\Phi_{\mathcal{A}}(a) = \begin{cases} \Phi_{\mathcal{A}_C}(a) \text{ if } a \in \mathcal{A}_C \\ \Phi_{\mathcal{A}_J}(a) \text{ if } a \in \mathcal{A}_J \end{cases} \tag{4.18}$$

where each function is defined as follow:

$$\Phi_{\mathcal{N}_P} : \mathcal{N}_P \longrightarrow T_{\mathcal{N}_P} \tag{4.19}$$
$$\Phi_{\mathcal{N}_A} : \mathcal{N}_A \longrightarrow T_{\mathcal{N}_A} \tag{4.20}$$
$$\Phi_{\mathcal{A}_C} : \mathcal{A}_C \longrightarrow T_{\mathcal{A}_C} \tag{4.21}$$
$$\Phi_{\mathcal{A}_J} : \mathcal{A}_J \longrightarrow T_{\mathcal{A}_J} \tag{4.22}$$

The following sections will provide details about the definitions of these functions, while their range is detailed below.

The range of these functions depends on the attributed associated with every node and arc. For each part, the considered attributes are the shape descriptor, the size, the component type, the pattern and the statistics for parts, hence

$$T_{\mathcal{N}_P} \quad = \quad Shape \times Size \times Component\_Type \times Pattern \times Part\_Statistics \qquad (4.23)$$

As described in section 4.2.3, spherical harmonics are the adopted shape descriptor represented with an histogram of 544 bins; the size values indicates the volume and the surface area of a part; the component type can be one of those illustrated in Figure 4.5 and explained in section 4.2.1; patterns are described by the type (i.e. linear translation, circular translation, circular rotation and reflection) and parameters whose type and number depend from the type of pattern (see Table 4.1); the part statistics refer to percentages of specific type of surfaces expressed as values between 0 and 1 and to the number of maximal faces expressed as natural values.

In case of node associated with a sub-assembly, the considered attributes encode the patterns present in the sub-assembly as list of repeated parts which form the pattern ($RP \subseteq \mathcal{N}_P$) and the characterization of the pattern. Moreover, attributes encodes assembly statistics, which are the numbers of patterns of repeated components of a specific type (e.g. linear patterns, circular translation, circular rotation and reflection). It follows that:

$$T_{\mathcal{N}_A} \quad = \quad Pattern\_List \times Component\_Type \times Assembly\_Statistics \qquad (4.24)$$

Concerning arcs, arcs representing assembly structure have no attribute (i.e. $T_{\mathcal{A}_S} = \emptyset$) while arcs that represent contacts and joints between parts have attributes that specify the permitted motion by their DOF.
In addition, in case of arcs representing contacts, we associate the *Component_Type* attribute, which indicates the type of contact, i.e. Surface, Curve, Point or UnSolved (in case of volumetric intersection), and, if the type is a surface, we indicate also the surface type by the *Face_Contact_Type* attribute. While in case of arcs representing joins, in addition to the DOF, we associate the *Joint_Type* attribute, which indicates the type of contacts that have originated the joints.
Thus, for arcs, the attributes sets are defined as follow:

$$T_{\mathcal{A}_C} \quad = \quad DOF \times Contact\_Type \times Face\_Contact\_Type \qquad (4.25)$$

$$T_{\mathcal{A}_J} \quad = \quad DOF \times Joint\_Type \qquad (4.26)$$

4.4 Creation of the Extended Assembly Model descriptor

CAD files do not contain all the required information to create their corresponding EAM descriptors, thus we perform reasoning processes on assembly models to extract the required information described in section 4.2.

CAD systems use different and proprietary file formats to store CAD models and associated knowledge. The file content strongly depends on the type of functionalities provided by specific CAD systems, therefore a generic retrieval system should not rely on 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) to read the assembly models and to get access to the associated information. Theoretically, this standard supports the representation of assembly models as well as the kinematic relationships between their components and their constraints. However, we have seen in section 1.4, that CAD systems sometime generate STEP files that do not incorporate the kinematic relationships and constraints.

Similarly, other information used all along the PDP may not be stored, or can be inaccurate. Consequently, our approach only relies on geometric data as well as on the hierarchical assembly structure of the CAD models, which are preserved in the STEP files. Moreover, since we aim to characterize assembly models also from the functional point of view and there may be numerous mechanisms in assemblies, in this thesis, we focus on the functional sets for motion transformation detailed in section 4.4.1.

4.4.1 The considered functional sets

Among the various functional sets, we currently focus on those whose function is to alter the amount of speed and torque generated by an input source and/or change the direction of the output source. In particular, we focus on speed reducers. They are present in a wide variety of mechanical systems, e.g. beam pumps, wind turbine and clutch transmission. In many cases, a single unit is not sufficient to reach the right variation. Hence, several units are present simultaneously and configured in different arrangements.

Most of the speed reducers are composed by a set of gears with different radius assembled with shafts. Bearings are used to support various loads so that the gears can be held in the proper alignment. Such components are also referred to as gear boxes, speed increasers, and gear reducers.
Shafts connect the speed reducer with input/output sources. Typically, their surface is smooth where bearings are located and with limited groves to install gears; if carving is not present, some slots are predisposed for inserting key parts, which prevent gear rotation.

Thanks to the ability of supporting radial load with possibility of low axial load and the suitability for high rotation speed, the most widespread kind of bearing employed in the considered mechanism are the groove ball bearings. They usually consist of an inner ring, outer ring, and rolling elements (balls or rollers) arranged in a circular pattern, and a cage, which holds the rolling elements at fixed intervals between the ring raceways (Figure 4.12). Tapered roller bearings are less common but possible.
When it is not suitable to fasten two parts by shrink-fitted, auxiliary parts are inserted to secure their position, e.g. spacers, c-clips and keys. Last, to ensure oil-tight, seals are present at the extremity of driven shaft.

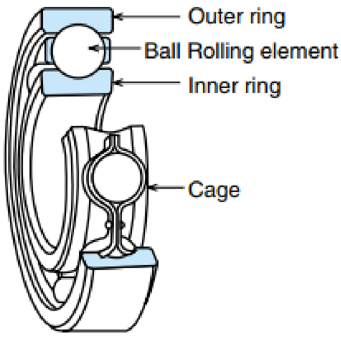


Figure 4.12: Bearing component and its constituting elements.

4.4.2 Enriched Assembly Model creation

The creation of the EAM results from the composition of several modules. Some of them can run in parallel, while others need the output of previous computations. The dependences of all the modules are illustrated in Figure 4.13.

The EAM creation starts reading a CAD model in STEP file format. Nodes and arcs of the EAM structural layer are created by the **structure creation** module detailed in section 4.4.3. The module **component relationship detection** analyzes the interactions between the parts to extract contacts and identify joints according to the steps (described in section 4.4.4) and, at the same time, for each created leaf, the part statistics are computed through the **part statistics computation** module (described in section 4.4.5). Repeated components and their regular patterns are detected by using the **repeated component pattern detection** process (described in section 4.4.6). Then, numerical values linked with the entire assembly are stored as attributes by the **assembly statistics computation** module (described in section 4.4.7).

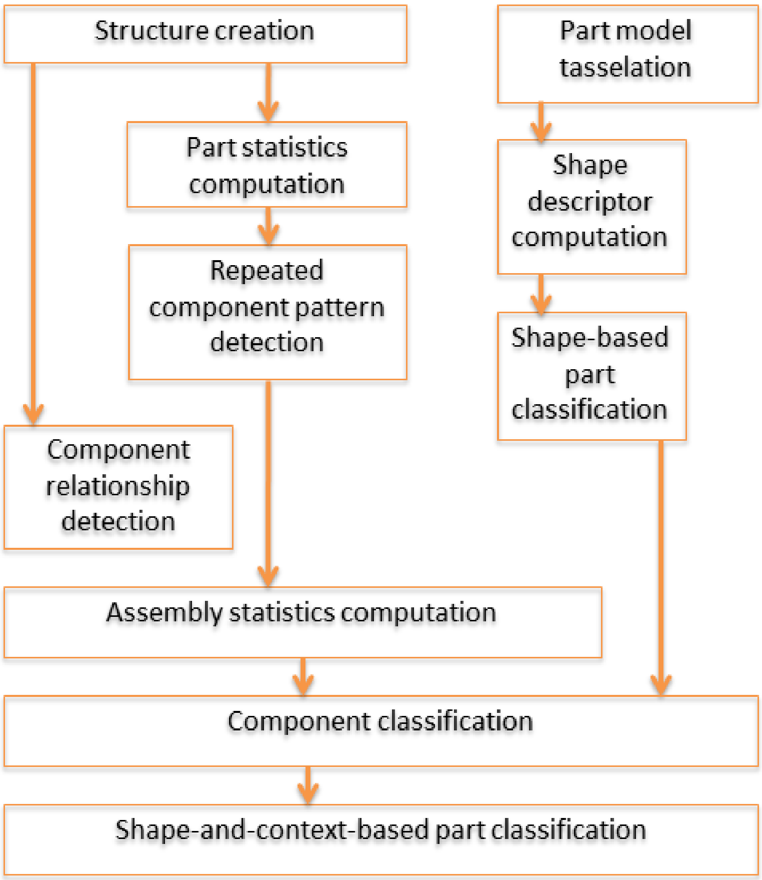


Figure 4.13: EAM creation process

For each part in the assembly, the corresponding tessellation (if not already present in the dataset) is computed by the **part model tessellation** module (described in section 4.4.8). This representation is necessary to compute some of the associated shape descriptors that are not directly computable from the B-rep representation but require a mesh representation. These operations are performed by the **shape descriptor computation** module using existing and available procedures [62, 87, 75]. Details are provided in section 4.4.9.

The shape descriptors are then the inputs for the **shape-based part classification** module, which associates a category to each assembly component (described in section 4.4.10). This classification aims at reducing the number of comparisons between parts for the shape similarity assessment, to possibly discard negligible parts (such as fastener) during the matching process and to better support the formulation of abstract queries. In the end, in order to overcome issues risen from simplified shapes of parts, the **component classification** and **shape-and-context-based part classification** modules exploit engineering knowledge to improve the classification. These modules are then described in section 4.4.11 and 4.4.12 respectively.

4.4.3 Module "Structure creation"

The procedure to detect the hierarchical structure of an assembly model traverses the assembly and creates the set of nodes associated with parts ($\mathcal{N}_P$) and sub-assemblies ($\mathcal{N}_A$) and the set of arcs related to the structure ($\mathcal{A}_S$).
It is an iterative procedure that for each sub-assemblies in the model reads its sub-components, i.e. its "children". As shown in Algorithm 1, the procedure takes as input the root of the assembly model *RootA*, creates its corresponding assembly node and initializes a queue with the children of the root of the assembly model. Then, until the queue is not empty, a *component* is extracted from the queue; if it has children, then it represents a sub-assembly, otherwise it is a part. In both the cases, corresponding sub-assembly or part nodes are created. In addition, an arc of structure type is created between each component and its children.

Algorithm 1 The structure creation algorithm

procedure TRAVERSEASSEMBLYMODEL(RootA)
    n_{Root} = New SubAssembly(RootA)
    $\mathcal{N}_A \leftarrow n_{Root}$
    $Q \leftarrow$ RootA.getChildren

    while $Q \neq \emptyset$ **do**
        component = GetFirstElementIn(Q)
        Remove component from Q
        n_f = GetFatherNodeOf(component)
        if component.getChildrenNumber $\neq 0$ **then**
            $Q \leftarrow$ component.getChildren
            n_c = New SubAssembly(component)
            $\mathcal{N}_A \leftarrow n_c$
        else
            n_c = New Part(component)
            $\mathcal{N}_P \leftarrow n_c$
        $\mathcal{A}_S \leftarrow (n_f, n_c)$
    Return $\mathcal{N}_P, \mathcal{N}_A, \mathcal{A}_S$

4.4.4 Module "Component relationship detection"

The procedure to compute the interface layer generates the arcs representing the contacts and the joints between parts ($\mathcal{A}_C$ and $\mathcal{A}_J$) and their set of attributes ($T_{\mathcal{A}_C}$ and $T_{\mathcal{A}_J}$). It takes in input the B-Rep model and relies on the following hypotheses:

(i) models include only rigid parts;

(ii) models should not have unrealistic configurations, i.e. volumetric interferences and clearances;

(iii) contacts do not change over time;

(iv) faces involved in the contact are mainly associated to analytic surfaces.

Condition (i) applies to mechanical parts, i.e. the type of objects we are considering, and allows the use of kinematic theory of rigid body, which does not consider deformation. Condition (ii) is generally true for product models ready to be manufactured and assembled, even if, because of approximation errors or other issues already discussed in section 1.2.3, it is possible to find volumetric interferences and clearances in CAD assembly models, which do not exist in the corresponding physical product. In order to manage also models that present volumetric intersections, we intend to figure out their probable real configuration. So far, our investigation is limited to volumetric interferences, based on the analysis of the intersecting volume and of the type of surfaces involved in this intersection.

The third condition (iii) implies that the relative motions of two parts do not add any new contact neither remove the old one. Only the portion of the contacts may change. In general, this condition does not hold for all the types of objects, as the one illustrated in Figure 4.14. In this thesis, we manage modifications of contacts as the one illustrated in Figure 4.14(a), where the faces involved in the contacts change, but their number and type are preserved. Differently, for the object in Figure 4.14(b) the contact between the green and red part can be present or not a different times.

(a) Contacts between gears

(b) Contacts in a so-called *Maltese cross*

Figure 4.14: Two parts whose contacts change over the time

At last, the fourth condition (iv) is not too restrictive, since most of the mechanical objects are mostly made by combinations of analytic surfaces. If this condition was not hold, we can decode parts in contacts anyway. More complex procedures will be required to assess DOF of parts including free-form surfaces, which are not addressed in the work of this thesis.

The pipeline to extract contacts, compute their DOFs and the resulting motions is represented in Figure 4.15 and its steps are described in the following paragraphs.

Figure 4.15: Pipeline for the interface layer definition

Detection of entities in contact

In this phase, we identify the pairs of parts which are in contact and those arising by volumetric interferences. For these detected pairs, the DOF will be computed in the phase "assignment of DOF for each contact". These processes are used in Algorithm 2, which returns the set of arcs $\mathcal{A}_C$ and its attributes $T_{\mathcal{A}_C}$, where, the generic attribute value associated with the arc $a_{i,j} = (n_i, n_j)$ is indicated as $t_{i,j}$, while $\mathcal{A}_C^{ij}$ and T^{ij} represent respectively the set of contact arcs between the nodes n_i and n_j and the set of their attributes.

Algorithm 2 The component relationship detection algorithm

procedure PARTCONTACTSDETECTION($\mathcal{N}_P$)

    for each $n_i, n_j \in \mathcal{N}_P$
        I = GetIntersection(n_i, n_j)
        if $I \neq \emptyset$ **then**
            if Volume(I) == 0 **then**
                $[\mathcal{A}_C^{ij}, T^{ij}]$ = GetContactArcsAndAttributes(I, n_i, n_j)
                $\mathcal{A}_C \leftarrow \mathcal{A}_C^{ij}$
                $T_{\mathcal{A}_C} \leftarrow T^{ij}$
            else
                entity = TryToSolveVolumetricIntersection(I, n_i, n_j)
                $\mathcal{A}_C \leftarrow a_{i,j} = (n_i, n_j)$
                $T_{\mathcal{A}_C} \leftarrow t_{i,j}$ = GetAttributesOfVolumetricIntersection(entity, n_i, n_j)

    Return $\mathcal{A}_C$ and $T_{\mathcal{A}_C}$

In general, two parts can have more than one contact, as we have seen in Figure 4.8, then to verify if two parts are in contact, we compute the intersection between each pair of parts of the assembly models (procedure *GetIntersection*). This intersection represents a set of entities (as faces, edges and vertices) that could not belong to the B-rep of the CAD model processed. Moreover, it is not regularized, i.e. it is not guaranteed that the intersection is a manifold body allowing to detect also partial contacts as in Figure 4.16, where the contact between parts (grey part) is only a portion of the face in the B-rep of the two parts.

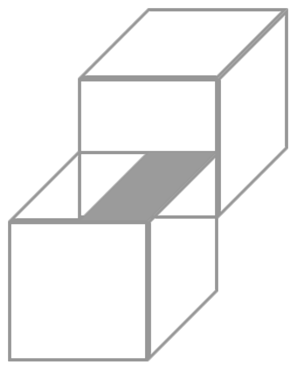
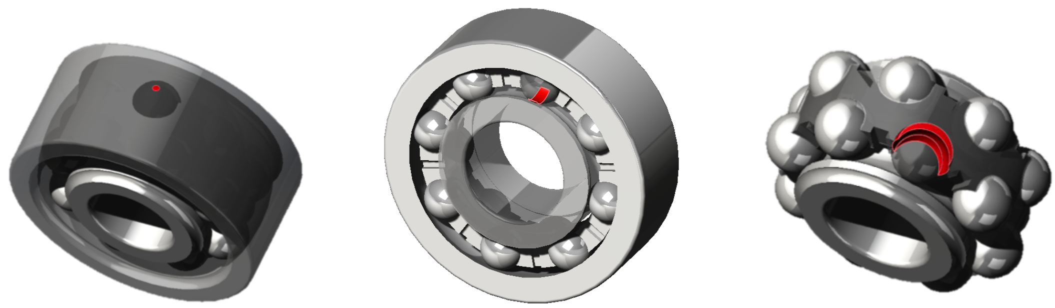


Figure 4.16: Two parts with a partial contact

If the intersection is not empty but its volume is null, then the two parts are in contact and their contact arcs and the set of attributes are computed by the procedure *GetContactArcsAndAttributes* detailed by the algorithms 3. Otherwise they form an interference, thus, their original configuration should be deduced (procedure *TryToSolveVolumetricIntersection* used in Algorithm 2) and then their DOF can be computed by the procedure *GetAttributesOfVolumetricIntersection* detailed by the algorithms 4.

The analysis of volumetric interferences is more awkward and so far, we have faced the intersection of a spherical part S with a generic part P (procedure *TryToSolveVolumetricIntersection*). Figure 4.17 reports examples of intersections with a spherical part, where in Figure 4.17(a) the contact should be a vertex, an edge in Figure 4.17(b) and a face contact in Figure 4.17(c).



(a) (b) (c)

Figure 4.17: Examples of volumetric intersections of spherical parts

Let $Int(S, P)$ be the resulting intersection between a spherical part S and another part P, then

- $Int(S, P)$ is resolved as **surface** if $\exists$ two faces in $Int(S, P)$ whose surfaces are two spheres with the same radius of S.
- $Int(S, P)$ is resolved as **curve** if $\exists$ a face in $Int(S, P)$ whose surface is a torus or a cylinder with the same radius of S.
- $Int(S, P)$ is resolved as **point** if $\exists$ a face in $Int(S, P)$ whose surface is a torus or a cylinder with radius different from the one of S.

These considerations try to solve some volumetric intersections arisen by wrong arrangement of parts and they do not consider the case of inaccurate tolerance size, as for instance, the intersection in Figure 4.17(c) caused by the fact that the radius of the sphere part S is bigger than the radius of the spherical surface of the part P.

Clearly, the problem of handling volumetric intersections is still an open issue, and we did a first effort trying to interpret the desired configuration, in order to improve the ability of the retrieval system and offering a system able of managing errors automatically without human corrections.

Once identified the elements in contact between two parts, we need to determine their DOF. This operation is discussed in the next paragraph.

Assignment of DOF for each contact

At this stage, we know the pairs of parts which arise interface and we need to compute their DOFs. This procedure depends on the kind of interface defined by the pairs of parts, i.e. contacts or volumetric intersection.
In the following, let (n_i, n_j) be the processed pair of nodes associated with the parts P_i and P_j in the assembly model.

In case P_i and P_j define a contact interface, for each face belonging to the interface, its corresponding DOF is assigned according to Table 4.2 by using the parameters which define the underling surface of the face, where R indicates a rotation, T a translation, the subscripts u, v and n the vector along which the rotations/translations are allowed. Here, R_{u+O} corresponds to a rotation along the vector formed by the directional versor u applied in the point O. These parameters will be set by the function *ParametersOf* in Algorithm 3 and Algorithm 4.

Table 4.2: DOF values according to the face contact type

Face contact 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}

For the DOF understanding in case of curve and point contacts more computations are necessary. Indeed, in this case, from the regular intersection we get just a line or a point (with their parameter) in the 3D space of the assembly model. This information is not sufficient to determinate which motions are blocked and which are free. For instance, considering the edge contact in Figure 4.18, the two parts can translate along x and y axes and rotate along y and z axes. To deduce this information we need to know the parameters of the surfaces that generate the contacts, in this example the normal of the planar face.

(a) 3D view (b) Projection view

Figure 4.18: Two view of a linear contact

This detection requires to project the curve obtained from the non-regular intersection on each faces of the parts involved in the contact to identify which faces produce the contact.
The complexity of this operation is rather high, and considering the frequency of only a linear and a vertex contact in mechanical objects, we opted for the assignment of DOF just in case of face contact.

These steps are illustrated in Algorithm 3, which takes as input two nodes n_i and n_j and their intersection I and returns the set of contact arcs for the pair of nodes n_i and n_j ($\mathcal{A}_C^{ij}$) and their set of attributes (T^{ij}).

Algorithm 3 The component relationship detection algorithm

procedure GetContactArcsAndAttributes(I, n_i, n_j)

  $\mathcal{A}_C^{ij}$ set of arcs
  T^{ij} set of attributes

  if $\exists$f $\in I$ **then**
    for each f $\in$ I
      $\mathcal{A}_C^{ij} \leftarrow a_{i,j} = (n_i, n_j)$
      $T^{ij} \leftarrow t_{i,j} =$(ParametersOf(f), "Surface", GetTypeOf(f))

  else if $\exists e \in I$ **then**
    $\mathcal{A}_C^{ij} \leftarrow a_{i,j} = (n_i, n_j)$
    $T^{ij} \leftarrow t_{i,j} =$($\emptyset$, "Curve", "NoFace")

  else if $\exists v \in I$ **then**
    $\mathcal{A}_C^{ij} \leftarrow a_{i,j} = (n_i, n_j)$
    $T^{ij} \leftarrow t_{i,j} =$($\emptyset$, "Point", "NoFace")

  Return [$\mathcal{A}_C^{ij}$, T^{ij}]

In case P_i and P_j define as interface a volumetric intersection, we exploit the results of the procedure *TryToSolveVolumetricIntersection* as illustrated in Algorithm 4, which returns the attributes for the contact arc defined by the nodes n_i and n_j when P_i and P_j define a volumetric intersection.
As in the case of contact interface, the DOF is assigned for configurations which originate a surface contact, while for curve and point contacts no DOF is assigned.

Algorithm 4 The component relationship detection algorithm

procedure GetAttributesOfVolumetricIntersection(entity, n_i, n_j)

    if entity == "Surface" **then**
        Return $t_{i,j}$ = (ParametersOf(entity), "Surface", "Sphere")
    if entity == "Curve" **then**
        Return $t_{i,j}$ = ($\emptyset$, "Curve", "NoSurface")
    if entity == "Point" **then**
        Return $t_{i,j}$ = ($\emptyset$, "Point", "NoSurface")
    else
        Return $t_{i,j}$ = ($\emptyset$, "UnSolved", "NoSurface"))

Computation of relative motion

We have detected the contacts between two parts and the DOF for each contact formed by a surface, now we need to combine these contacts in order to get the allowed motion between two parts.

According to the mechanical analysis [92], the contacts between two parts form a parallel kinematic chain and the DOF resulting from all the contacts can be computed composing the kinematic tensors of all the contacts.

Figure 4.19(a) shows an example of an assembly model while Figure 4.19(b) depicts its kinematic chain in the first row and its reduction in the second row. Contacts in the zoomed portion of Figure 4.19(a) are formed by two planar and one cylindrical faces, while the other junctions in the assembly involve only a cylindrical contacts. For this reason, the first kinematic chain has several arcs between some pairs of nodes.

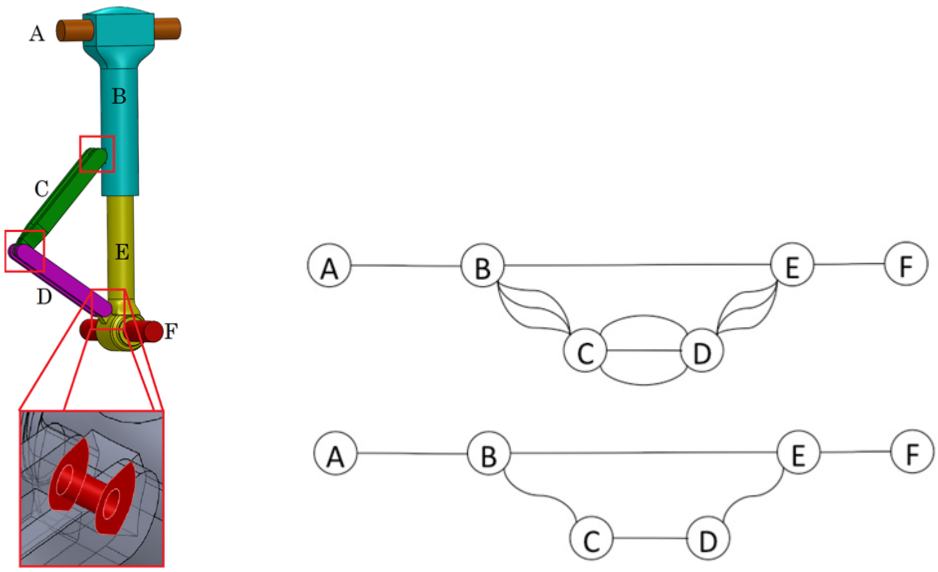


Figure 4.19: Assembly model example (a) and the reduction of its kinematic chain (b)

Focusing on the arcs between nodes B and C in the first row of Figure 4.19(b), the two planar and the cylindrical contacts are represented by the kinematic tensors expressed in equations 4.27, 4.28 and 4.29, where O_i are the center of the contacts between the parts.

$$\left\{T_{D/E}\right\}_1 = \left\{\begin{array}{c|c} 0 & v_u \\ 0 & v_v \\ w_n & 0 \end{array}\right\}_{O_1} \tag{4.27}$$

$$\left\{T_{D/E}\right\}_2 = \left\{\begin{array}{c|c} 0 & v_u \\ 0 & v_v \\ w_n & 0 \end{array}\right\}_{O_2} \tag{4.28}$$

$$\left\{T_{D/E}\right\}_3 = \left\{\begin{array}{c|c} 0 & 0 \\ 0 & 0 \\ w_n & v_n \end{array}\right\}_{O_3} \tag{4.29}$$

Then, the three kinematic tensors can be applied at the same point, and since the contacts form a parallel kinematic chain, their final kinematic tensor is expressed in the equation 4.30.

$$T_{D/E} = \left\{\begin{array}{c|c} 0 & 0 \\ 0 & 0 \\ w_n & 0 \end{array}\right\}_{O} \tag{4.30}$$

where O is the center of the resulting joint; a rotation in this case.

This example shows that to prevent a motion at the joint level, it is sufficient that a single contact does not allow it, i.e. to have a zero in the final kinematic tensor it is sufficient having a zero in same position in one kinematic tensor of the contacts. This characteristic is called *zero property* and it will be used to compute the DOF of the joint starting from contacts. Here, it is important to stress that this analysis has to be performed while considering the tensors applied at the same points.

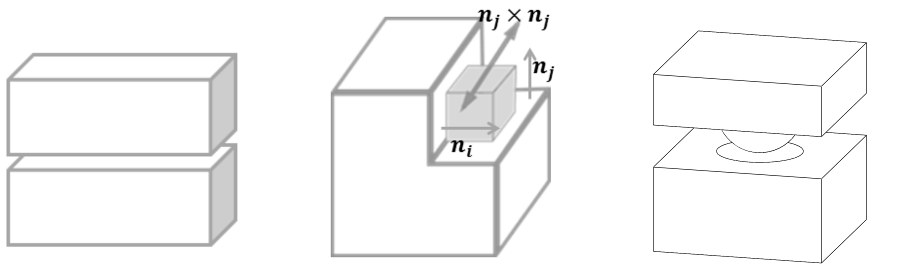
The procedure to compute the motion derived from all the contacts between two nodes n_i and n_j associated with the parts P_i and P_j in the assembly model is illustrated in Algorithm 5. This procedure returns the set of joint arcs $\mathcal{A}_J$ and the set of its attributes $T_{\mathcal{A}_J}$, where $t_{i,j}$ represents the attribute for the arc $a_{i,j} = (n_i, n_j)$.

For each pair of nodes (n_i, n_j) their contacts are read by the procedure *GetContactsBetween*, then the motion that they allow and the type of joint, i.e. the DOF and *Joint_Type* attributes, are computed by the *RelativeMotionComputation* procedure. It is based on the type and number of contacts, in particular, we have to distinguish the following cases:

(i) **There is only one contact arc between the nodes n_i and n_j** (Figure 4.20(a)). In this case, the DOF and the *Joint_Type* attributes of the joint arc are the same of the contact arc.

(ii) **The contact arcs between the nodes n_i and n_j are associated only with planar surfaces** (Figure 4.20(b)).
In this case, each planar face prevents a translation along its normal, thus between two planar faces non-parallel or coplanar with normal n_i and n_j respectively, only one translation is allowed, that is along the resulting vector $n_i \times n_j$. Since the joint arc arise by only planar contacts, the *Joint_Type* attribute will be *Planar*.

(iii) **The contact arcs between the nodes n_i and n_j are associated with different types of surfaces** (Figure 4.20(c)).
In this case, the joint resulting from the contacts between the parts is computed using the zero-property previously mentioned. Thus, only the DOFs allowed by **all** the contacts are permitted in the final motion. In this case, if the joint arc arise by contacts of the same type, its *Joint_Type* attribute will be inherited, otherwise it will be set up to *Mixed*.



(a) Single face (b) Only planar faces (c) Multiple faces of different types

Figure 4.20: Example of combinations of contacts

Algorithm 5 The component relationship detection algorithm

procedure ComputeJointLevel($\mathcal{N}_P$)

  for each $n_i \in \mathcal{N}_P$
    for each $n_j \in \mathcal{N}_P$
      contactList = GetContactsBetween(n_i, n_j)
      $t_{i,j}$ = RelativeMotionComputation(contactList)
      $\mathcal{A}_J \leftarrow a_{i,j} = (n_i, n_j)$
      $T_{\mathcal{A}_J} \leftarrow t_{i,j}$

  Return $\mathcal{A}_J$ and $T_{\mathcal{A}_J}$

4.4.5 Module "Part statistics computation"

The algorithm takes as input a node $n \in \mathcal{N}_P$ associated with a part P and returns $t_n \in Part\_Statistics$, i.e. a value of the set introduced in section 4.2.4. In Algorithm 6, plaP, cylP, conP, sphP, torP, ffP indicate respectively the percentage over the overall area of the faces of type planar, cylindrical, conical, spherical, toroidal and free form; while plaN, cylN, conN, sphN, torN, ffN indicate respectively the number of maximal faces of type planar, cylindrical, conical, spherical, toroidal and free form. Hence, this procedure represents the values of the attribute function $\Phi_{\mathcal{N}_P}(n)_{\restriction Part\_Statistics}$.

Algorithm 6 The part statistics computation algorithm

procedure GetPartStatistics(n)

  $t_n = ((plaP, plaN), (cylP, cylN), (conP, conN), (sphP, sphN), (torP, torN), (ffP, ffN)) = ((0.0, 0), (0.0, 0), (0.0, 0), (0.0, 0), (0.0, 0), (0.0, 0))$

  P = GetPartOf(n)
  faceList = GetFacesOf(P)

  for each face *in* faceList
    surfaceType = GetSurfaceTypeOf(face)
    area = GetAreaOf(face)

    switch surfaceType **do**
      case Plane: plaP += GetArea(face) and plaN = plaN +1
      case Cylinder: cylP += area and cylN = cylN +1
      case Cone: conP += area and conN = conN +1
      case Sphere: sphP += area and sphN = sphN +1
      case Torus: torP += area and torN = torN +1
      case Default ffP += area and ffN = ffN +1
  S = GetSurfaceAreaOf(compSurface)

  $t_n = ((\frac{plaP}{S}, plaN), (\frac{cylP}{S}, cylN), (\frac{conP}{S}, conN), (\frac{sphP}{S}, sphN), (\frac{torP}{S}, torN), (\frac{ffP}{S}, ffN))$

  Return t_n

4.4.6 Module "Repeated component pattern detection"

The procedure for the detection of regular patterns of repeated components is based on the work of Chiang et al. [29] and our generalization on assembly model [76].

The aim of this module is to define the attribute *Pattern_List* for the root node of an EAM. First of all, the set of repeated instances is identified, where in an assembly model a part is considered repeated if it is a different instance of the same object. Anyway, sometimes same parts are instantiated as different objects, to overcome this issue, we consider parts having identical values of volume and surface area as repetition of the same object. Then, for each list of repeated parts (ri), the center of gravity of each part is computed. The set of centers of gravity represents the input for the procedure defined by Chiang et al. [29], which identifies all the possible paths formed by the centroids. When a path of centroids satisfies a certain transformation (linear translation, circular translation, circular rotation or reflection), then a possible pattern (with its type and parameters) is identified and we need to verify that the entities of the corresponding repeated parts satisfy the same transformation, i.e. if they are arranged as specified by the detected possible pattern. This check verifies the correct orientation of the corresponding repeated parts and avoids to recognize configurations where the centroids are arranged in a pattern, but not their corresponding parts, as depicted in Figure 4.21.

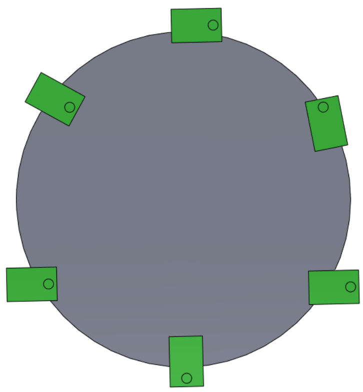


Figure 4.21: Repeated parts whose centers of gravity are aligned

We report the main steps of the method in Algorithm 7, which derives the values of the attribute function $\Phi_{\mathcal{N}_A}(n)_{\restriction Pattern\_List}$

Algorithm 7 The repeated component pattern detection algorithm

procedure $\textsc{ComputePatternArrangement}(\mathcal{N}_P)$

  $RepeatedInstances \leftarrow \text{ComputeInstances}(\mathcal{N}_P)$

  for each $ri \in RepeatedInstances$

    $GC \leftarrow \text{ComputeGravityCenter}(ri)$

    $(Pattern, RP) \leftarrow \text{ComputePossiblePattern}(GC)$

    $P \leftarrow \text{VerifyPositioning}((Pattern, RP))$

    $Pattern\_List \leftarrow P$

  Return $Pattern\_List$

4.4.7 Module "Assembly statistics computation"

Once most of the data necessary for the EAM creation are extracted, assembly statistics can be computed and associated as attributes to the root node in $\mathcal{N}_A$.
The data present in the assembly statistics regard the number of pattern for each type, i.e. linear translation, circular translation, circular rotation and reflection. Then, the procedure in Algorithm 8 to compute the assembly statistics takes as input the list of patterns detected by Algorithm 7 and counts the number of patterns for each type, where linTra, cirTra, cirRot and ref indicate respectively the number of linear translation, circular translation, circular rotation and reflection patterns.

Hence, this procedure assigns the values *Assembly_Statistics* to the attribute function $\Phi_{\mathcal{N}_A}(n)_{\restriction Assembly\_Statistics}$.

Algorithm 8 The assembly statistics computation algorithm

procedure GETASSEMBLYSTATISTIC(*Pattern_List*)

$Assembly\_Statistics =$
$(linTra, cirTra, cirTor, ref) =$
$(0, 0, 0, 0)$

for each pattern *in Pattern_List*
patternType = GetTypeOf(pattern)

switch patternType **do**
case linear translation: linTra = linTra +1
case circular translation: cirTra = cirTra +1
case circular rotation: cirTor = cirTor +1
case reflection: ref = ref +1
Return *Assembly_Statistics*

4.4.8 Module "Part model tessellation"

The EAM includes a classification of the parts of an assembly model that is based on the shape descriptors described in section 4.2.3. The computation of the shape descriptors (illustrated in section 4.4.9) is not based on B-rep representation, thus it requires to get a mesh representation of the parts which form an assembly model. For this operation we rely on functionalities present in most of the existing CAD systems. This kind of tools allow selecting a set of options, such as those illustrated in Figure 4.22. In our case, we set the deviation to 0.40 mm and the angle to 10 degree.

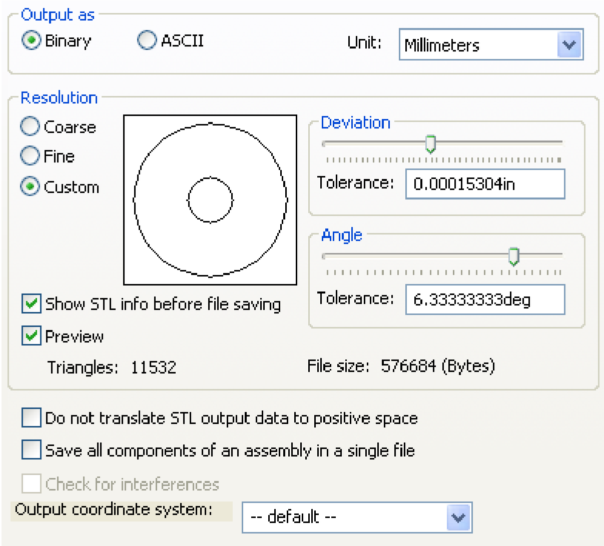


Figure 4.22: Options to tessellate a CAD model

4.4.9 Module "Shape descriptor computation"

In order to classify parts according to their shape, shape descriptors are required. As anticipated in section 4.2.3, different shape descriptors are able to capture some aspects of a model characterizing different features. We list in the following, the most suitable shape descriptors according to different classes of components as studied and explained by Jayanti [58].

- For *prismatic parts* the Light Field Descriptor [27] has considerably high precision for all recall values, while 3D spherical harmonics [62] perform well for values of recall after 0.3.
- *Thin-walled components* are characterized by higher surface areas and lower volume, then 3D shape distributions [87] and Surface Area and Volume perform better than more complex descriptors. For this class Light Field Descriptor presents some limits in their characterization.
- *Solid of revolution*s are well characterized by the view-based method (e.g. the Light Field Descriptor and 2D Shape distributions [95, 94]) and the 3D spherical harmonics are better than the D2 shape distributions.
- *Motor bodies* have comparable dimensions along the three axes. Hence the 3D Spherical Harmonics and Shape Histogram methods perform well for this category.

On the basis of these considerations and since minor variations in design features are especially challenging for view-based methods, our part classification is based on the following shape descriptors:

- 3D spherical harmonics;
- shape distribution (with D2 measure);
- inner distance;
- size values: surface area, volume;
- proportions among the minimum bounding box dimensions;

In our framework, the computation of 3D spherical harmonics, D2 shape distribution and the inner distance is demanded to available software tools, while the surface area, volume and proportions among the minimum bounding box dimensions computation is done starting from the B-rep available in the CAD models.
For the 3D spherical harmonics, we use the software tool and the default parameters as defined in [3], which approximates a shape with 64 harmonic coefficients. The same strategy is followed for the computation of shape distribution, computed by the procedure of [62], and for the inner distance [75].

At the end of this module, we store the 3D spherical harmonics as attribute of the part nodes, while the other shape descriptors are required as input of the next module.

4.4.10 Module "Shape-based part classification"

This module classifies parts according to their shape. The output of this module represents the attribute function $\Phi_{\mathcal{N}_P}(n)_{\restriction Componnet\_Type}$. The classification is based on a learning process defined by Rucco et al. [106] in which objects are described by a collection of shape descriptors considered the most suitable for mechanical objects [58]. In particular, the shape descriptors listed in section 4.4.9 are used.

In general, learning approaches strongly rely on the involved training set and our contribution in this module is represented by the definition of a proper training set. Indeed, the existing classified datasets are not fully appropriate for our purposes, as they aim at different goals, for instance the well-know benchmark of Jayanti et al. [58] classifies parts just according to shape characteristics neglecting functional aspects.

Considering our goal of classifying parts to discriminate those which might be components of mechanisms for motion transformation (as explained in section 4.4.1) from others negligible from this point of view and the fact that some parts are designed in a simplified manner, our class organization is somehow both functional and shape oriented. In particular, we enhance the "Functional Classification" of the National Design Repository [99], which consists of 70 models arranged in 7 classes, classifying 2354 elements (obtained from on-line repositories [2, 4] and engineering student's projects) in the 15 classes indicated in Figure 4.23. The validation of the class membership has been performed through interviews with domain experts, i.e. engineers and mechanical designers.

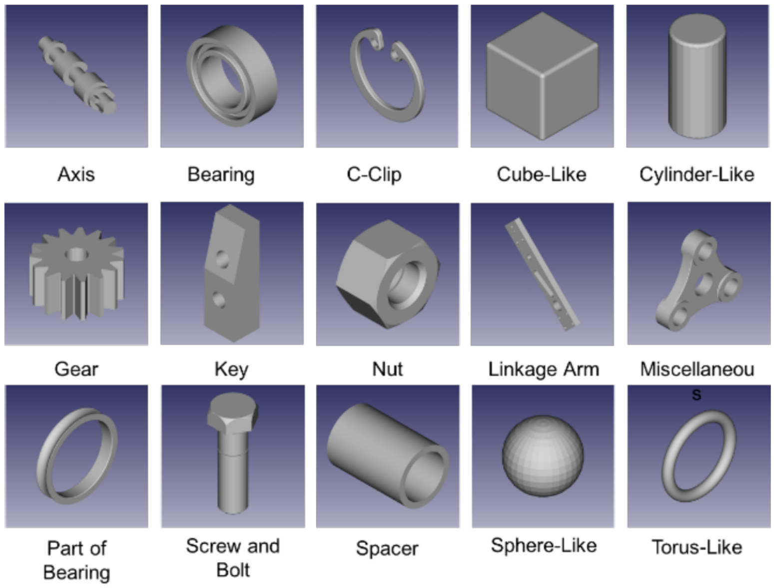
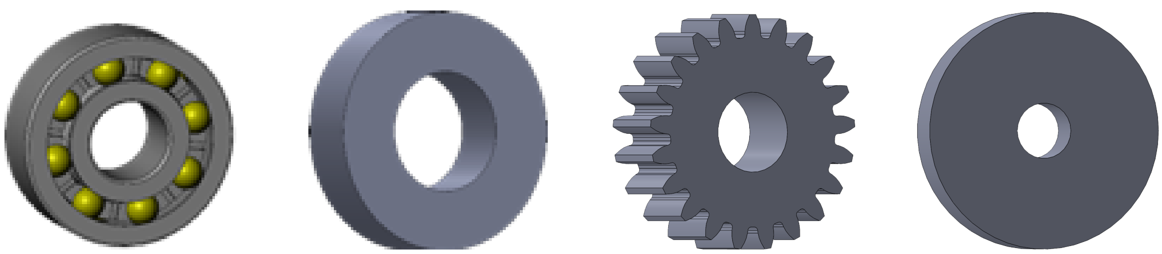


Figure 4.23: 15 classes of the training set

The training set has been organized according to the categories specified as attributes in the structural layer in section 4.2.1: bearing, c-clip, cylinder like, cube like, gear, key, linkage arm, nut, part of bearing, screw and bolt, shaft, spacer, sphere like, torus like and miscellaneous. As discussed in section 4.2.1, some of these classes are more geometry oriented (e.g. cylinder-like or torus-like) while others refer to specific types of mechanical artifacts (e.g. gear or shaft).

We have discussed in section 1.4 that the shape sometimes is not sufficient to fully characterize a component, then, when an object could correspond to different mechanical components, maybe because its shape is simplified and arduous to recognize, it is assigned to a geometry oriented class (e.g. cylinder-like instead of shaft). In addition, we classify simplified bearings as in Figure 4.24(b) and gears as in Figure 4.24(d) as a spacer.



(a) Detailed bearing (b) Simplified bearing (c) Detailed gear (d) Simplified gear

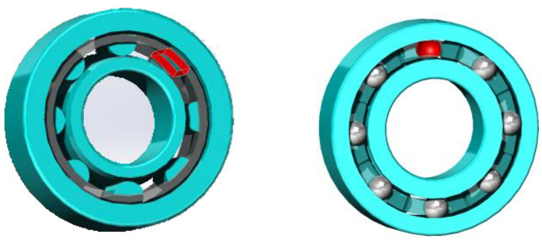
Figure 4.24: Examples of components with different shapes and functions

In this way, we want to provide a preliminary classification based solely on shape characteristic; the module **Shape-and-context-based part classification** described in section 4.4.12 will exploit the surrounding context to improve the classification.

4.4.11 Module "Component classification"

The **Component classification** module identifies assembly components that represent some functional sets characterized by the presence of specific types of components generally positioned according to some rules. The output of this module represents the attribute function $\Phi_{\mathcal{N}_A}(n)_{\upharpoonright Componnet\_Type}$.
This step exploits the shape-based part classification (section 4.4.10) and the relationships between parts automatically extracted from the assembly model through geometric reasoning (section 4.4.4) and explicitly stored in the Enriched Assembly Model (EAM) [43].
Since the EAM is represented by a graph structure, the identification algorithm is performed by graph matching, where subgraphs representing assembly components are compared with graphs representing predefined templates characterizing the mechanical components to be identified.

So far, we focused on bearing components. As described in section 4.4.1, they usually consist of an inner ring, outer ring, and rolling elements (balls or rollers) arranged in a circular pattern, and a cage, which holds the rolling elements at fixed intervals between the ring raceways. In the template definition, we do not exploit the cage part. This decision was led by the fact that its shape is very variable as the contacts which it rises with the rolling elements. An example of this situation is illustrated in Figure 4.25, where the cage has four different face contacts with the cylinder part (Figure 4.25(a)) and one face contact with sphere part (Figure 4.25(b)). This is a just an example, but there exists other configurations that may involve also curve contacts. Considering the variation of possible configurations with the cage, we decided to exclude this part in the template definition.

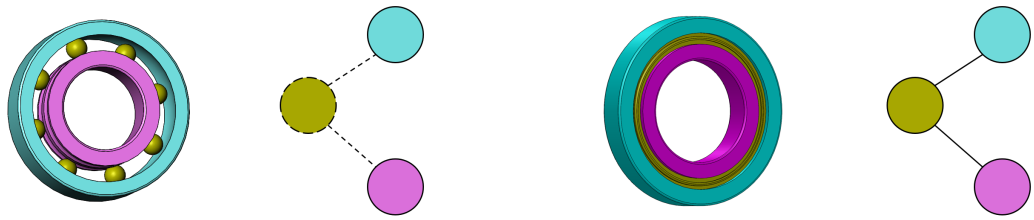


(a) (b)

Figure 4.25: Example of different type of contacts between cage and rolling elements

Figure 4.26 shows the two graphs used for the subgraph matching for the rolling ball bearing. In Figure 4.26(a) the yellow node identify all the balls that form the circular pattern. In this case the arcs identify a vertex or linear contact. While in Figure 4.26(b), the yellow node is associated with a single part of the CAD model and the arcs represent a surface contact.

The nodes of the bearing template graphs represent the main elements characterizing a bearing, which include the repeated elements (balls or rollers) arranged in a circular pattern or idealized with a toroidal/cylindrical shape, the inner and outer rings.
Arcs represent the interaction among these elements. For instance, balls are in contact by a point or a curve with the inner and outer rings, while rollers are always in contacts by a line and eventually two planar faces.



(a) (b)

Figure 4.26: Abstract bearing definition (a) Bearing with ball pattern; (b) bearing with idealized pattern

To focus on the most promising candidates, the matching consider parts classified as *part of bearing* for inner and outer rings, *sphere like* for balls, *cylinder like* for rollers, *torus like* and *spacer* in case of simplification of rolling elements. Details on the adopted matching method are reported in section 5.1.3.

4.4.12 Module "Shape-and-context-based part classification"

The **shape-and-context-based part classification** module assesses the shape-based part classification by analyzing the context of use of the part in the assembly model. Then, this module confirms or changes the value of the attributes $\Phi_{\mathcal{N}_P}(n)_{\restriction Componnet\_Type}$ previously set by the module **shape-based part classification**.

The analysis of the context of use of a part in an assembly relies on some a priori engineering knowledge related to generally present interactions between components in specific mechanisms. The rationale behind is that a specific mechanical component may perform its function within a functional set if it is positioned according to specific conditions with respect to certain classes of components. Hence, to determine if a component effectively belongs to the assigned class it is necessary to check with which types of components/parts the part interacts.

Analyzing this type of mechanism, we can identify a series of features that indicate how parts are in general assembled together. Here we report some of the rules generally valid for functional sets where a shaft is guided in rotation using bearings:

i In most of the configurations, a shaft is guided by two bearings.

ii The shaft and the bearings share the same axis.

iii The main technological solutions to limit the axial translation of the bearings are: a shoulder on the shaft (i.e. bigger diameter on the shaft), a spacer, a lock ring, a cap, a bearing housing, circlips (or elastic ring) which can be put on the two external planar surfaces of the bearing or on an inner ring of the bearing.

iv Bearings can be present between gears and shafts.

v When gears do not rotate along the shaft, then they are blocked either with grooves, keys, mechanical shrinking, or shrinking without elements (not visible in the CAD model).

vi Gears can have multiple cylindrical contacts except along the external surface, which can have only a linear contact to preserve the rotation.

vii The parts that rotate have their gravity center on the rotation axis to avoid extra loads due to dynamic effects.

viii Spacers are tube-like parts for which the outer (resp. inner) cylindrical surface is not in contact with something.

ix To prevent oil leak, seals are normally positioned at the extrema of the components along the shaft.

x Auxiliary elements along a shaft (e.g. spacers, bushings, gears, bearings) have cylindrical contact with the shaft and usually have no contact with screws or nuts.

Of course this list is not exhaustive. Taking into account these rules and the results of the shape-based classification, we apply the iterative procedure, illustrated in Algorithm 9, to verify the part classification. The procedure analyses the class of the elements in contact with the examined part and their type of contacts.

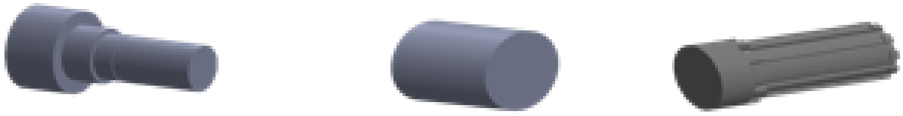
Currently, we are considering only parts that are in contact, but a proper context of use for a part should also include components/parts that interact indirectly with it, e.g. being in contact with a component/part that is in direct contact with the part.

Table 4.3 indicates for each class in the columns the categories of components/parts, expected to be in contact and checked to confirm or invalidate the classification proposed at the first shape-based classification phase.

Table 4.3: Expected contacts between component types

	Bearing	Gear	Shaft	Spacer
Bearing	✓	✓	✓	✓
Cylinder_like	✓	✓		✓
C-clip	✓	✓	✓	✓
Gear	✓	✓	✓	✓
Key		✓	✓	
Shaft	✓	✓		✓
Spacer	✓	✓	✓	✓

One important issue of the undertaken approach is that it needs to know the precision of the shape-based part classification. To identify trustful classified components, we identify classes with high and low recognition rates.
Analyzing common errors in the shape-based classification of assembly parts, we notice that the c-clip class has the lowest failure rate together with components of bearing when designed as sub-assembly (i.e. *part of bearing*, *sphere like*, *cylinder like*). Differently, bearings designed as parts can be misleading with parts representing seals or washers. Gears are generally well classified in presence of teeth, while in their simplified form they are recognized as spacers. The components more mismatched are shafts, which can be classified as *screws*, *cylinder like* or *miscellaneous*.
Figure 4.27 shows an example of three different shafts that are wrongly classified by considering their shape isolated from the context.



(a) (b) (c)

Figure 4.27: Shaft component pre-classified as *screws* (a), *cylinder like* (b) and *miscellaneous* (c)

We start the validation process analyzing components that should be recognized as bearing. If a part labeled as bearing has among its contacts those expected as indicated in Table 1, then the part is confirmed as *bearing*, otherwise its category is changed to *spacer*.
Then, we proceed with shaft components. If a part cannot be confirmed as *shaft* (i.e. it is not in contact with some of the elements indicated in Table 1), then its category is changed into the geometry class closest to a shaft, i.e. the *cylinder like* class. Then, we check parts classified as *screw*. For each part classified as *screw*, if it does not satisfy the screw contacts, we investigate if it is a shaft wrongly classified. If it does not pass neither the shaft control, then we label that part as *cylinder like* and it will be re-controlled later, when other components will be confirmed and can change the validation on this part.
After that we have correct soma parts, we pass to analyze the class that contains many parts wrongly classified: the *spacer* class. If the parts belonging to spacer class do not satisfy the expected contacts, we verify if those parts could be belong to the *gear* class. This because we have included in the training set of the shape-based part classification approach the simplified gears in the spacer class.
Finally, we recall the parts in the *cylinder like* class, which include also previously shafts and screws not confirmed, and we check if the corrected gears improve the identification of the shaft parts.

Algorithm 9 The shape-and-context based classification algorithm

```
1: procedure ShapeContextBasedClassification(Parts)
2:
3:     for each part ∈ parts.BearingClass
4:         if not HasBearingContact(part) then
5:             part is classified as Spacer
6:
7:     for each part ∈ parts.ShaftClass
8:         if not HasShaftContact(part) then
9:             part is classified as Cylinder_Like
10:
11:    for each part ∈ parts.ScrewClass
12:        if not HasScrewContact(part) then
13:            if HasShaftContact(part) then
14:                part is classified as Shaft
15:            else
16:                part is classified as Cylinder_Like
17:
18:    for each part ∈ parts.SpacerClass
19:        if not HasSpacerContact(part) then
20:            if HasGearContact(part) then
21:                part is classified as Gear
22:            else
23:                part is classified as Miscellaneous
24:
25:    for each part ∈ parts.CylinderClass
26:        if HasShaftContact(part) then
27:            part is classified as Shaft
```

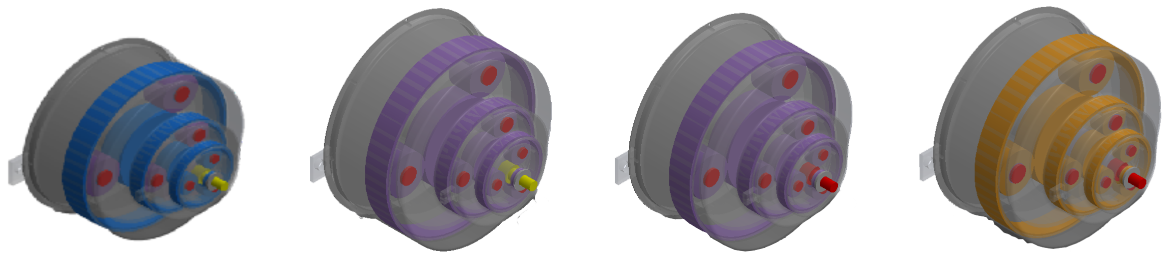
An example of the steps of the algorithm 9 is illustrated in Figure 4.28, where blue color identifies bearings, red is for shafts, yellow for screws, purple for spacers and orange for gears.

Figure 4.28(a) illustrates the classification of the parts at the beginning of the procedure. We can notice that all the gears are wrongly classified and, because of their simplified shape, they are recognize as bearing or spacer. In addition the front axis is classified as screw.

After passing the control in row 4 and 5 in Algorithm 9, the system identifies the wrong bearing and changes their classification in spacer (Figure 4.28(b)).

Next two steps (rows from 7 to 16) confirm axis parts and correct the front screw, whose results is displayed in Figure 4.28(c).

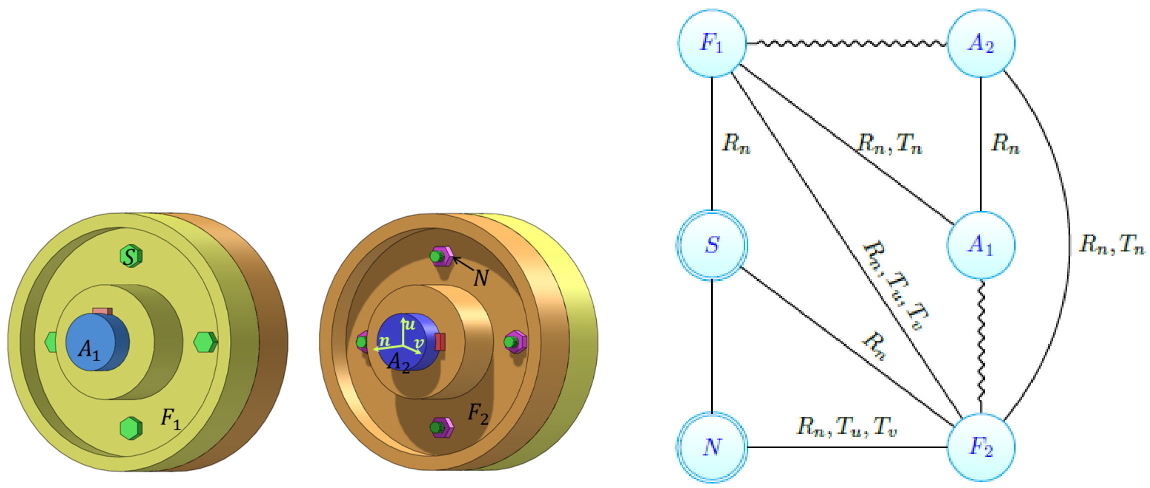
Last image in Figure 4.28(d) shows the classification of parts after the control in rows from 18 to 23, where the contacts of the gear parts are checked.



(a) First step (b) Second step (c) Third step (d) Fourth step

Figure 4.28: Example of the shape-and-context-based classification process

After the creation process, the EAM is represented as a graph, where all the extracted information is encoded as attributes of nodes and arcs. Figure 4.29 illustrates an example of the EAM graph structure created from a CAD model and enriched with semantic information extracted by the geometric reasoning algorithms. For readability purposes, only a part of the attributes is represented. The single line-circled nodes are associated with parts, while the double-line nodes are associated to parts belonging to regular patterns. The straight arcs connect two parts which are in contact, and the associated label indicates the remaining DOF. 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 these cases, we do not have labels specifying the corresponding degree of freedom.



(a) (b)

Figure 4.29: Example of a CAD model (a) and a part of its EAM descriptor (b)

4.4.13 Creation of EAM for the query model

As mentioned before, the graphic user interface provides functionalities to specify the model for the query, i.e. the model to be compared with the EAMs stored in the database. This process includes the creation of an EAM model for the query, in which the various layers can be fully or partially defined depending on the assembly features considered important for the comparison in the query specification. In this process, the EAM characteristics and associated values can be specified by the user or automatically computed from the provided query model, which can range from a complete 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 process applied to the models in the database. If in the query specification, some characteristics are relaxed, i.e. considered irrelevant, the corresponding evaluation modules are ignored. 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 constituting principal components and the related interface links. A dedicated user interface drives the user in this process. The user can add components specifying some associated attributes. He/she can insert the links between the components specifying their type, i.e. a structural arc or an interface arc and in the last case with the attribute specifying the related DOFs. The attributes that the user can specify are those described in the different layers of the EAM descriptor.

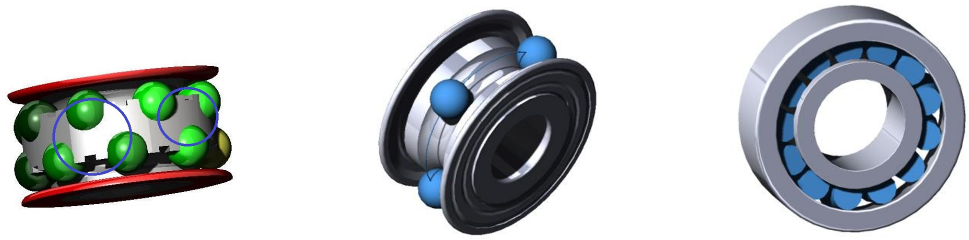
4.5 Conclusions: results and possible extension

This chapters defines the EAM and describes the procedures for its creation. Some steps collect data available in the B-rep of the assembly models and others extract the required information either by exploiting existing procedures or with a deeper analysis and reasoning.

The main contribution is in the following modules:

- repeated component pattern detection,
- component relationship detection,
- shape-based part classification,
- component classification,
- shape-and-context-based part classification.

Figure 4.30 shows some limits of the module for the **repeated component pattern detection**.



(a) (b) (c)

Figure 4.30: Example of limits in the detection of patterns of repeated elements

Balls in Figure 4.30(a) form many patterns of three spheres, while, from an human perspective, two circular pattens were more intuitive. This is due to the fact that, the procedure for the pattern detection defined by Chiang et al. [29] groups closest centers of gravity, while, in this case, our perception yields the patterns with a bigger number of elements.
After the due tests, one possible solution to this type of configurations is represented by grouping centers of gravity of repeated parts according to the bigger number which belong to the same plane.

In Figure 4.30(b) and 4.30(c), the rolling elements in the two bearings are not recognized as pattern. This happens because the rolling elements in both the cases are represented as a single part, where the spheres in Figure 4.30(b) are linked by a passing thread and the cylinders in Figure 4.30(c) are different shells of the same part. This problem cannot be overcome for the bearing in Figure 4.30(b), since the spheres cannot be seen as different elements, while it can be solved for the object in Figure 4.30(c) considering the shells of a parts as atomic element of an assembly model instead of the part itself. Hence, using this decomposition, the rolling cylinders can be seen as different repeated elements.

The module for the **component relationships detection** can be improved on two aspects; first of all clearance interferences could be considered in the detection. To achieve this information, computing the intersection between two parts is not sufficient, but other information such as the intersection of the corresponding bounding boxes could be required. Most of all it is important to study the tolerances for which a clearance is designed on purpose. Secondly, the solution of the problem of volumetric intersection could consider more configurations and not be limited to the intersections with a sphere.
In addition, relationships considered in the interface layer are not sufficient to capture the interaction of forces between part geometries. For instance, the fit condition between a shaft and a bore cannot be expressed by a spatial relationships since it does not provide functional design details such as contact pressure, contact force, rotational torque, rotational speed etc. [104]. Then, other relationships could be added and extracted in the proposed assembly model descriptor.

The modules for the **shape-based part classification** and the **component classification** are preparatory for the **shape-and-context-based part classification** module. The main contributions in the **shape based classification** and the **component classification** modules are represented by the definition of the training set and component templates respectively. Since engineering domain is very rich, we focus our attention on a set of devices for the alteration of speed and torque (i.e. gearboxes). This restriction allow us to extract some configuration rules, even if it is not too limited since gearboxes are included in many objects. As future work, the set of considered parts as the templates can be enhanced including other components.
In addition, to overcome problems due to the shape simplification of the parts, we can exploit other characteristics deriving from engineering knowledge, through which information can be deduced. In particular, in this thesis, we have used the surrounding context in term of contacts between parts, but in the future we can extend this analysis.

Finally, assembly statistics can be enlarged collecting other data as the number of parts for each type (gears, bearings, shaft and so on), in order to reduce the number of candidates to be computed with the query.

5

Assembly models comparison

The purpose of this chapter is to describe how two assembly models are compared using the proposed EAM descriptor. When models are represented by graphs, their comparison is performed by using graph matching, thus, in this chapter, we illustrate the method used to compute graph similarities and how to combine different criteria of similarity in the matching procedure.
In the end, we propose measures to assess the similarity between two compared assembly models. Since two assembly models can be similar according to several criteria, we define four different similarity measures, which are then combined according to the search objectives and criteria.

Contents

5.1 EAMs comparison

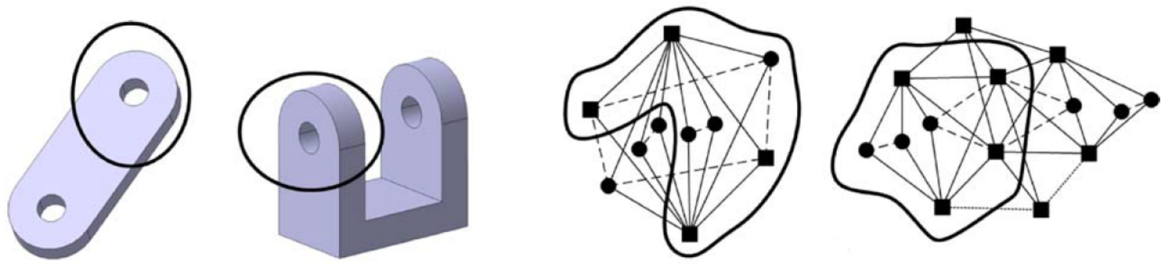
Section 4.3 illustrated the EAM of a 3D CAD model as an *attributed multi-graph*, where each node is associated with a component (part or sub-assembly) of the CAD model and the arcs encode specific relationships between them. In general, when models are represented by graphs, their comparison is performed by using graph matching, i.e. a process which aims to find a correspondence between nodes and arcs of two graphs according to some constraints [31]. These constraints can be more or less stringent and their satisfaction determines the level of similarity between the two objects that are compared.
Then, this section aim at defining the concept of graph similarity in section 5.1.1, how to manage the different similarity criteria in section 5.1.2 and, in the end, how to perform the matching between two EAMs in section 5.1.3. Basic notions and algorithms of graph theory are given in Appendix.

5.1.1 Graph similarity

In the following of this section, for simplicity, we refer only to *graphs* ($G = \mathcal{G}(\mathcal{N}, \mathcal{A})$), but the same considerations are valid also for *attributed multi-graphs* ($G = \mathcal{G}(\mathcal{N}, \mathcal{A}, \Phi_{\mathcal{N}}, \Phi_{\mathcal{A}})$) as the Enriched Assembly Model.

Given two graphs $G_i = \mathcal{G}(\mathcal{N}_i, \mathcal{A}_i)$ and $G_j = \mathcal{G}(\mathcal{N}_j, \mathcal{A}_j)$, where the generic node h^{th} in G_i is represented $n_i^h \in \mathcal{N}_i$ and the generic node k^{th} in G_j is represented $n_j^k \in \mathcal{N}_j$, we can state the problem of graph matching as finding a one-to-one mapping $f : \mathcal{N}_i \longrightarrow \mathcal{N}_j$ such that $(n_i^h, n_i^k) \in \mathcal{A}_i$ if and only if (iff) $(f(n_i^h), f(n_i^k)) \in \mathcal{A}_j$. This condition ensures the same topology in the two graphs, i.e. if two nodes n_i^k and n_i^h are adjacent in G_i, then also their corresponding nodes $f(n_i^k)$ and $f(n_i^h)$ have to be adjacent in G_j. If such a mapping f exists, it is called *graph isomorphism* and G_i is said to be isomorphic to G_j [119].
A weaker form of matching is representing by a *subgraph isomorphism*, which defines an isomorphism between G_i and a subgraph S_j of G_j.

Graph and subgraph isomorphisms express a strong condition of similarity and other matching conditions often are favourite [14] especially when the goal is the evaluation of similarity between non identical objects [15]. For instance, let us consider two objects as in Figure 5.1(a), where in their graph representations (Figure 5.1(b)) the squared and circle nodes indicate planar and cylindrical faces respectively and the arcs indicates if two faces are in contact. In this example, the two models have a local correspondence, highlighted in by the encircled nodes. In this example, the problem of finding local similarities is traduced in finding two subgraphs $S_i \subseteq G_i$ and $S_j \subseteq G_j$ such that there exists a graph isomorphism between S_i and S_j. This problem is known in literature as the *maximum common subgraph* (MCS) problem.



(a) (b)

Figure 5.1: Matching of local correspondence and their common subgraph [131]

Usually, the MCS problem is transposed into the problem of finding the *maximum clique* (MC) in a suitable graph [90]. This graph is the *association graph* introduced by Kozen [67] and Barrow and Burstall [10] as:

Attributed graph

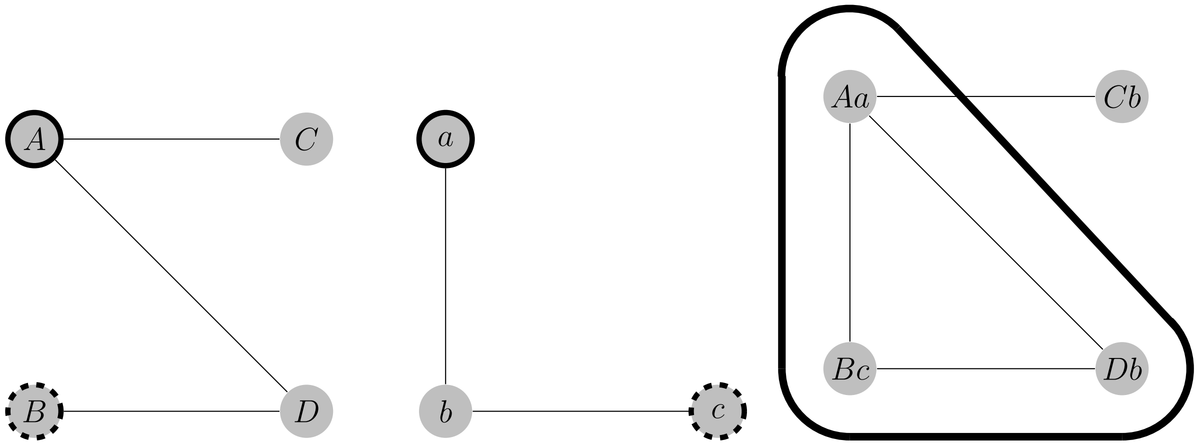
Definition 5.1.1. Given two attributed graphs:

- $G_i = \mathcal{G}(\mathcal{N}_i, \mathcal{A}_i, \Phi_{\mathcal{N}_i}, \Phi_{\mathcal{A}_i})$
- $G_j = \mathcal{G}(\mathcal{N}_j, \mathcal{A}_j, \Phi_{\mathcal{N}_j}, \Phi_{\mathcal{A}_j})$

their *association graph* $G_{i,j,c_N,c_A} = \mathcal{G}(\mathcal{N}_{i,j,c_N}, \mathcal{A}_{i,j,c_A})$ is defined by a set $\mathcal{N}_{i,j,c_N}$ of couples of *compatible* nodes and a set $\mathcal{A}_{i,j,c_A}$ of *compatible* arcs, where:

- $\mathcal{N}_{i,j,c_N}$ is a set of compatible nodes pairs $(n_i^h, n_j^k) \in \mathcal{N}_i \times \mathcal{N}_j$, where the compatibility is defined according to the values of the node attribute functions $\Phi_{\mathcal{N}_i}$ and $\Phi_{\mathcal{N}_j}$ and the criteria of similarity on nodes (i.e. c_N) selected by the user.
- $\mathcal{A}_{i,j,c_A}$ is a set of compatible arcs pairs $(a_i^h, a_j^k) \in \mathcal{A}_i \times \mathcal{A}_j$, such that $a_i^h \in \mathcal{A}_i$ iff $a_j^k \in \mathcal{A}_j$, where the compatibility is defined according to the values of the arc attribute functions $\Phi_{\mathcal{A}_i}$ and $\Phi_{\mathcal{A}_j}$ and the criteria of similarity on arcs (i.e. c_A) selected by the user.

The compatibility of nodes and arcs considered in this work is defined in section 5.1.2. The condition $a_i^h \in \mathcal{A}_i$ iff $a_j^k \in \mathcal{A}_j$ means that two association nodes are linked if their corresponding nodes are both connected (or not connected) in the original attributed graphs. An example is illustrated in Figure 5.2, which shows a simple example of two attributed graphs and their association graph. The nodes Aa, Bc, Cd and Db are constructed from G_1 and G_2 according to the same attributes indicated by the line used in the contour of the depicted node. Based on the consistency of connection characteristic, nodes Aa and Db are connected by an arc. Nodes Aa and Bc are connected because nodes A and B are not connected in G_1 and nodes a and c are also not connected in G_2.



(a) G_1 (b) G_2 (c) Association graph

Figure 5.2: Example of construction of an association graph

The following theorem establishes an equivalence between the graph isomorphism problem and the maximum clique problem [90].

Theorem 5.1.1. Let $G_i = \mathcal{G}(\mathcal{N}_i, \mathcal{A}_i, \Phi_{\mathcal{N}_i}, \Phi_{\mathcal{A}_i})$ and $G_j = \mathcal{G}(\mathcal{N}_j, \mathcal{A}_j, \Phi_{\mathcal{N}_j}, \Phi_{\mathcal{A}_j})$ two attributed graphs and let $G_{i,j,c_N,c_A} = \mathcal{G}(\mathcal{N}_{i,j,c_N}, \mathcal{A}_{i,j,c_A})$ be the corresponding association graph. Then, G_i and G_j are isomorphic iff the order of C is equal to $|\mathcal{N}_i| = |\mathcal{N}_j|$, where C is the maximum clique of G_{i,j,c_N,c_A} and $|\mathcal{N}_i|$ and $|\mathcal{N}_j|$ represent respectively the number of nodes in G_i and G_j.
In this case, any maximum clique C of G_{i,j,c_N,c_A} induces an isomorphism between G_i and G_j, and vice versa.

Proof. Suppose that the two graphs are isomorphic and let ϕ be an isomorphism between them. Then the subset of vertices of G_{i,j,c_N,c_A} defined as $C_\phi = \{(h, \phi(h)) : \forall h \in \mathcal{N}_i\}$ is clearly a maximum clique of order $|\mathcal{N}_i|$. Conversely, let C be an $|\mathcal{N}_i|$-vertex maximum clique of G_{i,j,c_N,c_A} and for each $(h, k) \in C$ define $\phi(h) = k$. Then, because of the way the association graph is constructed, it is clear that ϕ is an isomorphism between G_i and G_j. $\square$

Then to compare two EAMs, we need to find a maximum clique in a suitable association graph. A formal explanation of the association graph, that uses the EAM and different similarity criteria, is given in the next section 5.1.2, while the technique to solve the MC problem is reported in section 5.1.3.

5.1.2 Assembly similarity through association graph

In order to solve our matching problem in terms of MCS, we need to define a suitable *association graph* where looking for its maximum clique.
In this section, we present the definition and the construction of the association graph according to the similarity criteria selected in the query, then some examples of association graphs are shown.

In the following, we refer to the attributed multi-graph associated with the query model as $G_q = \mathcal{G}(\mathcal{N}_q, \mathcal{A}_q, \Phi_{\mathcal{N}_q}, \Phi_{\mathcal{A}_q})$ and to the attributed multi-graph associated with the k-th model to be compared as $G_k = \mathcal{G}(\mathcal{N}_k, \mathcal{A}_k, \Phi_{\mathcal{N}_k}, \Phi_{\mathcal{A}_k})$.
Their association graph is represented $G_{q,k,c_N,c_A} = \mathcal{G}(\mathcal{N}_{q,k,c_N}, \mathcal{A}_{q,k,c_A})$, the generic node i-th in G_q is represented $n_q^i \in \mathcal{N}_q$ and the generic node j-th in G_k is represented $n_k^j \in \mathcal{N}_k$.

The compatibility of nodes depends on different criteria chosen when specifying the query. In particular, the similarity conditions that can be set up referred to nodes concern: shape, size, component type and pattern type.
Similarly the compatibility on the arcs depends on the type of contacts and the DOF allowed, these aspects are described in the following paragraphs.
Each of these criteria is specified by the values of its attribute functions. For instance, the $\Phi_{\mathcal{N}_P}(n_q^i) \in Shape \times Size$, i.e. the first component of $\Phi_{\mathcal{N}_P}(n_q^i)$ denotes the shape attribute, while the second the size one. Then, to disambiguate the single attribute, in the following we indicate with $\Phi_{\mathcal{N}_*}(n)_{\restriction Node\_Attribute\_Set}$ the restriction of the attribute function $\Phi_{\mathcal{N}_*}(n)$ on a certain $Node\_Attribute\_Set$ and with $\Phi_{\mathcal{A}_*}(a)_{\restriction Arc\_Attribute\_Set}$ the restriction of the attribute function $\Phi_{\mathcal{A}_*}(a)$ on a certain $Arc\_Attribute\_Set$.

Compatibilities for nodes

Let see in the following the different compatibilities for the nodes.

Compatibility with respect to shape

Definition 5.1.2. Given

- $S_q^i = \Phi_{\mathcal{N}_P}(n_q^i)_{\restriction Shape} \in Shape = \mathbb{R}^{544}$,
- $S_k^j = \Phi_{\mathcal{N}_P}(n_k^j)_{\restriction Shape} \in Shape = \mathbb{R}^{544}$,

the two nodes n_q^i and n_k^j are considered *compatible according to the shape criterion* if $\chi_{Sh}(n_q^i, n_k^j) = 1$, where

$$\chi_{Sh}(n_q^i, n_k^j) = \begin{cases} 1 & \text{if} \quad ||S_q^i - S_k^j||_2 < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where ϵ represents the threshold set in the query.

Compatibility with respect to size

Definition 5.1.3. Given

- $(Vol_q^i, Sur_q^i) = \Phi_{\mathcal{N}_P}(n_q^i)_{\upharpoonright Size} \in Size$,
- $(Vol_k^j, Sur_k^j) = \Phi_{\mathcal{N}_P}(n_k^j)_{\upharpoonright Size} \in Size$,

the two nodes n_q^i and n_k^j are considered *compatible according to the size criterion* if $\chi_{Si}(n_q^i, n_k^j) = 1$, where

$$\chi_{Si}(n_q^i, n_k^j) = \begin{cases} 1 & \text{if} \quad (|Vol_q^i - Vol_k^j| < \epsilon_V) \text{ and} \\ & \quad (|Sur_q^i - Sur_k^j| < \epsilon_S) \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

where ϵ_V and ϵ_S represent the thresholds set in the query.

Compatibility with respect to *Component_Type*

Definition 5.1.4. Given

- $CT_q^i = \Phi_{\mathcal{N}_P}(n_q^i)_{\upharpoonright Component\_Type} \in Componet\_Type$,
- $CT_k^j = \Phi_{\mathcal{N}_P}(n_k^j)_{\upharpoonright Component\_Type} \in Componet\_Type$,

two nodes n_q^i and n_k^j are considered *compatible according to the component type criterion* if $\chi_{CT}(n_q^i, n_k^j) = 1$, where

$$\chi_{CT}(n_q^i, n_k^j) = \begin{cases} 1 & \text{if} \quad CT_q^i = CT_k^j \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Compatibility with respect to *Pattern_Type*

Definition 5.1.5. Given

- $PT_q^i = \Phi_{\mathcal{N}_P}(n_q^i)_{\upharpoonright Pattern\_Type} \in Pattern\_Type$,
- $PT_k^j = \Phi_{\mathcal{N}_P}(n_k^j)_{\upharpoonright Pattern\_Type} \in Pattern\_Type$,

two nodes n_q^i and n_k^j are considered *compatible according to the pattern type criterion* if $\chi_{PT}(n_q^i, n_k^j) = 1$

$$\chi_{PT}(n_q^i, n_k^j) = \begin{cases} 1 & \text{if} \quad PT_q^i = PT_k^j \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

Let $c_N = (\alpha_{Sh}, \alpha_{Si}, \alpha_{CT}, \alpha_{PT})$ be a vector which represents the criteria that can be selected, where α_{Sh}, α_{Si}, α_{CT} and α_{PT} represent respectively the activation of the shape, the size, the component type and the pattern criterion.

More precisely, the generic criterion $\alpha_* \in \{0,1\}$ and $\alpha_* = 1$ means that the criterion * is activated.
In order to reduce meaningless node association, two nodes can be associated according to the size criterion only if the shape or component type criteria are selected, i.e.,

$$\alpha_{Si} = 1 \iff (\alpha_{Sh} = 1) \text{ or } (\alpha_{CT} = 1) \tag{5.5}$$

With this notation, we can define the set of compatible nodes as:

$$\mathcal{N}_{q,k,c_N} = \left\{ \gamma_N(n_q^i, n_k^j, c_N), \quad n_q^i \in \mathcal{N}_q \text{ and } n_k^j \in \mathcal{N}_k \right\} \tag{5.6}$$

where

$$\begin{aligned} &\gamma_N : \mathcal{N}_q \times \mathcal{N}_k \times c_N \to \mathcal{N}_q \times \mathcal{N}_k \\ &\gamma_N(n_q^i, n_k^j, c_N) = \begin{cases} (n_q^i, n_k^j) & \text{if} \quad \chi_{Node}(n_q^i, n_k^j, c_N) = 1 \\ \emptyset & \text{if} \quad \chi_{Node}(n_q^i, n_k^j, c_N) = 0 \end{cases} \end{aligned} \tag{5.7}$$

with

$$\chi_{Node}(n_q^i, n_k^j, c_N) = \left\lfloor \frac{\alpha_{Sh}\chi_{Sh}(n_q^i, n_k^j) + \alpha_{Si}\chi_{Si}(n_q^i, n_k^j) + \alpha_{CT}\chi_{CT}(n_q^i, n_k^j) + \alpha_{PT}\chi_{PT}(n_q^i, n_k^j)}{\alpha_{Sh} + \alpha_{Si} + \alpha_{CT} + \alpha_{PT}} \right\rfloor \tag{5.8}$$

The symbol $\lfloor * \rfloor$ indicate the floor integer part of *.
The function χ_{Node} has value in $\{0,1\}$ and describes the compatibility of two nodes according to the criteria that can be chosen in the query, i.e. n_q^i and n_k^j are compatible according to c_N if $\chi_{Node}(n_q^i, n_k^j, c_N) = 1$.

Generally, the image of the function γ_N is a subset of $\mathcal{N}_q \times \mathcal{N}_k$ and its cardinality strictly depends on the specified criteria for the association of the nodes. The number of association nodes decreases as the criteria specify that more similarity conditions have to be satisfied.

Compatibility for arcs

In the following the different compatibilities for the arcs are introduced, where the generic arc in G_q between the node pair (n_q^i, n_q^j) is indicated as $a_q^{ij} \in \mathcal{A}_q$ and the generic arc in G_k between the node pair (n_k^h, n_k^l) is indicated as $a_k^{hl} \in \mathcal{A}_k$.

Compatibility with respect to contact

Definition 5.1.6. Two arcs a_q^{ij} and a_k^{hl} are considered *compatible according to the contact criterion* if $\chi_C(a_q^{ij}, a_k^{hl}) = 1$, where

$$\chi_C(a_q^{ij}, a_k^{hl}) = \begin{cases} 1 & \text{if} \quad (a_q^{ij} \in \mathcal{A}_{C_q}) \text{ and } (a_k^{hl} \in \mathcal{A}_{C_k}) \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

In section 4.2.2, we have seen that two parts may have multiple *contact information*, this means that multiple arcs can exist between the nodes which correspond to the two parts. Then, compatibility according to the allowed DOF of contacts criterion has to consider the set of contact arcs present between the two nodes. In particular, we indicate with $\mathcal{A}_{C_q}^{ij}$ the set of contacts arcs between the nodes n_q^i and n_q^j and with $\mathcal{A}_{C_k}^{hl}$ the set of contacts arcs between the nodes n_k^h and n_k^l.

Compatibility with respect to allowed DOF for contacts

Definition 5.1.7. Given

- $\mathcal{A}_{C_q}^{ij}$ and $\mathcal{A}_{C_k}^{hl}$
- $(C_q^{ij})_s = \Phi_{\mathcal{A}_C}(c_s)_{\restriction Contact\_Type} \in Contact\_Type$ for $c_s \in \mathcal{A}_{C_q}^{ij}$,
- $(T_q^{ij})_s \cup (R_q^{ij})_s = \Phi_{\mathcal{A}_C}(c_s)_{\restriction DOF}$ for $c_s \in \mathcal{A}_{C_q}^{ij}$,
- $(C_k^{hl})_t = \Phi_{\mathcal{A}_C}(c_t)_{\restriction Contact\_Type} \in Contact\_Type$ for $c_t \in \mathcal{A}_{C_k}^{hl}$,
- $(T_k^{hl})_t \cup (R_k^{hl})_t = \Phi_{\mathcal{A}_C}(c_t)_{\restriction DOF}$ for $c_t \in \mathcal{A}_{C_k}^{hl}$,

the set of arcs $\mathcal{A}_{C_q}^{ij}$ and $\mathcal{A}_{C_k}^{hl}$ are considered *compatible according to the allowed DOF for contacts criterion* if $\chi_{C_{num}}(\mathcal{A}_{C_q}^{ij}, \mathcal{A}_{C_k}^{hl}) = 1$, where

$$\chi_{C_{num}}(\mathcal{A}_{C_q}^{ij}, \mathcal{A}_{C_k}^{hl}) = \begin{cases} 1 & \text{if} \quad \Big[\Big(|(T_q^{ij})_s| = |(T_k^{hl})_t|\Big) \text{ and } \Big(|(R_q^{ij})_s| = |(R_k^{hl})_t|\Big)\Big] \\ & \qquad \forall c_s \in \mathcal{A}_{C_q}^{ij} \quad c_t \in \mathcal{A}_{C_k}^{hl} \quad \text{or} \\ & \quad \Big[(C_q^{ij})_s = \text{"}UnSolved\text{"}\Big] \quad \text{or } \Big[(C_k^{ij})_t = \text{"}UnSolved\text{"}\Big] \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

Compatibility with respect to the allowed DOF for joints

Definition 5.1.8. Given

- $J_q^{ij} = \Phi_{\mathcal{A}_C}(a_q^{ij})_{\restriction Joint\_Type} \in Joint\_Type$,
- $T_q^{ij} \cup R_q^{ij} = \Phi_{\mathcal{A}_J}(a_q^{ij})_{\restriction DOF}$,
- $J_k^{hl} = \Phi_{\mathcal{A}_C}(a_q^{hl})_{\restriction Joint\_Type} \in Joint\_Type$,
- $T_k^{hl} \cup R_k^{hl} = \Phi_{\mathcal{A}_J}(a_k^{hl})_{\restriction DOF}$,

two arcs a_q^{ij} and a_k^{hl} are considered *compatible according to the allowed DOF for joints criterion* if $\chi_{J_{num}}(a_q^{ij}, a_k^{hl}) = 1$, where

$$\chi_{J_{num}}(a_q^{ij}, a_k^{hl}) = \begin{cases} 1 & \text{if} \quad \Big[\Big(|T_q^{ij}| = |T_k^{hl}|\Big) \text{ and } \Big(|R_q^{ij}| = |R_k^{hl}|\Big)\Big] \quad \text{or} \\ & \quad \Big[J_q^{ij} = \text{"}UnSolved\text{"}\Big] \quad \text{or } \Big[J_k^{hl} = \text{"}UnSolved\text{"}\Big] \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

Concerning arcs, we have $c_A = (\alpha_C, \alpha_{C_{num}}, \alpha_{J_{num}})$ a vector which represents the criteria that can be selected, where α_C, $\alpha_{C_{num}}$ and $\alpha_{J_{num}}$ represent respectively the contact, the allowed DOF for contacts and the allowed DOF for joints criterion. Also for arcs, the generic filter $\alpha_* \in \{0, 1\}$ and $\alpha_* = 1$ means that the filter * is activated.

We can define the set of compatible arcs as:

$$\mathcal{A}_{q,k,c_A} = \Big\{\gamma_A(a_q^{ij}, a_k^{hl}, c_A), \qquad a_q^{ij} \in \mathcal{A}_q \text{ and } a_k^{hl} \in \mathcal{A}_k\Big\} \tag{5.12}$$

where

$$\begin{gathered}\gamma_A : \mathcal{A}_q \times \mathcal{A}_k \times c_A \to \mathcal{N}_{q,k,c_N} \times \mathcal{N}_{q,k,c_N} \\ \gamma_A(a_q^{ij}, a_k^{hl}, c_A) = \begin{cases} ((n_q^i, n_k^h), (n_q^j, n_k^l)) & \text{if} \quad \chi_{Arc}(a_q^{ij}, a_k^{hl}, c_A) = 1 \\ \emptyset & \text{if} \quad \chi_{Arc}(a_q^{ij}, a_k^{hl}, c_A) = 0 \end{cases}\end{gathered} \tag{5.13}$$

with

$$(n_q^i, n_k^h) \qquad \text{and} \quad (n_q^j, n_k^l) \in \mathcal{N}_{q,k,c_N} \tag{5.14}$$

$$a_q^{ij} = (n_q^i, n_q^j) \in \mathcal{A}_{C_q} \cup \mathcal{A}_{J_q} \tag{5.15}$$

$$a_k^{hl} = (n_k^h, n_k^l) \in \mathcal{A}_{C_k} \cup \mathcal{A}_{J_k} \tag{5.16}$$

$$\chi_{Arc}(a_q^{ij}, a_k^{hl}, c_A) = \left\lfloor \frac{\alpha_C \chi_C(a_q^{ij}, a_k^{hl}) + \alpha_{C_{num}} \chi_{C_{num}}(\mathcal{A}_{C_q}^{ij}, \mathcal{A}_{C_k}^{hl}) + \alpha_{J_{num}} \chi_{J_{num}}(a_q^{ij}, a_k^{hl})}{\alpha_C + \alpha_{C_{num}} + \alpha_{J_{num}}} \right\rfloor \tag{5.17}$$

The symbol $\Big\lfloor * \Big\rfloor$ indicate the floor integer part of *.

Since the contact similarity entails the allowed DOF for contacts similarity, which requires the similarity of the allowed DOF for joints, to avoid useless verifications, when a criterion of compatibility on the arcs is selected the other two are set to 0.

Once selected the comparison criteria (c_N for the nodes and c_A for the arcs), there exist **an unique** association graph (according to those criteria) and we refer to it as:

$$G_{q,k,c_N,c_A} = \mathcal{G}(\mathcal{N}_{q,k,c_N}, \mathcal{A}_{q,k,c_A}) \tag{5.18}$$

where $\mathcal{N}_{q,k,c_N} = \Big\{\gamma_N(n_q^i, n_k^j, c_N)\Big\}$ and $\mathcal{A}_{q,k,c_A} = \Big\{\gamma_A(a_q^{ij}, a_k^{hl}, c_A)\Big\}$.

The maximum cliques in the association graph G_{q,k,c_N,c_A} represent the common subgraphs between G_q and G_k according to the criteria c_N and c_A.

The generic h^{th} clique in the graph G_{q,k,c_N,c_A} is expressed as:

$$(C_{q,k,c_N,c_A})_h \subseteq G_{q,k,c_N,c_A} \tag{5.19}$$

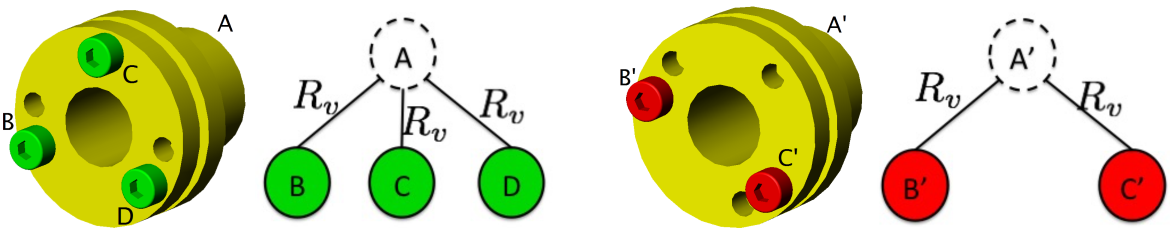
and the set of all the cliques for the association graph G_{q,k,c_N,c_A} is denoted as:

$$\mathcal{D}_{q,k} = \{(C_{q,k,c_N,c_A})_h\}_{h=1}^{Num_{qk}} \tag{5.20}$$

where Num_{qk} is the number of maximum cliques in the association graph G_{q,k,c_N,c_A}.

Example

Let see an example of assembly similarity through the association graph and the mechanism for its creation. In Figure 5.3(a) we have an assembly model representing a flange with three screws and a portion of its attributed multi-graph is depicted in Figure 5.3(b). For sake of readability, the root node corresponding to the entire assembly model is omitted and only the arcs of joint type are depicted. We compare this model with the one in Figure 5.3(c), whose portion of attributed multi-graph is depicted in Figure 5.3(d). The two models represent both a flange. The first has three screws while the second just two. In this example, the nodes of the two attributed multi-graphs in Figure 5.3(b) and 5.3(d) represent the part of the CAD models, same type of line indicates same value of the spherical harmonic shape descriptor (i.e. parts with similar shapes) and parts with the same color belong to patterns of a specific type (i.e. green for circular translation and red for linear translation). The arcs represent the joint contacts where the labels indicate the DOF allowed between two linked parts.



(a) Flange model with three screws (b) G_q (c) Flange model with two screws (d) G_k

Figure 5.3: Example of two assembly models and their EAM attributed multi-graphs G_q and G_k.

First query example

In the first query example, we suppose that the user looks for assemblies in which parts with similar shape are connected by the same joint relationships, i.e. the same degree of freedom. This means that the two arcs should have the same number of rotation and same number of translation. In this case, two nodes are put together in an association node according to their shape attribute (i.e. if the corresponding parts have similar shape), while the association arcs are added only if the joint arcs (between the related pairs of nodes in the attributed multi-graphs) have the same attributes.
With the notation previously introduced, the query can be expressed using the following criteria on nodes and arcs

$$c_N = (1,0,0,0) \tag{5.21}$$
$$c_A = (0,0,1) \tag{5.22}$$

Hence, nodes n_q^i and n_k^j is evaluated from the simplified $\gamma_N(n_q^i, n_k^j, c_N)$ function obtained from equation 5.8:

$$\gamma_N(n_q^i, n_k^j, c_N) = \begin{cases} (n_q^i, n_k^j) & \text{if} \quad \chi_{Sh}(n_q^i, n_k^j) = 1 \\ \emptyset & \text{if} \quad \chi_{Sh}(n_q^i, n_k^j) = 0 \end{cases} \tag{5.23}$$

Thus, the node A in G_q is associated only with the node A' in G_k, while the node $B, C, D \in G_q$ are associated with both B' and C' nodes of G_k (see Figure 5.4).

Concerning arcs, their compatibility is checked using the simplified γ_A function obtain from equation 5.17:

$$\gamma_A(a_q^{ij}, a_k^{hl}, c_A) = \begin{cases} ((n_q^i, n_k^h), (n_q^j, n_k^l)) & \text{if} \quad \chi_J(a_q^{ij}, a_k^{hl}) = 1 \\ \emptyset & \text{if} \quad \chi_J(a_q^{ij}, a_k^{hl}) = 0 \end{cases} \tag{5.24}$$

Thus, each pair of association node is linked by an arc if in the original attributed multi-graphs, the corresponding arcs have the same number of translations and rotations.
The association graph, resulting from these criteria, is illustrated in Figure 5.4, where there are six possible maximum cliques, i.e.

$$\begin{array}{rcl} (C_{q,k,c_N,c_A})_1 & = & \{AA', BB', CC'\} \\ (C_{q,k,c_N,c_A})_2 & = & \{AA', BC', DB'\} \\ (C_{q,k,c_N,c_A})_3 & = & \{AA', CB', DC'\} \\ (C_{q,k,c_N,c_A})_4 & = & \{AA', CC', DB'\} \\ (C_{q,k,c_N,c_A})_5 & = & \{AA', BB', DC'\} \\ (C_{q,k,c_N,c_A})_6 & = & \{AA', BC', CB'\} \end{array} \tag{5.25}$$

Each clique represents a possible subgraph matching between the two attributed multi-graphs G_q and G_k.

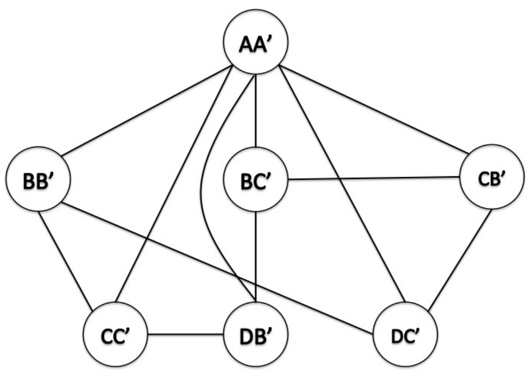


Figure 5.4: Association graph for the objects in Figure 5.3 built with shape and joint criteria of similarity

Second query example

In the second example, the query is for parts with similar shape, the same type of pattern and linked by same joints, thus the type of pattern is also involved in the computation of the association graph, through $\alpha_{PT} = 1$. In this case, the filter is represented as:

$$c_N = (1, 0, 0, 1) \tag{5.26}$$
$$c_A = (0, 0, 1) \tag{5.27}$$

Thus the nodes are associated according to the following compatibility function:

$$\gamma_N(n_q^i, n_k^j, c_N) = \begin{cases} (n_q^i, n_k^j) & \text{if } \left\lfloor \frac{\chi_{Sh}(n_q^i, n_k^j) + \chi_{PT}(n_q^i, n_k^j)}{2} \right\rfloor = 1 \\ \emptyset & \text{if } \left\lfloor \frac{\chi_{Sh}(n_q^i, n_k^j) + \chi_{PT}(n_q^i, n_k^j)}{2} \right\rfloor = 0 \end{cases} \tag{5.28}$$

Back to the example of Figure 5.3, the nodes associated according to the shape are the ones discussed in the previous example, while, for the pattern condition, $\chi_{PT}(A, A') = 1$ since they do not belong to any pattern, and $\chi_{PT}(B, B') = \chi_{PT}(B, C') = \chi_{PT}(C, B') = \chi_{PT}(C, C') = \chi_{PT}(D, B') = \chi_{PT}(D, C') = 0$ since the type of pattern for these pairs of nodes is different.

The pattern similarity criterion produces the following node compatibilities:

$$\begin{aligned} \chi_{Node}(A, A') &= \left\lfloor \frac{\chi_{Sh}(A, A') + \chi_{PT}(A, A')}{2} \right\rfloor \\ &= \left\lfloor \frac{1+1}{2} \right\rfloor = \left\lfloor 1 \right\rfloor = 1 \end{aligned} \tag{5.29}$$

and

$$\begin{aligned} \chi_{Node}(B, B') &= \left\lfloor \frac{\chi_{Sh}(B, B') + \chi_{PT}(B, B')}{2} \right\rfloor \\ &= \left\lfloor \frac{1+0}{2} \right\rfloor = \left\lfloor 0.5 \right\rfloor = 0 \end{aligned} \tag{5.30}$$

Similarly, $\chi_{Node}(B, C') = \chi_{Node}(C, B') = \chi_{Node}(C, C') = \chi_{Node}(D, B') = \chi_{Node}(D, C')$. Hence, this query produces only an association node and consequently no arc can be created. The association graph, resulted from these criteria, is illustrated in Figure 5.5, where it is quite obvious that there are no solutions that satisfy all the activated criteria.

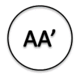


Figure 5.5: Association graph for the objects in Figure 5.3 built with shape, pattern and joint criteria of similarity

5.1.3 Matching algorithm

We have defined the association graph given the attributed multi-graph representations of two assembly models and their similarity criteria. Now we need to find its maximum cliques.

The most famous clique detection algorithm was developed in 1973 by Bron and Kerbosh [19]. This kind of problem is known to be NP-complete, hence, exact graph matching has exponential time complexity in the worst case [31]. Then, to resolve the MC problem, some heuristic algorithms are used [21, 31].

Among the various techniques proposed for the identification of the MC in the association graph, we decided to use two different methods. An heuristic method is used to reduce the complexity of an exact graph matching, and in particular, our procedure exploits the *simulated annealing* technique [16, 17, 131].

Moreover, to validate the adequacy and the usefulness of the information included in the proposed assembly descriptor, we decided to test our framework using also an exact graph matching method. This decision is led by the fact that the correctness and the precision of an exact method can underline possible inability and limits of the EAM descriptor which can be less evident using an heuristic approach.

In the following, we summarize the heuristic algorithm based on the simulated annealing approach and the adopted exact algorithm based on Bron and Kerbos's method.

Heuristic algorithm

The selected heuristic approach for the clique detection is based on the simulated annealing process. The term annealing refers to a physical process to obtain a pure lattice structure, where a solid is first heated up in a heat bath until it melts, and next cooled down slowly until it solidifies. Kirkpatrick et al. [66] used the simulated annealing process to solve a combinatorial optimization problem, i.e. given a function f they aim to find the solution which minimizes f.

A subgraph S can be characterized as a clique by the number of its nodes ($|N|$) and arcs ($|A|$) if it satisfies the following equation:

$$\frac{|N|(|N|-1)}{2} = |A|, \tag{5.31}$$

then, the function f we aim at minimizing is the following:

$$f(S) = \frac{|N|(|N|-1)}{2} - |A|. \tag{5.32}$$

This function is used in the simulated annealing procedure (illustrated in Algorithm 10), which takes as input a graph G and a subgraph S. Three parameters T_i and T_e $DecreasingParameter$, which simulate respectively the initial temperature, the final temperature and the speed of cooling in the annealing process, are set in the algorithm as $T_i = 1000$, $T_e = 0$ and $DecreasingParameter = 0,99$ based on the evaluations proposed in [131].

At any necessary iteration a new subgraph ($NewSubGraph$) is considered, by substituting a node with another in the association graph (by the Swap procedure).
If the value of $f(NewSubGraph)$ is minor than $f(S)$, then $NewSubGraph$ is the new possible solution, otherwise $NewSubGraph$ can be accepted anyway as the new possible solution according to the probability defined in the work of Metropolis [82], i.e. $e^{\frac{-d}{tao}}$, where d represents the difference between the new and the old values of f and tao represents the current temperature.

Algorithm 10 Simulated annealing procedure

```
procedure SimulatedAnnealing(G, S)
    tao = Ti
    f = f(S)

    while tao > Te & f > 0 do
        NewSubGraph = Swap(G, S)
        f_New = f(S)
        d = f_New - f

        if d <= 0 then
            S = NewSubGraph
            f = f_New

        else
            prob = Exp(-d / tao)
            random = Random(0,1)

            if prob - random > 0 then
                S = NewSubGraph
                f = f_New

        tao = DecreasingParameter * tao

      return S;
```

Exact algorithm

For the exact algorithm of the clique detection we applied the Eppstein-Strash algorithm [38]. This algorithm represents an improved version of the algorithm by Tomita [121], which is in turn based on the Bron-Kerbosch algorithm for the detection of all maximal cliques in graphs [19].

The algorithm of Eppstein-Strash improves Tomita's algorithm by using the concept of degeneracy. The degeneracy of a graph G is the smallest number d such that every subgraph of G contains a node of degree at most d. Moreover, every graph with degeneracy d has a degeneracy ordering: a linear ordering of the vertices such that each node has at most d neighbors after it in the ordering. Eppstein-Strash algorithm first computes the degeneracy ordering; then for each node n in the order, starting from the first, the algorithm of Tomita is used to compute all cliques containing v and v's neighbors. Other improvements depend on the use of adjacency lists for data representation. For more details we refer to [38].

5.2 Similarity assessment

When searching for similar models, it is crucial providing solutions ranked according to the "closeness" to the query. Generally, the closeness is expressed by exploiting a measure, which defines the *dissimilarity* of two models as a metric, i.e. a function d between two spaces X, $d : X \times X \longrightarrow \mathbb{R}$, which satisfies the following properties for each x, y and $z \in X$ [13]:

- $d(x, y) \geq 0$ (non-negativity)
- $d(x, y) = 0$ iff $x = y$ (reflexivity)
- $d(x, y) = d(y, x)$ (symmetry)
- $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality)

The metric properties are important in the use of retrieval systems and should be considered when a similarity measure is defined.
The non-negativity is an essential property if we want to change the measure from *how much two models are dissimilar* to *how much two models are similar*. Moreover, it ensures that also the similarity measure is positive. Indeed, once the distance d is defined, then the *similarity measure* between two models can be defined in terms of their dissimilarity, for instance as:

$$\begin{aligned} similarity &= \frac{1}{1+d} && (5.33) \\ similarity &= 1 - d \qquad \text{if } d \in [0,1] && (5.34) \end{aligned}$$

Reflexivity ensures that just identical models can be zero-dissimilar (or similar at 100%).
Symmetry is another suitable property for any retrieval system. If we compare a model A with a model B, we expect to have the same result if we compare B with A.
While the triangle inequality is useful for indexing a database [133]. Anyhow, in case of partial and local similarities, the triangle inequality can raise some inaccuracy [123]. For example, a man is partially similar to a centaur, a centaur is partially similar to a horse, but the man is completely dissimilar to the horse [13].
For this reason, we define a measure of similarity between assembly models which is not a metric, but that satisfies non-negativity, reflexivity and symmetry properties.

We require also invariance to rotations and translations, since this characteristic guarantees that identical models with different reference frames are considered equal. Thus, in the case of assembly model similarity, the measure has to take into consideration not only the shape of constituent components but also how they are connected together.
In addition, supposing we have a similarity measure between two models, if only a minor component or a minor sub-assembly changes in one of the two assemblies, then the measure between the two models should not change too much.

As said, the similar elements of two assemblies correspond to the common subgraphs between the two graphs associated with the query and the compared (target) model and only a *portion* of the models indicates a local similarity (see section 1.3.1). Thus, the similarity is computed on the subgraphs corresponding to the detected cliques opportunely weighted to obtain the local, partial and global similarity measures.

To evaluate both partial and global similarities on the local one retrieved, we define the similarity measure considering the association graph portion identified by the clique detection opportunely weighted as discussed in subsection 5.2.5.

In order to define a measure on a retrieved clique $(C_{q,k,c_N,c_A})_h$, we define a set of measures ($\mathcal{S}$), where each of them characterizes a single aspect of similarity between two assembly models. In particular, we define the following similarity measures:

$$\mathcal{S} = [\mu^{shape}, \mu^{joint}, \mu^{position}, \mu^{structure}] \tag{5.35}$$

Let see in the following subsections the definition of every single measure, and then, in section 5.2.5, how to combine them to achieve single measures useful for the ranking of retrieved models.

5.2.1 Shape similarity measure: μ^{shape}

The shape similarity measure $\mu^{shape}((C_{q,k,c_N,c_A})_h)$ is based on the shape descriptor of each node involved in the clique. The shape layer of the EAM, as discussed in section 4.2.3 stores the 3D spherical harmonics and the size values, these last ones manage information about the dimensions of a part. Since two objects can have exactly the same shape with different dimension, we decided to use only the 3D spherical harmonics in the evaluation of shape similarity of a part, while the size values should be considered for a further separated measure, as the size measure, which could be used for a refinement of the results.

Two 3D spherical harmonics can be compared in different ways. A method is proposed by Kazhdan et al. [63], where they consider the implications of anisotropy on shape matching functions. Anyway, this evaluation is complex and not suitable for our purpose. Still Kazhdan et al., one year before in [62], illustrated some properties of the spherical harmonics, in particular, they demonstrated that the L_2-distance is the most appropriate for the evaluation of the similarity.

Using L_2-distance, we define the shape similarity of a clique $(C_{q,k,c_N,c_A})_h$ as the average of the similarity of each node in the clique.

μ^{shape}

Definition 5.2.1. Let (n_q^i, n_k^j) be a generic node in the clique $(C_{q,k,c_N,c_A})_h$.
Given $S_q = \Phi_{\mathcal{N}_P}(n_q^i)_{\restriction Shape} \in Shape$ and $S_k = \Phi_{\mathcal{N}_P}(n_k^j)_{\restriction Shape} \in Shape$, the *measure of shape similarity* is

$$\mu^{shape}((C_{q,k,c_N,c_A})_h) = \frac{1}{|\mathcal{N}_{c_h}|} \sum_{(n_q^i, n_k^j) \in (C_{q,k,c_N,c_A})_h} \left[1 - \left| \frac{S_q}{||S_q||_2} - \frac{S_k}{||S_k||_2} \right| \right] \tag{5.36}$$

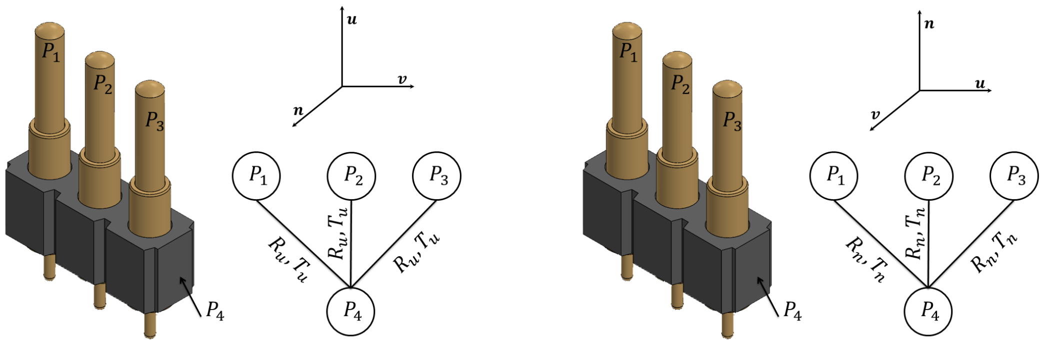
where $|\mathcal{N}_{c_h}|$ represents the number of nodes in the clique $(C_{q,k,c_N,c_A})_h$.

Since the spherical harmonics are normalized, then the norm L-2 has values in $[0, 1]$. This characteristic guarantees that μ^{shape} satisfies the non-negativity, reflexivity and symmetry properties.

5.2.2 Joint similarity measure: μ^{joint}

To assess how much two assemblies are similar in terms of the relative DOF among their parts, we define a measure on the joints, $\mu^{joint}((C_{q,k,c_N,c_A})_h)$.
As illustrated in section 4.2.2, a joint can be risen from contacts of different types (Surface, Curve, Point or UnSolved). In case of joints deriving from contacts of type "Surface", we are able to compute the degree of freedom allowed to the two linked parts, otherwise we have just the information of the type of joint.
Since for joint arcs we have two different types of information (i.e. attributes), we define $\mu^{joint}((C_{q,k,c_N,c_A})_h)$ as a combination of two other measures, one based on the similarity given by the joint of type "Surface" $\mu^{joint}_{surface}((C_{q,k,c_N,c_A})_h)$ and another that considers "Curve" and "Point" type $\mu^{joint}_{curve,point}((C_{q,k,c_N,c_A})_h)$.

The $\mu^{joint}_{surface}((C_{q,k,c_N,c_A})_h)$ should not be affected by different reference frames, i.e. if two assembly models, that identify the same object, are embedded in different reference frames or simply rotated or translated, then their similarity measure should be the same. Anyway, the information about the DOF in the EAM depends on the reference frame of the assembly model. This results that a simple comparison between the DOF of the corresponding elements is not appropriate. For example, if we evaluate the distance between the axes defined by the DOF between the parts P_1 and P_4 in Figure 5.6(a) with the one between the same parts in Figure 5.6(b), i.e. the angle defined by the axis u and n, we will have a variation of 90 degrees even if the objects are the same.



(a) Query model

(b) Target model

Figure 5.6: Example of the same object embedded in two different reference frame

On the contrary, the variations of each pair of axes defined by the DOF between the parts (P_1, P_4), (P_2 P_4) and (P_3 P_4) are the same in both the configurations. Then, to make the similarity measure independent from the reference frame, we consider the variations of the rotation and of translation axes defined by the DOF of a part with all the parts in contact. Therefore, we compute $\mu^{joint}_{surface}$ on the nodes of the cliques instead of using the arcs.

We manage the configurations in which we have several angles defined by the axes of the DOF using matrices to represent all the possible variations of the rotation/translation angles defined by a part. Each element of these matrices identifies the inner product of a pair of axes, as specified in definition 5.2.2.

Rotation and translation variation matrices

Definition 5.2.2. Given

- $Rot(n) = \{\bigcup_h R^h \quad \forall n^h : (n, n^h) \in \mathcal{A}_J\}$ the set of all the joint rotations with respect to a node n,
- $Tra(n) = \{\bigcup_h T^h \quad \forall n^h : (n, n^h) \in \mathcal{A}_J\}$ the set of all the joint translations with respect to a node n,

$Var_{Rot}(n)$ is the matrix of the variations of rotations related to the node n, where its generic element is defined as: $(Var_{Rot}(n))_{i,j} = r_i \cdot r_j$,
where $r_i, r_j \in Rot(n)$.

$Var_{Tra}(n)$ is the matrix of the variations of translations related to the node n, where its generic element is defined as: $(Var_{Tra}(n))_{i,j} = t_i \cdot t_j$,
where $t_i, t_j \in Tra(n)$.

The choice of using the inner product is led by the fact that the axes are normalized, then it corresponds to the cosine of the angle between the considered axes. We do not use the vector product, which represents the sine of the angle between the considered axes, since angles with different spans can have the same sine values, for instances $sin(30) = sin(150) = 0.5$. We do not have the same problem with the cosine since the considered angles are included in $(0, \pi]$.

The final variation for a node in the clique is computed by the average of this matrix elements, where with *average* of a matrix we mean the arithmetic mean of the elements in the matrix divided by the number of elements, i.e.

$$\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} x_{i,j} \qquad\qquad \forall x_{i,j} \in M \tag{5.37}$$

where N is the dimension of the matrix M.
Using this approach, when a single arc is incident to a node, its variation is 1.
Moreover, it also takes into consideration the cases presenting contacts and joints having type "UnSolved". When we have a contact or a joint of "Unsolved" type, we do not have information about its DOF and the number of axis generated by the DOF which concur to define the variation matrix is the only difference between the two models. We use this information to distinguish these cases, dividing the average of each variation matrix by the number of rows of the matrix.

Considering joints arisen from curves or points, we do not have any DOF to compare. In this case, we discriminate situations where joints are formed by the same type of contacts. Then we define a joint measure $\mu^{joint}_{curve,point}((C_{q,k,c_N,c_A})_h)$ that assigns the maximum values 1 if the joints are of the same type (both Curve or Point) and a lower value if the joints are of different type. The lower value is set to 0.8 and it was chosen empirically in order to decrease the measure, but not too much.

We define the following joint measures:

μ^{joint}

Definition 5.2.3.

$$\mu^{joint}((C_{q,k,c_N,c_A})_h) = \begin{cases} \mu^{joint}_{surface}((C_{q,k,c_N,c_A})_h) & \text{if } \mu^{joint}_{curve,point}((C_{q,k,c_N,c_A})_h) = 0 \\ \mu^{joint}_{curve,point}((C_{q,k,c_N,c_A})_h) & \text{if } \mu^{joint}_{surface}((C_{q,k,c_N,c_A})_h) = 0 \\ \frac{\mu^{joint}_{surface}((C_{q,k,c_N,c_A})_h)+\mu^{joint}_{curve,point}((C_{q,k,c_N,c_A})_h)}{2} & \text{otherwise} \end{cases} \tag{5.38}$$

$\mu^{joint}_{surface}$

Definition 5.2.4.

$$\mu^{joint}_{surface}((C_{q,k,c_N,c_A})_h) = \frac{1}{|\mathcal{N}_{c_h}|} \sum_{(n^i_q,n^j_k)\in(C_{q,k,c_N,c_A})_h} \left[1 - \frac{d_{Rot}((n^i_q,n^j_k)) + d_{Tra}((n^i_q,n^j_k))}{2}\right] \tag{5.39}$$

where,

- $|\mathcal{N}_{c_h}|$ represents the number of nodes in the clique $(C_{q,k,c_N,c_A})_h$
- $d_{Rot}((n^i_q,n^j_k)) = ||\frac{\sigma(Var_{Rot}(n^i_q))}{|Rot(n^i_q)|} - \frac{\sigma(Var_{Rot}(n^j_k))}{|Rot(n^j_k)|}||$
- $d_{Tra}((n^i_q,n^j_k)) = ||\frac{\sigma(Var_{Tra}(n^i_q))}{|Tra(n^i_q)|} - \frac{\sigma(Var_{Tra}(n^j_k))}{|Tra(n^j_k)|}||$
- $\sigma(M)$ is the average of matrix M,
- $Var_{Rot}(n^i_q)$ is a matrix of the variation of rotations, related to the node n^i_q of the query model. If $Rot(n^i_q) = \emptyset$ then we assign $\frac{\sigma(Var_{Rot}(n^i_q))}{|Rot(n^i_q)|} = 1$
- $Var_{Rot}(n^j_k)$ is a matrix of the variation of rotations, related to the node n^j_k of the target model. If $Rot(n^j_k) = \emptyset$ then we assign $\frac{\sigma(Var_{Rot}(n^j_k))}{|Rot(n^j_k)|} = 1$
- $Var_{Tra}(n^i_q)$ is a matrix of the variation of translation, related to the node n^i_q of the query model. If $Tra(n^i_q) = \emptyset$ then we assign $\frac{\sigma(Var_{Tra}(n^i_q))}{|Tra(n^i_q)|} = 1$
- $Var_{Tra}(n^j_k)$ is a matrix of the variation of translation, related to the node n^j_k of the target model. If $Tra(n^j_k) = \emptyset$ then we assign $\frac{\sigma(Var_{Tra}(n^j_k))}{|Tra(n^j_k)|} = 1$.

$\mu^{joint}_{curve,point}$

Definition 5.2.5.

$$\mu^{joint}_{curve,point}((C_{q,k,c_N,c_A})_h) = \frac{1}{|\mathcal{N}_{c_h}|} \sum_{((n^i_q,n^j_k),(n^l_q,n^h_k))\in(C_{q,k,c_N,c_A})_h} \left[1 - d_{edge}((a^{ij}_q, a^{lh}_k))\right] \quad (5.40)$$

where,

- $|\mathcal{N}_{c_h}|$ represents the number of nodes in the clique $(C_{q,k,c_N,c_A})_h$
- $a^{ij}_q = (n^i_q, n^j_q)$ and $a^{lh}_k = (n^l_k, n^h_k)$
- $d_{edge}((a^{ij}_q, a^{lh}_k)) = \begin{cases} 1 & \text{if } J^{ij}_q \neq J^{hl}_k \\ 0.8 & \text{Otherwise} \end{cases}$
- $J^{ij}_q = \Phi_{\mathcal{A}_J}(a^{ij}_q)_{\restriction Joint\_Type} \in Joint\_Type$
- $J^{hl}_k = \Phi_{\mathcal{A}_J}(a^{hl}_k)_{\restriction Joint\_Type} \in Joint\_Type$

For sake of comprehension, we compute $\mu^{joint}_{surface}((C_{q,k,c_N,c_A})_h)$ for the clique in Figure 5.7 depicted from the comparison of the graph G_q and G_k depicted in Figure 5.8, where R_x and R_y denote a rotation along axis x and y respectively.

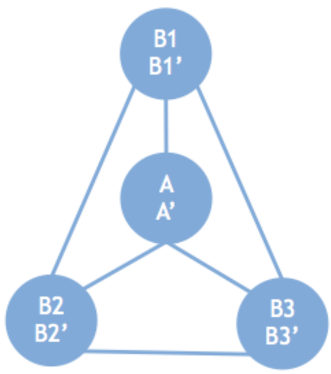


Figure 5.7: A possible clique resulting from the comparison of G_q and G_k of Figure 5.8

The evaluation of joint similarity is achieved computing, for each node in the clique, the difference of the reciprocal arrangement of the directions of the allowed translations and rotations in the original attributed graphs. Said differently, starting from a node in the associated graph (Figure 5.7), if we consider the rotations and the translations of the corresponding nodes in the compared graphs (Figure 5.8), we have:

$$\begin{aligned} Rot(B1) = Rot(B2) = Rot(B3) = Rot(B1') &= \{R_x\} \\ Rot(B2') = Rot(B3') &= \{R_y\} \\ Rot(A) &= \{R_x, R_x, R_x\} \\ Rot(A') &= \{R_x, R_y, R_y\} \end{aligned}$$

(a) G_q (b) G_k

Figure 5.8: Example of two attributed graphs with different attributes on arcs

Now, we compute the variation of the angles between every pair of axes in $Rot(n)$ for each node $n \in \{A, A', B1, B2, B3, B1', B2', B3'\}$.

$$Var_{Rot}(A) = \begin{array}{c|ccc} & x=(1,0,0) & x=(1,0,0) & x=(1,0,0) \\ \hline x=(1,0,0) & (1,0,0)\cdot(1,0,0) & (1,0,0)\cdot(1,0,0) & (1,0,0)\cdot(1,0,0) \\ x=(1,0,0) & (1,0,0)\cdot(1,0,0) & (1,0,0)\cdot(1,0,0) & (1,0,0)\cdot(1,0,0) \\ x=(1,0,0) & (1,0,0)\cdot(1,0,0) & (1,0,0)\cdot(1,0,0) & (1,0,0)\cdot(1,0,0) \end{array}$$

$$= \begin{array}{c|ccc} & x=(1,0,0) & x=(1,0,0) & x=(1,0,0) \\ \hline x=(1,0,0) & 1 & 1 & 1 \\ x=(1,0,0) & 1 & 1 & 1 \\ x=(1,0,0) & 1 & 1 & 1 \end{array}$$

$$Var_{Rot}(A') = \begin{array}{c|ccc} & x=(1,0,0) & y=(0,1,0) & y=(0,1,0) \\ \hline x=(1,0,0) & (1,0,0)\cdot(1,0,0) & (1,0,0)\cdot(0,1,0) & (1,0,0)\cdot(0,1,0) \\ y=(0,1,0) & (0,1,0)\cdot(1,0,0) & (0,1,0)\cdot(0,1,0) & (0,1,0)\cdot(0,1,0) \\ y=(0,1,0) & (0,1,0)\cdot(1,0,0) & (0,1,0)\cdot(0,1,0) & (0,1,0)\cdot(0,1,0) \end{array}$$

$$= \begin{array}{c|ccc} & x=(1,0,0) & y=(0,1,0) & y=(0,1,0) \\ \hline x=(1,0,0) & 1 & 0 & 0 \\ y=(0,1,0) & 0 & 1 & 1 \\ y=(0,1,0) & 0 & 1 & 1 \end{array}$$

$$Var_{Rot}(B1) = \begin{array}{c|c} & x=(1,0,0) \\ \hline x=(1,0,0) & (1,0,0)\cdot(1,0,0) \end{array}$$

$$= \begin{array}{c|c} & x=(1,0,0) \\ \hline x=(1,0,0) & 1 \end{array}$$

$$Var_{Rot}(B2') = \begin{array}{c|c} & y=(0,1,0) \\ \hline y=(0,1,0) & (0,1,0)\cdot(0,1,0) \end{array}$$

$$= \begin{array}{c|c} & y=(0,1,0) \\ \hline y=(0,1,0) & 1 \end{array}$$

So we have:

$$\begin{aligned}
\frac{\sigma(Var_{Rot}(B1))}{|Rot(B1)|} &= \frac{1}{1} = 1 \\
\frac{\sigma(Var_{Rot}(B2'))}{|Rot(B2')|} &= \frac{1}{1} = 1 \\
\frac{\sigma(Var_{Rot}(A))}{|Rot(A)|} &= \frac{1}{3} = 0.33 \\
\frac{\sigma(Var_{Rot}(A'))}{|Rot(A')|} &= \frac{0.55}{3} = 0.18
\end{aligned}$$

In the end, since no translation occurs, the distance $d_{Tra}(*,*)$ is 0.

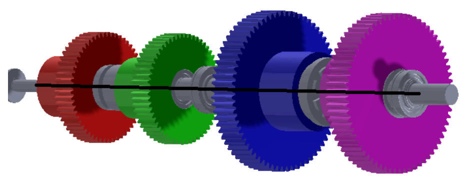
Following the definition 5.2.5, for the clique C in the example of Figure 5.7, we have

$$\begin{aligned}
\mu^{joint}_{surface}(C) &= \frac{1}{4}\Big[\Big(1 - \frac{d_{Rot}(B1,B1') + d_{Tra}(B1,B1')}{2}\Big) + \Big(1 - \frac{d_{Rot}(B2,B2') + d_{Tra}(B2,B2')}{2}\Big) \\
&\quad + \Big(1 - \frac{d_{Rot}(B3,B3') + d_{Tra}(B3,B3')}{2}\Big) + \Big(1 - \frac{d_{Rot}(A,A') + d_{Tra}(A,A')}{2}\Big)\Big] \\
&= \frac{1}{4}\Big[\Big(1 - \frac{0+0}{2}\Big) + \Big(1 - \frac{0+0}{2}\Big) + \Big(1 - \frac{0+0}{2}\Big) + \Big(1 - \frac{0.15+0}{2}\Big)\Big] \\
&= \frac{1}{4}\Big[1+1+1+0.92\Big] = 0.98
\end{aligned}$$

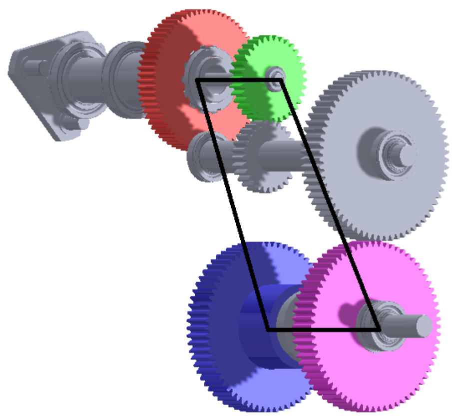
Finally, we have $\mu^{joint} = 0.98$ since in the considered example there was no contact of curve or point type, i.e. $\mu^{joint}_{curve,point} = 0$.

5.2.3 Position similarity measure: $\mu^{position}$

Another salient characteristic which increases or decreases the level of similarity between two assembly models is the relative arrangement of the assembly components. For instance, if we consider the assemblies in Figure 5.9, they have high values of shape and joint similarities and what discerns the two models is the arrangement of the parts. The pattern information in the EAM cannot disambiguate these configurations, since the colored parts are not repetitions of the same part. Then, here we need a criterion to characterize the position of not repeated parts.



(a)



(b)

Figure 5.9: Example of assembly models with similar parts arranged in different configurations

To solve this issue, we consider the *directional versor* between the center of gravity of a part P and the center of gravity of each part P_i in the clique which is not in contact with P. The use of versors instead of segments which links two centers of gravity makes the measure size independent. Anyway even using versors, their comparison is affected by the reference frame. To overcome this problem, the relative position similarity is computed following the same approach adopted for the computation of the joint similarity in case of surface type. In this case, the directional versors are the analogous of the rotation/translation axes defined by the DOF used in $\mu^{joint}_{surface}$.

$\mu^{position}$

Definition 5.2.6.

$$\mu^{position}((C_{q,k,c_N,c_A})_h) = \frac{1}{|\mathcal{N}_{c_h}|} \sum_{(n^i_q,n^j_k)\in(C_{q,k,c_N,c_A})_h} 1 - d_{Dir}((n^i_q, n^j_k)) \tag{5.41}$$

where,

- $|\mathcal{N}_{c_h}|$ represents the number of nodes in the clique $(C_{q,k,c_N,c_A})_h$
- $d_{Dir}((n^i_q, n^j_k)) = |\sigma(Var_{Dir}(n^i_q)) - \sigma(Var_{Dir}(n^j_k))|$
- $\sigma(M)$ is the average of a matrix M
- $Var_{Dir}(n^i_q)$ is a matrix of the variation of the directional versors between the centers of gravity of the parts corresponding to the node n^i_q and the nodes n^l_q such that $(n^i_q, n^l_q) \notin \mathcal{A}_C$,
- $Var_{Dir}(n^j_k)$ is a matrix of the variation of the directional versors between the centers of gravity of the parts corresponding to the node n^j_k and the nodes n^l_k such that $(n^i_q, n^l_q) \notin \mathcal{A}_C$.

5.2.4 Structure similarity measure: $\mu^{structure}$

As discussed in section 1.2.2, an assembly model can have different product structures according to the user design intent, then we aim to define the structure measure $\mu^{structure}((C_{q,k,c_N,c_A})_h)$.

The EAM encodes the hierarchical structure of an assembly model by the arcs $\mathcal{A}_S$ which represent the relation "made-of" among the components of a model. In this case, for each pair of associated nodes (n_q^i, n_k^j) and (n_q^l, n_k^h) in the clique, we compare the structural relations of the pairs of nodes (n_q^i, n_q^l) and (n_k^j, n_k^h) that have generated the association node in the clique. It means that we need to verify if the nodes n_q^i, n_q^l belong (or not belong) to the same sub-assembly in G_q and if the nodes n_k^j, n_k^h belong (or not belong) to the same sub-assembly in G_k.

Using a distance, which assigns 1 if the pair of nodes $(n_q^i,\, n_q^l)$ has a different relation from the pair $(n_k^j,\, n_k^h)$ or 0 otherwise, we define the structure similarity measure of a clique $(C_{q,k,c_N,c_A})_h$ as follow:

$\mu^{structure}$

Definition 5.2.7.

$$\mu^{structure}((C_{q,k,c_N,c_A})_h) = \frac{1}{|\mathcal{N}_{c_h}|^2} \sum_{((n_q^i,n_k^j),(n_q^l,n_k^h))\in(C_{q,k,c_N,c_A})_h} \left[1 - d_{Str}\Big((n_q^i, n_k^j), (n_q^l, n_k^h)\Big)\right] \tag{5.42}$$

where,

- $|\mathcal{N}_{c_h}|$ represents the number of nodes in the clique $(C_{q,k,c_N,c_A})_h$
- $$d_{Str}(((n_q^i, n_k^j), (n_q^l, n_k^h)) = \begin{cases} 1 & \text{if } \Big[(\exists n_q^* \in \mathcal{N}_q) \text{ and } (\nexists n_k^* \in \mathcal{N}_k)\Big] \text{ or} \\ & \quad \Big[(\nexists n_q^* \in \mathcal{N}_q) \text{ and } (\exists n_k^* \in \mathcal{N}_k)\Big] \text{ such that} \\ & \\ & \quad \Big[\Big((n_q^i, n_q^*), (n_q^l, n_q^*) \in \mathcal{A}_{S_q}\Big) \text{ and} \\ & \quad \Big((n_k^j, n_k^*), (n_k^h, n_k^*) \in \mathcal{A}_{S_k}\Big)\Big] \\ 0 & \text{Otherwise} \end{cases}$$

5.2.5 Combination of similarity measures for different type of similarities

So far, we have defined measures of similarity for the various aspects characterizing an assembly. Then, we need to combine them in order to provide an unique measure for ranking the models in the retrieval framework. The composition of the similarity measures should also consider the possibility to weight each similarity value for a certain relevance factor chosen in the query or during the browsing of the results.

In addition, since these values rely only on local similarity, i.e. they express how much the matching portion of the two assembly models is similar, their simple combination does not provide information on how the parts are globally or partially similar. This can be achieved by weighting the local measure according to the matched portion of the query and of the target models.
We call this weight *coverage factor*, which is the number of nodes retrieved in the clique over the number of nodes in the query and in the target models.

In the following we indicate with C_h the clique $(C_{q,k,c_N,c_A})_h$.

Local, partial and global similarity

Definition 5.2.8. Being

- $G_q = \mathcal{G}(\mathcal{N}_q, \mathcal{A}_q, \Phi_{\mathcal{N}_q}, \Phi_{\mathcal{A}_q})$ the attributed multi-graph representation of the EAM of the query model $\mathcal{M}_q$
- $G_k = \mathcal{G}(\mathcal{N}_k, \mathcal{A}_k, \Phi_{\mathcal{N}_k}, \Phi_{\mathcal{A}_k})$ the attributed multi-graph representation of the EAM of the target model $\mathcal{M}_t$,
- $(C_{q,k,c_N,c_A})_h$ the MC of the association graph resulting from the G_q and G_k according to the criteria c_N and c_A,

The measure of similarity between models $\mathcal{M}_q$ and $\mathcal{M}_k$ can be evaluated in three different ways:

- *local similarity* using the similarity measure η:

$$\begin{aligned} \eta \quad &: \quad \mathcal{D}_{q,k} \times \mathcal{W} \longrightarrow \mathbb{R} \\ \eta(C_h, w) \quad &= \quad \frac{\sum_{i \in \mathcal{M}} w^i \mu^i(C_h)}{\sum_{i \in \mathcal{M}} w^i} \end{aligned} \tag{5.43}$$

with $\mathcal{M} = \{shape, joint, position, structure\}$ and
$\mathcal{W} = \{w^{shape}, w^{joint}, w^{position}, w^{structure}\}$ where $\forall\ w^i \in \mathcal{W},\ w_i \in [0, 1]$

- *partial similarity* using the similarity measure φ:

$$\begin{aligned} \varphi \quad &: \quad \mathcal{D}_{q,k} \times \mathcal{W} \longrightarrow \mathbb{R} \\ \varphi(C_h, w) \quad &= \quad \frac{|C_h|}{|\mathcal{N}_q|} \eta(C_h, w) \end{aligned} \tag{5.44}$$

- *global similarity* using the similarity measure ϕ:

$$\begin{aligned} \phi \quad &: \quad \mathcal{D}_{q,k} \times \mathcal{W} \longrightarrow \mathbb{R} \\ \phi(C_h, w) \quad &= \quad \frac{2|C_h|}{|\mathcal{N}_q| + |\mathcal{N}_k|} \eta(C_h, w) \end{aligned} \tag{5.45}$$

The assembly local similarity measure proposed is defined as the weighted average of the single similarity measures. The weights are chosen in the query or set during the browsing of the results. An interesting further study concerns the capability of assigning the most suitable weights according to a particular usage scenario.

In the end, global and partial similarities are defined starting from the local similarity and balancing it over the number of components of the query and of the compared models. A more refined weight could consider the "relevance" of the parts, for instance in terms of their volume or of their class membership (i.e. a rivet could be negligible respect to a gear).

5.3 Conclusion

In this chapter, we have seen that to compare two assembly models through their EAMs we need to solve a MCS problem. This problem is translated in a MC problem, which is NP-complete. Hence, to solve it, an heuristic algorithm has been proposed. In addition, to validate the correctness and the usefulness of the data included in the EAM, an exact matching method has been included as well.

The MC problem relies on the construction of an association graph, which associates nodes and arcs according to several criteria of compatibility that have been described in this chapter. Reducing the number of nodes and arcs in the association graph decreases the complexity of the MC problem. Then in the future, other criteria of association can be studied. For instance, the association graph could have several levels of details. To a first skimming, nodes corresponding to the same functional set can be collapsed in a single association node, and if a clique including it has been found then the matching can go in a deeper level of detail analyzing the functional sets similarity.

In addition, since two assembly models can be similar according to different criteria, in this chapter, we have proposed measures, which assess the assembly similarity according to different criteria. In particular, we have defined shape, joint, position and structure similarity measures. While the measures of shape, joint and structure evaluate information introduced in the EAM, the position measure has been introduced to distinguish arrangement of not repeated elements. Thus a possible enhancement of the EAM could concern the inclusion of information about the mutual position of the components in the assembly models. For instance, we could introduce if two shafts are coaxial or parallel. Moreover, we think that this information can be added in the compatibility conditions for the construction of the association graph, which can help to reduce the number of arcs. In the end, other measures could be further defined as a *size measure* which assesses two models according to their dimensions, at the current status of this thesis the size is only involved in the association graph creation.

Next chapter will report some results of the comparison between assemblies based on the concepts defined in this chapter, showing examples of searches according to different criteria and the values of the introduced measures.

6

Results and discussion of the developed prototype

This last chapter demonstrates the effectiveness of the proposed retrieval system and discusses the accuracy of the defined similarity measures. In particular, this chapter introduces the dataset used for the tests, it details the developed prototype and how the results are proposed to users. Finally, it presents and discusses results on several test cases.

Contents

6.1 Model dataset organization

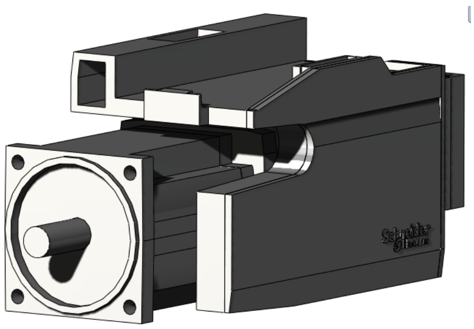
The evaluation of the effectiveness of a retrieval system, i.e understanding if it yields to accurate and satisfying results, can be measured in terms of precision and recall, as discussed in section 3.1.2. This assessment requires a ground-truth, i.e. a collection of models and a set of meaningful queries over which the models are labeled as “relevant” or “non relevant”. The most known mechanical shape benchmarks in literature are the Princeton Shape Benchmark (PSB) [112], the National Design Repository (NDR) [99] and the Engineering Shape Benchmark (ESB) [58]. These benchmarks are not proper for our purpose, since they classify just parts and do not consider assembly models, and so far, no public database exists to evaluate and compare assembly retrieval systems [51, 96].

This lack is due to some difficulties to:

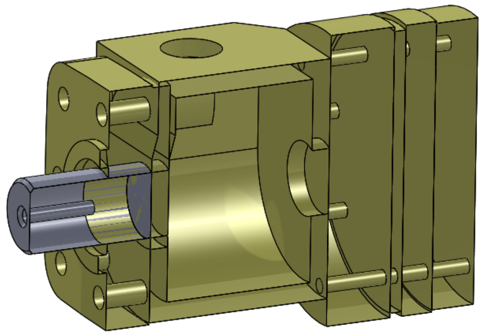
(i) **Assign relevant/irrelevant label to target models according to a query**
It is not straightforward identifying the assembly models that should be relevant (i.e. retrieved) according to a specified query. Main reasons that originate this complexity can be identified in the different types of similarity of a model, the purpose of the retrieval (i.e. the desired information to access) and the background of the user who queries the database. Each of these elements may influence the identification of a model as relevant or not. Moreover, the assembly complexity can require some time and specific engineering knowledge to browse through the components checking their similarities.

(ii) **Define partial similarity**
The complexity of the identification of relevant models increases when the query is included in the target model. For instance, Chen et al. [28] assumed that a model is relevant according to a query if there exists a sub-assembly in the target model which is similar to the query model. This practice may facilitate the creation of an assembly benchmark, but it raises the models considered as false-positive, i.e. two models that represent the same object can be considered one relevant and the other non relevant simply according to their hierarchical structure. Anyhow, this practice can be reasonable only if the user specifies the structure as a similarity criterion.

(iii) **Get realistic models**
Most of online repositories do not provide complex assemblies and many of the available models are inaccurate, e.g. with unrealistic simplifications or with sub-assemblies components collapsed in single parts, which makes the comparison quite difficult. Figure 6.1 shows two gear pump models downloaded from GrabCad [2]. In the first case, the model is reduced to a part, while in the second case, it has only the external components, i.e. from the section view, we can see that there is nothing inside the model. This kind of models is not suitable to evaluate the similarity of assembly models. They can be used just to compare the external shape of an assembly model, with no relevance on the part relationships and their interaction.



(a) Gear pump model reduced in single a part



(b) Gear pump model with simplified components

Figure 6.1: Example of two gear pump models downloaded from GrabCad [2]

Despite these difficulties, some benchmarks have been defined. To test assembly retrieval systems, the dataset of models proposed by Iwaya et al. [55] was organized by twenty students of the Polytechnic School of the University of Sao Paulo and the Santa Caterina State University. Hu et al. [51] composed a dataset of 614 assemblies containing 5100 parts of which 2814 are unique, while Chen et al. [28] collected 2249 parameterized assembly models and 10062 parts model, and two graduate students were invited from Mechanical Engineering Department to label all assemblies in the library as relevant or irrelevant to the queries.
More recently, Katayama and Sato [60] have evaluated their approach over 3 models (a clutch, a die and a gear) with 5 different structures.

Anyhow, these benchmarks are not public, then it is arduous to try to compare our method with other ones present in the literature. Hence, in this thesis, we built our own dataset of assembly models over which we can assess the effectiveness of the proposed approach.

We have defined our dataset of assembly models focusing on gearboxes. This decision is led by the fact that gearboxes are widely used in industrial applications [72]. Indeed, their presence inside many assembly models (as helicopters, wind turbines, bucket wheel excavators, tracked loaders and milling machines) makes them suited to evaluate partial and local retrieval. Moreover, according to the arrangement of the gears, gearboxes can be classified in different categories (e.g. simple gear train, compound gear train, reverted gear train and planetary gear train) [72]. This variety of configurations is useful for the effectiveness evaluation because it offers the possibility of highlighting the retrieval system capabilities of exploiting the arrangement of the parts.
In addition, to test the precision of the proposed retrieval system, i.e. its ability to discard non relevant models, we have included in our dataset other models whose shape and functionality are different from the ones of gearboxes.

In this work, we collected 140 assembly models focusing on the quality of the models to minimize problems deriving from inaccuracies and unrealistic simplification. Table 6.1 illustrates the used dataset. In the future, to ease further comparisons with our work, we aim to make available the dataset and define its own ground truth.

Table 6.1: 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	**140**

6.2 Developed prototype

The framework illustrated in chapter 3 has been implemented to validate the assembly descriptor presented in chapter 4 and the evaluation of assembly similarities described in chapter 5.
In the following sections, we provide details on the development environment (section 6.2.1) and on the user interface for the visualization and browsing of the results (section 6.2.2).

6.2.1 Development environment

CAD systems use different proprietary file formats to store all the information specified by users when modeling specific parts. The content of the files 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 “Standard for the Exchange of Product model data”, known as STEP format, as representation format of the assembly models and to get access to the associated information.

STEP format has numerous Application Protocols (APs), which specify what kind of information is managed. Nowadays, commercial CAD systems mainly support the following APs:

- **AP 203: Configuration controlled 3D designs**
  It is used to exchange geometry, product structure and configuration management data. Edition 2 adds tolerances, construction history, layers and colors [46].
- **AP 214: Core data for automotive mechanical design processes**
  It is used to exchange mechanical geometry, product structure, configuration management assemblies, suppliers, tolerances and other information. It includes drawing exchange ensuring that a complete manufacturing technical data package can be exchanged [46].

Theoretically, this standard supports the representation and exchange of the kinematic relationships between components of an assembly model and its constraints. However, most of CAD systems do not contain the latter ones and generate files that do not incorporate kinematic relationships and constraints. Moreover, during the PDP, interference information may be not stored on purpose, because the simulation process simplifies assembly models removing negligible components and this suppression can produce loss of consistency in the constraint definitions [111].

To allow the access to the information present in STEP file, we exploit the Application Programming Interface (API) of a commercial CAD software. In particular, we employed SoldWorks®, since it allows to access at the list of components (parts and sub-assemblies) of an assembly model, to get the parent and children of a component in the assembly structure and to read the B-rep of each part and their mating information.
Alternative solutions have been analyzed such as CATIA, whose documentation is available only after the purchase of the software; ACIS, which offers the advantage of being cross-platform but it does not have a native graphical user interface; and FreeCAD, which provides information about the parts of an assembly model without their sub-assembly grouping.

Hence, the method described in the previous chapters has been integrated as a plugin of the CAD system SolidWorks®, where a plug-in is a software component that adds a specific set of features to an existing software application. The plugin is registered and activated when SolidWorks®starts, becoming visible in the Taskpane tab.

The Graphical User Interface (GUI) of the plug-in was realized in C# by using Microsoft Visual Studio 2013, while the creation of the EAM description has been implemented in a multi-module prototype system, as described in section 4.4.2, in Microsoft Visual C# 2013 and exploiting the API of SolidWorks®.

A snapshot of the developed interface is illustrated in Figure 6.2, where different sheets are shown.

In the first sheet *Assembly Descriptor* (Figure 6.2(a)), the user can compute the EAM either of a single or of a set of CAD models and choose where to store these EAM descriptors.

In the *Search Criteria* (Figure 6.2(b)) sheet, the user can set several similarity options, as described in the following:

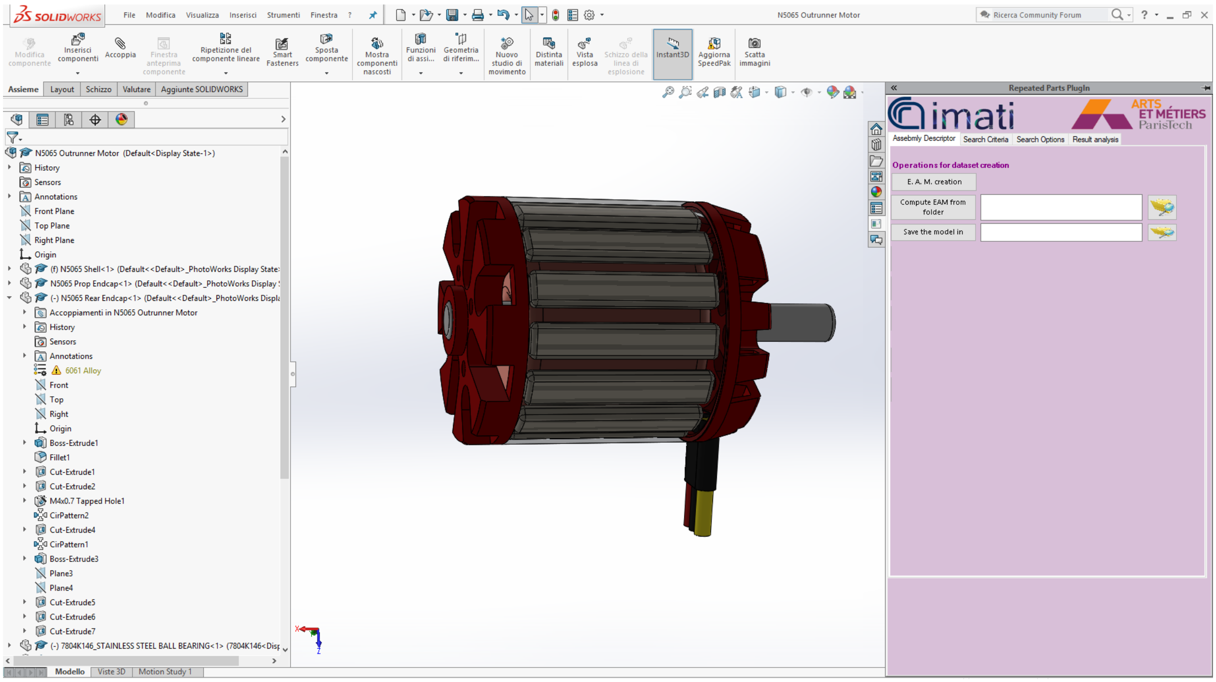
(i) a **pattern filter**, which limits the matching procedure on target models whose number of patterns of a certain type belongs to the range specified by the user. In addition, it specifies if the shape of the elements of the patterns has to be considered during the matching procedure,

(ii) the **similarity criteria**, which are divided into similarity on parts and on assembly relationships that correspond to the vectors c_N and c_A described in section 5.1.2. Note that the system enables the setting of a threshold value for the shape and the size criteria of the parts, which represents the percentage of similarity to be satisfied, e.g. shape similarity at 80%,

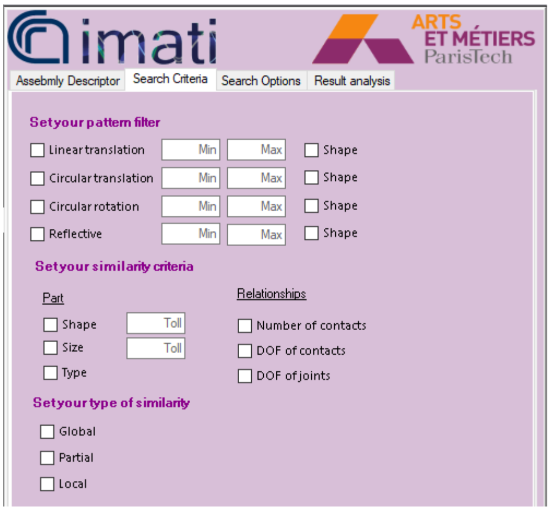
(iii) the **type of similarity**, which specify the measure (global, partial or local as defined in section 5.2.5) to use for the ranking of the models.

In the sheet *Search Options* (Figure 6.2(c)), the EAM of the query model and the folder including the EAMs of target models are selected.
So far, all the results are stored and can be inspected in the sheet *Result analysis* (Figure 6.2(d)). In this visualization, the user selects the name of the query model and of the target model, then the retrieved sets of similar components are listed with their corresponding measures.

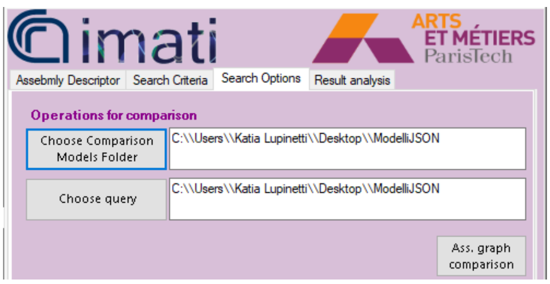
The different sheets are further detailed in section 6.2.2.



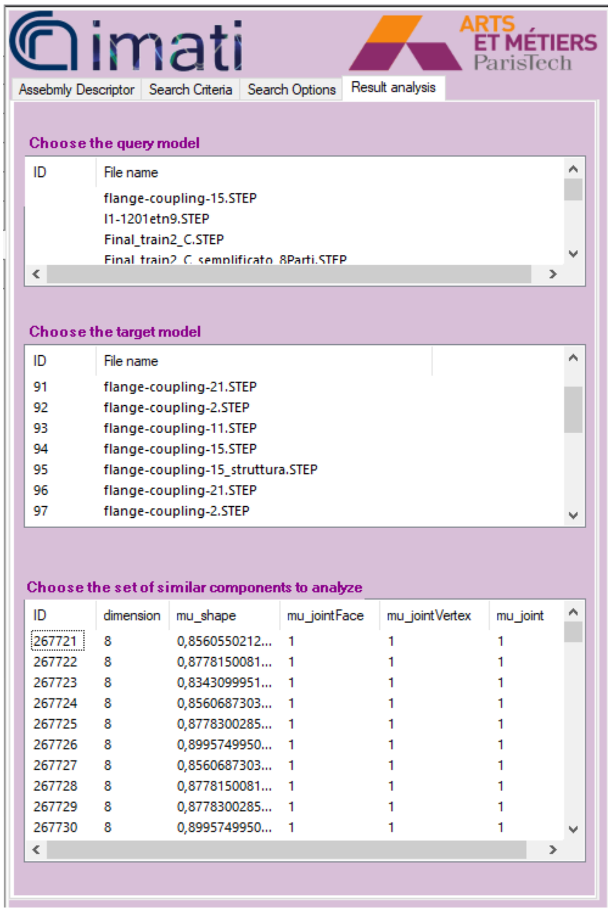
(a) Assembly descriptor sheet



(b) Search criteria sheet



(c) Search options sheet



(d) Result analysis sheet

Figure 6.2: Illustration of the developed user interface

Once the data have been computed, the EAM are stored in JavaScript Object Notation (JSON) format. JSON is a lightweight data-interchange format, based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999 [1]. This format is completely language independent and is based on two universal data structures: a collection of name/value pairs and an ordered list of values. These properties make JSON an ideal data-interchange language.
Figure 6.3 shows the EAM in JSON format of the model used as query for the example in section 6.3.1. Here we can observe that the EAM is translated in a list of nodes with several attributes specified by a key-value (Figure 6.3(a)) and a list of arcs defined by their source and target node and their attributes (Figure 6.3(b)).
In the example in Figure 6.3(a) a node part, its statistics and shape information are displayed; while in Figure 6.3(b) an arc of joint type is illustrated where its DOF contains arrays of allowed translations and rotations, in particular a rotation associated to a planar face is detailed.

(a) Attributes on part node

(b) Attributes on joint arc

Figure 6.3: Example of EAM in JSON format

The matching and the similarity assessment modules are developed by using Java and are invoked during the retrieval as a jar file.

6.2.2 Visualization and browsing of the results

The obtained results are stored by using Mysql as database system and for their analysis a browser view has been implemented. A first simple way to represent the results is illustrated in the sheet of the UI in Figure 6.2(d). Since this type of display is not very intuitive and user-friendly to browse the results, we developed multiple dynamic web pages that are based on HTML5, jQuery, Ajax and PHP, where a X3D library is used for the model visualization. In this way, the user will not see a list of names, but a 3D overview of the target models with their matched components highlighted.

Selected a query, for each retrieved model a frame is created (see Figure 6.4), which reports the name of the target model, a 3D view of it and an histogram. In the 3D view the matched components are colored in blue while the rest of the model in red. The bars of the histogram indicate the values of the local (orange bar), global (green) and partial (purple) similarity measures. These measures are computed as discussed in section 5.2.5. Right now, the weights are manually added.

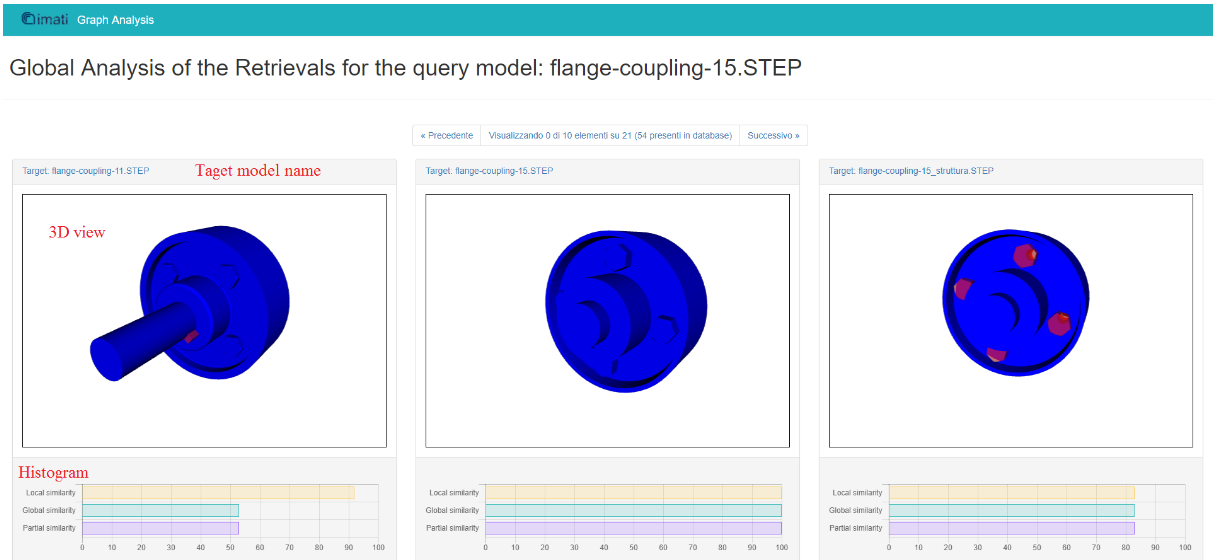
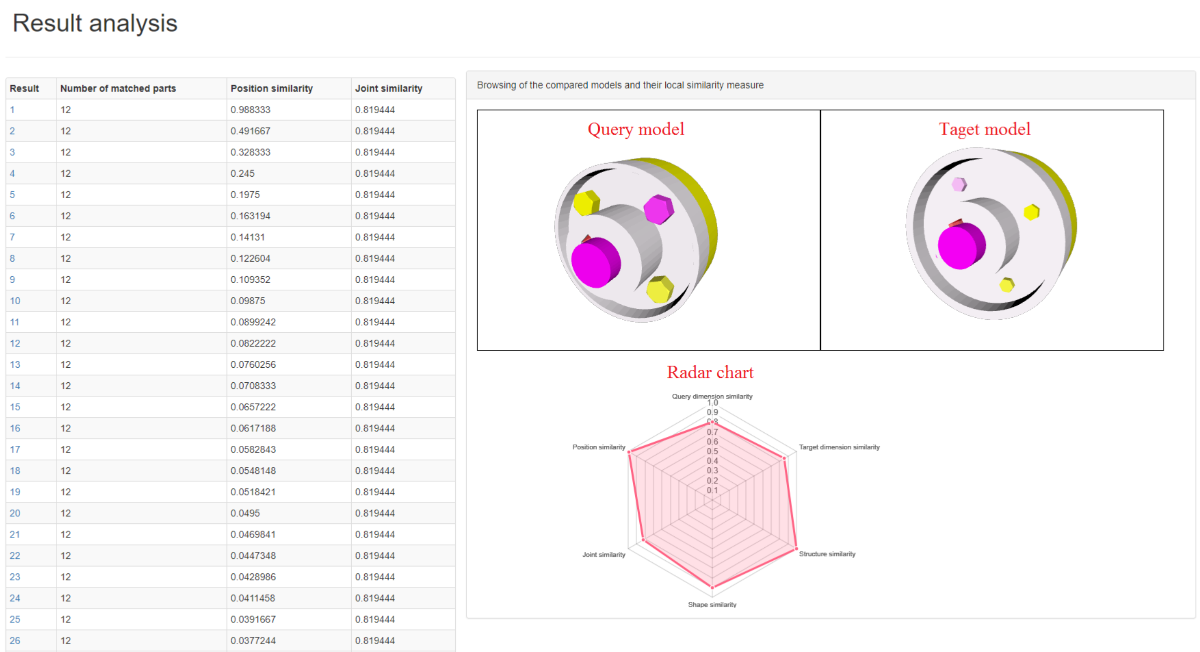


Figure 6.4: Illustration of the target models for a given query

Each model can be further analyzed in another page (see Figure 6.5), where the query and the target models are displayed. In this 3D view, each pair of matched parts is highlighted by a different color, and the values of the single measures are reported in a radar chart. In the chart, each axis identifies a different criterion of similarity defined in section 5.2, i.e. μ^{shape}, μ^{joint}, $\mu^{position}$, $\mu^{structure}$.

In addition, we report also two values which indicate the number of matched elements over the query model ($\frac{|N_{clique}|}{|N_{query}|}$) and the number of matched elements over the target model ($\frac{|N_{clique}|}{|N_{target}|}$). This values are indicated in the radar chart as *query dimension similarity* and *target dimension similarity* respectively. We have added this information in the similarity chart, to improve the informative content. If the values of all the similarities are close, then we know that almost all the elements in the query and the target model are matched. On the contrary, if the partial and the global similarities are lower than the local one, we know that not all the query or target model are matched, but we do not know in which measure. For this reason, we decided to add also these two values to the chart of similarity measures which provide details on the comparison results of the two assembly models.


Result analysis

Result	Number of matched parts	Position similarity	Joint similarity
1	12	0.988333	0.819444
2	12	0.491667	0.819444
3	12	0.328333	0.819444
4	12	0.245	0.819444
5	12	0.1975	0.819444
6	12	0.163194	0.819444
7	12	0.14131	0.819444
8	12	0.122604	0.819444
9	12	0.109352	0.819444
10	12	0.09875	0.819444
11	12	0.0899242	0.819444
12	12	0.0822222	0.819444
13	12	0.0760256	0.819444
14	12	0.0708333	0.819444
15	12	0.0657222	0.819444
16	12	0.0617188	0.819444
17	12	0.0582843	0.819444
18	12	0.0548148	0.819444
19	12	0.0518421	0.819444
20	12	0.0495	0.819444
21	12	0.0469841	0.819444
22	12	0.0447348	0.819444
23	12	0.0428986	0.819444
24	12	0.0411458	0.819444
25	12	0.0391667	0.819444
26	12	0.0377244	0.819444

Figure 6.5: Analysis of the matching between two models

6.3 Results and discussion

In this section we illustrate some results of the test carried out to validate the proposed EAM descriptor as well as the matching method. In particular, sections 6.3.1, 6.3.2 and 6.3.3 illustrate search examples with different query models and different similarity settings. While section 6.3.4 provides an overview of the benefits of using different similarity criteria to reduce the number of association nodes and thus the complexity for the clique detection.

6.3.1 Flange

The first example is based on a mechanical flange, i.e. a component used to connect two objects.
In this example the query model is illustrated in Figure 6.6. The model has four screws and fours nuts arranged in a circular translation pattern, two main flanges, two shafts and two keys. All the parts are organized in a flat structure, i.e. without any sub-assemblies. The model does not present any volumetric intersection and each screw and the corresponding nut are in contact through an idealized cylindrical face.

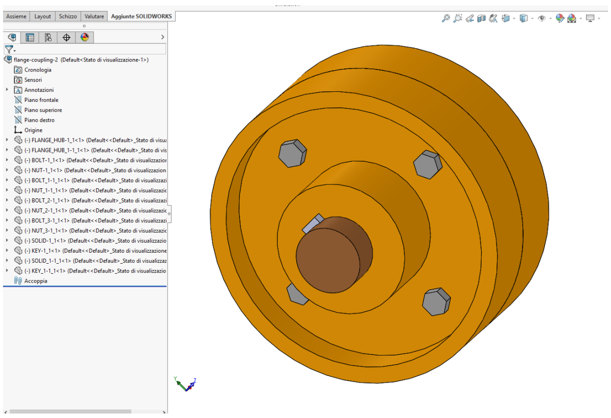


Figure 6.6: Query model

The selected criteria for the search are indicated by the vectors $c_N = (\alpha_{Sh}, \alpha_{Si}, \alpha_{CT}, \alpha_{PT})$ and $c_A = (\alpha_C, \alpha_{C_{num}}, \alpha_{J_{num}})$ introduced in section 5.1.2. In particular, we have that:

$$c_N = (1, 0, 1, 1), \tag{6.1}$$
$$c_A = (0, 0, 1), \tag{6.2}$$

where 1 and 0 indicate if the criterion is activated or not respectively.
This means that the nodes have to be compatible according to the shape ($\alpha_{Sh} = 1$), the component type ($\alpha_{CT} = 1$) and the pattern type ($\alpha_{PT} = 1$). In this case, the threshold ε for the shape is set up to 0.20, thus two components should have shapes similar at 80% according to the values of their 3D spherical harmonics.
In the end, the criterion of similarity for the arcs ($\alpha_{J_{num}} = 1$) means that two pairs of compatible nodes should have the same number of allowed rotations and translations, or one of them should be labeled as "UnSolved". Since all the contacts in the query model are solved, only the target can present this characteristic.

Results of this first search are illustrated in Figure 6.7, where in the set of weights to compute the global (green bar), partial (purple bar) and local (orange bar) similarity measures (according to the definition 5.2.8) is:

$$\begin{aligned} \mathcal{W} &= \{w^{shape}, w^{joint}, w^{position}, w^{structure}\} \\ &= \{1, 1, 1, 1\}. \end{aligned} \tag{6.3}$$

This means that the measures μ^{shape}, μ^{joint}, $\mu^{position}$ and $\mu^{structure}$ are used with the same importance to compute the final values of similarity (global, partial and local). In addition, considering a scenario of reusing an existing model, in this view, the models are ordered by their global similarity values.

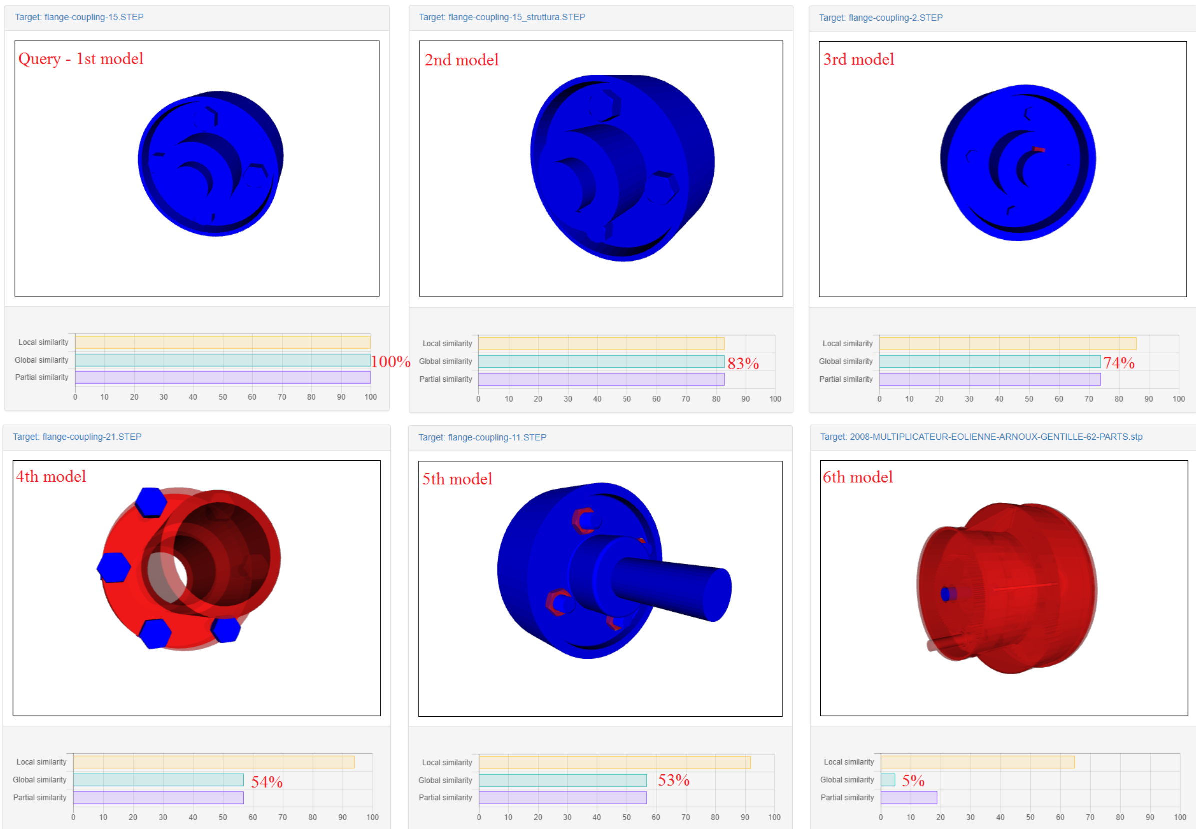


Figure 6.7: Set of retrieved models for the query model in Figure 6.6

The first model in Figure 6.7 corresponds to the query model, it is therefore straightforward that all its measures have top values (100%) since it perfectly matches itself.

The second model has the same components of the query model, i.e. same number of parts, same shape and same contacts but organized in a different ways. In the query model the structure is flat, while in this target model, the set of screws and the set of nuts are gathered forming two sub-assemblies. Thus, its $\mu^{structure}$ is less than 1 and this factor decreases the final value of the local similarity measure. Since all the components of the query and of the target model are matched, the values of the partial and global similarities corresponds to the local one, i.e. $\frac{2|C_h|}{|\mathcal{N}_q|+|\mathcal{N}_k|} = 1$ and $\frac{|C_h|}{|\mathcal{N}_q|} = 1$ in the definition 5.2.8.

The values of the single measures are reported in the radar chart of Figure 6.8, where we can observe, as expected, that all the measures are equal to one except the structure similarity, whose value is 0.35.

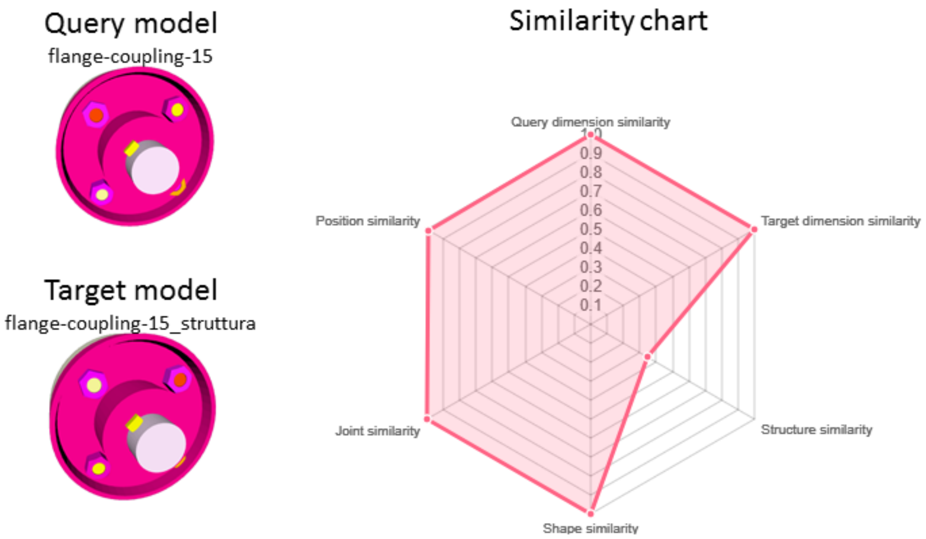


Figure 6.8: Example of matching inspection for the second model in Figure 6.7

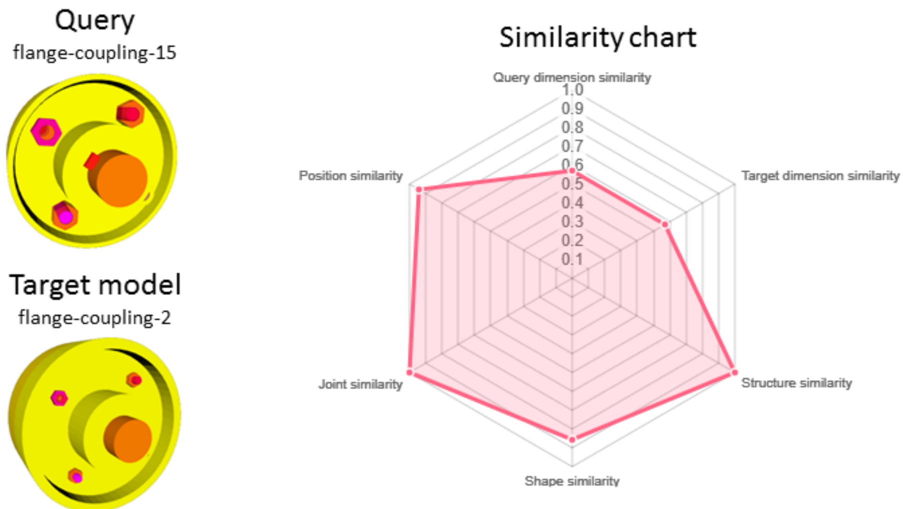
The third model of Figure 6.7 is a flange whose screws and nuts present a volumetric intersection, then they are matched thanks to the use of the attribute "UnSolved". Differently from the arcs in the query models, the arcs in the target models of type "UnSolved" have no the number of allowed rotations and translations, then according to the definitions 5.2.4 and 5.2.5, this difference affects their similarity at the level of the joint (μ^{joint}). In this example, the number of matched components is twelve and the number of components in the query and in the target models is fourteen, then the partial and the global measures are lower than the local one according to the same factor, i.e. $\frac{2|C_h|}{|\mathcal{N}_q|+|\mathcal{N}_k|} = \frac{2\cdot 12}{14+14} = 0.85$ and $\frac{|C_h|}{|\mathcal{N}_q|} = \frac{12}{14} = 0.85$ in the definition 5.2.8.

In Figure 6.9, we illustrate the use of the attributed "Unsolved" in the definition 5.1.7 and 5.1.8 of compatible arcs. If we do not handle the volumetric interferences in the construction of the association graph, then we will match less parts increasing the value of the local measure. Indeed, Figure 6.9(a) shows measures obtained varying the joint condition in 5.1.8, with the following definition of $\chi_{J_{num}}$:

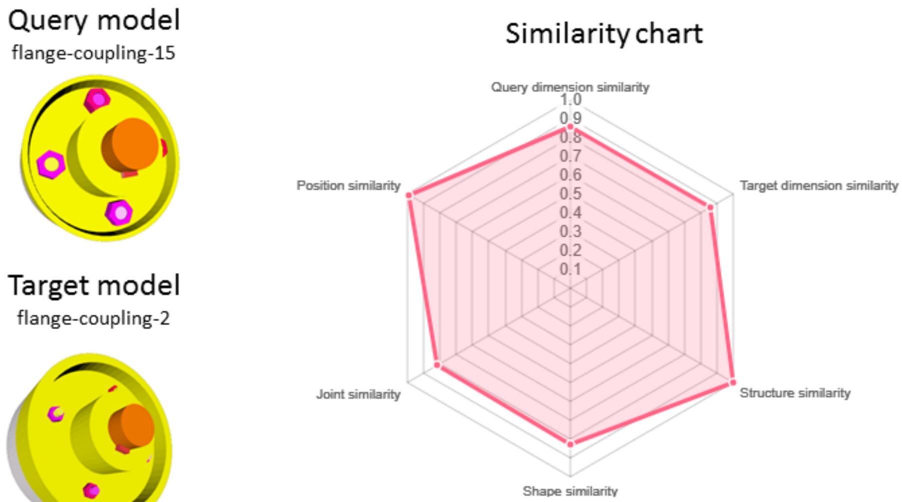
$$\chi_{J_{num}}(a_q^{ij}, a_k^{hl}) = \begin{cases} 1 & \text{if} \quad (|T_q^{ij}| = |T_k^{hl}| \text{ and} |R_q^{ij}| = |R_k^{hl}|) \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

In this case, two association nodes are linked by an arc if and only if the corresponding nodes in the attributed graphs are linked by a joint arc, which allows the same number of translations and rotations, i.e. no association arc is inserted between association nodes that represent parts, which share a volumetric intersection. Then, using the definition 6.4, the joint compatibility is stricter that using the definition 5.2.8, indeed the portion of matched elements over the query and the target model is lower.

Discarding some elements in the matching allows to increase the values of the local similarity measures, since these possible matchings are not identical to the elements of the query and thus they add values minor than 1 to be counted in the average of the single measures. For this reason, shape, joint, and structure similarity in Figure 6.9 are higher than the ones reported in Figure 6.9(b) obtained with the joint compatibility as reported in 5.1.8. We think that the similarity values obtained in Figure 6.9(b) are more appropriate in a retrieval context, since in this way, similar configurations are not completely discarded but simply retrieved with a lower similarity assessment and then the user can chose the solution that better fits his/her purposes.



(a) Example of matching inspection for the third model in Figure 6.7 using the equation 6.4 as joint compatibility

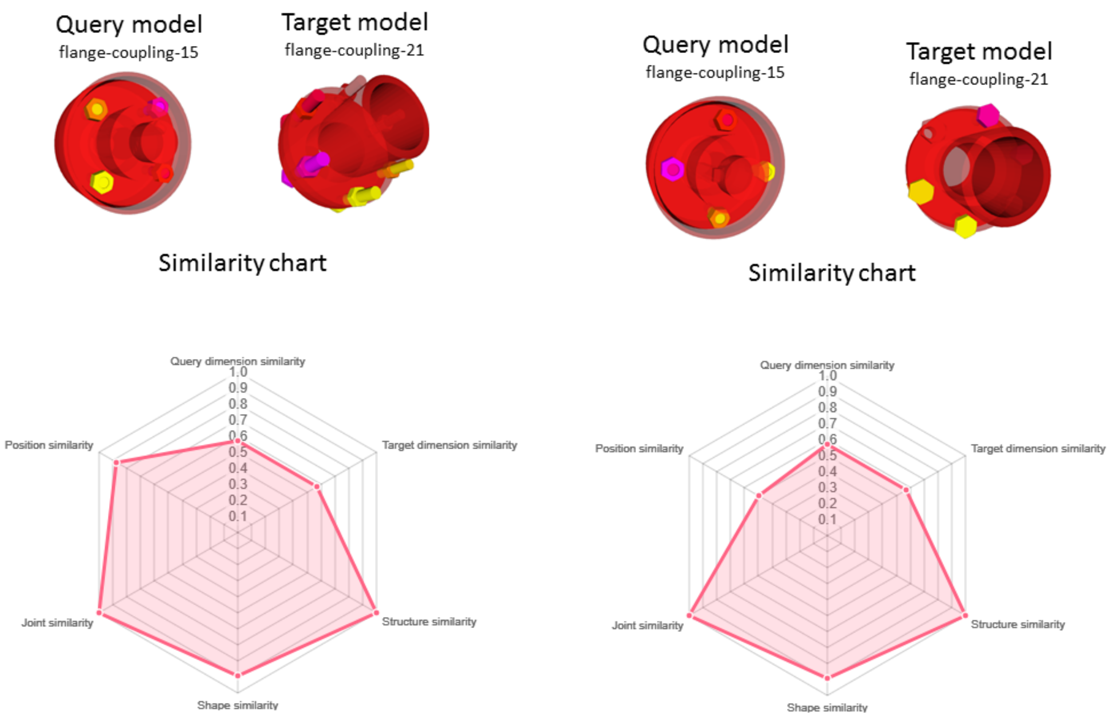


(b) Example of matching inspection for the third model in Figure 6.7 using the equation 5.1.8 as joint compatibility

Figure 6.9: Impact of the use of the "Unsolved" arc type on the matching results: (a) without using "Unsolved"; (b) using "Unsolved"

The fourth and the fifth models in Figure 6.7 have very similar measures, however at a glance, the coverage of the target models seams different. Actually, the coverage of the models is measured according to the number of matched elements and the two models have the same number of matched elements: four screws and four nuts for the fourth model against four screws, two main flange, a shaft and a key in model fifth one.
An evaluation using the volume may improve the visual perception similarity, but we think that, in general, it is more meaningful to consider the relevance of the matching parts, i.e. fastener elements should be less important than a shaft. Of course, this kind of consideration requires a study on the significance for each component category for the different type of mechanical assembly models. In addition, before to investigate the relevance of the components, the set of values of the attribute *Component_Type* should include more class of objects.

The single measures of the fourth model of figure 6.7 are presented in Figure 6.10. In particular, in this example we report two different cliques, which correspond to two different possible solutions, i.e. two sets of similar components.

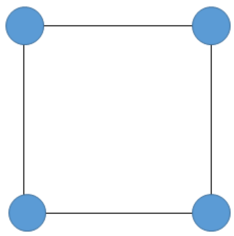


(a) (b)

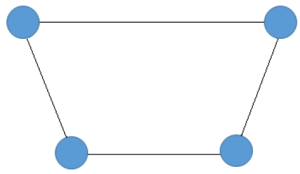
Figure 6.10: Two different results for the fourth model in Figure 6.7 where the arrangement of the parts is different and detected by different values of the position measure

Both the query and the target models have a circular pattern of screws and nuts, but with a different number of repeated elements (i.e. four in the query model and six in the target model). Thus, in the target model, it is not possible to find four equidistant screws and nuts that cover an entire circumference. The most similar solution that can be found has a constant distance between the elements of pattern (Figure 6.11(b)). This type of information is highlighted in the similarity evaluation by the position measure.

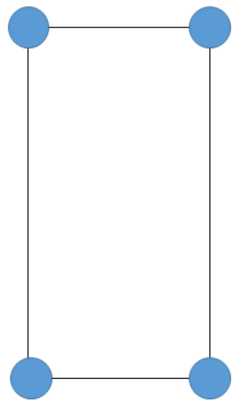
Indeed, in the first case, where the distance between the set of screws and nuts is constant as in the query model, the position value is higher than in the second where the screws and the nuts are arranged as depicted in Figure 6.11(c), losing the regular arrangement that characterize the query model. Then, the position similarity measure helps the ranking of the retrieved objects proposing before the ones with a regular arrangement as the one present in the query model.



(a) Baricenters of the screws of the query model in Figure 6.10



(b) Baricenters of the matched screws in the target model in Figure 6.10(a)



(c) Baricenters of the matched screws in the target model in Figure 6.10(b)

Figure 6.11: Representation of the screw baricenters for the models in figure Figure 6.10

The fifth model of figure 6.7 is very similar to the query model, the shape of its parts differs only for the shafts and the main flanges which have a border thicker than the query one. However, in this model the screws and the nuts present clearances, then these components are not in contact. The fact that not all the components are matched is reflected by the partial and global measures that are lower than the local one. This difference indicates that the matched parts are very similar, but do not cover all the query model and neither the target model. This is confirmed analyzing the values of the single measures reported in Figure 6.12. As expected, the values of joint, position and structure are very high, while the value of the shape similarity highlights small differences in the matched components. The most significant variation is in the number of matched components. In this example, both the query and the target models have fourteen parts and ten of them are matched, then the number of matched parts over the query and the target is 0.57 and these values affect negatively partial and global measures.

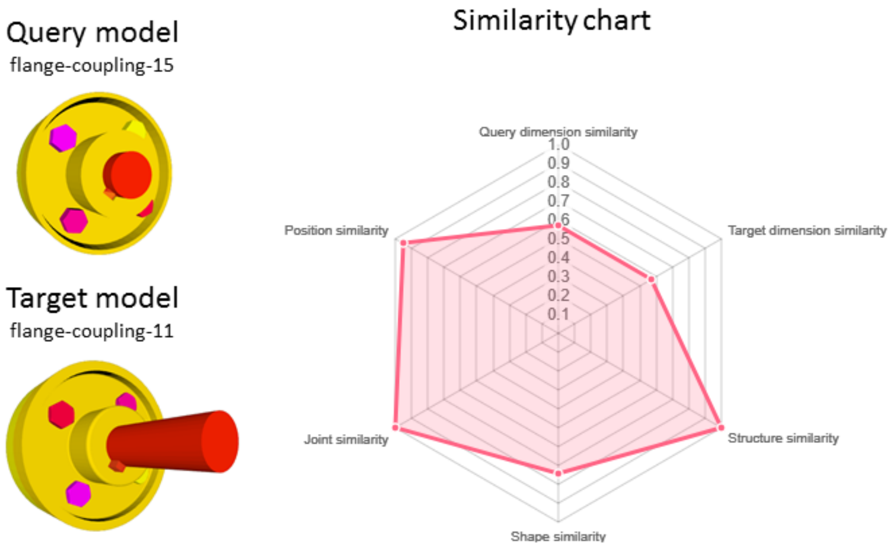


Figure 6.12: Example of matching inspection for the fifth model in Figure 6.7

Still analyzing the fifth model of figure 6.7, what hinders a full match is the type of contact. In particular, the fours nuts in the target models are not in contact with the screws and the key and the shaft present different contacts. Indeed, as illustrated in Figure 6.13, the key and the shaft in the query model are in contact by three planar faces, thus a translation is allowed, while in the target model the two parts are in contact by four planar faces and no motion are allowed. The key in the query model has three planar contacts against the four of the target. This difference rises a different DOF between the key and the shaft in the two models. Indeed, in the query model the key and shaft can translate without losing any face in contact, while in the target model the two parts are blocked.

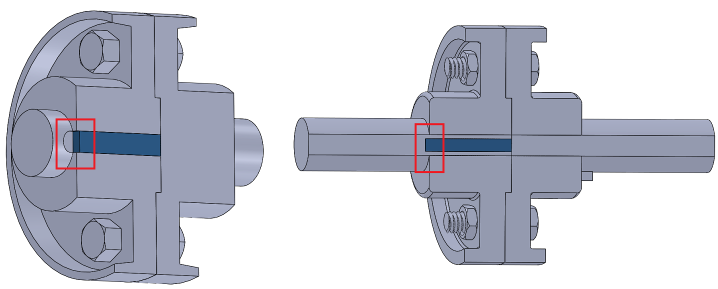
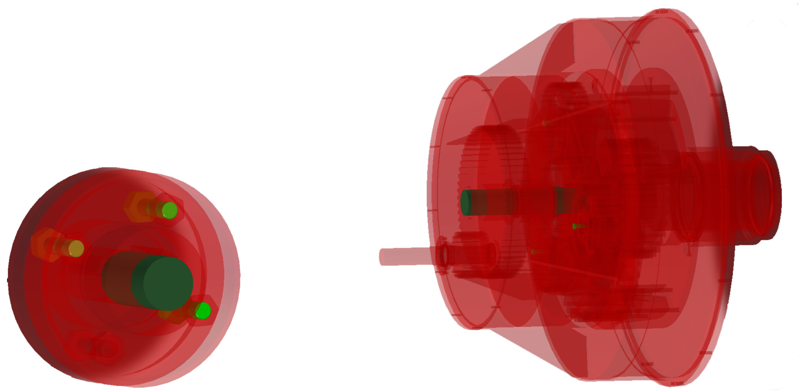


Figure 6.13: Different contacts between the key and the shaft of the query (left) and target (right) assembly models

Finally, from the values of the different types of similarity measures, the user can easily understand that the sixth model is not suitable for his/her purpose of reusing an existing flange model. Indeed, the partial similarity measure is very low indicating that not many elements of the query model have not been matched. Indeed, inspecting this model in Figure 6.14, we can observe that the matching parts are just three screws and a shaft, and they are not sufficient to determinate a flange model.



(a) (b)

Figure 6.14: Example of matched elements for the sixth model in Figure 6.7

6.3.2 Planetary gearboxes

In this section, we report the example of a user who is looking for assembly models similar to a planetary gearbox. This functional set has several sun-planet and ring planet gear pairs [72] as the model depicted in Figure 6.15, which has been used as query model with the following vectors of similarity criteria. Notice that the structure of this assembly model is flat, i.e. there is no organization in sub-assemblies.

$$
\begin{aligned}
c_N &= (0, 0, 1, 1), &\quad (6.5)\\
c_A &= (0, 0, 1). &\quad (6.6)
\end{aligned}
$$

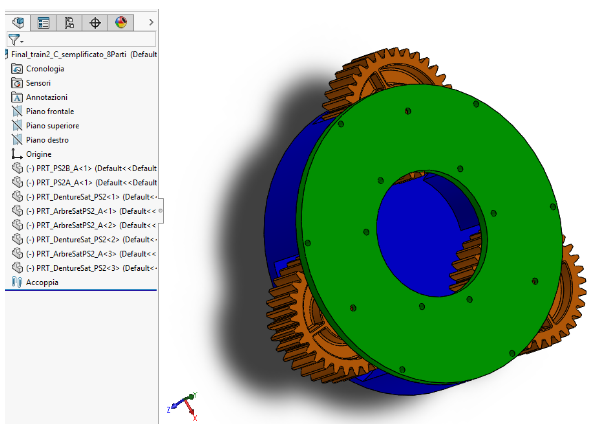


Figure 6.15: Query model

Differently from the previous case, for this search, the shape is not considered ($\alpha_{Sh} = 0$) and the nodes have to be compatible according to the component type ($\alpha_{CT} = 1$) and the pattern type ($\alpha_{PT} = 1$). Again, the criterion of similarity for the arcs ($\alpha_{J_{num}} = 1$) means that two pairs of compatible nodes should have the same number of allowed rotations and translations, or one of them should be labeled as "UnSolved".

Results of this search are illustrated in Figure 6.16, where the models are ordered by their partial similarity values and the set of weights to compute the global (green bar), partial (purple bar) and local (orange bar) similarity measures (according to the definition 5.2.8) is:

$$\begin{aligned} \mathcal{W} &= \{w^{shape}, w^{joint}, w^{position}, w^{structure}\} \\ &= \{1, 1, 1, 0\}. \end{aligned} \tag{6.7}$$

This means that the measures μ^{shape}, μ^{joint} and $\mu^{position}$ are used with the same importance to computed the final values of similarity (global, partial and local), while the $\mu^{structure}$ is not considered for their evaluation. Note that in the visualization of Figure 6.16 some blue parts are hardly visible, since other parts hide them and the transparency on the non-matched parts has to be improved according to the complexity of the assembly model.

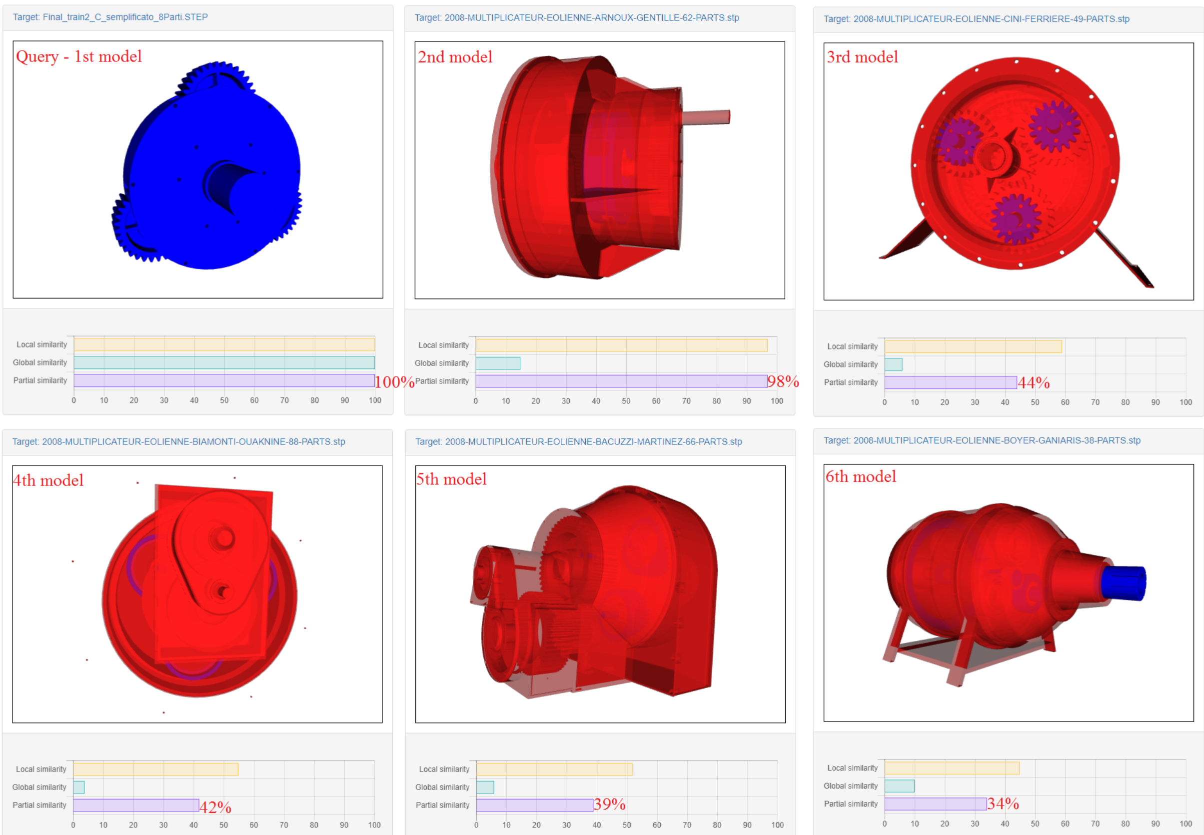


Figure 6.16: Retrieved models ordered according to the partial similarity criteria for the query model in Figure 6.15

In general, for all the retrieved models, we can observe that global measure is much lower that the others. This suggests that the query model is included in the target models.

The second model has high values of local and partial similarities, this suggest that the single similarity values are high and that the entire query is included in the target model. Indeed, the similarity chart in Figure 6.17, confirms indeed that all the similarity measures have almost tops values and only the target coverage is low, due to the fact that the query model is entirely included in a bigger target model. In figure 6.16 the parts are mostly in red, since the blue matched parts are inside the model. Again, in Figure 6.17 the pairs of matched parts in the query and the target model have the same color, but the actual implementation does not allow a proper visualization of the internal parts.

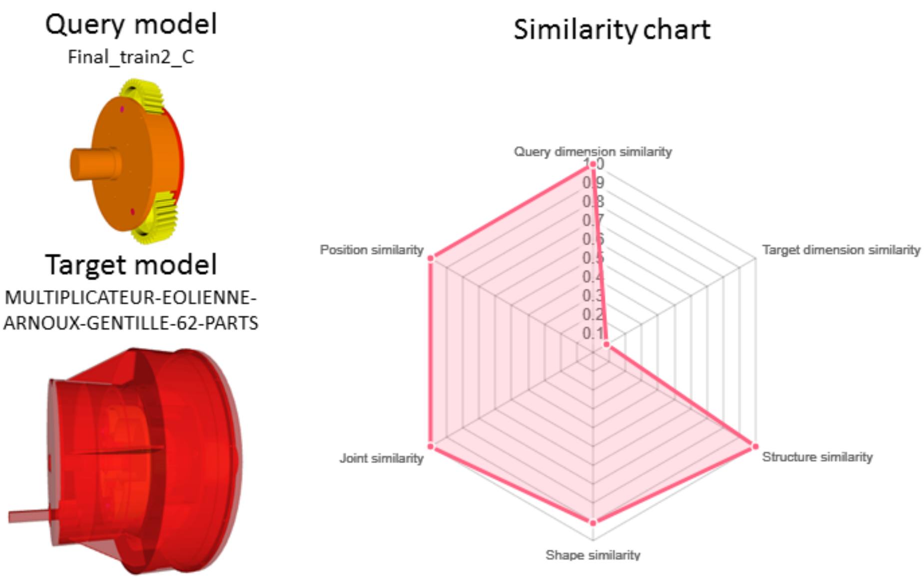


Figure 6.17: Example of matching inspection for the second model in Figure 6.16

In the third retrieved model, we can observe that the local similarity is higher than the partial one. This is due to the fact that not all the components of the query model are matched, but the ones that are considered compatible have high level of similarity. Analyzing deeper results in Figure 6.18, we can see that six parts are considered similar, thus the *query dimension similarity* is $6/8 = 0.75$, while the *target dimension similarity* is $6/103 = 0.058$. The matched components are the three gears and three axis, whose shapes are similar to the ones in the query model, then the μ^{shape} has an high value. μ^{joint} is equal to zero since no contacts are presents between the matched components. Since all the components are disconnected, there is no variation of incident rotation/translation to compare. This definition affects negatively the final value of the local similarity measure when the weight w^{joint} is not null. Anyhow, we think that assigning $\mu^{joint} = 1$ when no contact is present can be misleading for the user in his model analysis.

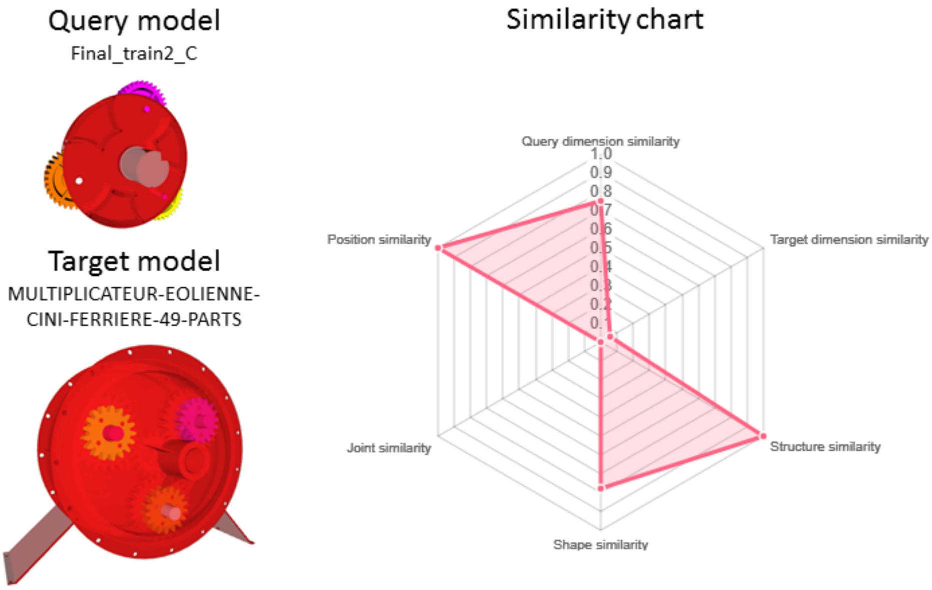


Figure 6.18: Example of matching inspection for the third model in Figure 6.16

The fourth model of Figure 6.16 is inspected in Figure 6.19. From the similarity chart we can see, that its similarity values are comparable to those in Figure 6.18, i.e. the main difference is represented by the value of μ^{shape}. Indeed, the retrieved planar gears of this model have no contacts, as the previous example, and they are recognized thanks to the attribute *Component_Type* of the EAM, which identifies three simple rings as gears exploiting the surrounding context of the components. Note that including shape compatibility, i.e. imposing $c_N[1] = 1$, this configuration would be probably not retrieved (i.e. depending on the chosen shape similarity threshold), since the shapes of the planar gears are quite different from the ones proposed in the query model.

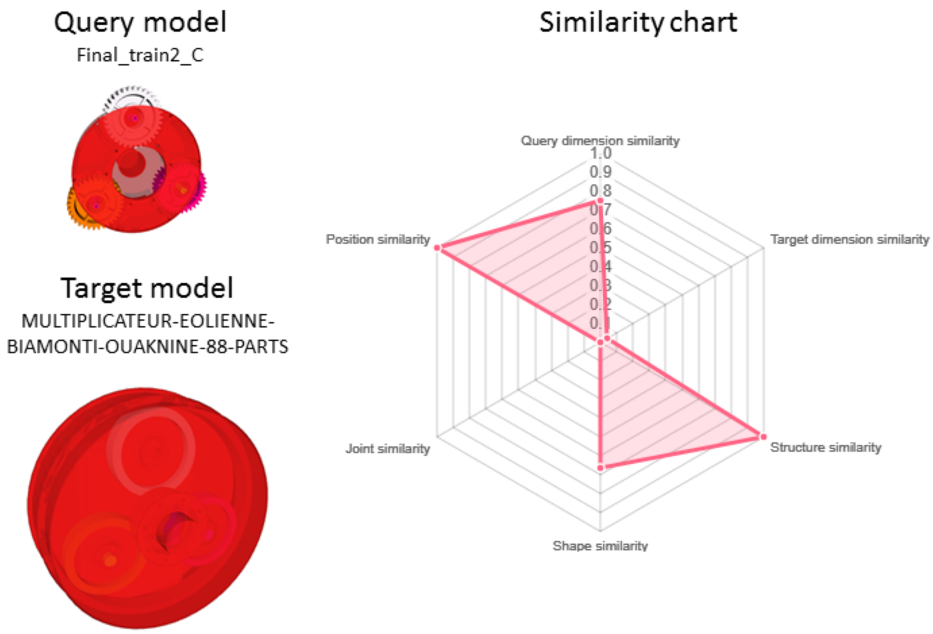


Figure 6.19: Example of matching inspection for the fourth model in Figure 6.16

For the last two models of Figure 6.16, the same considerations as for the model in Figure 6.18 hold, i.e. three gears and three shafts are retrieved whose shape is similar to the ones in the query model. Again the μ^{joint} is zero since there is no contact between the retrieved parts.

6.3.3 Bearing

For the third example, we aim to retrieve bearing assembly models as the one depicted in Figure 6.20, where its structure is flat. This model presents volumetric intersection between its rolling elements and the outer ring. Using the analysis described in section 4.4.4, these volumetric interferences are solved as punctual contacts.

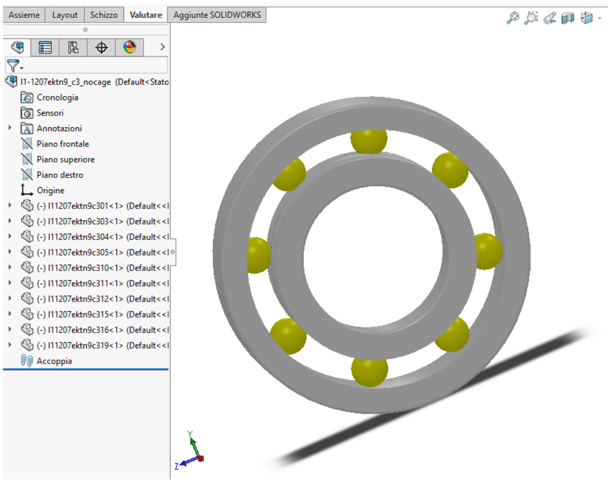


Figure 6.20: Query model of a ball bearing

The similarity criteria for this search are represented by the following vectors.

$$c_N = (1, 0, 0, 1), \tag{6.8}$$
$$c_A = (0, 0, 1). \tag{6.9}$$

The criteria indicate that two parts are considered similar is they have similar shape ($\alpha_{Sh} = 1$) and belong to the same type of pattern ($\alpha_{PT} = 1$).
Again, the criterion of similarity for the arcs ($\alpha_{J_{num}} = 1$) means that two pairs of compatible nodes should have the same number of allowed rotations and translations, or one of them should be labeled as "UnSolved".

Results of this search example are illustrated in Figure 6.21, where the models are ordered by their partial similarity values and the set of weights to compute the global (green bar), partial (purple bar) and local (orange bar) similarity measures (according to the definition 5.2.8) is:

$$\begin{aligned} \mathcal{W} &= \{w^{shape}, w^{joint}, w^{position}, w^{structure}\} \\ &= \{1, 1, 1, 1\}. \end{aligned} \tag{6.10}$$

This means that the measures $\mu^{shape}, \mu^{joint}, \mu^{position}$ and $\mu^{structure}$ are used with the same importance to compute the final values of similarity (global, partial and local).

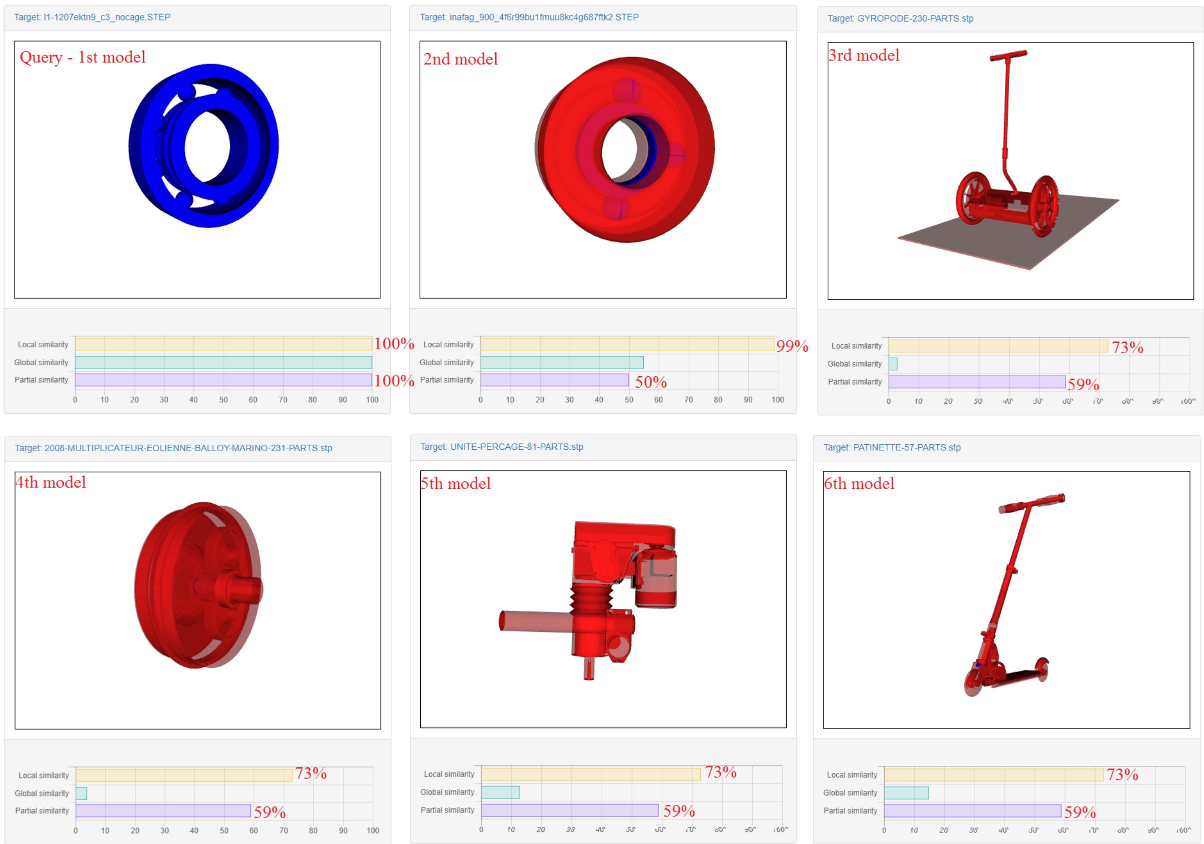


Figure 6.21: Set of retrieved models for the query model in Figure 6.20

The first model of Figure 6.21 represents the query model and its similarity measures have top values since it perfectly match itself.

The second model is analyzed in Figure 6.22. From this model, we can observe that the proposed matching method is able to manage query models bigger than target one, i.e. with a greater number of parts. Note that most works present in the state of the art assume that the query model has the same number of components of the target models or it is included in. Then, differently from the previous examples where the query was included in the targets, in this case we have that the value of $\%\frac{|N_{clique}|}{|N_{query}|} = \frac{5}{8}$ is lower that $\%\frac{|N_{clique}|}{|N_{target}|} = \frac{5}{10}$, where the matched elements are four spheres and the outer ring.

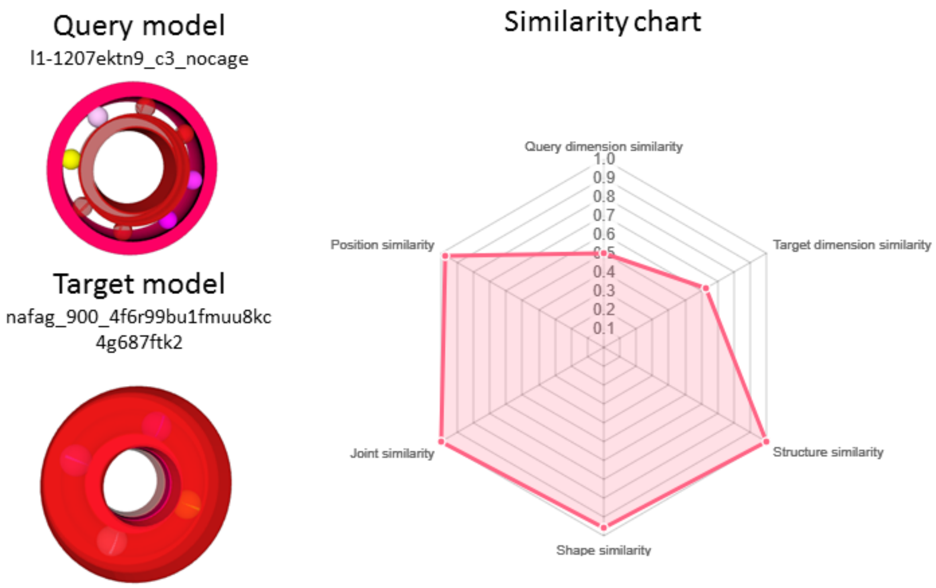
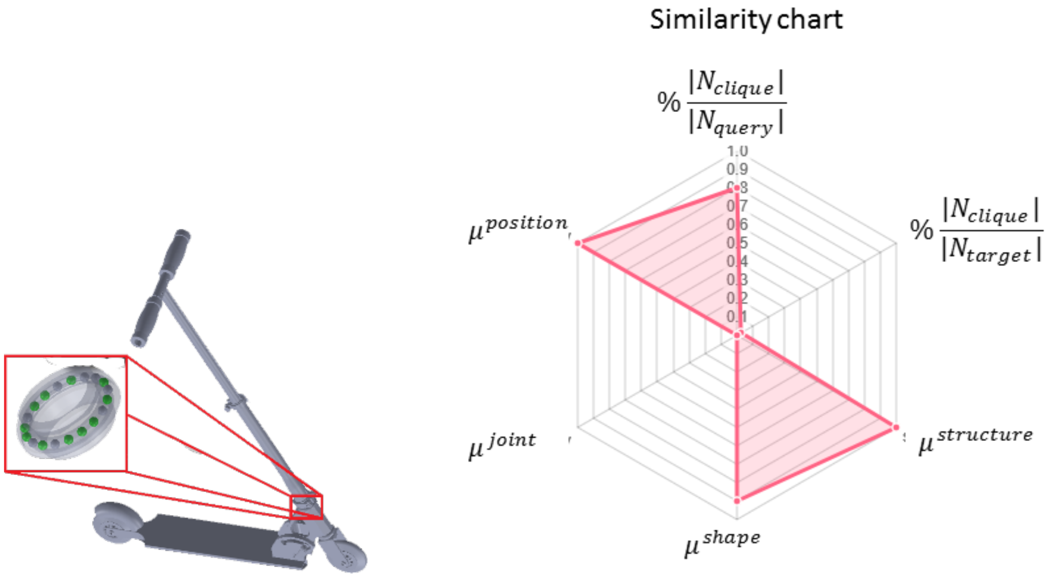


Figure 6.22: Example of matching inspection for the second model in Figure 6.21

Apparently, the other models in Figure 6.21 seam to have no affinity with the query model. This is due to the fact, that the bearing is often a small component in a bigger object, and without a deeper inspection of the target models, it is challenging to visualize something that is inside the models. Figure 6.23 illustrates a zoom of the last model of Figure 6.21 and its similarity chart, where we can see the that the matched components are represented by the spheres. The outer and inner ring are not matched, since their shape is quite different from the ones of the query model.
Since there are no contacts among the retrieved parts, the μ^{joint} is zero, while $\%\frac{|N_{clique}|}{|N_{target}|}$ is very close to zero because also in this case we have that $|N_{target}| >> |N_{clique}|$.



(a) (b)

Figure 6.23: Example of matching inspection for the sixth model in Figure 6.21

The same observations hold for the other models in Figure 6.21. Considering this set of examples, in the future, an optimized view should be developed, in order to allow also visualization of small internal components.

6.3.4 Combination of similarity criteria

To demonstrate that only a single criterion is not sufficient for a meaningful matching, we report in Figure 6.24 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. It corresponds to the third model in Figure 6.7. Each column reports the number of matched nodes according to the specified vectors c_N and c_A introduced in section 5.1.2 that represent the similarity criteria on the nodes ($c_N = (\alpha_{Sh}, \alpha_{Si}, \alpha_{CT}, \alpha_{PT})$) and on the arcs ($c_A = (\alpha_C, \alpha_{C_{num}}, \alpha_{J_{num}})$).

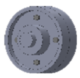
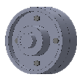
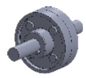
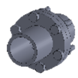
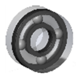
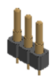
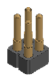
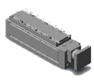
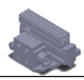
 **Query**	**Considered Model**	c_N = (0,0,0,0) c_A = (0,0,1)	c_N = (1,0,0,0) c_A = (1,0,0)	c_N = (1,0,0,0) c_A = (0,0,1)	c_N = (1,0,0,1) c_A = (0,0,1)
A		14	14	14	14
B		14	8	8	8
C		8	8	6	0
D		4	0	0	0
E		9	0	0	0
F		8	0	0	0
G		8	6	3	0
H		9	11	6	0
I		4	3	3	0
L		9	7	7	0
M		7	0	0	0
N		3	0	0	0

Figure 6.24: Number of matched nodes according to several criteria of similarity

More precisely, in the third column, we report the results using $c_N = (0,0,0,0)$ and $c_A = (0,0,1)$, i.e. no criteria limits the association between the nodes and association arcs are created if the numbers of rotations and translations in the joint level of the corresponding attributed graphs 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. Thus only 8 parts are matched (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 values obtained by imposing $c_N = (1,0,0,0)$ and $c_A = (1,0,0)$. These conditions verify the component shape similarity and if a contact is present between the corresponding parts. The threshold for the shape compatibility is set to 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. Since we translate the retrieval problem in a MCS problem, 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 fifth column reports results using $c_N = (1,0,0,0)$ and $c_A = (0,0,1)$, criteria, i.e. two parts must have similar shape (with threshold at 0.35) and association arcs are created if the numbers of rotations and translations in the joint level of the corresponding attributed graphs are the same. 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. The shape criterion matches the screws and all the nuts. The combination of these two criteria returns just the four screws and the two nuts disconnected.

The sixth column corresponds to the results obtained by specifying as condition $c_N = (1,0,0,1)$ and $c_A = (0,0,1)$, i.e. two nodes are associated according to their pattern and shape criteria, while association arcs are created if the numbers of rotations and translations in the joint level of the corresponding attributed graphs are the same. In this case, models that do not present in their statistics the number of pattern types as the ones in the query model are discard. 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 the obtained results, it becomes clear that the adopted criteria are good for retrieving similar models and that a single search key is not sufficient to search efficiently. Thus, combining several similarity criteria reduces the number of nodes in the association graph, then implicitly, the complexity of the matching algorithm can be reduced.

6.4 Conclusion

In this chapter, we have presented the developed prototype of the proposed retrieval system. To assure openness, we have based our development on non-proprietary CAD format. In articular, we have used the STEP file format as input, while the storage of the computed EAMs uses JSON format, which provide a schema to map objects (in our case an attributed multi-graph) in textual files and vice-versa.

Through these examples, we have noticed the importance of managing volumetric intersections in the assembly model descriptor. Even if it is not possible to solve intersections between models and deduce their desired configuration without errors, the information of "possible contact" improves the results of retrieved models.
On clearances, similar considerations should be addressed, analyzing if a clearance conveys an engineering meaning or if it is originated by errors, as in the fifth proposed flange example. Differently from volumetric intersections, we cannot assume that two parts with a clearance should be in contact in their real configuration, then the study to resolve these arrangements have to be more precise and accurate, for instance considering also the types of components involved in the clearance.
The results also show how the attributed of component type deduced by the shape-and-context-based classification allows to overcome limits due to the shape simplification in the DMU. Indeed, classifying components not only by their shape but also by their contacts permits to retrieve similar models even in different stages of their PDP (e.g. when the shapes are idealized and when the model is completely detailed).

The proposed method is also able to retrieve local similarity made of disconnected components. This is possible since in the definition of the association graph, two association nodes are connected if they have the same relationship, where "same relationship" indicates also that both the original pairs of nodes are non in contact.
Always about the matching, we want to remind that, the proposed matching method requires that two models have at least three elements to be identified as similar, if only a couple of elements is supposed to be retrieved, then our matching procedure do not retrieve any similar components. This limit is due to the fact that we set the minimum dimension of the clique at three. This decision is not limiting, since we aim at comparing big assembly models. Allowing to retrieve also single nodes or just a pair of nodes (so only with a relationship) would increase the entropy of the system, introducing a lot of meaningless results.

By these tests, we can confirm the usefulness of the information extracted from CAD models and stored in the EAM files for the retrieval system. A further improvement can be represented by the addition of another type of arc, which encodes the mutual arrangement of components that are not in contact. Indeed, observing that the position measure improves the value of similarity, it is reasonable to add other information that may also reduce the size of the association graph improving the matching procedure.

The proposed results demonstrate the ability of the our method to retrieve assembly models according to different criteria and handling several issues; anyhow, in the future, in order to validate unquestionably the effectiveness of the proposed method, we need to provide a precision and recall evaluation. We have done a first step in the validation of the proposed framework creating a suitable set of assembly models, but for the precision and recall evaluation we need to formulate a set of queries and to define their relevant models, i.e. we need an appropriate ground-truth. So far, what limits us in this definition is that an optimal definition of an assembly ground-truth should involved many users that have an active role in the definition of what should be retrieved. In the end, a formal comparison with other works which assess assembly similarity is hard since a common ground-true for assembly models is not available.

Conclusion and perspectives

New product designs can benefit from the reuse of existing solutions and of the associated information, thus it is useful to have good instruments for the retrieval of CAD models. Nowadays, CAD systems are integrated with Product Data Management (PDM) systems, which allow to track and control data related to existing products. These systems are very appropriate to handle text searches (e.g. material or requirements), but offer limited capabilities for geometry-based searches (e.g. shape or joint based searches).

To overcome these limitations, the general objective of this thesis was the proposition of an assembly retrieval system able to compare assembly models according to different similarity criteria. The information is organized in a multi-layer structure, the so-called Enriched Assembly Model (EAM), through which an assembly model is characterized according to four data layers: the structure layer, the interface layer, the shape layer and the statistic layer.
The structure layer encodes the hierarchical assembly structure of the CAD model as specified by the designer. In addition, it includes information regarding the type of components and the arrangements of repeated parts. The interface layer specifies the contacts existing between two parts and their equivalent joint. The shape layer aims to characterize the shape of assembly components by the use of spherical harmonics descriptors and their size by using volume and surface area information. In the end, statistic layer has the purpose of filtering large datasets and reducing the number of models to be compared by using several numerical values, such as the number of patterns of repeated components of a specific type.

This descriptor has been designed using a graph-based structure and the comparison of two assembly models has been translated in a graph matching problem. Using the Maximum Common Subgraph (MCS) matching and exploiting the percentage of matching components over the query model and the target model, we are able to assess different types of similarity: global (the whole query model is similar to the whole target model), partial (the query model is included in the target model) and local (the query and the target model share some similar components). In the end, since assembly models can be similar according to different criteria, we have defined a set of measures to evaluate the different types of similarities.

Some conclusions and future works to improve the proposed assembly retrieval approach have been already discussed in the previous chapters, then this final part of the manuscript mostly focuses on the possible extension and perspectives, arranging the discussion in the following three main points:

(i) the definition of a proper assembly descriptor,

(ii) the comparison of two EAMs by graph matching procedure,

(iii) the improvement of user experience.

The definition of a proper assembly descriptor

Our main concern for the definition of an assembly descriptor was the capability of capturing the characterizing features of an assembly and the possibility of extracting automatically the required data from CAD model. This objective was challenging because of the presence of simplification and unrealistic configurations often present during the design of assembly models. In this thesis, we extracted meaningful information, including, but not limited to, the classification of assembly components, the arrangement of repeated parts, and the contacts between parts.
In the future, the EAM can be enhanced with the following information.

- **Functional set**
  Extra effort should be put into defining other useful and meaningful templates for the identification of components in assembly models. There exist many functional sets to be included in the proposed system, anyhow it is necessary a detailed study on these models to figure out what are the best characteristics that describe them. Indeed, their characterization requires on the one hand the engineering knowledge about their technical specification and on the other hand the design experience concerning the possible simplifications adopted during the different phases of the PDP.

- **Overall shape of an assembly**
  Beside the descriptors to characterize the shape of the single parts of an assembly model, it worths to consider also the overall shape of an assembly resulting from the union of the assembly (or sub-assembly) parts. This would support the retrieval of assemblies that are globally similar according to their external shape. Moreover, it could facilitate the use of scanned data for the query specification, thus permitting to retrieve the candidate CAD assemblies corresponding or at least similar to a physical product.

- **Different representations**
  So far, we have based our matching on geometric information present in the CAD models. The geometry has been analyzed using the B-rep representation of assembly parts and mesh representation derived from the tessellation of the single CAD parts. In the future, it worths to investigate what other descriptors can be used to enrich the EAM and thus to allow multi-modal searches. In this way, after defining node compatibility according to the new descriptors, the proposed framework should be able to retrieve models not only using CAD models, but also allowing the use of different inputs.
  For instance, in addition to the shape descriptors used as node attributes, other meaningful signatures can be derived from pictures of the assembly parts obtained by cameras or by computer tomography. From the scientific point of view, including descriptors derived from images involves several interesting problems as the camera calibration (e.g. where to locate the objects with respect to camera, how to detect hidden portions of the parts), the management of multi-views (e.g. how many images are necessary to define a meaningful descriptor), as well as image analysis (e.g. contour detection or points of interest).

Moreover, with respect to EAM construction, we can improve the following aspects.

- **To improve the identification of functional sets**
  Currently, the component classification of functional sets, like bearings, exploits assembly hierarchical structure, i.e. it identifies sub-assemblies similar to pre-defined templates. Besides increasing the set of considered templates, an important improvement is represented by the ability to reach this goal without the use of the structure information. In this way, also components that are not structured in a proper way can be recognized.

- **To extend reasoning processes on free-form surfaces**
  All the reasoning procedures are mainly limited to consider analytic surfaces. Anyhow, it is not uncommon that parts are modeled with free-form surfaces. Then, in order to provide a retrieval system able to manage as much configurations as possible, we should enlarge our analysis (e.g. the detection of DOF for parts in contact) considering also this type of surfaces. In particular, in this thesis, we have managed the interfaces arisen by contacts between two parts if the involved surfaces are analytics or free-form that describe sphere parts. A possible extension in the EAM construction concerns the management of more complex free-form surfaces that can be involved in the description of contacts between two parts. The main problem dealing with these surfaces regards the detection of the DOF that they produce.

- **To solve volumetric interferences and clearances**
  Among the possible interfaces, we have faced the analysis of some volumetric interferences, in particular, we have studied how to deduce the correct contacts between two parts if no positioning errors occurs when one of the involved part describes a spherical object. In addition to these cases, we aim to recover the wished configuration condition in more complex configurations than those currently treated.
  Finally, among the possible interfaces, also the clearances should be managed in the future. Anyway, their analysis is quite complicated since they do not identify always unrealistic arrangements as for the volumetric interferences. In this case, some reasoning can be done combining the distance between the faces of two parts and also their component type to find out if the clearance is a "real clearance" or if it represents a missed contact.

- **Integration in PDM/PLM systems**
  The proposed retrieval framework has been designed to be a stand-alone system, so that it can process also not organized dataset of 3D models. If this system will be integrated into PDP or PLM systems, other useful data can be included in the reasoning process to analyze complicated situations hardly recognizable using pure geometric information. For instance, knowing that for a specific product some of its components are acquired from a certain supplier can be used to select a proper set of templates for the identification of its components.

The comparison of two EAMs by graph matching procedure

So far, we compare each model in the database with the query model. Some of them are immediately discarded by the use of some filters, such as the number of patterns of a certain type. In the future, we aim to exploit these filtering characteristics to develop an indexing system of the database in order to ease the selection of the models, which are meaningful candidates for a certain query, avoiding in this way a number of unnecessary checks.

Concerning the actual comparison procedure, since the EAM has been represented as an attributed multi-graph, the comparison of two assembly models has been traduced into a graph matching problem. In order to assess different types of similarity, it is necessary to detect the common subgraphs between two assembly models and to solve a MCS problem. Considering the current techniques to solve MCS problem, it has been reduced into a Maximum Clique (MC) problem. An interesting manner to reduce the complexity of the MC problem would be to reduce the size of the association graph on which the MC problem is based. To reduce its size, in the future, we aim to investigate two possibilities.

- **Reduce the number of association nodes**
  The first way for reducing the number of nodes to be matched could be to discard unimportant components, such as bolts and nuts. This possibility should be carefully investigated to identify which are the less important components and how to treat the relationships among the remaining components once the meaningless ones are not considered.
  Another option to reduce the number of nodes in the association graph is to gather together (as a single association node) meaningful sets of nodes, as the nodes associated with repeated parts in the patterns or objects that are integral each other, i.e. the mutual distances between the parts remain the same during the motion of the objects. In this way, the matching would be performed with a smaller association graph and, if the matching retrieves a possible solution, then it has to be analyzed deeper. To reach this goal, besides managing macro-nodes in the descriptor representation and relative procedures for the update of its relationships, we need also to understand how to identify meaningful sets of components and how to manage them in the similarity measure definitions.

- **Reduce the number of association arcs**
  The size of the association graph can be reduced also decreasing the number of arcs. To this purpose, we aim to extract other useful information from assembly model to better characterize the arrangement of the parts. In particular, we are thinking to use additional relationships in the EAM, which encode the mutual position of the components in the assembly models, such as if two components (e.g. two different shafts) are coaxial or parallel in the 3D space of the assembly model.

The improvement of user experience

For this task, we have faced the problem of defining partial query, providing a suitable visualization for the browsing of the results and finally the proposition of measures, which evaluate assembly similarity according to different similarity criteria.
Additional efforts should address the following perspectives.

- **Evaluate combinations of measure weight**
  So far, the weights used to combine the set of measures to compute the similarity has been set by the user. To discover which weights best fulfill user's purposes, it is necessary to investigate how weight combinations affect the final score and, most of all, it is essential to include the user feedback for indicating which results are considered pertinent for the specific query to eventually collect data for the weight specification.

- **The semantic definition of a query model**
  An interesting research topic, linked with the query definition, is represented by the interpretation of an engineering textual request such that it can be translated into a graph structure to be used as an abstract query in our retrieval framework.

- **Result inspection**
  Even if, in general, the provided visualization capabilities offer the possibility to identify the components in the retrieved assembly models that are similar to the query one, in the case of complex assemblies this visualization is still not sufficient, therefore further studies are required to improve the comprehension in such cases.

- **Data set browsing**
  Using the reciprocal comparison of all the models in the data set can facilitate the browsing of the dataset by visualizing similar models to the one selected according to all the measures defined.

Publications

Journal

- **Mutli-criteria retrieval of CAD assembly models**
  Katia Lupinetti, Franca Giannini, Marina Monti, Jean-Philippe Pernot
  Journal of Computational Design and Engineering, Elsevier,
  In printing

- **Identification of similar and complementary subparts in B-rep CAD models**
  Franca Giannini, Katia Lupinetti, Marina Monti
  Journal of Computing and Information Science in Engineering, Vol. 17, Iss. 4, 2017

- **Regular patterns of repeated elements in CAD assembly model retrieval**
  Katia Lupinetti, Lisa Chiang, Franca Giannini, Marina Monti, Jean-Philippe Pernot
  Computer-Aided Design and Applications Vol. 14, Iss. 4, 2017, Pages 516-525

Conference

- **Identification of functional sets in mechanical assembly models**
  Katia Lupinetti, Franca Giannini, Marina Monti, Jean-Philippe Pernot
  Proceeding of ICIDM2017 conference, Milan, Italy, 2017 July 17-19

- **CAD Assembly retrieval and browsing**
  Matteo Rucco, Katia Lupinetti, Franca Giannini, Marina Monti, Jean-Philippe Pernot
  Proceeding of PLM conference, Seville , Spain, 2017 July 10-12

- **Identification of functional components in mechanical assemblies**
  Katia Lupinetti, Franca Giannini, Marina Monti, Jean-Philippe Pernot
  Procedia CIRP, Vol. 60, 2017, Pages 542-547

- **Regular patterns of repeated elements in CAD assembly model retrieval**
  Katia Lupinetti, Lisa Chiang, Franca Giannini, Marina Monti, Jean-Philippe Pernot
  Proceeding of CAD'16 conference, Vancouver, Canada, 2016 June 27-29, Pages 147-151

- **Automatic extraction of assembly component relationships for assembly model retrieval**
  Katia Lupinetti, Franca Giannini, Marina Monti, Jean-Philippe Pernot
  Procedia CIRP, 2016, Pages 472-477

- **CAD assembly descriptors for knowledge capitalization and model retrieval**
  Katia Lupinetti, Franca Giannini, Marina Monti, Jean-Philippe Pernot
  Proceeding of TMCE conference, Aix-en-Provence, France, 2016 May 9-13
  Tools and Methods of Competitive Engineering, Pages 587-598

Submitted

- **Part classification with supervised machine learning**
  Matteo Rucco, Franca Giannini, Katia Lupinetti, Marina Monti
  Submitted to Artificial Intelligence for Engineering Design, Analysis and Manufacturing

Appendix

Basic notions of graph theory

Definition 0.0.1. A **graph** is an ordered pair $G = (N, A)$ composed by

- a finite set N of nodes, or vertices,
- a finite set A of arcs, or edges.
- for each pair of nodes $u, v \in N$, there exists at most one arc $a \in A$ joining the nodes u and v, i.e. $a = (u, v)$.

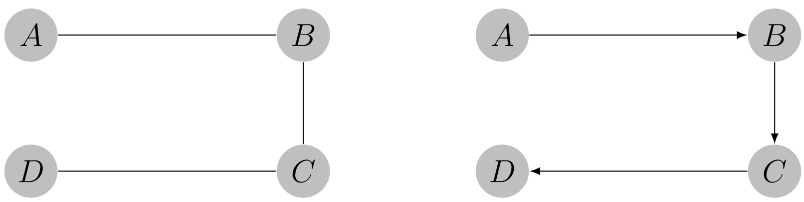
Definition 0.0.2. A graph $G = (N, A)$ has an **adjacency matrix** M defined as follows,

$$M_{i,j} = \begin{cases} 1 \text{ if } (a_j, a_j) \in A, \\ 0 \text{ if } otherwise. \end{cases} \tag{11}$$

There exist two kinds of graphs, the undirected and the directed graph.

Definition 0.0.3. A graph is **undirected** if all arcs have no orientation, i.e. given $u, v \in N$ two nodes, then the arc $a = (u, v) \in A$ and the arc $a' = (v, u) \in A$ represent the same arc, thus the relation expressed by an arc is symmetric, see Figure 25(a).

Definition 0.0.4. A graph is **directed** if all arcs have an orientation, i.e. given $u, v \in N$ two nodes, then the arc $a = (u, v) \in A$ and the arc $a' = (v, u) \in A$ do not represent the same arc, thus the relation expressed by an arc is not symmetric, see Figure 25(b).



(a) Undirected graph (b) Directed graph

Figure 25: Example of undirected and directed graph.

Definition 0.0.5. Given a graph G = (N, A), if there exists an arc $a \in A$ connecting nodes u and v (i.e., a=(u,v)), then the following definitions are given:

- u and v are **adjacent**,
- a is **incident** in both u and v.

Another important concept concerning graph theory is the **degree** notion.

Definition 0.0.6. Given an undirected graph $G = (N, A)$, the **degree** of a node $u \in N$ is the number of its incident arcs.

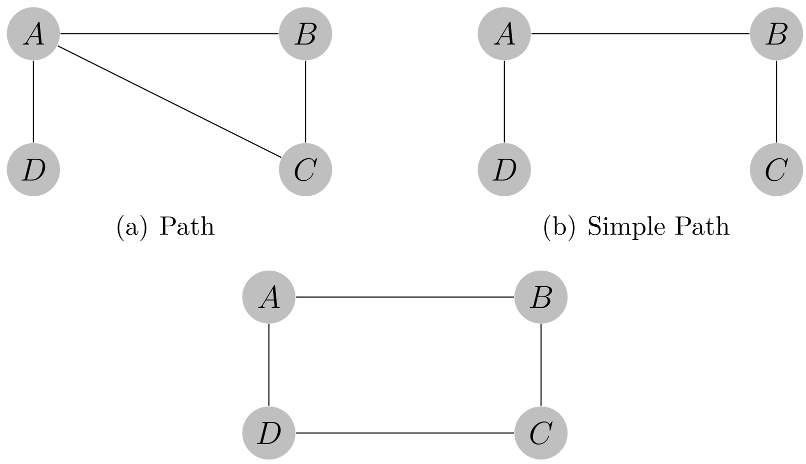
Given a directed graph $G = (N, A)$, the **degree** of a node $u \in N$ is given by the sum of the **inner degree** and the **outer degree**, where

- the inner degree of a vertex v is the number of inner arcs in v,
- the outer degree of a vertex v is the number of outer arcs in v.

Definition 0.0.7. A **path** in a graph $G = (N, A)$ is a sequence of nodes $[u_1, \cdots, u_n]$ such that for each consecutive pair of nodes (u_i, u_{i+1}) there exists an arc $a = (u_i, u_{i+1}) \in A$, see Figure 26(a).

In particular:

- If all the nodes in the sequence $[u_1, \cdots, u_n]$, excepted the first and the last, are distinct, then the path is simple, see Figure 26(b).
- A simple path with the first and last node coincidents is a *cycle*, see Figure 26(c).



(a) Path

(b) Simple Path

(c) Cycle

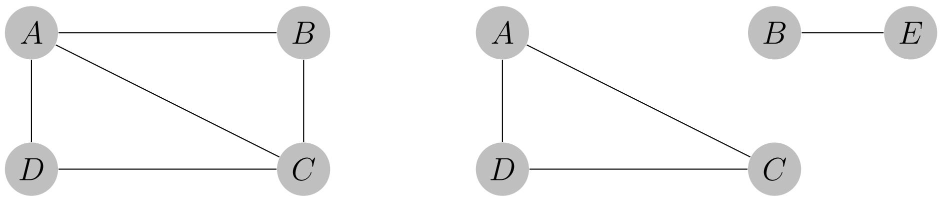
Figure 26: Example of graph paths.

Definition 0.0.8. An undirected graph $G = (N, A)$ is **connected** iff for each pair of nodes $u, v \in N$ there exists a path that joins the nodes, see Figure 27.

Definition 0.0.9. An undirected $G = (N, A)$ is **complete** if each node $u \in N$ is connected with all the nodes in $N \setminus \{u\}$, see Figure 28.

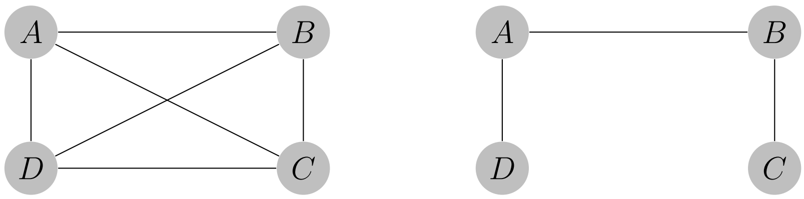
Definition 0.0.10. A **subgraph** of a graph $G = (N, A)$ is a graph $S = (N', A')$ such that $N' \subseteq N$ and $A' \subseteq A$, see Figure 29.

Definition 0.0.11. A clique C in a graph $G = (N, A)$ is a subset of node $N' \subseteq N$, such the subgraph $S = (N', A')$ is complete, see Figure 30.



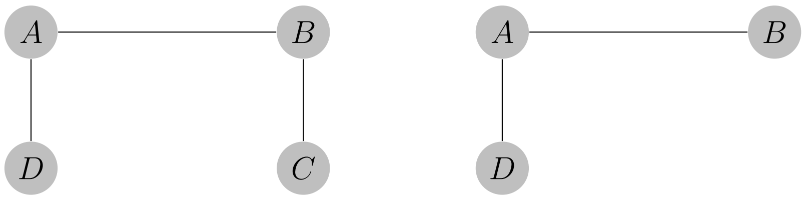
(a) Connected (b) Not connected

Figure 27: Example of connected and not connected graph.



(a) Complete (b) Not complete

Figure 28: Example of complete and not complete graph.



(a) Graph (b) Subgraph

Figure 29: Example of subgraph.

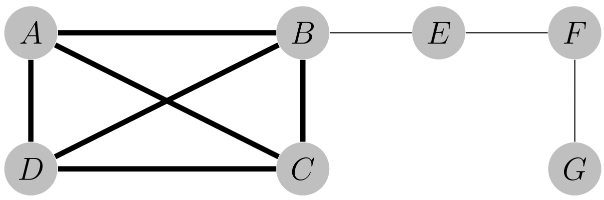


Figure 30: The subgraph with thicker arcs are a clique.

A clique of a graph $G = (N, A)$ can be characterized through the following property

Proposition 0.0.1. If a *subgraph* $S = (N', A')$ of a graph $G = (N, A)$ is a clique of G then

$$\frac{k(k-1)}{2} = \sum_{i,j: a_{ij} = (n_i, n_j) \in A'} M_{i,j}. \tag{12}$$

where k denotes the number of nodes of S, i.e. $k = |N'|$.

Definition 0.0.12. An **attributed graph** G is given by a quadruple $G = (N, A, \Phi_N, \Phi_A)$, where N is the set of nodes, A is the set of graph arcs, $\Phi_N : N \longrightarrow T_N$ and $\Phi_A : A \longrightarrow T_A$ are the node and the arc attribute functions, with T_N and T_A sets of node and arc attributes of G.

Definition 0.0.13. An **attributed subgraph** S of G is a quadruple $G = (N_S, A_S, \Phi_{N_S}, \Phi_{A_S})$, where $N_S \subseteq N$, $A_S \subseteq A$, Φ_{N_S} and Φ_{A_S} are induced by Φ_N and Φ_A respectively.

Definition 0.0.14. A **graph isomorphism** between two graphs $G_1 = (N_1, A_1)$ and $G_2 = (N_2, A_2)$ is a bijective function $f : N_1 \longrightarrow N_2$ such that:
for all the arcs $a_1 = (n_1, n_1') \in A_1$, there exists an arc $a_2 = (f(n_1), f(n_1')) \in A_2$. Moreover, for all arcs $a_2 = (n_2, n_2') \in A_2$, there exists an arc $a_1 = (f^{-1}(n_2), f^{-1}(n_2')) \in A_1$.

Definition 0.0.15. If $f : N_1 \longrightarrow N'$ is a graph isomorphism between the graphs G_1 and G', and G' is a subgraph of G_2, then f is called **subgraph isomorphism** from G_1 to G'.

Definition 0.0.16. A **common subgraph** of G_1 and G_2 is a graph G, $G \subseteq G_1$ and $G \subseteq G_2$, such that there exists a subgraph isomorphism from G to G_1 and from G to G_2.

Definition 0.0.17. A **maximum common subgraph** of G_1 and G_2, denoted as MCS_{G_1,G_2} is a common subgraph G such that there exists no other common subgraph having more nodes than G. The MCS_{G_1,G_2} is not necessarily unique.

Definition 0.0.18. A **graph isomorphism** between two attributed graphs $G_1 = (N_1, A_1, \Phi_{N_1}, \Phi_{A_1})$ and $G_2 = (N_2, A_2, \Phi_{N_2}, \Phi_{A_2})$ is a bijective function $f : N_1 \longrightarrow N_2$ such that:

1. $\Phi_{N_1}(n) = \Phi_{N_2}(f(n)) \ \forall n \in N_1$.
2. For all the arcs $a_1 = (n_1, n_1') \in A_1$, there exists an arc $a_2 = (f(n_1), f(n_1')) \in A_2$ such that $\Phi_{A_1}(a_1) = \Phi_{A_2}(a_2)$. Moreover, for all arcs $a_2 = (n_2, n_2') \in A_2$, there exists an arc $a_1 = (f^{-1}(n_2), f^{-1}(n_2')) \in A_1$ such that $\Phi_{A_1}(a_1) = \Phi_{A_2}(a_2)$.

Definition 0.0.19. Given two graphs $G_m = (N_m, A_m)$ and $G_n = (N_n, A_n)$ their *association graph Ass_{mn}* is defined by two sets $Ass_{mn} = (\mathcal{H}_N, \mathcal{H}_A)$ where:

- $\mathcal{H}_N$ is a set of nodes pairs $(n_m^i, n_n^j) \in N_m \times N_n$,
- $\mathcal{H}_A$ is a set of edges pairs $(a_m^i, a_n^j) \in A_m \times A_n$, such that $a_m^i \in A_m \iff a_n^j \in A_n$.

Bibliography

[1] Introducing json. http://json.org/. Accessed: 22/09/2017.

[2] Public library GrabCAD. http://www.grabcad.com/. Last accessed January 2018.

[3] Rotation invariant shape descriptors. http://htmlpreview.github.io/?https://github.com/mkazhdan/ShapeSPH/blob/master/descriptors.html. Last accessed Septermber 2017.

[4] Tracepart repository. http://www.traceparts.com/. Last accessed January 2018.

[5] Standard glossary of software engineering terminology. *IEEE Software Engineering Standards & Collection*, 1990.

[6] T. Alam. Kinematics of machines. https://mechanical-engg.com/blogs/entry/949-types-of-kinematic-pairs/.

[7] G. F. Ashby. *Multidimensional models of perception and cognition*. Psychology Press, 2014.

[8] M. Attene, S. Marini, M. Spagnuolo, and B. Falcidieno. Part-in-whole 3D shape matching and docking. *The Visual Computer*, 27(11):991–1004, 2011.

[9] J. Bai, S. Gao, W. Tang, Y. Liu, and S. Guo. Design reuse oriented partial retrieval of CAD models. *Computer-Aided Design*, 42(12):1069 – 1084, 2010.

[10] H. G. Barrow and R. M. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. 4(4):83–84, Jan. 1976.

[11] M. Bertoa and A. Vallecillo. Usability metrics for software components. In *8th international workshop on quantitative approaches in object-oriented software engineering (QAOOSE'2004), Oslo, Norway*, 2004.

[12] D. Bespalov, W. C. Regli, and A. Shokoufandeh. Local feature extraction and matching partial objects. *Computer-Aided Design*, 38(9):1020–1037, 2006.

[13] S. Biasotti, A. Cerri, A. M. Bronstein, and M. M. Bronstein. Quantifying 3D shape similarity using maps: Recent trends, applications and perspectives. In *Eurographics (State of the Art Reports)*, pages 135–159, 2014.

[14] S. Biasotti, S. Marini, M. Mortara, G. Patane, M. Spagnuolo, and B. Falcidieno. 3D shape matching through topological structures. In *International Conference on Discrete Geometry for Computer Imagery*, pages 194–203. Springer, 2003.

[15] S. Biasotti, S. Marini, M. Spagnuolo, and B. Falcidieno. Sub-part correspondence by structural descriptors of 3D shapes. *Computer-Aided Design*, 38(9):1002–1019, 2006.

[16] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of combinatorial optimization*, pages 1–74. Springer, 1999.

[17] I. M. Bomze, M. Budinich, M. Pelillo, and C. Rossi. Annealed replication: a new heuristic for the maximum clique problem. *Discrete Applied Mathematics*, 121(1):27–49, 2002.

[18] A. Brière-Côté, L. Rivest, R. Maranzana, and É. de technologie supérieure de Montréal. Comparing 3D CAD models: Uses, methods, tools and perspectives. *Computer-Aided Design & Applications*, 2011.

[19] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[20] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. Partial similarity of objects, or how to compare a centaur to a horse. *International Journal of Computer Vision*, 84(2):163, Jul 2008.

[21] H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone, and M. Vento. A comparison of algorithms for maximum common subgraph on randomly connected graphs. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 123–132. Springer, 2002.

[22] B. Bustos, T. Schreck, M. Walter, J. M. Barrios, M. Schaefer, and D. Keim. Improving 3D similarity search by enhancing and combining 3D descriptors. *Multimedia Tools and Applications*, 58(1):81–108, 2012.

[23] A. Cardone and S. Gupta. Similarity assessment based on face alignment using attributed vectors. *Computer-Aided Design and Applications*, 3(5):645–654, 2006.

[24] A. Cardone, S. K. Gupta, A. Deshmukh, and M. Karnik. Machining feature-based similarity assessment algorithms for prismatic machined parts. *Computer-Aided Design*, 38(9):954–972, 2006.

[25] A. Cardone, S. K. Gupta, and M. Karnik. A survey of shape similarity assessment algorithms for product design and manufacturing applications. *Journal of Computing and Information Science in Engineering*, 3(2):109–118, 2003.

[26] S. K. Chandrasegaran, K. Ramani, R. D. Sriram, I. Horváth, A. Bernard, R. F. Harik, and W. Gao. The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design*, 45(2):204 – 228, 2013.

[27] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. In *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library, 2003.

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

[29] L. Chiang, F. Giannini, and M. Monti. *Identification of Regularities in CAD Part and Assembly Models*, pages 355–365. Springer International Publishing, Cham, 2016.

[30] C.-H. Chu and Y.-C. Hsu. Similarity assessment of 3D mechanical components for design reuse. *Robotics and Computer-Integrated Manufacturing*, 22(4):332 – 341, 2006.

[31] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298, 2004.

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

[33] A. S. Deshmukh, S. K. Gupta, M. V. Karnik, and R. D. Sriram. A system for performing content-based searches on a database of mechanical assemblies. In *ASME 2005 International Mechanical Engineering Congress and Exposition*, pages 411–423. American Society of Mechanical Engineers, 2005.

[34] G. Döllner, P. Kellner, and O. Tegel. Digital mock-up and rapid prototyping in automotive product development. *J. Integr. Des. Process Sci.*, 4(1):55–66, Jan. 2000.

[35] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based 3D shape retrieval. In *ACM SIGGRAPH 2010 Talks*, page 5. ACM, 2010.

[36] M. El-Mehalawi and R. A. Miller. A database system of mechanical components based on geometric and topological similarity. part i: representation. *Computer-Aided Design*, 35(1):83–94, 2003.

[37] M. El-Mehalawi and R. A. Miller. A database system of mechanical components based on geometric and topological similarity. part ii: indexing, retrieval, matching, and similarity assessment. *Computer-Aided Design*, 35(1):95–105, 2003.

[38] D. Eppstein and D. Strash. Listing all maximal cliques in large sparse real-world graphs. *Experimental Algorithms*, pages 364–375, 2011.

[39] B. Falcidieno, F. Giannini, J.-C. Léon, and J.-P. Pernot. Processing free form objects within a Product Development Process framework. In *Advances in Computers and Information in Engineering Research*, volume 1, pages 317–344. ASME-Press, 2014.

[40] M. A. Farooq, R. Kirchain, H. Novoa, and A. Araujo. Cost of quality: Evaluating cost-quality trade-offs for inspection strategies of manufacturing processes. *International Journal of Production Economics*, 188:156 – 166, 2017.

[41] T. A. Funkhouser and P. Shilane. Partial matching of 3D shapes with priority-driven search. In A. Sheffer and K. Polthier, editors, *Symposium on Geometry Processing*, volume 256 of *ACM International Conference Proceeding Series*, pages 131–142. Eurographics Association, 2006.

[42] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[43] F. Giannini, K. Lupinetti, and M. Monti. Identification of similar and complementary subparts in B-rep mechanical models. *Journal of Computing and Information Science in Engineering*, 17(4):041004, 2017.

[44] J. Goutsias and H. J. Heijmans. Fundamenta morphologicae mathematicae. *Fundamenta Informaticae*, 41(1, 2):1–31, 2000.

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

[46] S. A. Handbook. Iso 10303, version 3, 2006.

[47] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 203–212, New York, NY, USA, 2001. ACM.

[48] M. Hirz, W. Dietrich, A. Gfrerrer, and J. Lang. Overview of virtual product development. In *Integrated Computer-Aided Design in Automotive Development: Development Processes, Geometric Fundamentals, Methods of CAD, Knowledge-Based Engineering Data Management*, pages 25–50. Springer Berlin Heidelberg, 2013.

[49] T. Hong, K. Lee, and S. Kim. Similarity comparison of mechanical parts to reuse existing designs. *Computer-Aided Design*, 38(9):973–984, 2006.

[50] S. Hou and K. Ramani. Dynamic query interface for 3D shape search. In *Proceedings of DETC*, volume 4, pages 347–355, 2004.

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

[52] IBM. Speed product development with integrated digital mock-up solutions. `ftp://ftp.software.ibm.com/software/applications/plm/resources/05_PLM_001119_DMU_Whitepaper_LR_1.pdf`. Accessed: 14/06/2017.

[53] A. Çiçek. Similarity and scaling assessments of mechanical parts using adjacency relation matrices. *Journal of Materials Processing Technology*, 206(1-3):106–119, 2008.

[54] C. Y. Ip, W. C. Regli, L. Sieger, and A. Shokoufandeh. Automated learning of model classifications. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 322–327. ACM, 2003.

[55] L. H. Iwaya, R. S. Rosso, and M. d. S. Hounsell. A design for assembly application with dynamic information retrieval from case database. *IFAC Proceedings Volumes*, 46(7):186–191, 2013.

[56] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Shape-based searching for product lifecycle applications. *Computer-Aided Design*, 37(13):1435–1446, 2005.

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

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

[59] H.-W. Jung, S.-G. Kim, and C.-S. Chung. Measuring software product quality: A survey of iso/iec 9126. *IEEE software*, 21(5):88–92, 2004.

[60] K. Katayama and T. Sato. Matching 3D CAD assembly models with different layouts of components using projections. *IEICE TRANSACTIONS on Information and Systems*, 98(6):1247–1250, 2015.

[61] K. Katayama and T. Sato. A matching method for 3D CAD models with different assembly structures using projections of weighted components. *Journal of Information Processing*, 25:376–385, 2017.

[62] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on geometry processing*, volume 6, pages 156–164, 2003.

[63] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Shape matching and anisotropy. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 623–629. ACM, 2004.

[64] J. L. Kelley. *General topology.* Courier Dover Publications, 2017.

[65] K.-Y. Kim, H. Yang, and D.-W. Kim. Mereotopological assembly joint information representation for collaborative product design. *Robotics and Computer-Integrated Manufacturing*, 24(6):744–754, 2008.

[66] S. Kirkpatrick, D. G. Jr., and M. P. Vecchi. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[67] D. Kozen. A clique problem equivalent to graph isomorphism. *SIGACT News*, 10(2):50–52, July 1978.

[68] H. W. Kuhn. The hungarian method for the assignment problem. *50 Years of Integer Programming 1958-2008*, pages 29–47, 2010.

[69] H. Laga, M. Mortara, and M. Spagnuolo. Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes. *ACM Transactions on Graphics (TOG)*, 32(5):150, 2013.

[70] Z. Li, X. Zhou, and W. Liu. A geometric reasoning approach to hierarchical representation for B-rep model retrieval. *Computer-Aided Design*, 62:190–202, 2015.

[71] Z. Li, X. Zhou, W. Liu, Q. Niu, and C. Kong. A similarity-based reuse system for injection mold design in automotive interior industry. *The International Journal of Advanced Manufacturing Technology*, 87(5-8):1783–1795, 2016.

[72] X. Liang, M. J. Zuo, and Z. Feng. Dynamic modeling of gearbox faults: A review. *Mechanical Systems and Signal Processing*, 98:852–876, 2018.

[73] M. Lin, T. Shao, Y. Zheng, N. Mitra, and K. Zhou. Recovering functional mechanical assemblies from raw scans. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[74] Y.-J. Liu, X. Luo, A. Joneja, C.-X. Ma, X.-L. Fu, and D. Song. User-adaptive sketch-based 3D CAD model retrieval. *IEEE Transactions on Automation Science and Engineering*, 10(3):783–795, 2013.

[75] Y.-S. Liu, Y. Fang, and K. Ramani. Idss: deformation invariant signatures for molecular shape comparison. *BMC bioinformatics*, 10(1):157, 2009.

[76] K. Lupinetti, L. Chiang, F. Giannini, M. Monti, and J.-P. Pernot. Regular patterns of repeated elements in CAD assembly model retrieval. *Computer-Aided Design and Applications*, 14(4):516–525, 2017.

[77] L. Ma, Z. Huang, and Y. Wang. Automatic discovery of common design structures in CAD models. *Computers & Graphics*, 34(5):545 – 555, 2010.

[78] C. D. Manning, P. Raghavan, and H. Schütze. *Boolean retrieval*, page 1–17. Cambridge University Press, 2008.

[79] C. D. Manning, P. Raghavan, and H. Schütze. *Evaluation in information retrieval*, page 139–161. Cambridge University Press, 2008.

[80] M. Mantyla. *An introduction to solid modeling.* Computer Science Press, 1988.

[81] C. H. Mendonça. *The System Verification Breakdown Method*, pages 65–72. Springer London, London, 2007.

[82] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[83] T. Miura and S. Kanai. 3D shape retrieval considering assembly structure. In *Proceeding of Asian Symposium for Precision Engineering and Nanotechnology 2009 (ASPEN 2009)*, pages 11–13, 2009.

[84] R. Morris. Information analysis. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.8229&rep=rep1&type=pdf`.

[85] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[86] R. L. Norton. *Design of Machinery, no. 3.* McGraw-Hill Higher Education, 2003.

[87] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, 2002.

[88] J. Pakkanen, P. Huhtala, T. Juuti, and T. Lehtonen. Achieving benefits with design reuse in manufacturing industry. *Procedia CIRP*, 50:8–13, 2016.

[89] S. C. Park and J. W. Oh. Kinetic model extraction from a geometric model. *Computer-Aided Design and Applications*, 12(3):338–343, 2015.

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

[91] J.-P. Pernot, F. Giannini, and C. Petton. Thin part identification for CAD model classification. *Engineering Computations*, 32(1):62–85, 2015.

[92] S. Pommier and Y. Berthaud. *Mécanique générale.* Dunod, 2010.

[93] J. Pu, K. Lou, and K. Ramani. A 2D sketch-based user interface for 3D CAD model retrieval. *Computer-aided design and applications*, 2(6):717–725, 2005.

[94] J. Pu, K. Lou, and K. Ramani. A 2D sketch-based user interface for 3D CAD model retrieval. *Computer-aided design and applications*, 2(6):717–725, 2005.

[95] J. Pu and K. Ramani. A 3D model retrieval method using 2D freehand sketches. *Computational Science–ICCS 2005*, pages 17–77, 2005.

[96] F. Qin, S. Gao, X. Yang, M. Li, and J. Bai. An ontology-based semantic retrieval approach for heterogeneous 3D CAD models. *Advanced Engineering Informatics*, 30(4):751–768, 2016.

[97] F.-w. Qin, L.-y. Li, S.-m. Gao, X.-l. Yang, and X. Chen. A deep learning approach to the classification of 3D CAD models. *Journal of Zhejiang University SCIENCE C*, 15(2):91–106, 2014.

[98] J.-F. Rameau and P. Serré. Computing mobility condition using groebner basis. *Mechanism and Machine Theory*, 91:21–38, 2015.

[99] W. C. Regli, C. Foster, E. Hayes, C. Y. Ip, D. McWherter, M. Peabody, Y. Shapirsteyn, and V. Zaychik. National design repository project: a status report. In *International Joint Conferences on Artificial Intelligence (IJCAI) and AAAI/SIGMAN Workshop on AI in Manufacturing Systems, Seattle, Washington, USA*, 2001.

[100] R. Renu and G. Mocko. Retrieval of solid models based on assembly similarity. *Computer-Aided Design and Applications*, 13(5):628–636, 2016.

[101] R. S. Renu and G. Mocko. Computing similarity of text-based assembly processes for knowledge retrieval and reuse. *Journal of Manufacturing Systems*, 39:101–110, 2016.

[102] A. A. Requicha. Representations for rigid solids: theory, methods, and systems. 12(4):554–562, 1980.

[103] R. Riascos, L. Levy, J. Stjepandić, and A. Fröhlich. Digital mock-up. In J. Stjepandić, N. Wognum, and W. J.C. Verhagen, editors, *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, pages 355–388. Springer International Publishing, 2015.

[104] U. Roy and B. Bharadwaj. Design with part behaviors: behavior model, representation and applications. *Computer-Aided Design*, 34(9):613–636, 2002.

[105] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[106] M. Rucco, F. Giannini, K. Lupinetti, and M. Monti. Part classification with supervised machine learning. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *(*):*, Submitted.

[107] C. Sammut and G. I. Webb, editors. *True Positive*. Springer US, Boston, MA, 2010.

[108] A. Shahwan. *Processing Geometric Models of Assemblies to Structure and Enrich them with Functional Information.* PhD thesis, Grenoble, 2014.

[109] A. Shahwan, G. Foucault, J.-C. Léon, and L. Fine. Towards automated identification of functional designations of components based on geometric analysis of a DMU. In *GTMG2011-12èmes Journées du Groupe de Travail en Modélisation Géométrique*, page 61, 2011.

[110] A. Shahwan, G. Foucault, J.-C. Léon, and L. Fine. Deriving Functional Properties of Components from the Analysis of Digital Mock-ups. *Engineering Computations*, page 16, 2014.

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

[112] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling Applications, 2004. Proceedings*, pages 167–178. IEEE, 2004.

[113] Silvia, S. Marini, M. Spagnuolo, and B. Falcidieno. Sub-part correspondence by structural descriptors of 3D shapes. *Computer-Aided Design*, 38(9):1002 – 1019, 2006.

[114] A. K. Swain, D. Sen, and B. Gurumoorthy. Extended liaison as an interface between product and process model in assembly. *Robotics and Computer-Integrated Manufacturing*, 30(5):527–545, 2014.

[115] P. M. Swamidass. *Encyclopedia of production and manufacturing management.* Springer Science & Business Media, 2000.

[116] D. Systems. Creating the future. `https://www.3ds.com/informazioni-su-3ds/management/bernard-charles/`. Accessed: 14/06/2017.

[117] J. W. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia tools and applications*, 39(3):441–471, 2008.

[118] S. Tao and Z. Huang. Assembly model retrieval based on optimal matching. In *Software Engineering and Knowledge Engineering: Theory and Practice*, pages 327–336. Springer, 2012.

[119] R. L. R. C. S. Thomas H. Cormen, Charles E. Leiserson. *Introduzione agli algoritmi e strutture dati.* McGraw-Hill, 2005.

[120] B. Toche, G. Huet, G. McSorley, and C. Fortin. A product lifecycle management framework to support the exchange of prototyping and testing information. *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1259–1270, 2010.

[121] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42, 2006.

[122] J. U. Turner, S. Subramaniam, and S. Gupta. Constraint representation and reduction in assembly modeling and analysis. *Robotics and Automation, IEEE Transactions on*, 8(6):741–750, 1992.

[123] A. Tversky and I. Gati. Similarity, separability, and the triangle inequality. *Psychological review*, 89(2):123, 1982.

[124] D. G. Ullman. *The mechanical design process*, volume 2. McGraw-Hill New York, 1992.

[125] P. Wang, Y. Li, J. Zhang, and J. Yu. An assembly retrieval approach based on shape distributions and Earth Mover's distance. *The International Journal of Advanced Manufacturing Technology*, 86(9-12):2635–2651, 2016.

[126] P. Wang, J. Zhang, Y. Li, and J. Yu. Reuse-oriented common structure discovery in assembly models. *Journal of Mechanical Science and Technology*, 31(1):297–307, 2017.

[127] W. Wetzlinger, A. Auinger, and M. Dörflinger. Comparing effectiveness, efficiency, ease of use, usability and user experience when using tablets and laptops. In *International Conference of Design, User Experience, and Usability*, pages 402–412. Springer, 2014.

[128] D. E. Whitney. *Mechanical assemblies: their design, manufacture, and role in product development*, volume 1. Oxford University Press on Demand, 2004.

[129] R. D. Yang, X. Fan, D. Wu, and J. Yan. Virtual assembly technologies based on constraint and dof analysis. *Robotics and Computer-Integrated Manufacturing*, 23(4):447–456, 2007.

[130] C. Yiu Ip and W. C. Regli. Content-based classification of CAD models with supervised learning. *Computer-aided Design and Applications*, 2(5):609–617, 2005.

[131] C. F. You and Y. L. Tsai. 3D solid model retrieval for engineering reuse based on local feature correspondence. *The International Journal of Advanced Manufacturing Technology*, 46(5-8):649–661, 2010.

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

[133] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity search: the metric space approach*, volume 32. Springer Science & Business Media, 2006.

[134] J. Zhang, Z. Xu, Y. Li, S. Jiang, and N. Wei. Generic face adjacency graph for automatic common design structure discovery in assembly models. *Computer-Aided Design*, 45(8–9):1138 – 1151, 2013.

[135] L. Zhen-Ba, B. Shu-Hui, Z. Kun, G. Shu-Ming, H. Jun-Wei, and W. Jun. A survey on partial retrieval of 3D shapes. *Journal of Computer Science & Technology*, 28(5):836–851, 2013.

IDENTIFICATION DE CARACTERISTIQUES DE FORME ET DE STRUCTURE D'ASSEMBLAGES DE PIECES CAO POUR LA RECHERCHE DANS DES BASES DE DONNEES

RESUME :

Au cours des dernières années, de nombreux produits ont été conçus en utilisant des logiciels de CAO (Conception Assistée par Ordinateur) qui permettent de définir des modèles numériques 3D. Ces logiciels sont utilisés dans de nombreux domaines, tels que l'automobile, la marine, l'aérospatiale et plus encore. Au sein d'une entreprise qui utilise ces systèmes, il est possible d'avoir accès à des modèles CAO de produits déjà développés. Ceci est particulièrement intéressant lors de la conception de produits similaires pour permettre l'accès la connaissance des éventuels problèmes et leurs solutions. Par conséquent, il est utile de disposer de solutions technologiques capables d'évaluer les similitudes de différents produits afin que l'utilisateur puisse récupérer des modèles existants et avoir ainsi accès à des informations utiles pour la nouvelle conception.

Le concept de similarité a été largement étudié dans la littérature et il est bien connu que deux objets peuvent être similaires de plusieurs façons. Ces multiples possibilités rendent complexe l'évaluation de la similarité entre deux objets. À ce jour, de nombreuses méthodes ont été proposées pour l'identification de différentes similitudes entre les pièces, mais peu de travaux abordent ce problème pour le traitement d'assemblages de pièces. Si l'évaluation de la similarité entre deux pièces a beaucoup de points de vue, quand on va examiner des assemblages de pièces, les combinaisons de similarité augmentent vertigineusement puisqu'il y a plus de critères à considérer.

Sur la base de ces exigences, nous proposons de définir un système qui permette la récupération des assemblages des pièces similaires en fonction de multiples critères de similarité. Pour ce faire, il faut avoir un descripteur qui peut gérer les informations nécessaires pour caractériser les différentes similitudes entre les deux modèles. Par conséquent, l'un des points principaux de ce travail est la définition d'un descripteur capable de coder les données nécessaires à l'évaluation des similarités. De plus, certaines des informations du descripteur peuvent être disponibles dans le modèle CAO, tandis que d'autres doivent être extraites de manière appropriée. Par conséquent, des algorithmes sont proposés pour extraire les informations nécessaires pour remplir les champs du descripteur. Enfin, pour une évaluation de la similarité, plusieurs mesures entre les modèles sont définies, de sorte que chacune d'entre elle évalue un aspect particulier de leur similarité.

Mots clés :
Descripteurs de formes et d'assemblages, traitement de modèles géométriques, caractérisation et récupération de modèles CAO, filtrage hiérarchique

IDENTIFICATION OF SHAPE AND STRUCTURAL CHARACTERISTICS IN ASSEMBLY MODELS FOR RETRIEVAL APPLICATIONS

ABSTRACT:

The large use of CAD systems in many industrial fields, such as automotive, naval, and aerospace, has generated a number of 3D databases making available a lot of 3D digital models. Within enterprises, which make use of these technologies, it is common practice to access to CAD models of previously developed products. In fact, designing new products often refers to existing models since similar products allow knowing in advance common problems and related solutions. Therefore, it is useful to have technological solutions that are able to evaluate the similarities of different products in such a way that the user can retrieve existing models and thus have access to the associated useful information for the new design.

The concept of similarity has been widely studied in literature and it is well known that two objects can be similar under different perspectives. These multiple possibilities make complicate the assessment of the similarity between two objects. So far, many methods are proposed for the recognition of different parts similarities, but few researches address this problem for assembly models. If evaluating the similarity between two parts may be done under different perspectives, considering assemblies, the viewpoints increase considerably since there are more elements playing a meaningful role.

Based on these requirements, we propose a system for retrieving similar assemblies according to different similarity criteria. To achieve this goal, it is necessary having an assembly description including all the information required for the characterizations of the possible different similarity criteria between the two assemblies. Therefore, one of the main topics of this work is the definition of a descriptor capable of encoding the data needed for the evaluation of similarity adaptable to different objectives. In addition, some of the information included in the descriptor may be available in CAD models, while other has to be extracted appropriately. Therefore, algorithms are proposed for extracting the necessary information to fill out the descriptor elements. Finally, for the evaluation of assembly similarity, several measures are defined, each of them evaluating a specific aspect of their similarity.

Keywords:

Shape and assembly descriptors, Geometric model processing,

Model characterization and retrieval, Hierarchical filtering

IDENTIFICAZIONE DI CARATTERISTICHE DI FORMA E DI STRUTTURA IN ASSEMBLATI CAD PER APPLICAZIONI FINALIZZATE ALLA RICERCA DI MODELLI ASSEMBLATI IN DATABASE

SOMMARIO:

L'ampio uso di sistemi CAD in molti settori industriali, come quello automobilistico, navale e aerospaziale, ha generato un ampio numero di database di modelli di prodotti che rendono disponibile un'enorme quantità di dati digitali 3D. All'interno delle aziende che utilizzano questi sistemi, è frequente avere accesso ai modelli CAD di prodotti sviluppati in precedenza, poiché la progettazione di nuovi prodotti sovente fa riferimento a modelli già esistenti per riutilizzare soluzioni esistenti o per conoscere in anticipo eventuali problemi e relative soluzioni legati a prodotti simili. Diventa pertanto utile disporre di soluzioni tecnologiche che siano in grado di valutare le similarità di diversi prodotti, in modo che l'utente possa recuperare modelli già esistenti e di conseguenza avere accesso alle informazioni associate utili per la nuova progettazione.

Il concetto di similarità è stato ampiamente studiato in letteratura ed è risaputo che due oggetti possano essere simili sotto svariati punti di vista. Queste molteplici possibilità rendono complessa la valutazione della similarità tra due oggetti. Finora, molti metodi sono stati proposti per il riconoscimento di diverse similarità di parti, tuttavia pochi lavori hanno indirizzato questo problema trattando modelli assemblati. Se la valutazione della similarità di due parti può avvenire sotto molti punti di vista, quando si parla di assemblati i possibili criteri di similarità aumentano sensibilmente poiché vi sono più elementi da valutare.

Sulla base di queste necessità, il nostro obiettivo è la definizione di un sistema per il recupero di assemblati simili secondo vari criteri di similarità. A tale scopo è necessario avere a disposizione una descrizione del modello assemblato che includa tutte le informazioni che servono per definire tutti i possibili criteri di similarità tra due assemblati. Pertanto, uno dei punti principali di questo lavoro è la definizione di un descrittore in grado di codificare i dati necessari per la valutazione di similarità. Inoltre, alcune delle informazioni presenti del descrittore possono essere disponibili nel modello CAD mentre altre devono essere opportunamente estratte. Sono quindi proposti degli algoritmi per l'estrazione automatica delle informazioni necessarie al riempimento dei campi del descrittore. Infine, per una valutazione della similarità, sono state definite diverse misure di similarità, in modo che ognuna valuti un particolare aspetto di similarità.

Parole chiave:
Descrittori di forma e di assemblati, Elaborazione geometrica dei modelli, Caratterizzazione e recupero di modelli, Filtraggio gerarchico