

A reverse engineering method for DMU maturity management: use of a functional Reeb graph

Guillaume Herlem¹, Guillaume Ducellier¹, Pierre-Antoine Adragna¹, Alexandre Durupt² and Sébastien Remy¹

Abstract In a development process, Digital Mock-Up (DMU) is the referential view of the developed product, all along the product states (“as-design”, “as-manufactured”, etc.). In case of long lifetime products, such as boats or planes, each released product has its own DMU. During the use, maintenance operations are made. Some of the modifications are made “on field” and are not always reported to the DMU which is no longer mature. The product lifecycle is impacted and not efficient anymore. This paper focuses on a systematized methodology for checking the maturity level of a DMU by confronting it to the real product it is intended to represent. DMU inconsistencies correspond to unreported product components modifications. The first step of the process is a global comparison between the DMU and a digitization of the in-use product using the Reeb graph. The used comparison criterion is the topology. The second step is to identify the kept DMU components in the digitized dataset and thus deduce inconsistencies. That step implies a new shape descriptor combining topological and functional descriptions of a mechanical part: the Functional Reeb Graph.

Keywords DMU, PLM, Reverse Engineering, Reeb Graph, CAD, Shape Matching

1. Introduction

In the industry, PLM (Product Lifecycle Management) solutions are widely used for supporting product development projects. They are intended to be the answer to collaborative design and integrated engineering. It implies many actors with various kinds of expertise and often located in different geographical places. The modern

¹ Université de Technologie de Troyes, Troyes, France

{guillaume.herlem, pierre_antoine.adragna, guillaume.ducellier, sebastien.remy}@utt.fr

² Université de Technologie de Compiègne, Compiègne, France

alexandre.durupt@utc.fr

way to develop a product then is in extended mode: partners share and manage design information with massive use of 3D models. The project relies on a referential representation: the **DMU (Digital Mock-Up)** which is basically a representation of the product **BOM (Bill of Materials)** by an assembly of CAD models. It gathers different views of the product development related to expertise domains. The DMU is intended to be a referential representation during the different states of the product development: “as-design”, “as-manufactured”, “as-used”, etc. For mass consuming products (few years), a common DMU is used for all produced products. But for long lifetime products (dozens of years) such as aeronautical ones for example, each produced product has its own DMU. Even if the design or manufacturing phases are well handled, the “as-used” state can be more uncertain. In fact, unplanned maintenance activities can be made “on-field” on a long lifetime product, generating variations that are not always reported to the DMU. The PLM is no longer efficient. The inconsistency of the DMU is a problem which impacts further planned modifications: design activities based on non-up-to-date DMU data generates mistakes and additional delays caused by an a posteriori on-field data discrepancy checking. This paper focuses on a reverse engineering approach for identifying DMU inconsistencies in a discrepancy checking context. We propose to acquire a virtual view of the in-use product and confront it to its digital representation. We present a new way to describe CAD models for identifying components in a digitized product assembly. The aim is to isolate the implied component(s) causing the maturity differential with the in-use product. The process is intended to be integrated in a CAD environment. In section 2, we briefly present related works. The proposed solution is presented in section 3. In section 4, we discuss on our solution.

2. Related Works

Our problematic is partially covered by 3 scientific research domains: (1) the global aim is to check the DMU and its level of maturity, an activity close to “discrepancy checking”, (2) our approach of the problem (retrieving virtual data from a real object) is similar to “reverse engineering” approaches and (3) identifying 3D models (identifying components) in a scanned dataset is linked to “shape matching” studies.

2.1 *DMU Discrepancy Checking*

DMU discrepancy checking is a kind of 3D inspection which consists in confronting a manufactured model to its digital representation. The purposes can be multiple: checking unplanned component and validate the manufacturing as in [1] and [2] or update the design after manufacturing a physical prototype as in [3]. 3D inspection is well-covered and widely used in the industry to check mechanical parts. A 3D model is retrieved by generating surfaces on a digitized point cloud (using

laser scanner for example) from a real model. Then, the 3D inspection resides in the analysis of the deviation between the original design and the captured one. DMU discrepancy checking is the application to assembly models. The recent works use other means than laser scanner like augmented reality ([2], [3]) or on-site pictures ([1]). The proposed solutions are applied to static systems like plants or piping. There is not any application on mechanical systems (with mobility and different poses between the DMU and the reality).

2.2 Reverse Engineering

Retrieving data from a real part or a product into a virtual engineering framework is known as **Reverse Engineering (RE)**. The main objective is to enable redesign or new design activities based on a physical object. In the scientific literature, RE is a well-covered domain. The input data is generally a point cloud (acquired as in 3D inspection for instance). The first studies, initiated with [4], were based on a pure geometrical approach with the fitting and assembling of surfaces on the point cloud. The rebuild model is automatically generated from the point cloud and that fully-geometrical approach is well-implemented in commercial software. But the model is generally frozen and does not enable redesign activity without tedious work. Another kind of approach exists: rebuilding a model using features. Studies like [5] propose to retrieve a 3D model by fitting manufacturing features or functional features in the point cloud. The features are geometrical sets driven by parameters. The result is close to a common 3D model handled in CAD software. Thus, redesign activity is possible. For now, all those solutions have been fully-tested on single part component. A new scientific and industrial challenge is to apply them to the components assembly context in order to get a “virtual view” of an in-use product assembly.

2.3 Shape Matching

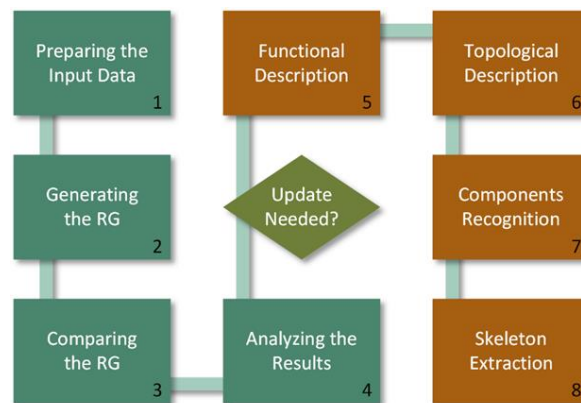
Comparing two 3D models is covered in the “shape matching” (or “shape retrieval”, “shape indexing”) methods. It consists in defining a (dis)similarity level between the models by comparing them using their shape signature. The most common purpose is to retrieve similar shapes in a 3D models database. And the main challenge of shape matching is to describe in a compact but highly descriptive way the shape of a 3D model. That shape signature is obtained with the use of a shape descriptor, mainly applied on the 3D model meshed surface. A shape descriptor is a 3D model discretization based on an abstraction of the model. A way to check the discrepancy between two products assembly representations is to identify common components between them. Shape descriptors have that potential possibility which has not been tested yet.

Among the different shape descriptors in the scientific literature, the **Reeb Graph (RG)** initiated in [6], is a well-covered graph-based representation with a high descriptive power and a high potential of evolution, as in [7], [8] and [9]. The generation principle is simple³. A scalar function is applied on each vertex of the meshed model. The vertices values are sorted into groups of same values intervals. A RG node is attributed to each group of vertices (the node is the barycenter of the mesh portion) and a RG edge connects two RG nodes if the corresponding mesh portions are connected. The RG nodes gather attributes: computed local information like surface area or curvature. The 3D models comparison is then simplified to a 1D graphs comparison. The choice of the mathematical function enables the RG to be insensible to the object orientation and pose. That property is useful for our context. Indeed, an in-use product may have a different pose and a different orientation (due to its installation, its use or its wear for example) from the setup defined in its DMU which can be arbitrary.

2.4 Conclusion on Related Works

In our study of the related works, we identify a shape descriptor with a high potential of application for our purpose: (1) comparing a DMU and a digitized dataset from a real product, (2) identifying DMU components in the dataset. For the first step, it can be simplified as the comparison of two 3D models, as in shape matching methods. It is possible to extract an external envelop from a DMU and mesh it. And a digitized point cloud from a real object can be easily meshed too. The comparison between two meshes is well-covered. The Reeb graph (RG) (and its extensions) is a powerful tool, suitable for the meshed assemblies' comparison. For the second step, it appears that topology (and local embedded geometrical information) is not enough. We need to retrieve components in a meshed assembly model. In terms of shape matching, it is similar to partial shape matching: the component model is here

Fig. 1. This paper proposition of a reverse engineering process for DMU maturity management



³ More details on the RG generation are available in [7].

a subpart of the assembly model. And even if it can be applied to simple shapes, mechanical parts are more complex: as stated in [10], the RG of a meshed components assembly model is not the “assembly” of the RG of each component model. The contacts between part components generate undefined zones.

3. Proposed Solution

3.1 Main Principle and DMU Inconsistencies Definition

We consider two input data:

- A point cloud digitized from the real product
- A product DMU at an unknown maturity state

For being compared on a common criterion, the shape, the two inputs are meshed giving two 3D representations of the product. We propose a process in two steps assisted by a product expert:

1. Determining the presence of at least one inconsistency,
2. If at least one inconsistency, identifying the component(s) generating that inconsistency(ies)

That steps are made of activities, numbered 1 to 4 for the first step and numbered 5 to 8 for the second step (Fig. 1).

Our proposition is, at first, to consider a product assembly as a unique 3D model. So a component modification results in a modification of the topology of the assembly model. Our contribution is to use the study of the topology variation as an indicator of the maturity state of the DMU (among other possible indicators but not presented in this paper). Thus, an inconsistency is assumed to be:

A component which is $\left\{ \begin{array}{l} \textit{added} \\ \textit{lacking} \\ \textit{replaced} \\ \textit{moved} \end{array} \right.$ in the *real product* compared to the *DMU*.

And then, for the first step, in a global approach, we propose to use the aMRG comparison presented in [8]. Our paper does not detail that first step, as we treat the meshed DMU and the meshed scanned data as two simple 3D models. The graphs generation and comparison are the same as the aMRG solution. In this section, we consider that the process is applied to an in-use product which has been modified and its DMU has not been updated. That suspicion has been confirmed to the expert by the first step of the process (the global comparison using aMRG).

In this contribution, we present the second step of our process: the recognition of the DMU components in the meshed scan (Activities 5 to 7). The expert focuses on the meshed scan area generating the dissimilarity score.

*Our process is supported by a **CPM (Core Product Model)** extension originally introduced in [10]. The extension is made for supporting our approach by managing the new product representation: abstractions based on topology and kinematics and used in the descriptor presented in the next section. This paper does not present the data model.*

3.2 A New CAD Model Descriptor

As stated in section 2.4, the RG is not suitable “as-is” for identifying a component in a meshed assembly. For supporting our identification purpose, we propose to derivate a new RG shape descriptor (more specific to CAD models) from a simple definition of a mechanical part:

“Functional features linked by Topological features”

In our study, we choose to restraint functional features to be the sets of geometrical elements of a part aimed to enable mobility with compliant features of another part. We only consider the functional surfaces linked to the kinematic domain. The other geometrical elements are simply considered as links between functional features, as

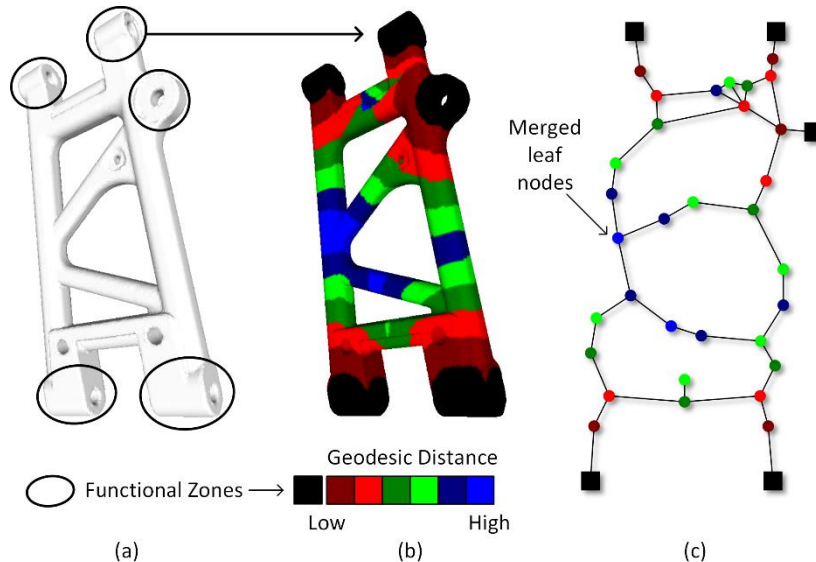


Fig. 2. The FRG generation on a lower triangle of a suspension

topological features. This descriptor, the **FRG (Functional Reeb Graph)**, is a direct illustration of that basic definition. We extend the current works on the RG by a knowledge addition: the creation of a new graph node, the functional node. That node is aimed to describe the “undefined zones” detailed in [10]. In fact, in the case of a mechanical product, they are the areas corresponding to the kinematic pairs of the product. We proposed to inspire the new descriptor generation from [9]: using feature points as sources for the RG generation. In our case, the feature points are the centroids of the functional zones of a part, the functional nodes. The principle then is to link the different functional zones by travelling along the topology of the model. For that step, we use the approach of the geodesic distances computation presented in [7] and [8]. An example of the generation of an FRG on mechanical part is displayed in Fig. 2.

The computation of the FRG follows this procedure:

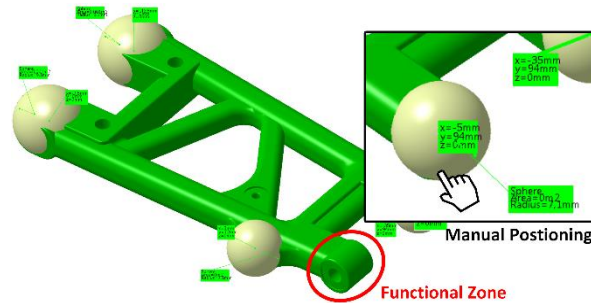
1. For each functional node (from functional zones, Fig. 2a), the geodesic distances to all the mesh vertices are computed
2. Each mesh vertex is attached to the closer functional node, giving a set of “under influence” vertices for all functional node
3. For each set of vertices:
 - 3.1. The vertices are sorted by group of geodesic distance intervals (Fig. 2b)
 - 3.2. A topological node is attributed to each group of vertices (as in [7] and [8])
 - 3.3. An edge links two nodes if the vertices groups are connected (as in [7] and [8])
4. The connected sets are linked by merging the corresponding leaf nodes (Fig. 2c)

3.3 *The Inconsistency Identification*

This section presents the aim of the FRG: enabling the inconsistency identification (activities 5 to 8 in Fig. 1).

Functional Description (5) Using a custom user interface, the expert defines the kinematic pairs locations on his focus on the meshed scanned product. He uses enclosing spheres that can be stretched (Fig. 3). Not all the system is described. Our use case in Fig. 4 is a mechanical part of a vehicle suspension. The previous step of the process enables to locate the dissimilarity area so only that area is treated, not all the vehicle or all the chassis. From this enclosing shapes (Fig. 4a), which are the

Fig. 3. The localization of the functional zones of a mechanical part



functional zones (darker mesh areas in Fig. 4b), the centroids are computed giving the functional nodes.

Topological Description (6) This activity corresponds to the FRG generation: the topological links between the functional zones are computed (Fig. 4b), generating topological nodes and edges of the FRG. The same generation as described in 3.2 is applied to the meshed assembly. The resulting FRG then is a direct assembly of the FRG of the components as they are based on the functional zones which are undefined zones with previous RG alternatives.

Components Recognition (7) When the meshed scanned assembly have been described with the FRG, an algorithm is ran to identify the DMU components that are present in the real product scan. That algorithm is similar to partial shape matching basic procedures which can be found in the scientific literature. The variation is the addition of the number of functional nodes as a primary search criterion. For that activity, we assumed that all DMU components are described with the FRG and accessible by querying a database with our graph structure. And they are retrievable under a meshed form. A list of matching results is presented to the product expert who validates the right components. The Fig. 4c illustrates the recognized lower

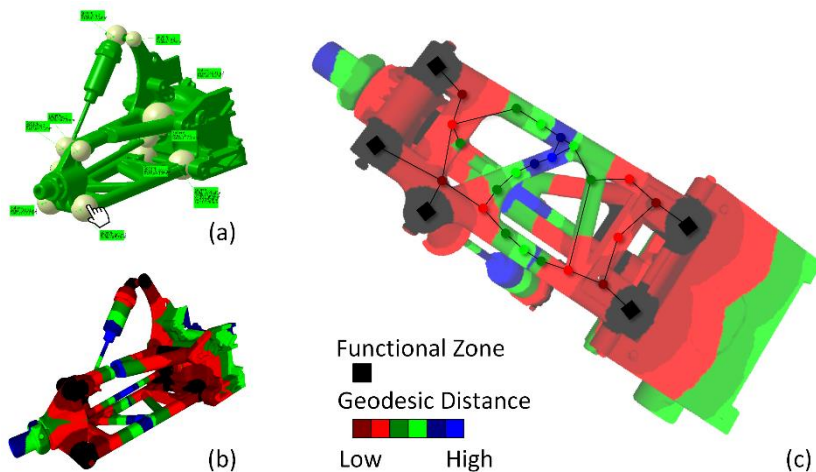


Fig. 4. The identification of the lower triangle in the meshed assembly scan using the FRG

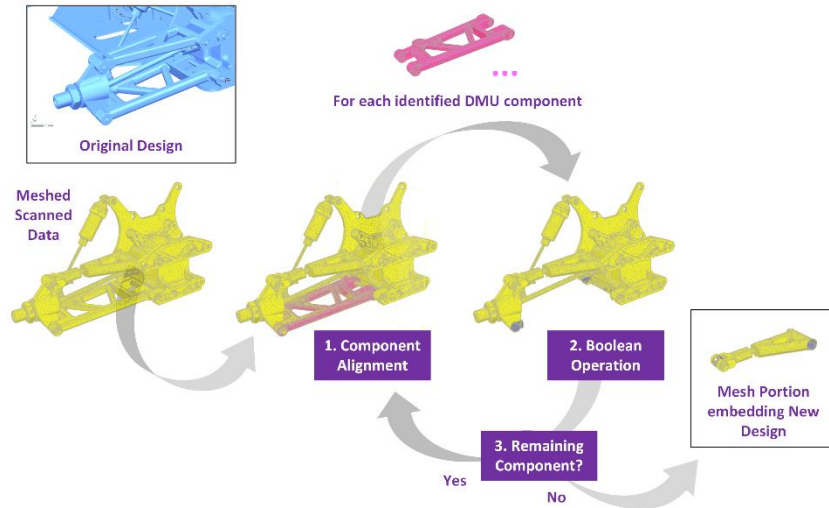


Fig. 5. The suppression of the meshed scan portions corresponding to recognized components

triangle in the meshed assembly scan. Using the nodes locations, each recognized and validated component mesh is coarsely aligned in the assembly mesh and then finely fitted using best-fit algorithms. The equivalent meshed scanned portion is then suppressed (Fig. 5).

Skeleton Extraction (8) All the mesh areas of the identified components have been suppressed. The remaining mesh portion(s) (Fig 5) correspond to unrecognized component(s): DMU inconsistency(ies) as defined in section 3.1. The skeleton extraction from that remaining mesh is still a work in progress activity. Its aim is detailed in the further works in section 4.

4. Conclusion

This paper proposes an approach for checking DMU discrepancy. The main contribution is the development of a new CAD model descriptor based on the Reeb graph (and evolutions) shape descriptor: the Functional Reeb Graph. As a knowledge addition to the topological description, we proposed to take into account, in a basic way, the purpose of a mechanical part: the “kinematic” intent. The FRG allows identifying the inconsistencies of a product DMU as we defined them: unreported component modifications.

Discussion The proposed process enable to isolate a partial meshed model corresponding to the unrecognized component which can be a useful basis for a reverse engineering process with a CAD commercial software. In the methodology setup resides two strong constraints: the studied system needs to be a kinematical one with a high exposition of its components and a product expert must be implied.

Further Works The next step is to improve the information stored in the functional node. At the moment, it is only the centroid of a functional zone of a component. In the future, information on the kinematic pair the functional zone is intended to enable will be stored in the node. Further works will be on the information extraction from the isolated mesh part corresponding to the implied component. We strongly believe it is possible to store functional features and topological features directly into the nodes. The final stage of our approach will be the generation of a geometrical skeleton of the component enclosed in the remaining mesh portion of the scanned product. That skeleton will be a support for the redesign activity in a CAD environment for updating the DMU.

Acknowledgments The study presented in this paper is possible thanks to the European Regional Development Fund (ERDF) and the Champagne-Ardenne Region providing NUM3D equipment.

References

- 1 Georgel, PF, Schroeder, P, Navab, N (2009) Navigation tools for viewing augmented CAD models. *IEEE Computer Graphics and Applications* 29(5):65-73
- 2 Kahn, S, Wuest, H, Stricker, D, Fellner, DW (2010) 3D discrepancy check via augmented reality. In: 9th IEEE International Symposium on Mixed and Augmented Reality 2010: Science and Technology, ISMAR '10, p. 241-242, IEEE, Seoul, South Korea.
- 3 Webel, S, Becker, M, Stricker, D, Wuest, H (2007) Identifying differences between CAD and physical mock-ups using AR. In: Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '07, p. 281-282, IEEE, Nara, Japan
- 4 Várady, T, Martin, RR, Cox, J (1997) Reverse engineering of geometric models - An introduction. *CAD Computer Aided Design* 29(4):255-268
- 5 Durupt, A, Remy, S, Ducellier, G (2010) KBRE: A knowledge based reverse engineering for mechanical components. *Computer-Aided Design and Applications* 7(2):279-289
- 6 Reeb, G (1946) Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. In : *Comptes Rendus de L'Académie des Sciences*, vol 222, p. 847-849, Paris, France
- 7 Hilaga, M, Shinagawa, Y, Kohmura, T, Kunii, TL (2001) Topology matching for fully automatic similarity estimation of 3D shapes. In: Proceedings of the 28th International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01, p. 203-212, ACM, New York, NY, USA
- 8 Tung, T, Schmitt, F (2005) The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modeling* 11(1):91-120
- 9 Tierny, J, Vandeborre, JP, Daoudi, M (2009) Partial 3D shape retrieval by Reeb pattern unfolding. *Computer Graphics Forum* 28(1):41-55
- 10 Herlem, G, Adragna, P-A, Ducellier, G, Durupt, A (2012) DMU Maturity Management as an Extension of the Core Product Model. *Product Lifecycle Management: Towards Knowledge-Rich Enterprises* 388:192-201