Management of Associations within Digital Mock-ups for Improved Collaboration

Borhen LOUHICHI, PhD; Pr. Louis RIVEST, PhD

Department of Automated Production Engineering

Ecole de technologie superieure, Montreal, Canada

Abstract. In the current collaborative and extended-enterprise environment, product definition must be exchanged between original equipment manufacturers (OEM) and other partners who contribute to the evolution of product definition. In many cases, an information package, which is associated with other objects in the DMU, must thus be extracted from the OEM's Digital Mock -up (DMU) and be sent to partners and suppliers to allow them to perform the required work on their work packages (WP). The objective of our research is to maintain the associations between objects when a work package is removed from the DMU of the originator, modified by the subcontractor, and then reinserted into the DMU. We propose an association management model to transpose the initial work package associations towards the modified work package, to reconcile these associations with the DMU and then propagate changes towards the latter to maintain consistency.

Keywords: Digital Mock-up, work package, Associations, transposition of associations, CAD data, Change propagation, reconciliation matrix.

1 Introduction

The collaborative development for Product Lifecycle Management (PLM) is an industry-lead research program focused on developing a collaborative environment for improved information management in the product development process while maintaining confidential data security [1]. It is composed of five tasks: information sharing management, collaborative platform security, PLM process modeling, product management interface development and PLM simulation.

The work presented in this paper concerns the first of these tasks, that of information management sharing, especially CAD (Computer Aided Design) data sharing among partners. A product's evolution requires the contribution of several partners. Therefore, data exchange between a project's partners is pervasive and critical. This data is often associated with other data in a global digital mockup controlled by the project leader (the OEM's DMU (Original Equipment Manufacturers of the Digital Mock-up)), which cannot be shared for various reasons, such as confidentiality. Once the value is added and brought to the work-package, it must be reintegrated into the OEM DMU. The new associations between work package and digital mock-up lose

adfa, p. 1, 2011.

their coherence. There is as yet no solution to systematically manage the evolution of associations. Our general objective is to develop a conceptual model that will allow efficient management of the associations that relate a WP (work package) to an OEM's DMU in order to facilitate product information sharing with partners and suppliers. A major challenge to achieving this objective is to define a model of a WP and its context such that the various associations can be managed systematically. The association management model proposed here captures associations between the initial work package (iWP), extracted from the DMU, and the initial state of the OEM digital mock-up (iDMU), and then control the modifications performed on the initial work package (iWP) to obtain the modified work package (mWP). This management model will then help reconcile the associations between the modified work package (mWP) and the digital mock-up while propagating changes to the iDMU to obtain a modified DMU (mDMU) and maintain consistent information and associations.

This paper is organized as follows. First, the problem at hand is structured so as to establish our starting point and assumptions. A literature review covering data management, information sharing and the propagation of change is presented next. Next, the possible scenarios changes of the work-package are presented. The proposed approach and algorithms of our associations' management tool, including the user interface, are detailed next, followed by illustrating example. The paper ends with our conclusions and perspectives on the future of this work.

2 The state of the art

Several studies have focused on the management of associations between objects in a digital mock-up. Fortin et al. [2] developed dynamic links that provide a continuous dynamic exchange of data among different trades, which provides feedback from manufacturing to design. The work of Toche et al. [3] is focused on the implementation of a communication model that allows 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. [4] developed a system containing generic inter-view links called "Universal Linking of Engineering Objects" (ULEO) that form relationships between technical objects from different professions (design, manufacturing, quality). The ULEO system allows for the construction of a unique multi-view model. Yassine et al. [5] developed a system that provides improved management between heterogeneous elements that make up the product development process. In other work, Yassine et al. [6] developed the DSM (Design Structure Matrix). 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. In the context of data sharing and the propagation of changes between CAD objects, Chen et al. [7] developed a mechanism designed for better management and transfer of information between different product design phases using a top-down method. Tremblay et al. [8] [9] proposed a conceptual model to manage associations and propagate changes in a business-focused application. Giguère et al. [10] [11] presented manual propagation of changes, in an assembly during the modification of a part, as a tedious process that leads easily to data inconsistencies. With the objective to protect the data during a transfer between partners, Mun et al. [12] 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 to share only information necessary to change propagation.

This literature review relates to two different aspects. The first aspect consists of managing associations between heterogeneous objects (design, production...) [2][3][4] or managing processes and tasks [5][6]. The second aspect is finding better concepts and methods enabling better data sharing and change propagation between CAD objects [7] [8] [9] [10] [11] [12]. In most pieces of literature, 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 second aspect addressed in the literature seems closest to this work. Indeed, we can learn from the methods used in this second aspect to solve some issues of the problem as contemplated: transposition of the associations and change propagation.

3 Work Package Evolution Scenarios

During its evolution, an object can assume different versions. The original object (O-Object) that is sent from an OEM to a partner as part of an iWP is likely to differ from the next version of the object being returned (R-Object) to the OEM by the partner. Each version is either an enhancement of the previous one or it is replaced by a completely new version. The basic objective here is to establish correspondence (associated) entities between the O- and R-Objects. The first, rather trivial step is to associate entities that ‘survived’ the evolution. Such entities can be either geometrically identical (same underlying equations, domain, position) or share the same ID.

Several technological scenarios can arise during an object’s evolution. These technological scenarios are translated into three geometric scenarios that manage the associations between the entities, hereafter limited to faces for ease of discussion, constituting the objects.

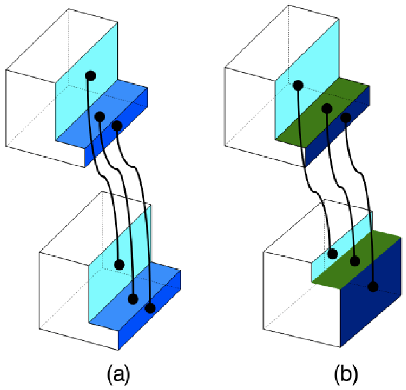


Fig. 1. (a) Absolutely identical objects and (b) Structurally identical objects; valid associations.

- Valid associations: Here the matching between the entities of the R-Object and those of the original O-Object is obvious if all entities' IDs are preserved. The R and O objects are either *absolutely identical* (Figure 1 (a)), that is to say, the object is not modified, or structurally identical (Figure 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.
- Partially undetermined associations: With this case, some entities are orphans upon first analysis. This arises 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/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 (Figure 2 (a)), m = n (Figure 2 (b)), and m> n (Figure 2 (c)). In this figure, the faces shown in white are paired (matched) based on geometric equivalence or ID persistence, which reduces the search space for matching unpaired entities, that is, those for which associations are undetermined. Algorithms to transform these 'grouped entities associations' into 'individual entities associations' are proposed in the next section.

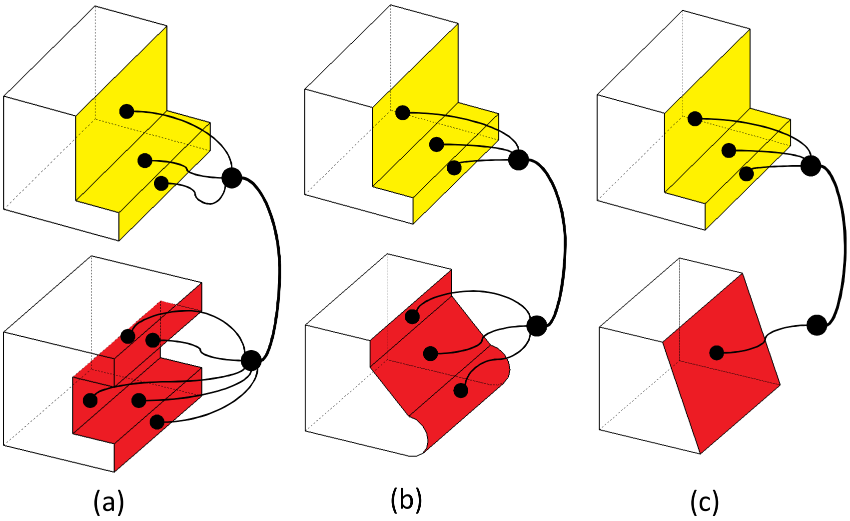


Fig. 2. Partially undetermined associations a) m <n; b) m = n; c) m> n.

- Totally Undetermined associations: In this case, all entities' IDs are destroyed and their cardinality may vary as well. This can occur when there is a passage through a neutral format or when an entire object is replaced by a new one. All constituting entities are therefore considered new. Correspondence is established, in a first step at least, between the complete groups of entities of the two objects: original and returned (Figure 3). Of course, some analysis can be conducted in order to pair entities based on their geometric definition and thus transform this extreme case into a 'Partially undetermined associations' case.

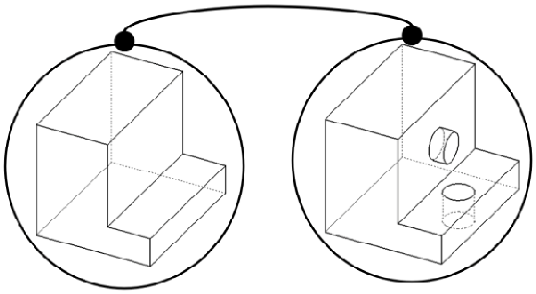


Fig. 3. Undetermined associations

4 Association management model

The association management model enables digital mock-up/work packages to be reconciled after their modification and assists the user in propagating the modifications (Figure 4).

The association management model operating mechanism works in three steps, as introduced earlier:

- Capture initial associations between the initial work package (iWP), extracted from the DMU and then sent to the partner, and the initial state of the OEM digital mock-up (iDMU);
- Identify the correspondence between elements of the initial work package (iWP) and those of the modified work package (mWP), as per scenarios discussed above; this mapping then enables the transposing of associations captured between the iDMU and the iWP to the mWP so as to express reconciled associations; and
- Reconciled associations between the modified work package (mWP) and the initial digital mock-up are used to propagate changes to the DMU in order to obtain a modified DMU (mDMU) and maintain consistent information.

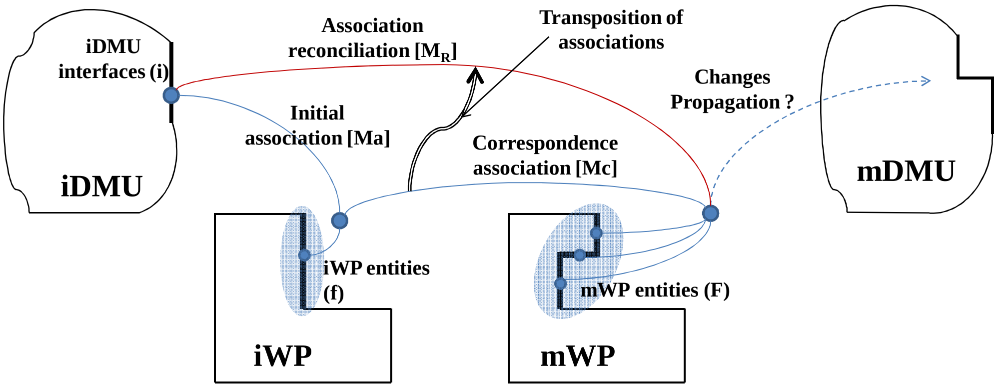


Fig. 4. Complete associations' management cycle from iDMU to propagation of changes to the mDMU

4.1 Mathematical Representation

The proposed mathematical representation enables modeling of the relevant associations between the iDMU, the iWP and the mWP, so as to propagate changes to the DMU (Figure 4), with user involvement, in order to ensure geometric consistency. At the beginning, the iWP is associated with the iDMU. A matrix (the Association Matrix, [Ma]) is used to represent these initial associations. Next, the evolution of the iWP towards the mWP is analysed, using algorithms described below, and the corresponding entities are associated through a correspondence matrix ([Mc]). Combining these matrixes makes it possible to transpose the initial associations between an iDMU and an iWP into reconciled associations between that iDMU and that 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.

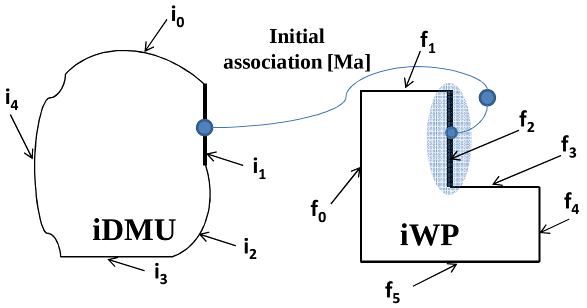


Fig. 5. Association between an initial digital mock-up and an initial work package

Associations between the iWP and the iDMU (Figure 5) are modeled as follows:

$$(f) = [M_a](i) \tag{1}$$

With:

- (f) as the vector of the geometric entities (faces, edges, vertices) of the iWP, which can play a role (a functional relationship, a geometric constraint...) to link the work package to the iDMU.

- (i) as the vector of geometric entities that define the interfaces of the iDMU that are linked 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 are considered.

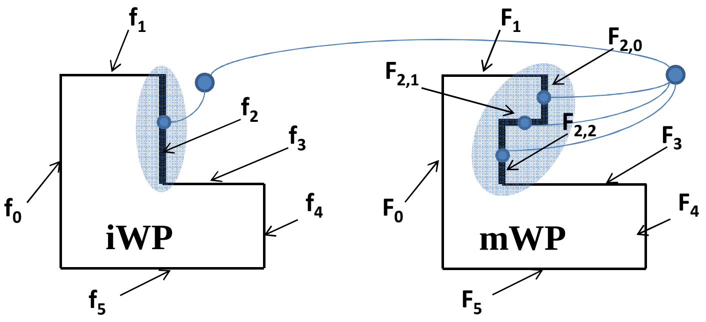
- $[M_a] = [ma_{ij}]$ is the initial association matrix that defines associations between the iDMU interfaces and the entities of the iWP, such that the ij^{th} element of the matrix $[M_a]$, $ma_{ij} = 1$ if the entity f_j of iWP is associated with entity i_i of iDMU, and equal to zero otherwise, as illustrated below.

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

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

4.2 Representing correspondence associations between iWP and mWP entities

The correspondence associations (Figure 6) between the iWP and the mWP associate, generally speaking, m entities from the iWP to n entities of the mWP, where one group of entities can play the role of the other in its same context (functional links, geometric constraints, etc.). For example, the face F_0 (under the part in Figure 6) from the mWP can play the role of the initial entity f0 from iWP. Similarly, the modified faces $F_{2,0}$, $F_{2,1}$, $F_{2,2}$ can replace the initial face f_2. In Figure 6, these correspondences are shown in a table. Establishing this correspondence is no trivial task, and the proposed correspondence algorithm [13] was developed. This algorithm enables automatic calculation of the correspondence matrix between iWP entities and entities of the mWP. For process to occur, we must identify, for each entity of the iWP that carries an association with the iDMU, its 'equivalent' in the mWP, which is not a trivial task. Establishing the correspondence matrix thus requires reconciling orphan entities (an entity of mWP is considered to be an "orphan" if its identifier is not found in the iWP) of the mWP to iWP entities, by analyzing both the neighboring entities of the orphan and its geometric characteristics. For example, in Figure 6 the group of orphan entities in the mWP, $F_{2,0}$, $F_{2,1}$ and $F_{2,2}$,have F_1 (paired with f_1) and F_3 (paired with f_3) neighbours, and the entity f_2 in the iWP has f_1 and f_3 as neighbours. So, the group of orphan entities ($F_{2,0}$, $F_{2,1}$ and $F_{2,2}$) in the mWP is paired with the entity f_2 in the iWP. If we have more than one solution (when several entities share the same neighbouring list in an iWP), geometric characteristics are used to reach the correct solution.



Correspondence	
f_0	F_0
f_1	F_1
f_2	$F_{2,0}$, $F_{2,1}$, $F_{2,3}$
f_3	F_3
f_4	F_4
f_5	F_5

Fig. 6. Correspondences between iWP and mWP entities.

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

$$(F)=[M_c](f) \tag{3}$$

where:

- (f) is the vector of the geometric entities (faces, edges and vertices) of the iWP; and

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

To manipulate smaller vectors, we include only the entities that can play a role (functional relationship, geometric constraint, etc.) in linking the iWP and the iDMU.

- $[M_c]$ is the correspondence matrix between the iWP and the mWP. We will have the ijth element of the matrix $[M_c]$ $mc_{ij} = 1$ if entity f_i of the iWP corresponds to entity F_j of the mWP, i.e., if 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 part shown in Figure 6, the correspondence matrix is shown in Equation 4.

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

In this case (Eq. 4):

- F_0 in the mWP corresponds to (replace the) f_0 in the iWP;
- F_1 in the mWP corresponds to f_1 in the iWP;
- $F_{2,0}$, $F_{2,1}$ and $F_{2,2}$ in the mWP corresponds to f_2 in the iWP;
- F_3 in the mWP corresponds to f_3 in the iWP;
- F_4 in the mWP corresponds to f_4 in the iWP;
- F_5 in the mWP corresponds to f_5 in the iWP;

To work in the reverse, if we consider the case where iWP in the previous case is the mWP in this case and the mWP in the previous case is the iWP in this case, the equation 2 becomes:

$$\begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} F_0 \\ F_1 \\ F_{2,0} \\ F_{2,1} \\ F_{2,2} \\ F_3 \\ F_4 \\ F_5 \end{pmatrix} \tag{5}$$

Eq. 5 ➔ $$\begin{cases} f_0 = F_0 \\ f_1 = F_1 \\ f_2 = F_{2,0} + F_{2,1} + F_{2,2} \\ f_3 = F_3 \\ f_4 = F_4 \\ f_5 = F_5 \end{cases}$$

- $f_0 = F_0$ means f_0 in the mWP corresponds to (replace the) F_0 in the iWP;
- $f_2 = F_{2,0} + F_{2,1} + F_{2,2}$ means f_2 in the mWP corresponds to $F_{2,0}$, $F_{2,1}$ and $F_{2,2}$ in the iWP;

4.3 Representing reconciled associations between iDMU and mWP

The reconciled associations express how the DMU can 'connect' to the modified work package; they are obtained as follows:

Equation 1 and Equation 3 ➔

$$\left.\begin{aligned}(F) &= [M_c](f) \\ (f) &= [M_a](i)\end{aligned}\right\} (F) = [M_r](i) \quad \text{avec} \quad [M_r] = [M_c] \times [M_a] \tag{6}$$

$[M_r]$ 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 is associated with interface entity j.

Thus, to establish our associations' management model, three matrices are used: the initial association matrix, the correspondence matrix and the reconciliation matrix. The association matrix is easily extracted from the starting point (iWP/iDMU). The correspondence matrix, however, must be obtained using a specific algorithm that is developed to find the correspondences between the mWP and the iWP and it's not the object of this paper [13].

4.4 Change propagation

Propagating change 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) or pseudo-imprints [9] [10] [11] [12]. That assistance implies that we must locate the areas of the DMU that need to be modified and provide users with the support they need (sketches and pseudo-imprint) to apply the changes. Another way to assist users is to provide solutions in the form of connectivity maps [5]. In our study, we drew on the latter solution and chose to provide the user with an association reconciliation matrix based on two other matrices: a correspondence matrix $[M_c]$ and an initial association matrix [M_a]. The reconciliation matrix enables us to determine, for each mWP entity, if it matches any of the DMU entities and if yes, which one(s). The user is also advised, via context messaging, of the principal changes that should be made to the DMU to maintain the consistency of the associations with the modified version of the work package.

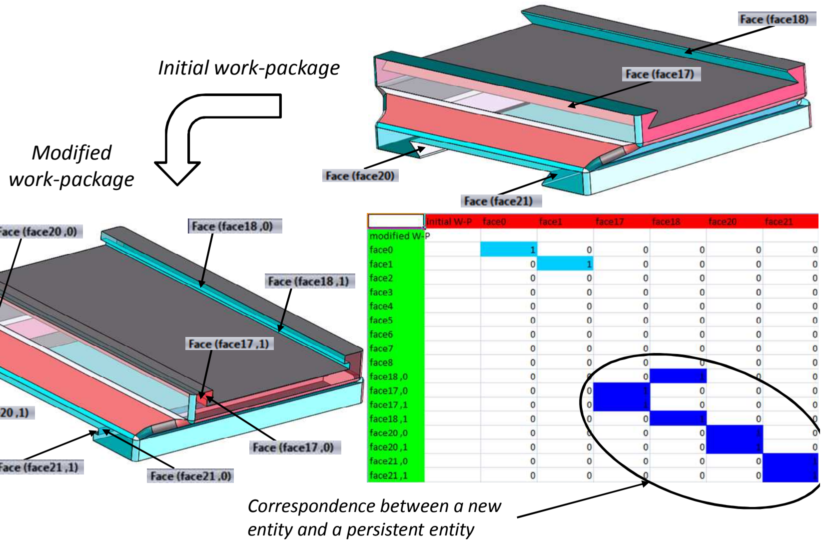
5 Illustration

To illustrate and validate this approach, we take a milling machine table as a model. The modification scenario of the work package is: change in technological solution (T shaped to dovetail slide linkage).

In the first phase, the initial association matrix [M_a] between the iWP and the iDMU is determined. In this example the associations are assembly constraints established within the DMU prior to extraction of the WP. This step consists of scanning the various assembly constraints and identifying the entities in the iDMU that match those in the iWP.

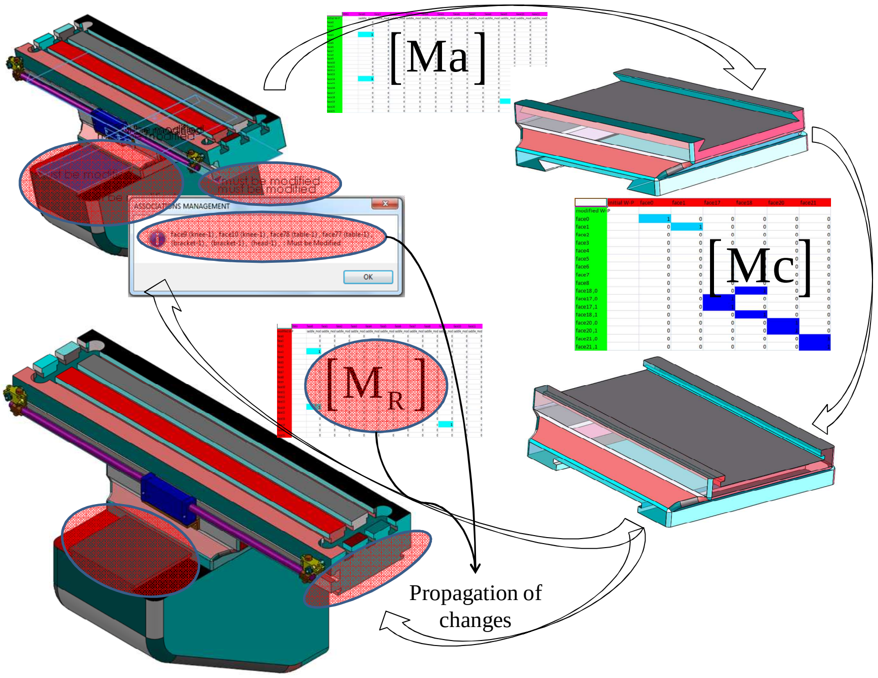
In the second step, correspondences between entities of the iWP and of the mWP are established. The orphan entities of the mWP are reconciled through the correspondence algorithm, process that enables the correspondence matrix to be generated (as an Excel spreadsheet). This step is essential for the transposition of associations towards the mWP. The reconciliation results, the correspondence between the two versions of WP entities (initial and modified), as well as the correspondence matrix [M_c] are shown in figure 7.

The third step consists in reconciling the associations between the mWP and the mDMU. The reconciliation matrix [M_r] is calculated by multiplying the correspondence matrix [M_c] by the initial association matrix [M_a]. The reconciliation matrix is provided to the user in an Excel spreadsheet so that the user is able to make the necessary changes to the DMU in order to obtain consistent associations (Figure 8). A mDMU entity must be modified if it has an association with a mWP entity that has been modified. Therefore, we provide the user with decision making support so that he/she can make the changes required to obtain consistent associations and geometry. The user is alerted with a message indicating which entities need to be modified. An annotation “Must be modified” is placed on each of these (Figure 8).



modified W-P	Initial W-P	face0	face1	face17	face18	face20	face21
face0		1	0	0	0	0	0
face1		0	1	0	0	0	0
face2		0	0	0	0	0	0
face3		0	0	0	0	0	0
face4		0	0	0	0	0	0
face5		0	0	0	0	0	0
face6		0	0	0	0	0	0
face7		0	0	0	0	0	0
face8		0	0	0	0	0	0
face18 ,0		0	0	0	1	0	0
face17 ,0		0	0	1	0	0	0
face17 ,1		0	0	1	0	0	0
face18 ,1		0	0	0	1	0	0
face20 ,0		0	0	0	0	1	0
face20 ,1		0	0	0	0	1	0
face21 ,0		0	0	0	0	0	1
face21 ,1		0	0	0	0	0	1

Fig. 7. Entity reconciliation and reconciliation matrix construction.



face9 (knee-1) , face10 (knee-1) , face78 (table-1) , face77 (table-1) , (bracket-1) , (bracket-1) , (head-1) , : Must be Modified

Fig. 8. Complete association management cycle with transposition and reconciliation.

6 Conclusion and perspectives

We have presented an association management model that functions in three steps. First, it captures the initial associations between the iWP/iDMU in the associations Matrix [M_a]. Next, it controls the modifications from the iWP to the mWP using the correspondence algorithm to calculate the correspondence Matrix [M_c]. This step makes it possible to transpose the initial associations to the mWP. Finally, the third step involves the reconciliation matrix [M_r], found by multiplying [M_a] and [M_c], to reconcile the associations and propagates changes to obtain a mDMU that is consistent with the mWP. Recall that our objective is not to eliminate user intervention but rather to assist users through an efficient management model. This work provides decision making support for the user which allows him/her to associate the mWP with the mDMU and to 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 information consistency. This work is part of the process to solve the totally indetermination association scenario by developing an algorithm that transforms indeterminate cases into partially undetermined cases. This last algorithm is used to detect the similarity between work-packages (initial and final) and to match the similarities' faces (Figure 13). The correspondence between other faces is determined by use of the same algorithm that was used to solve the partially undetermined associations case.

7 Bibliography

1. http://www.mcgill.ca/plm2-criaq/project/.

2. Fortin, C., Toche, B., McSorley, G., & Huet, G. (2010). 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, MMEP'10, Cambridge, UK.

3. Toche, B., Huet, G., McSorley, G., & Fortin, C. (2010). 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, IDETC/CIE 2010, Montreal, Canada.

4. Zimmermann, J. U., Haasis, S., & Van Houten, F. J. A. M. (2002). ULEO - Universal linking of engineering objects. CIRP Annals - Manufacturing Technology, 51(1), 99-102.

5. Yassine Ali A., Whitney D., Daleiden S., Lavine J., Connectivity maps: modeling and analysing relationships in product development processes. Journal of Engineering Design, 14(3), 377-394, 2003.

6. Yassine Ali A., Whitney Daniel E., Whitney, Lavine Jerry, Tony Zambito (2000), DO-IT-RIGHT-FIRST-TIME (DRFT) APPROACH TO DSM RESTRUCTURING. ASME 2000 International Design Engineering Technical Conferences, September 10-13, 2000, Baltimore, Maryland.

7. X. Chen, S. Gao, Y. Yang, S. Zhang, Multi-level assembly model for top-down design of mechanical products, Computer Aided Design, In Press, Available online 24 December 2010.

8. 8.T.G.Tremblay, L.Rivest, O.Msaaf, R.Maranzana, 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, 18 - 22 September, 2006.

9. Tremblay, T.G. (2006). La propagation du changement dans un contexte PLM : vers la maîtrise du rôle des associations dans le développement de produits et des processus associés, Master Thesis, École de technologie supérieure, Montréal, Canada.

10. Giguère, F., Rivest, L., Desrochers, A., & Maranzana, R. Les caractéristiques contextuelles : une solution pour accroître la productivité en CAO. Revue Internationale de CFAO et d'Informatique Graphique, 17(1-2), 105-117 (2001).

11. F. Giguere, L. Rivest and A. Desrochers. Improving design productivity and product data consistency, Feature Based Product Life-Cycle Modelling, René Soenen and Gustav J. Olling, ed., Hardbound, Kluwer Academic Publisher, pp.77-91 (2002).

12. D. Mun, J. Hwang, S. Han, Protection of intellectual property based on a skeleton model in product design collaboration, Computer Aided Design 41, 641-648 (2009).

13. B. Louhichi, Louis Rivest, “Maintaining Consistency between CAD : Elements in Collaborative Design using Association Management and Propagation”, submitted to Journal of Engineering Design, November 2011.