

This paper has been published in Computers in Industry, Vol. 65, Issue 1, 2014, Pages 124-135.  
DOI: 10.1016/j.compind.2013.08.003

Borhen LOUHICHI, Pr. Louis RIVEST.

## **Maintaining Consistency between CAD Elements in Collaborative Design using Association Management and Propagation**

Borhen LOUHICHI, PhD; Pr. Louis RIVEST, PhD  
Department of Automated Production Engineering  
Ecole de technologie supérieure, Montreal, Canada  
1100, rue Notre-Dame Ouest  
Montréal, Québec  
Canada H3C 1K3  
+1 (514)396-8984  
[borhen.louhichi@etsmtl.ca](mailto:borhen.louhichi@etsmtl.ca), [louis.rivest@etsmtl.ca](mailto:louis.rivest@etsmtl.ca)

### **Abstract**

The complete definition of a product often requires the collaboration of various partners. Data sharing and exchange between partners has thus become an important task throughout a product's entire life cycle. Even while subsets of the product definition are exchanged (as work packages) and modified by various partners, the global product definition must remain consistent. This paper focuses on maintaining consistency between Computer-Aided Design (CAD) work packages and the global product Digital Mock-Up (DMU). The approach is designed to ensure better management of the associations between objects when a work package is extracted from the global DMU, modified by a partner, sent back to the originator and then re-inserted into the global DMU, which must be modified in turn so as to maintain consistency. To this end, we propose an association management model for the digital mock-up (Digital Mock-up Association Management Model, DMU-AMM) that transposes the associations that exist between a DMU and a work package, including package extraction and modifications, to ultimately guide the evolution of the DMU so as to reconcile the associations between a modified DMU and the modified work package and thereby maintain consistency.

**Keywords.** Digital Mock-up, Work package, Associations, Transposition of associations, CAD data, Change propagation, Reconciliation matrix.

## 1. Introduction

In the current collaborative and extended-enterprise environment, data exchange between project partners is pervasive and critical. An evolving product definition must be exchanged between original equipment manufacturers (OEMs) and various partners, who all contribute to the evolution of a product's definition. The OEM needs to preserve confidentiality while maintaining consistency in the complete product definition. Information-filled work packages (WPs) are extracted from the OEM's Digital Mock-up (DMU) and sent to partners to allow them to perform the required work on these packages. These information/work packages, which contain CAD and possibly other types of files, should be inserted back into the original OEM DMU after modification by a partner. With today's CAD tools, numerous dependencies, here designated as associations, relate information objects located in a file to other objects located in other files. For example, a shaft diameter value in a parametric CAD file can be associated with a calculation result performed in a spreadsheet; similarly, interface elements between work packages, such as a plane, can be defined in a skeleton model and exploited by many other CAD models. At some point in the preliminary or detailed design of complex products, such as airplanes, thousands of such associations exist. However, few solutions available today are capable of systematically managing these associations between a WP and its OEM DMU; restricting the ability to quickly and easily create WPs and reconcile them with the DMU when necessary. As a consequence, WPs may be sent with information missing, the context cannot be maintained; and reinsertion of the modified WP into the OEM DMU leads to tedious manual work and/or inconsistencies. No solution allows the systematic management of the evolution of associations.

Our objective is to promote efficient management of the associations that relate a WP to the OEM DMU in order to facilitate product information sharing among partners. A major challenge toward achieving this objective is to define a conceptual model of a WP and its context such that the various associations can be systematically managed.

The paper proposes a Digital Mock-up Association Management Model (DMU-AMM) that can capture associations between the initial work package (iWP), extracted from the DMU, and the initial state of the OEM digital mock-up (iDMU), monitor the modifications performed on the initial work package (iWP) so as to obtain the modified work package (mWP), as well as reconcile associations between the modified work package (mWP) and the digital mock-up when propagating changes to the iDMU to obtain a modified DMU (mDMU) and maintain consistent information.

This work was performed within an industry-led research program called the Collaborative development for Product Lifecycle Management [1], which aims at developing a collaborative environment for better information management in the product development process while maintaining confidential data security.

This paper is organized as follows. First, a conceptual approach is proposed, to structure the problem at hand. A literature review follows, covering data management, information sharing and the propagation of change. The proposed approach and algorithms utilized to provide the desired

management tool enabling the reconciliation of associations and assisting users are detailed next, followed by the model validation results. The conclusions and perspectives for this work are presented at the end.

## 1. Terminology

The following terminology is defined to help describe our approach in a consistent, clear manner.

*Object*: a generic term which includes: model, assembly, part, features, geometric elements and documents [2].

*Entity*: any of the smaller elements that makes up the object and can be used to formalize procedural knowledge (face, edge, circle radius...). Entities are themselves objects [2].

*Global Digital Mock-up*: acts as a repository for the numerical definition of the product and is the result of the collaboration between several partners; as a minimum, in our working context, it gathers the CAD data that defines the product.

*Work Package*: data package that includes parts or an assembly of parts. It is intended to be modified by a design partner and reinserted into its context (the Global Digital Mock-up).

*Association*: a generic term used to express a dependency between objects; an association is a relation, a link or a constraint [2].

*Relation*: a dependency between two objects that does not involve procedural knowledge [2].

*Link*: a dependency between two objects that requires procedural knowledge to execute a specific task [2].

*Constraint*: a dependency between two objects (usually low-level objects, and thus entities) that involves a formalized knowledge (coincidence, parallelism ...); for example, using a CAD system to define parts and assemblies basically involves defining constraints [2].

*Skeleton model*: skeleton model data should be considered as the initial underlying geometry for CAD modeling. The use of a skeleton model allows for conceptual product modeling before the creation of any solid geometry and enables the control of direct parent-child relationships between solid features in parts or between parts and/or subassemblies. The elements of the skeleton model are geometric entities: datum, curves, and sketches, and non-geometric entities: parameters, equations, and rules [3].

## 2. Proposed conceptual approach

In this section, the CAD elements' consistency maintenance problem described above is examined. First, we consider the associations that relate objects from the initial work package to objects in the modified work package; three cases are delineated. Next, we focus on associating objects from the modified work package to the DMU where the evolution of these objects is propagated to the DMU.

### 2.1 First Step: Work Package Evolution

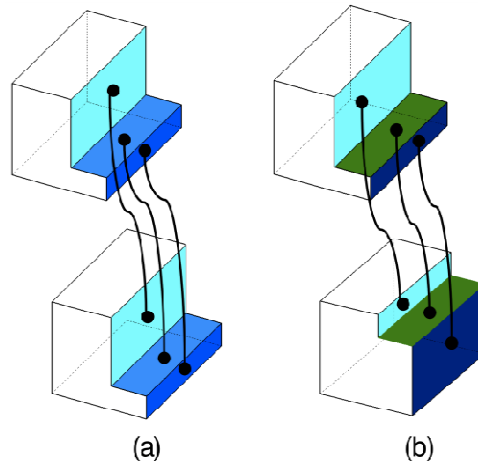
Whatever the motivation behind the work package evolution, it is expressed through CAD object versions as modifications to geometry. The *original* object (O-Object) being sent from an OEM to a partner as part of an iWP will differ from the next version of the object being *returned* (R-Object) to the OEM by the partner. Each R-Object is either an evolution of a previous version or it is a completely new version. The objective here is to associate, or to establish correspondence between entities that belong to the O- and R-Objects. Several cases can be considered.

Hoffman et al. [4] [5] distinguish three categories of shape (model) update: 1) *Shape changes*; 2) *Changes of parameters, dimensions, and constraints*; and 3) *Changes of attributes*. According to those authors, the effect of editing the shape of any shape geometric element is composed of one of the following events:

- (1) The element has moved,
- (2) The element has been deleted,
- (3) The element has been joined with another element,
- (4) The element has been split into several new elements,
- (5) The element has been enlarged or restricted,
- (6) The element has been created.

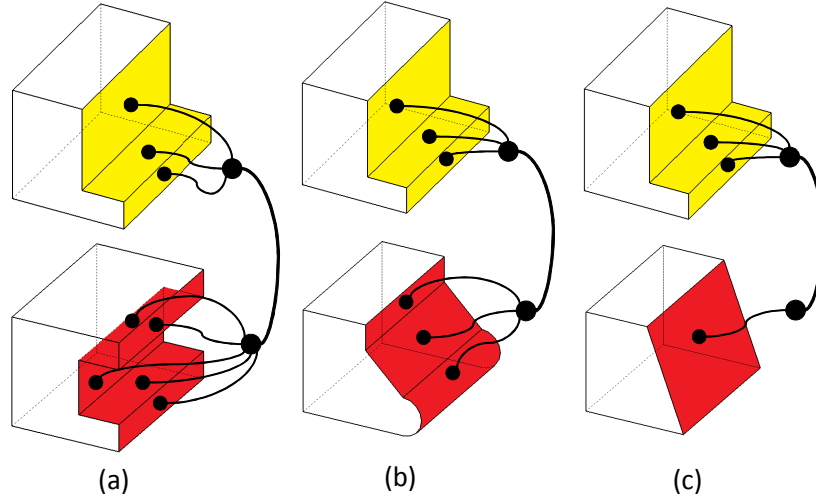
Hoffman et al.'s classification of changes focuses on elements (or entities); we focus on the associations between entities. We distinguish three cases. To simplify the discussion, objects are hereafter limited to faces.

- *Case 1 - Valid associations:* In this case, the matching between entities of the R-Object and those of the original O-Object is obvious if all the entities' ID's are preserved. The R and O objects are either *absolutely identical* (Fig. 1 (a)), that is to say, the object is not modified, or *structurally identical* (Fig. 1 (b)), if the object is modified but its constituting entities keep their cardinalities and identities. The correspondence between entities is therefore trivial and the associations are valid.



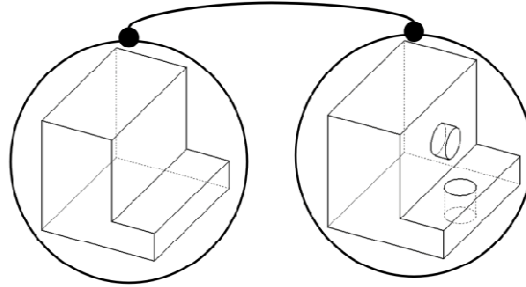
**Fig. 1.** (a) Absolutely identical object and (b) structurally identical objects; valid associations.

- *Case 2 - Partially undetermined associations:* In this case, some entities are orphans upon first analysis. This occurs either because the number of entities differs between the O- and the R-Objects, or because some entities find no 'trivial' equivalent via geometric equivalence or ID persistence. The former happens when  $m$  entities of the O-Object are destroyed and replaced by  $n$  entities on the R-Object. We therefore establish the correspondence between entities by grouping them. Three cases can be distinguished based on cardinality:  $m < n$  (Fig. 2 (a)),  $m = n$  (Fig. 2 (b)), and  $m > n$  (Fig. 2 (c)). In Fig. 2, the faces shown in white are paired (matched) based on geometric equivalence or ID persistence, which greatly reduces the search for matching unpaired entities, that is, for entities whose associations are undetermined. Algorithms to transform these 'grouped entities associations' into 'individual entities associations' are proposed in the next sections.



**Fig. 2.** Partially undetermined associations a)  $m < n$ ; b)  $m = n$ ; c)  $m > n$ .

- *Case 3 - Undetermined associations:* In this case, all entities' IDs are destroyed and their cardinality may vary as well. This may occur when there is a passage through a neutral format or when the entire object is replaced by a new one. All constituting entities are therefore considered to be new. The correspondence is established, in a first step at least, between the whole groups of entities of the two objects: original and returned (Fig. 3). Of course, some analysis can be conducted so that entities can be paired based on their geometric definitions, thereby transforming an extreme case into a 'Partially undetermined associations' case.



**Fig. 3.** Undetermined associations.

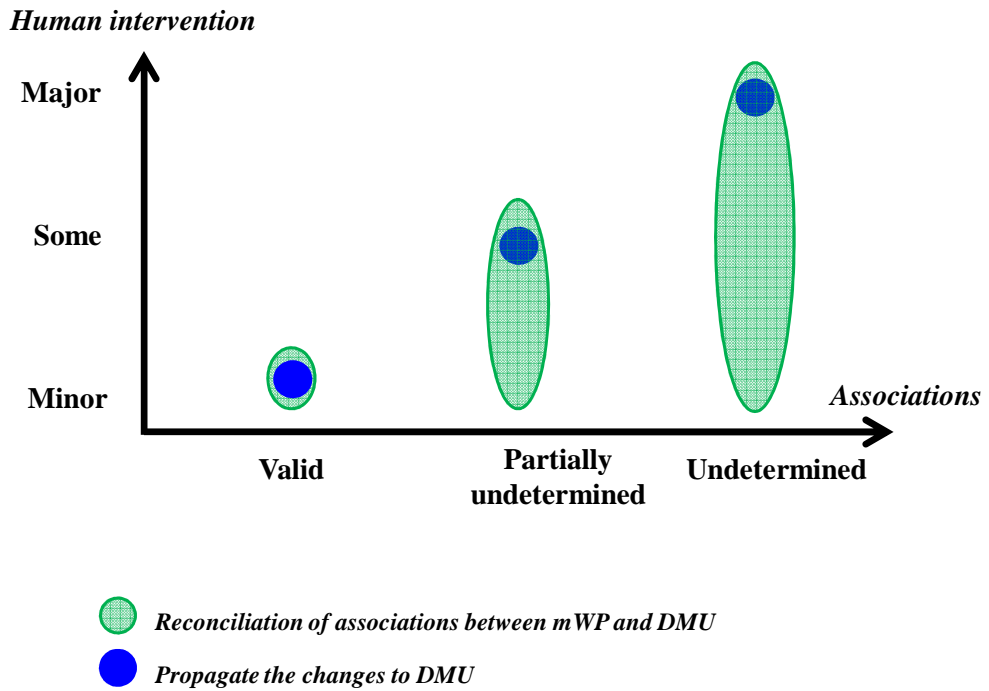
## 2.2 Second Step: Work package reinsertion into DMU's: scenarios

The reinsertion of the R-Object (modified work package) into the digital mock-up implies the reconciliation of associations (between the Work Package and the Digital Mock-up) and the propagation of changes in the DMU to preserve data consistency. The complexity of the reinsertion work depends on the evolution from the iWP to the mWP and on how associations can be handled between mWPs and DMUs (Fig. 4):

- If the O- and the R-Objects are structurally identical, associations between them are valid and the reinsertion is trivial, since associations between the R-Object and the DMU are

similar to those between the O-Object and the DMU, and hence are easily reconciled. User intervention may only require a very minor update (change of dimension, etc.) of the DMU.

- In the case when there are partially undetermined associations (between O- and R-Objects), entities are aggregated to be associated as groups of entities. However, associations between the R-Object and the DMU are expected at the individual entities' level. In this case, the propagation of changes towards the digital mock-up requires some user intervention.
- If the R-Object is such that all associations are undetermined, the proposed solution is to transform the situation into an easier one where some entities will be re-associated, while others will remain orphaned. To achieve this, comparison algorithms perform this re-association of similar entities [6][7][8]. However, describing this specific algorithm is not within the scope of this paper.



**Fig. 4.** Complexity of reinserting mWPs into the DMU in relation to the type of evolution scenario.

Fig. 4 summarizes the three levels of complexity in reconciling associations and propagating changes, according to the work package change scenario. Recall that the objective is not to eliminate user intervention but rather to assist the user via a management model.

### 3. State of the art

This paper can be positioned with respect to research work belonging to three themes: 1) the management of associations between heterogeneous objects; 2) the management of associations within assemblies; and 3) CAD data sharing and change propagation.

### **3.1 Management of associations between heterogeneous objects**

Fortin et al. [9] developed dynamic links allowing a synchronous evolution of product development from design to production. These dynamic links take the form of equity, occurrence and reference links. They provide a continuous dynamic data exchange between different expertises, allowing feedback from manufacturing to reach all the way to design. In the same context, the work of Toche et al. [10] is focused on the implementation of a communication model enabling interoperability between the prototyping phase and the design phase. Their model is based on mapping coding and decoding through a central interface exchange. Zimmermann et al. [11] developed a system containing generic inter-view links called "*Universal Linking of Engineering Objects*", which forms relationships between technical objects from different professions (design, manufacturing, quality). The system allows for the construction of a unique multi-view model.

Hoffman et al. [4] developed a "CAD and product master model" to manage mapping between different model views (CAD, machining process planning (MPP), geometric dimensioning and tolerancing (GD&T)) and to maintain consistency between them. According to Hoffman, coordinating the views (CAD, GD&T, etc.) is the responsibility of the master model.

Yassine et al. [12] developed a system that provides improved management of the heterogeneous elements that make up the product development process. The system is based on a connectivity map that captures the dependency relationships. Yassine et al. [13] developed the DSM (Design Structure Matrix) based on the DRFT approach (Do-it-Right-First-Time). The DSM is a compact representation of the design process information. It is a design plan showing the order of tasks as well as the necessary checks required in the design process. The DSM increases the efficiency of the process and reduces production time. The conception of the DSM-DRFT model was inspired by other studies that used the DSM, but from other points of view: process [14], product [15] and organization [16] [17].

### **3.2 Management of associations within CAD assemblies**

To improve CAD assembly management, Fouda et al. [18] proposed a precedence graph generation system for product components. The associations between components are harnessed; physical contacts are described, as well as insertion order, stability and relative motion. The precedence graph is generated based on four rules: minimization of the number of repurposings; maximization of the number of possible assembly sequences; the stability of sub-assemblies; and the priorities among the components. Li et al. [19] proposed a fast assembly system based on TAFs (typical assembly features). TAF concepts are developed for functionality. The mechanism is based on assembly features and the possible constraints between them. It enables the complexity of the assembly process to be reduced and effectively manages its components. Mascle et al. [20] developed the SCAP (Security Content Automation Protocol) capable of creating a geometric and technological mock-up from various assembly components. SCAP consists of four modules: the first module assigns a signature to each component of the assembly, describing its assembling priority, its materials, etc.; the second assigns each component a signature to decipher the number of its instances in the assembly and how they are dispatched (in a circular fashion, linear, etc.); the third module specifies the process used to assemble the component (welding, gluing, etc.); and the fourth module identifies sub-assemblies and their instances. The SCAP system improves the assembly and disassembly time of a product while automatically deducing certain characteristics of the assembly process.

In the context of the product and assembly relationships' management in the concurrent engineering and product lifecycle management (PLM) domain, Demoly et al. [21] [22] have

developed a novel approach to integrate assembly process engineering information and knowledge in the early phases of the product development process. Their proposed approach – called SKeLeton geometry-based Assembly Context Definition (SKL-ACD) – enables the control of the product modelling phase by introducing skeleton entities consistent with the product relationships and the assembly sequence planning information [21]. SKL-ACD is based on six skeleton entities: Point entity, Line (or straight) entity, Plane entity, Coordinate system entity, Constraint entity, Parameter entity.

### ***3.3 CAD data sharing and change propagation***

Data sharing and the propagation of changes between CAD objects are addressed by Chen et al. [23] via a multi-level mechanical assembly design. Their mechanism improves the management and transfer of information between different product design phases using a top-down method. The top-down process is refined to fit the developed multi-level model [24] [25] [26] [27]. The mechanism is based on inheritance; it maintains the link between an original object and a target object to ensure the accurate propagation of changes. Tremblay et al. [2] proposed a conceptual model to manage associations and propagate changes in a business-focused application. The model decomposition is based on three abstraction levels: view (design ...), document (part ...) and feature (geometric ...), exploiting the concepts of aggregation and decomposition of each of the various layers. Some years earlier, Giguère et al. [28] showed how the manual propagation of changes in an assembly during the modification of a part is a tedious process that easily leads to data inconsistencies. To address these issues, Giguère et al. proposed a solution based on contextual features, enabling the assisting and automating of modeling tasks while reusing the knowledge acquired. The propagation operates from a reference characteristic to a target characteristic by way of a derivation link.

With the objective of protecting data during a transfer between partners, Mun et al. [3] proposed a CAD information-sharing method based on skeleton models. This method guarantees the individual intellectual proprietary rights of each company contributing to the product design and that only the information required for the change propagation will be shared.

### ***3.4 Synthesis***

This literature review relates to three different aspects. The first aspect consists of managing the associations between heterogeneous objects (design, production...) [9][10][11] or in managing processes and tasks [12][13][14][15][16][17][4][5]. The second aspect consists of managing assemblies, based on graphs or systems that enable the geometric and technological modeling of various elements of an assembly. This allows more system stability during the assembly and disassembly phases [18][19][20] [21] [22]. The third aspect relates to concepts and methods that provide for better data sharing and change propagation between CAD objects [2][3][23][24][25][26][27][28]. In most of the references cited, dependencies are manipulated between higher-level objects (Part design, Product, Bill of Materials). In this work we focus on CAD data so as to maintain associations when a work package is extracted from the DMU, modified by a partner and then reinserted. In CAD, associations are most often geometric and parametric constraints established between geometric entities (faces, edges, vertices). We therefore manage and manipulate geometric entities of the object. The third aspect addressed in the literature appears to be closest to this work. Indeed, we can learn a great deal from the methods used in this third aspect, and apply them to solve some issues of the problem as



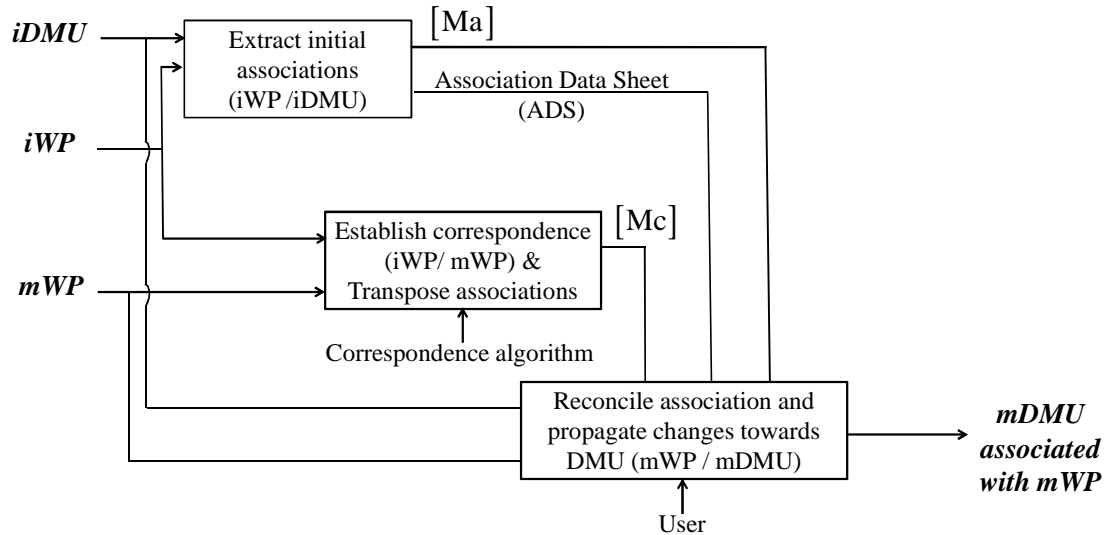
contemplated: transposition of the associations, change propagation, and the use of skeletons as data objects for the re-association of objects.

#### 4. Association management model (DMU - AMM)

The association management model (DMU - AMM) enables digital mock-up/work packages to be reconciled after their modification and assist the user in propagating the modifications made. The association management model operating mechanism works in three steps, as introduced earlier:

- Capture of the initial associations between the initial work package (iWP) extracted from the DMU and then sent to a partner (or partners), and the initial state of the OEM digital mock-up (iDMU);
- Identification of the correspondence between elements of the initial work package (iWP) and those of the modified work package (mWP), as per three cases discussed above; this mapping enables the transposing of the associations captured between the iDMU and the iWP to the mWP, thereby expressing the reconciled associations; and
- Reconciliation of the associations between the modified work package (mWP) and the initial digital mock-up is used to propagate changes to the DMU to obtain a modified DMU (mDMU) and maintain consistent information.

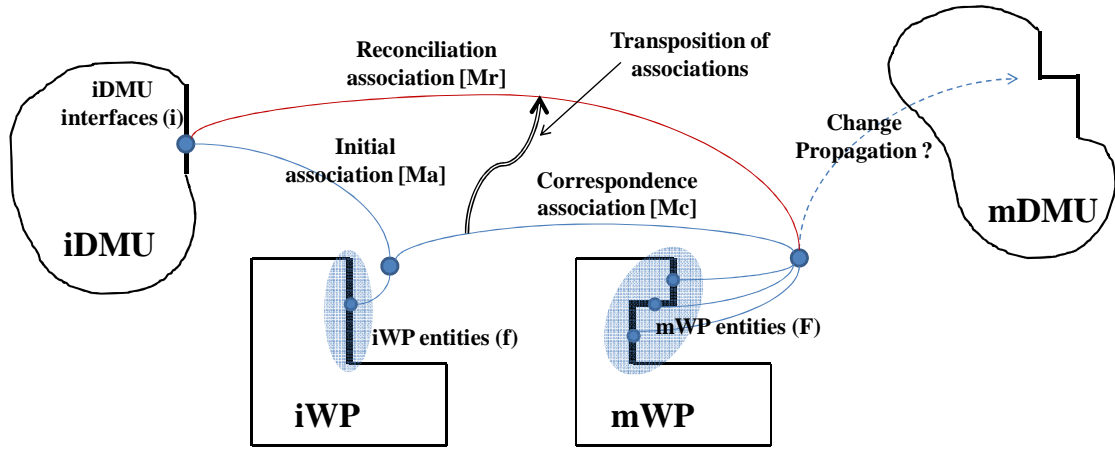
The inputs to this model are the initial digital mock-up (iDMU), the initial work package (iWP) and the modified work package (mWP). The correspondence algorithm is used to establish the correspondence between entities of the initial and modified work packages and to transpose the associations from entities of the iDMU towards the modified work package. The model output is the modified mock-up associated to the modified work-package (Fig. 5).



**Fig. 5.** SADT (Structured Analysis and Design Technique) Diagram – Association management model (DMU - AMM).

#### 4.1 From Initial associations to Change propagation

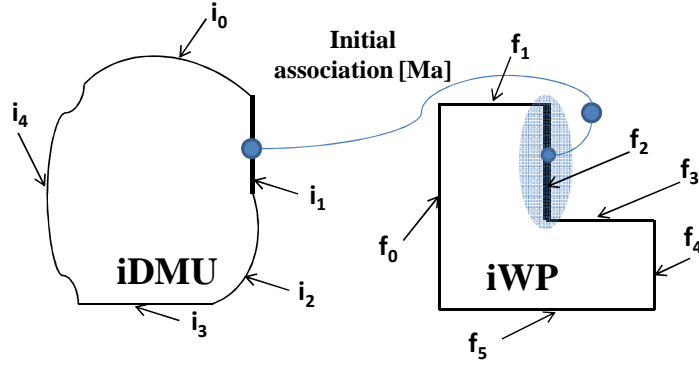
The proposed mathematical representation enables modeling of the relevant associations between iDMU, iWP and mWP, so as to propagate changes to the DMU (Fig. 6), with user involvement, in order to ensure geometric consistency. Prior to its extraction, the iWP is associated with the iDMU. The *Association matrix* ([Ma]) is used to represent these initial associations. Next, the evolution of the iWP towards the mWP is analysed, using the correspondence algorithm described below, and corresponding entities are associated through a *Correspondence matrix* ([Mc]). Combining these matrices enables transposing of the initial associations between the iDMU and the iWP into reconciled associations between the iDMU and the mWP. We will thus manipulate three matrices: the association matrix [Ma], the correspondence matrix [Mc], and the reconciliation matrix [Mr]. These steps are detailed next.



**Fig. 6.** Full associations' management cycle from the iDMU to the propagation of changes to the mDMU.

#### 4.2 Representing the initial associations between the iWP and the iDMU

In a CAD environment, the associations established between objects are in the form of geometric assembly constraints, constraints between geometric reference elements (such as planes or axes), associations by skeletons (diagrams used to generate parts in the context of an assembly) and feature-to-feature associations. In order to simplify our explanation, only assembly constraints between geometric entities are used as associations between the iDMU and the iWP, and only face-type geometric entities are used to describe the approach; also, in our simple example, the DMU contains two parts while the WP only contains one.



**Fig. 7.** Association between initial digital mock-up and the initial work package.

Associations between the iWP and the iDMU (Fig. 7) are modeled as follows:

$$(f) = [Ma](i) \quad (1)$$

where:

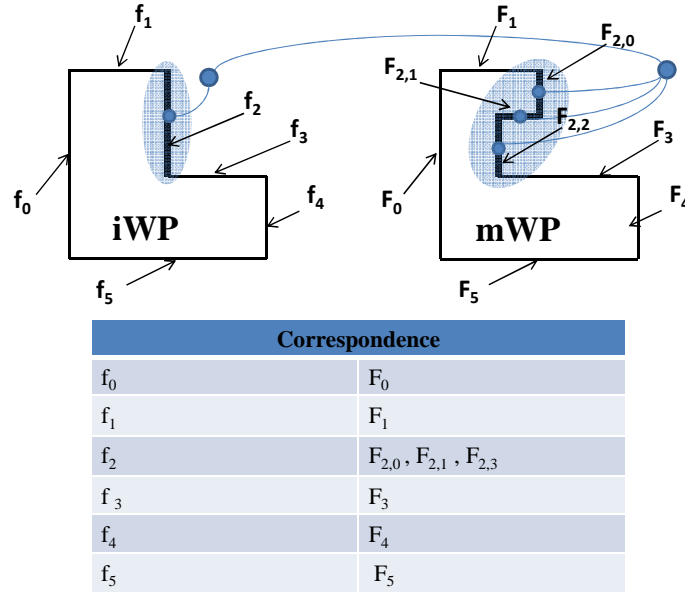
- $(f)$  is the vector of the geometric entities (faces, edges, vertices) of the iWP;
- $(i)$  is the vector of geometric entities (that define the interfaces) of the iDMU that are associated to the iWP. To make the vector  $(i)$  as small as possible, only geometric object entities of the iDMU that are linked to the iWP can be considered; and
- $[Ma] = [ma_{ij}]$  is the initial association matrix that defines associations between the iDMU interfaces and the entities of the iWP, such that  $ma_{ij} = 1$  if entity  $f_j$  of the iWP is associated with entity  $i_i$  of the iDMU, and equal to zero otherwise, as illustrated below.

$$\begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_0 \\ i_1 \\ i_2 \\ i_3 \\ i_4 \end{pmatrix} \quad (2)$$

In this case, (Eq. 2):  $f_1 = i_1$  if the entity (face)  $f_1$  of the iWP is associated with entity  $i_1$  of the iDMU.

### 4.3 Representing correspondence associations between the iWP and the mWP

The correspondence associations (Fig. 8) between the iWP and the mWP associate, generally speaking,  $m$  entities from the iWP to  $n$  entities of the mWP. For example, the modified faces  $F_{2,0}$ ,  $F_{2,1}$ , and  $F_{2,2}$  can replace the initial face  $f_2$ . In Fig. 11, these correspondences are shown in a table. Establishing these correspondences is no trivial task, and the proposed Correspondence algorithm is presented below.



**Fig. 8.** Correspondences between entities of the iWP and the mWP.

The correspondence associations between both versions of the work package are represented as a matrix:

$$(F) = [Mc](f) \quad (3)$$

where:

- $(f)$  is the vector of geometric entities (faces, edges and vertices) of the iWP, and
- $(F)$  is the vector of geometric entities (faces, edges and vertices) of mWP.

To manipulate smaller vectors, we need only include the entities that play a role (functional relationship, geometric constraint, etc.) to associate the iWP with the iDMU.

- $[Mc]$  is the correspondence matrix between the iWP and the mWP. We will have  $mc_{ij} = 1$  if entity  $f_i$  of iWP corresponds to the entity  $F_j$  of mWP; that is to say, entity  $F_j$  can assume the role of  $f_i$  in the association with the DMU (functional links, geometric constraints,

etc.). In general terms, the cardinality of these correspondences between entities is of type  $m:n$ , such that  $m=n$ ,  $m>n$  or  $m<n$ . For the example shown in Fig. 8, the correspondence matrix is shown in Eq. 4.

$$\begin{pmatrix} F_0 \\ F_1 \\ F_{2,0} \\ F_{2,1} \\ F_{2,2} \\ F_3 \\ F_4 \\ F_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} \quad (4)$$

In this case, the correspondence algorithm yields the following (Eq. 4):

- $F_0$  in the mWP corresponds to (replace the)  $f_0$  in the iWP through the persistent id;
- $F_1$  in the mWP corresponds to  $f_1$  in the iWP through the persistent id;
- $F_{2,0}$ ,  $F_{2,1}$  and  $F_{2,2}$  in the mWP correspond to  $f_2$  in the iWP through their common neighborhood;
- $F_3$  in the mWP corresponds to  $f_3$  in the iWP through the persistent id;
- $F_4$  in the mWP corresponds to  $f_4$  in the iWP through the persistent id; and
- $F_5$  in the mWP corresponds to  $f_5$  in the iWP through the persistent id.

#### 4.4 Representing reconciled associations between the iDMU and the mWP

The reconciled associations express how the DMU can ‘connect’ to the modified work package; and are obtained as follows:

$$\text{Eq. 1 and Eq. 2: } \begin{cases} (F) = [Mc](f) \\ (f) = [Ma](i) \end{cases} \left\{ (F) = [Mr](i) \quad \text{where} \quad [Mr] = [Mc] \times [Ma] \right. \quad (5)$$

$[Mr]$  is the reconciliation matrix of associations between the mWP and the iDMU. We will have  $mr_{ij} = 1$  if entity  $F_i$  of the mWP can be associated with interface entity  $i_j$ .

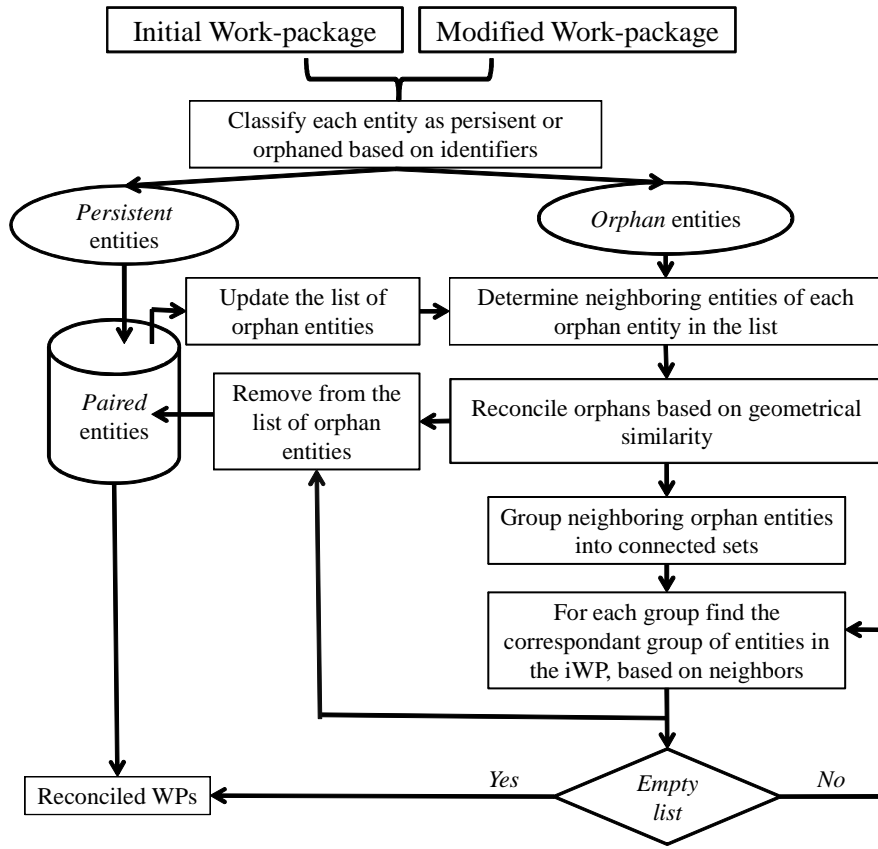
Thus, to establish our association management model, three matrixes are used: the initial association matrix, the correspondence matrix and the reconciliation matrix. The association matrix is rather easily extracted from the starting point (iWP/iDMU). The correspondence matrix, however, must be obtained using the correspondence algorithm, presented next.

#### 4.5 Correspondence algorithm

This section presents the algorithm enabling automatic calculation of the correspondence matrix between iWP entities and entities of the mWP. For this, we must identify, for each entity of the iWP that carries an association with the iDMU, its corresponding entity in the mWP, which is not a trivial task.

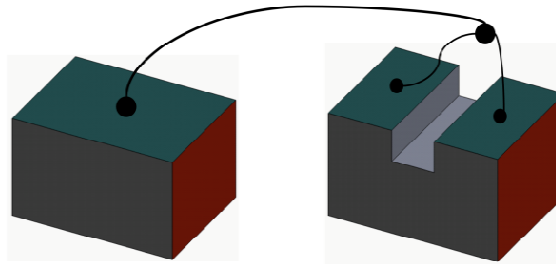
At the beginning, each entity is considered to be an ‘orphan’ since it does not have a corresponding counterpart in the object it is compared to. As soon as a corresponding counterpart is detected, the entity is ‘paired’. As a first step, pairing of entities is performed using their identifiers. If an entity from the two compared objects bear the same identifier, the entity is persistent, be it identical in both objects or not; this entity is persistent from the iWP to the mWP. For example, in Fig. 8,  $f_0$  (from iWP) is paired to  $F_0$  based on identifiers. On the other hand, some entities of mWP have an identifier that is not found in the iWP. For example,  $F_{2,1}$  (from mWP) remains an orphan based on the identifiers test. Establishing the correspondence matrix thus requires reconciling orphan entities of the mWP to entities of the iWP, which is carried out by analyzing both the neighboring entities of the orphan and its geometric characteristics. Here, only face-type entities are used to develop the correspondence algorithm. The steps in the matching correspondence algorithm are as follows (Fig. 9).

- All entities are examined and those that have a persistent identifier are classified as persistent entities and are therefore considered as paired entities.
- Entities that are not persistent are orphans; for each of them, the list of neighboring entities must be determined.



**Fig. 9.** Correspondence algorithm.

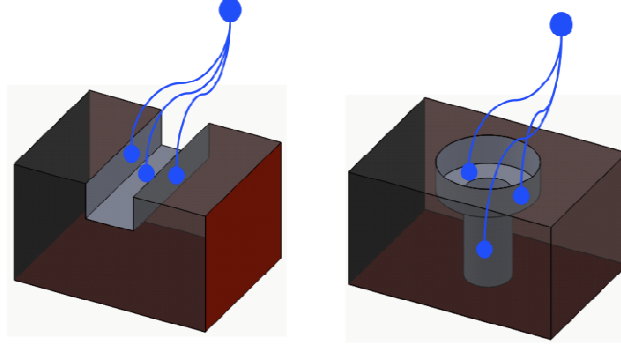
- Reconcile orphan entities using the similarity of their geometric definitions (types and equations of underlying surfaces) and common neighbors, such as the example shown in Fig. 10, where the top face has been split into two faces. Once the entities are reconciled, the lists of neighbors are updated for the next iteration.



**Fig. 10.** Reconciliation of the top face from the iWP with both top faces of the mWP.

- When the previous step has converged, neighboring orphan faces are grouped into connected sets. Fig. 11 shows two examples of such grouping. This grouping greatly

facilitates the coming step where each group of entities will be connected to a group formed by one or more faces from the iWP.

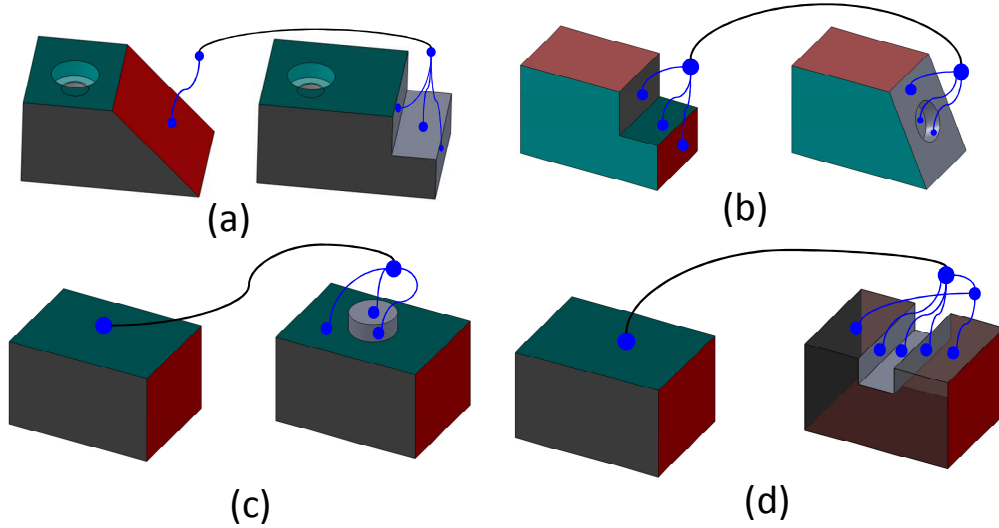


**Fig. 11.** Grouping of neighboring orphan entities (connected faces).

- Each group of connected orphan entities from the mWP has a list of neighbors that is used to identify entities from the iWP that share the same neighbors. The following cases (Fig. 12) need to be addressed:
  - (a) A group of  $m$  entities from iWP is destroyed and replaced by a group of  $n$  entities in the mWP, where  $m=1$  and  $n \geq 1$ . In this case, the neighboring list of the group of connected entities from mWP is the same as the neighboring list of one entity from the iWP. The solution is obtained by direct comparison between the lists of neighbors (Fig. 12 (a)). If we have more than one solution (when several entities share the same neighboring list in the iWP), geometric characteristics are used to reach the correct solution.
  - (b) A group of  $m$  entities from the iWP is destroyed and replaced by a group of  $n$  entities in the mWP, where  $m > 1$  and  $n \geq 1$ . In this case, no single entity from the iWP has a list of neighbors that is a perfect match with the list of neighbors of the group from the mWP. We therefore have to combine multiple entities from the iWP until their aggregated list of neighbors fits perfectly with the list from the mWP (Fig. 12 (b)).
  - (c) A group of orphan entities from the mWP have a single entity in the list of neighbors, and this entity has a corresponding entity  $E_e$  in the iWP. In such a case, the group of entities from the mWP is redefined so as to include the neighboring entity within the group; the correspondence is then established between this augmented group of entities from mWP and the entity  $E_e$  from the iWP (Fig. 12 (c)).
  - (d) A group of orphan entities from the mWP have several neighbors (as distinct from case (c)). In this case, there is no match between the list of neighbors of the group from the mWP and the list of neighbors of a single entity from the iWP (as this is not case (a)). Even if we combine lists of neighbors from multiple entities of the iWP, no solution is found (which means that this is not case (b)). The group of entities from the mWP is therefore redefined to include one of their neighboring entities within the group. At this stage, multiple groupings are considered. The correspondence is then searched for this augmented group of entities from the mWP. Each of these groupings has a corresponding entity in the iWP with the exact same list of



neighbors. The geometric characteristics are used to identify the best solution (Fig. 12 (d)).

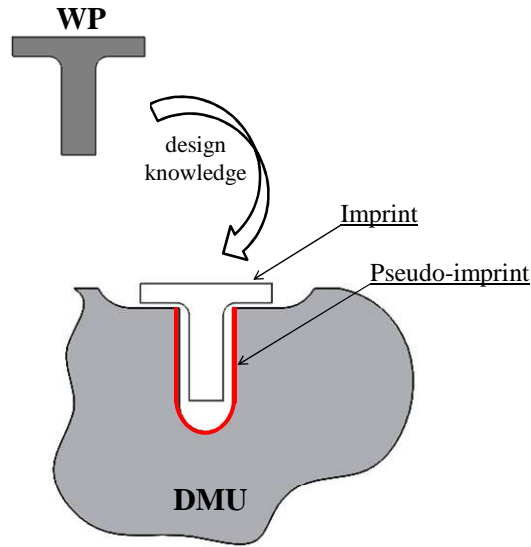


**Fig. 12.** Scenarios addressed by the correspondence algorithm.

The correspondence algorithm as described above leads to defining the correspondence matrix calculation  $[Mc]$ . As stated in section 4.4, combining  $[Ma]$  and  $[Mc]$  defines  $[Mr]$  which enables, in turn, transposing the initial associations between the iDMU and iWP into reconciled associations between the iDMU and the mWP. The final step requires propagating the changes to obtain the mDMU.

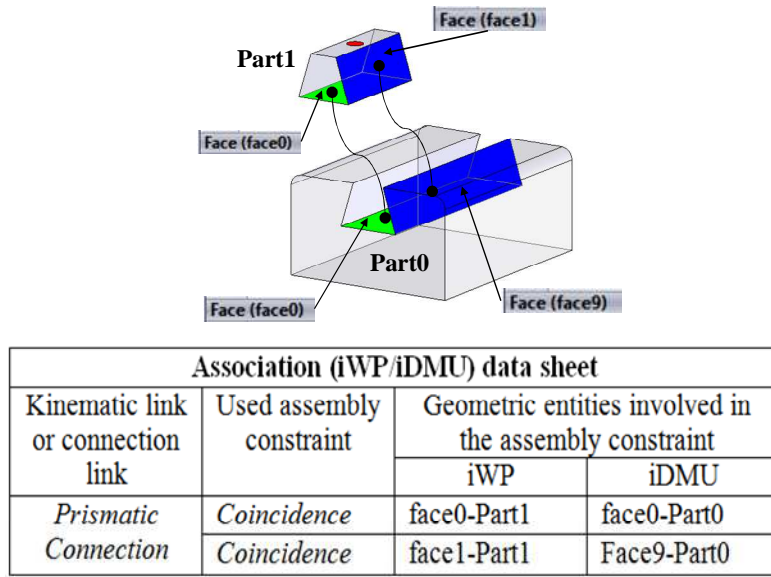
#### 4.6 Change propagation

Propagating changes to the mDMU is the final step of the proposed association management model. Several solutions can be considered to help users make the changes required to obtain coherent associations and geometry between the mWP and the mDMU. User assistance may be provided in the form of skeletons (sketches) [3] [28]. Another way is to provide the user (product architect, or integrator) with a pseudo-imprint that is information-specific to a target feature [28]. This imprint is automatically obtained by the application of design knowledge to an imprint (Fig. 13). The imprint is the information specific to a reference feature determined by extraction from a group of geometric entities in the design context [28]. A pseudo-imprint can thus be used to propagate changes from the WP (imprint) to the DMU (pseudo-imprint), assuming that the design knowledge was previously captured.



**Fig. 13.** Imprint/Pseudo-imprint [28].

In our case, this design knowledge is not captured; we can nevertheless identify the imprint in the mWP and locate areas of the DMU that need to be modified. Another way to assist the user is to provide solutions in the form of connectivity maps [12]. With regard to this work, we can draw on the latter solution and provide the user with an association reconciliation matrix based on two other matrices: the correspondence matrix [Mc] and the initial association matrix [Ma]. An Association (iWP/iDMU) Data Sheet (Fig. 14) is also provided to help the user manage associations and propagate changes. The principle of the Association Data Sheet (ADS) is inspired from Demoly et al.'s work [21] (Skeleton geometry-based assembly context definition) that integrates assembly process engineering information and knowledge in the early phases of the product development process. In our case, the ADS is a skeleton assembly entity that contains information about the association between the iWP and the iDMU and is used as design knowledge to assist the user.



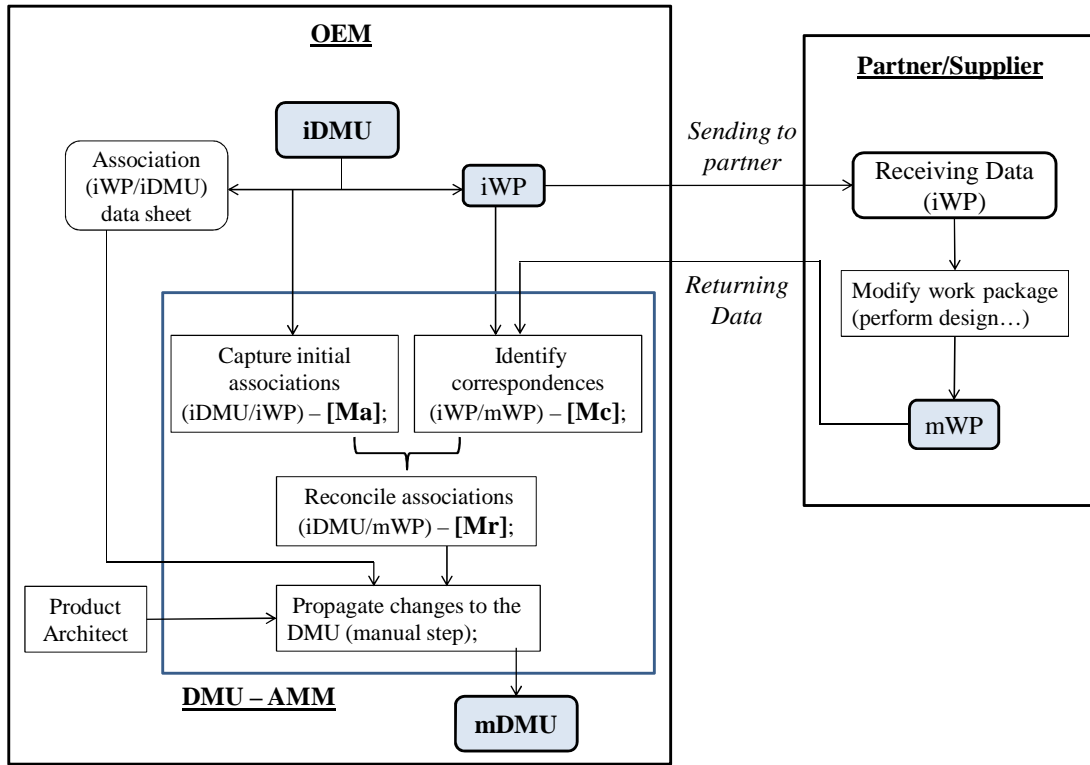
**Fig. 14.** Association (iWP/iDMU) data sheet.

The reconciliation matrix and the Association Data Sheet help determine, for each mWP entity that belongs to vector (F), if it is associated to entities of the iDMU and if yes, which one. The user's attention is thus brought, through context messaging, to the entities of the DMU where changes are needed in order to maintain consistency with the mWP. The user utilizes the ADS to verify if the connection links are consistent between the mWP and the mDMU after change propagation.

## 5. Illustration and validation

The collaborative design process requires exchanging data between original equipment manufacturers (OEMs) and partners who contribute to the evolution of product definition. Data must be reintegrated into the OEM DMU after modification by the partner. The steps of the collaborative process using the proposed Association Management Model are summarized as follows (Fig. 15):

- Extract the iWP and its association data sheet (ADS) from the iDMU;
- Send the iWP to the partner;
- The partner adds value to the WP and returns the mWP to the OEM DMU;
- The DMU-AMM is used to capture initial associations (iDMU/iWP), identify the correspondence (iWP/mWP) and reconcile associations (iDMU/mWP); and
- Propagate changes to the DMU (by the user) by using the reconciliation matrix and the association (iWP/iDMU) data sheet (ADS).



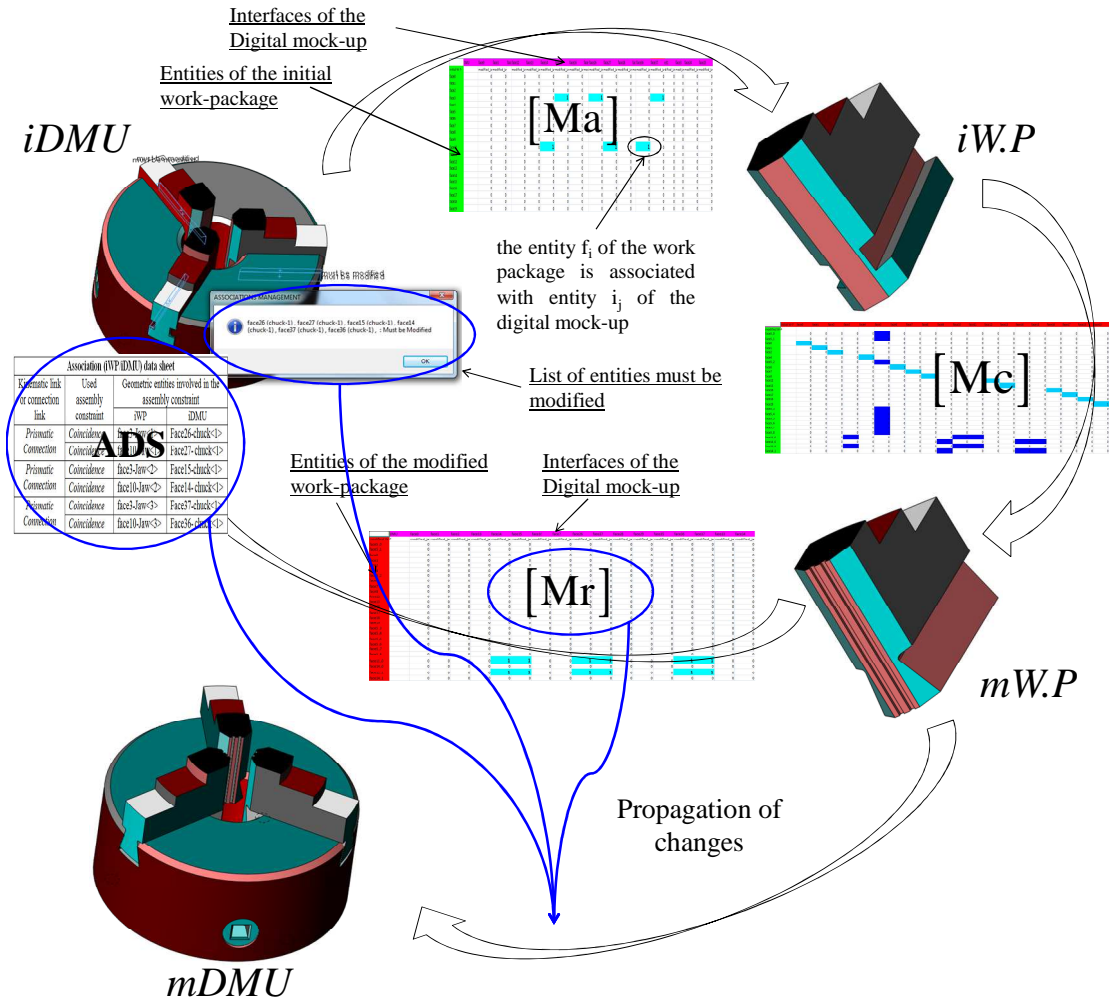
**Fig. 15.** Collaborative Process using the DMU-Association Management Model.

To illustrate and validate the DMU-AMM, a DMU made of a three-jaw chuck is considered (Fig. 16). The chuck itself belongs to a work package. The modification scenario covers two needs:

- A change in the technological solution: T shaped to dovetail slide linkage;
- Add texture to the front face so as to improve gripping.

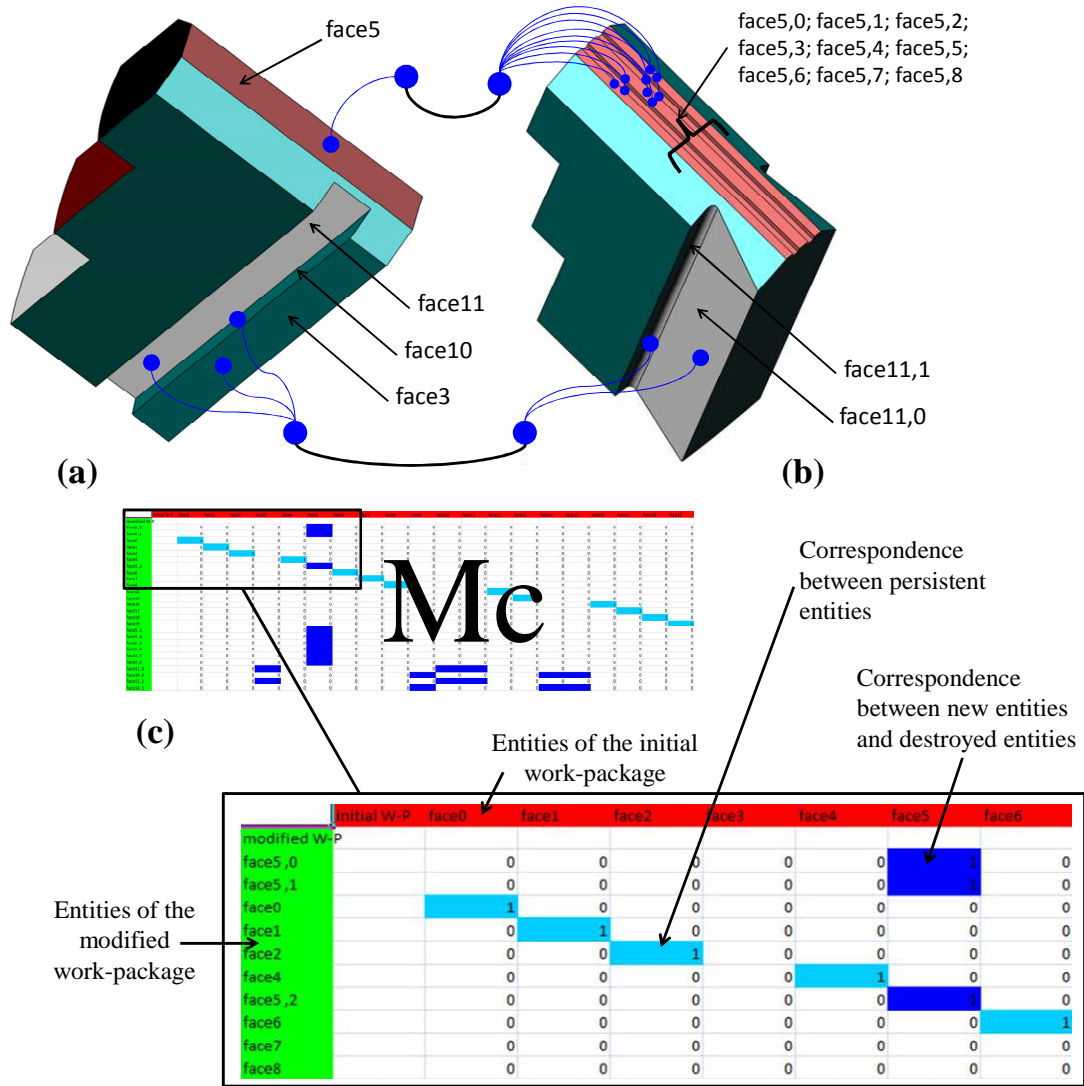
In the first step, the initial association matrix  $[Ma]$ , between the iWP and the iDMU is extracted (as an Excel spreadsheet), as well as the ADS. The associations are assembly constraints established within the DMU prior to extraction of the WP. This step consists of scanning the various assembly constraints to identify the entities in the iDMU and in the iWP that are associated to one another.

In the second step, correspondences between entities of the iWP and of the mWP are established. The entities of the mWP are reconciled through the correspondence algorithm. This enables the correspondence matrix  $[Mc]$  to be generated (as an Excel spreadsheet). This step is essential for the transposition of initial associations towards the mWP.



**Fig. 16.** Complete association management cycle with transposition and reconciliation.

The reconciliation result, the correspondence between two versions of WP entities (initial and modified), as well as the correspondence matrix [Mc] are shown with more details in Fig. 17. Light blue cells show persistent entities (based on identifiers). Dark blue cells show correspondences between new entities and destroyed entities. For example, the front face from the iWP (face5) has been replaced by a series of faces (face5,0; face5,1; face5,2; face5,3; face5,4; face5,5; face5,6; face5,7; face5,8) to enhance gripping. In addition, the group of faces face3; face10; and face11 has been replaced by the group of faces face11,0; and face11,1 in the mWP.



**Fig. 17.** Entities' reconciliation and reconciliation matrix construction.

The third step consists of reconciling the associations between the mWP and the mDMU. The reconciliation matrix  $[Mr]$  is calculated by multiplying the correspondence matrix  $[Mc]$  by the initial association matrix  $[Ma]$ . The reconciliation matrix is provided to the user in an Excel spreadsheet so that the user can make the necessary changes to the DMU in order to obtain consistent associations (Fig. 16) and control the validity of the changes according to the association data sheet. An mDMU entity will probably be modified if it has an association with an entity of the mWP that has been modified. Therefore, we provide the user with decision-making support, enabling him/her to make the changes as required to obtain consistent associations and geometry. The user is alerted with a message indicating which entities are most likely to be modified. An annotation is placed on each of them (Fig. 16).

## 6. Conclusion and perspectives

In this paper, we presented an association management model that facilitates collaborative design by maintaining consistency between CAD elements. First, it captures the initial associations of the iWP/iDMU in the associations matrix [Ma]. Second, it controls the modifications from the iWP to the mWP using the correspondence algorithm to calculate the correspondence, indicated in Matrix [Mc]. This enables the initial associations to be transposed to the mWP. Finally, the reconciliation matrix [Mr], found by multiplying [Ma] and [Mc], is used to reconcile the associations and identify the changes to propagate to obtain an mDMU that is consistent with the mWP. The objective is not to eliminate user intervention but rather to assist the user by means of an efficient management model. This work provides decision making support for the user; allowing the user to associate the mWP with the mDMU and identify the interface entities of the mDMU that must be modified in order to consistently match the mWP. This contribution will facilitate the data exchange between partners contributing to the evolution of a product, while maintaining the consistency of information.

The DMU-AMM is a tool that facilitates the integration of a modified work-package to the DMU and allows the user to save time and reduce errors, especially if the WP contains a large number of parts. Indeed, it allows the automatic detection of areas of the DMU that may have to be modified and helps users to reconcile associations and verify their modification by using the ADS.

## Acknowledgments

The authors would like to thank the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

## Bibliography

- [1] <http://www.mcgill.ca/plm2-criaq/project/>.
- [2] Tremblay T.G., Rivest L., Msaaf O. and Maranzana R., The role of associations in CAD and PLM for handling change propagation during product development, 13th ISPE International Conference On Concurrent Engineering: Research and Applications, Antibes, French Riviera, (2006) 18 - 22 September 2006.
- [3] Mun D., Hwang J. and Han S., Protection of intellectual property based on a skeleton model in product design collaboration, Computer Aided Design 2009; 41, 641-648.
- [4] Hoffmann C.M. and Joan-Arinyo R., CAD and the product master model, Computer-Aided Design 1998; 30, 905-918.

- [5] Hoffmann C.M. and Joan-Arinyo R., Distributed maintenance of multiple product views, *Computer-Aided Design* 2000; 32, 421-431.
- [6] Gao W., Gao S.M., Liu Y.S., Bai J. and Hu B.K., Multiresolutional similarity assessment and retrieval of solid models based on DBMS, *Computer-Aided Design* 2006; 38, 985–1001.
- [7] Hong T., Lee K. and Kim S., Similarity comparison of mechanical parts to reuse existing designs, *Computer-Aided Design* 2006; 38, 973–984.
- [8] Brière-Côté A., Rivest L. and Maranzana R., A Three-Step Approach for Structuring 3D CAD Model Comparison Scenarios, 8th International Conference on Product Lifecycle Management, Technische Universiteit Eindhoven, The Netherlands (2011), 11th – 13th July 2011.
- [9] Fortin, C., Toche, B., McSorley, G., and Huet, G., Supporting Product Development Value Streams through Manufacturing Process Management. Paper presented at the Proceedings of the 1st International Conference on Modelling and Management of Engineering Processes (2010), MMEP'10, Cambridge, UK.
- [10] Toche, B., Huet, G., McSorley, G., & Fortin, C., A Product Lifecycle Management Framework to Support the Exchange of Prototyping and Testing Information. Paper presented at the Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (2010), IDETC/CIE 2010, Montreal, Canada.
- [11] Zimmermann J. U., Haasis S., and Van Houten F. J. A. M., ULEO - Universal linking of engineering objects. *CIRP Annals - Manufacturing Technology* (2002), 51(1), 99-102.
- [12] Yassine A., Whitney D., Daleiden S. and Lavine J., Connectivity maps: modeling and analysing relationships in product development processes. *Journal of Engineering Design* (2003), 14(3), 377-394.
- [13] Yassine A., Whitney D. E., Lavine J. and Zambito T., DO-IT-RIGHT-FIRST-TIME (DRFT) APPROACH TO DSM RESTRUCTURING. *ASME 2000 International Design Engineering Technical Conferences* (2000), September 10-13, 2000, Baltimore, Maryland.
- [14] Yassine A., Falkenburg D. R. and Chelst K., Engineering Design Management: An Information Structure Approach. *International Journal of Production Research* 1999, vol. 37, no. 13, pp. 2957-2975.
- [15] Pimmler T. U. and Eppinger S. D., "Integration Analysis of Product Decompositions", *Proceedings of the ASME Sixth International Conference on Design Theory and*



- Methodology, Minneapolis, MN, Sept., 1994. Also, M.I.T. Sloan School of Management (1994), Cambridge, MA, Working Paper no. 3690-94-MS, May 1994.
- [16] McCord K. R. and Eppinger S. D., "Managing the Integration Problem in Concurrent Engineering", M.I.T. Sloan School of Management, Cambridge, MA, Working Paper no.3594, 1993.
  - [17] Whitney D. E., Dong Q., Judson J. and Mascoli G., "Introducing Knowledge-Based Engineering Into an Interconnected Product Development Process", M.I.T. Center for Innovation in Product Development, Cambridge, MA, White Paper Jan. (1999) 27.
  - [18] Fouda P., Danloy J., L'Eglise T., Lit P. De, Rekiek B. and Delchambre A., "A Heuristic to Generate a Precedence Graph Between Components for a Product Family", International Symposium on Assembly and Task Planning Soft Research Park (2001), Fukuoka, Japan, May 28-29.
  - [19] Li G. D., Zhou L. S., An L. L., Ji J. F., Tan C. B. and Wang Z. G., "A system for supporting rapid assembly modeling of mechanical products via components with typical assembly features", *Int. J. Adv. Manuf. Technol.* 2010; 46:785–800.
  - [20] Mascle C., Jabbour T. and Maranzana R., "Assembly features for mechanical product data", *IEEE International Symposium on Assembly and Task Planning* (1997), Marina del Rey, CA - August.
  - [21] Demoly F., Toussaint L., Eynard B., Kiritsis D. and Gomes S., Geometric skeleton computation enabling concurrent product engineering and assembly sequence planning, *Computer-Aided Design* 2011; Vol. 43, no 12, pp. 1654-1673.
  - [22] Demoly F., Yan X.-T., Eynard B., Kiritsis D. and Gomes S., Integrated product relationships management: a model to enable concurrent product design and assembly sequence planning, *Journal of Engineering Design* 2012; Vol. 23, No. 7, pp. 544–561.
  - [23] Chen X., Gao S., Yang Y. and Zhang S., Multi-level assembly model for top-down design of mechanical products, *Computer Aided Design* 2012; vol.44 (10), 1033-1048.
  - [24] Fenves Fenves SJ., Foufou S., Bock C. and Sriram RD., CPM2: a Core Model for Product Data. *Journal of Computing and Information Science in Engineering* 2008; 8: 014501–6.
  - [25] Rachuri S., Han Y-H., Foufou S., Feng SC., Roy U., Wang F., A model for capturing product assembly information. *Journal of Computing and Information Science in Engineering* 2006; 6:11–21.

- [26] Chen X., Gao S., Yang Y. and Zhang S., The skeleton in the multi-level assembly model for top-down innovation design of mechanical product (2009). International conference on product lifecycle management.
- [27] Zhang S., Chen X., Gao S. and Yang Y., A framework for collaborative top-down assembly design. In: Proceedings of the ASME international design engineering technical conference and computers and information in engineering conference (2007); Vol 6, p. 139–49.
- [28] Giguere F., Rivest L. and Desrochers A., Improving design productivity and product data consistency, Feature Based Product Life-Cycle Modelling (2002), René Soenen and Gustav J. Olling, ed., Hardbound, Kluwer Academic, pp.77-91.



Borhen Louhichi is a Postdoctoral researcher at Ecole de Technologie Supérieure in Montreal, Canada and Associate Professor at the Institute of Applied Sciences and Technology of Sousse, Tunisia. He has obtained a Ph.D. in Mechanical Engineering from National Engineering School of Monastir, Tunisia. He was a researcher at Université du Québec à Trois-Rivières, Canada and the University of Liege, Belgium. His research interests include CAD/CAM/CAE integration, PLM, Inspection and Tolerancing.



Louis Rivest became professor at Ecole de Technologie Supérieure in Montreal, Canada, after spending a few years in the aerospace industry. He has obtained a Ph.D. from Ecole Polytechnique de Montreal in 1993, and a bachelor in Mechanical Engineering in 1988. His research centers on the models, methods, tools and processes supporting complex product development. Currently, his work is applied primarily in the aerospace field. His teaching and research activities thus relate to CAD, Product Data Management and Product Lifecycle Management.