Advances in Manufacturing Technology XXXV
M. Shafik and K. Case (Eds.)

doi:10.3233/ATDE220590

Change Propagation and Associations Reconciliation Based on Feature Correspondences

Ameni ELTAIEF[1], Borhen LOUHICHI[1], Sébastien REMY[2] and Benoit EYNARD[3]
[1] *University of Sousse (ISSATSo, LMS), Sousse 4023, Tunisia*
[2]*LASMIS, Université de Technologie de Troyes, France*
[3]*Department of Mechanical Engineering, Université de Technologie de Compiègne, Roberval Institute, France*

Abstract. CAD systems have undergone enormous progress. They are becoming more and more intelligent and automated. But mechanical products are also more and more complex. which makes their management more delicate, and especially the management of associations and the propagation of changes affecting one of the components in systems designed using Computer-Aided Design software's. Therefore, to propagate this change easily, it is important to predict the Change Propagation Path. In this context, this paper focus on a mothed based on establishing correspondences between the assembly Components features to facilitate the associations reconciliation and then change propagation. Thanks to this method, by specifying the affected features in one component, it is easy to predict which other features (in other assembly parts) are likely to be affected too. To this end, the authors proposed an Association *Reconciliation Algorithm* as well as a *Correspondences Algorithm* which are mainly based on the assembly mates, the topological and geometrical relationships existing between components. The results of those algorithms will be used later one to determine the change propagation paths and the change propagation tree.

Keywords: Assembly Management, Change propagation, Propagation tree, Features, Associations, Correspondences, Assembly Mates, Parent feature.

1. Introduction

It is important to clearly define the terms that will be used as a basis for this research. A dependency is a relationship that makes a given object dependent on another. According to Giguère et al. [1], a technological link is a link expressing a dependency relationship between information items. In the context of an assembly, many authors ([2-4]) use the term Mating relation to describe a dependency between two components of an assembly. To describe the type of dependency more precisely, they use the term constraint. While Association is a generic term that refers to the family of terms that includes relationship, constraint, and link. An association expresses a dependency between elements or entities, whereas a relationship establishes an explicit and abstract dependency between two technical objects (without involving any specific or formalized knowledge). The interpretation and use of a relationship require human intervention. For example, the document defining the tooling used in the manufacture of a given part is linked by a relation to the document that defines the part. However, the knowledge required to define the tooling from the part definition is not formalized in this relation.

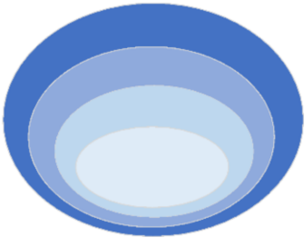


Figure 1. Different dependencies levels

Although, the link can be defined as a tool that joins, unifies, establishes a logical relationship or dependency. It is the dependency between two elements that involve a formalized knowledge to perform a given task. Links can be managed automatically by a software program, but they may occasionally require human intervention. The know-how associated with a link is totally mastered and formalized, contrary to that implied by a relationship. Links within assemblies are tools used in CAD systems. Assembly links are established between different components. For instance, they allow components to be aligned and positioned within each other [3].

Finally, the constraint concretely and precisely describes a type of dependency between two entities. A constraint is indecomposable, it is a low-level know-how. The level of abstraction of a constraint is lower than for a link. A link is an organized aggregation of constraints. A constraint constitutes a condition to be respected to validate the model [3]. For example, a perpendicularity constraint can be applied between two lines, as an operation to be performed. The different levels of dependencies are illustrated by the figure fig.1.

In the context of the management and propagation of modifications within an assembly, dependencies have been used as a tool to express whether a change in the value of one element (of the assembly) can generate changes in the values of other parameters or other elements [5]. Indeed, a dependency between two parameters of a product is defined as: "the direct effect of the change of the value of one parameter on another" [5-6]. DSM /DMM are tools that help to capture and visualize the existing dependencies in a product model ([7-9]).

Dependency graphs have also been used in the field of engineering change management. In the field of change management, graph theory is used to model and visualize the various dependencies between elements of a system. This type of graph is often called a "Dependency Graph" but the term "Dependency Network" is also commonly used. The matrices, DSM or DSM/DMM are like a dependency graph in that they contain the same data (dependency values). Indeed, a matrix adjacent to a graph (with which a graph is constructed) coincides with the definition of a DSM. Thus, we find the use of these two types of data structures (matrices and dependency graphs) in the ECM methods.

While in CAD, associations are defined as geometric and parametric constraints established between geometric entities (faces, edges, vertices). Ma et al. [10] distinguish two types of associations: geometric and non-geometric. However, Tremblay [11], arranges the associations in 3 types: relation, links and constraints. To characterize these different types of associations, four properties have been determined: the Cardinality which defines the number of entities used, the Direction, the temporality which is a characteristic that shows that an association is persistent, transient, or semi-persistent, the last property is the aggregation and the decomposition: the aggregation is the grouping of a set of associations in order to facilitate its manipulation at a higher level of abstraction. Decomposition is the opposite operation. In the work of Louhichi et al. [12] associations are assembly constraints established within the DM (CAD Assembly). According to Chen et al. [13] an association describes a group of semantically structured links. In the context of an assembly, several authors ([14-16]) use the term contact relationship to describe a dependency between two components of an assembly. In the context of the product and assembly relationships' management in the concurrent engineering and product lifecycle management (PLM) domain, Demoly et al. [17] have developed a novel approach to integrate assembly process engineering information and knowledge in the early phases of the product development process. Eltaief et al. have proposed different approaches for a better management of the Digital Mockup (DMU) and propagation of changes to the assembly after a modification that affects one of its components [18-19]. For an analysis of the risk assessment of an engineering change, a matrix-based approach is proposed [20].

2. Proposed approach

2.1. Initial associations Implementation

The term association is a generic term that has already been defined and that expresses a dependence between elements or entities. Moreover, it has already been mentioned that the term association designates the highest level of knowledge, while the word constraint describes the lowest level of knowledge. Consequently, it is proposed to start from the lowest level (assembly constraints) until reaching the highest level and establish associations between the different Features of the different components of the assembly.

In fact, the assembly constraints used are generally standard constraints that link two geometric or topological entities. From these constraints, associations between features will be determined. The extracted constraint data is then stored in the database. These data are the basis on which the algorithm of association determination between Features is founded. Indeed, for each entity involved in a constraint, the parent Feature or the associated Feature can be determined. Then the two associated constraint entities will give two associated Features and so on until all constraints are completed. Thus, a list of Features associated in pairs is obtained. Consequently, a sorting method is used to form a list of associated Features. Because a Feature can have more than one other Feature associated with it, this way we obtain lists of associated Features. Each list contains features that are related to each other and have interactions and influences on each other. Therefore, changes made to one of the two Features that belong to the same list or that are "associated" must necessarily be propagated to the other.

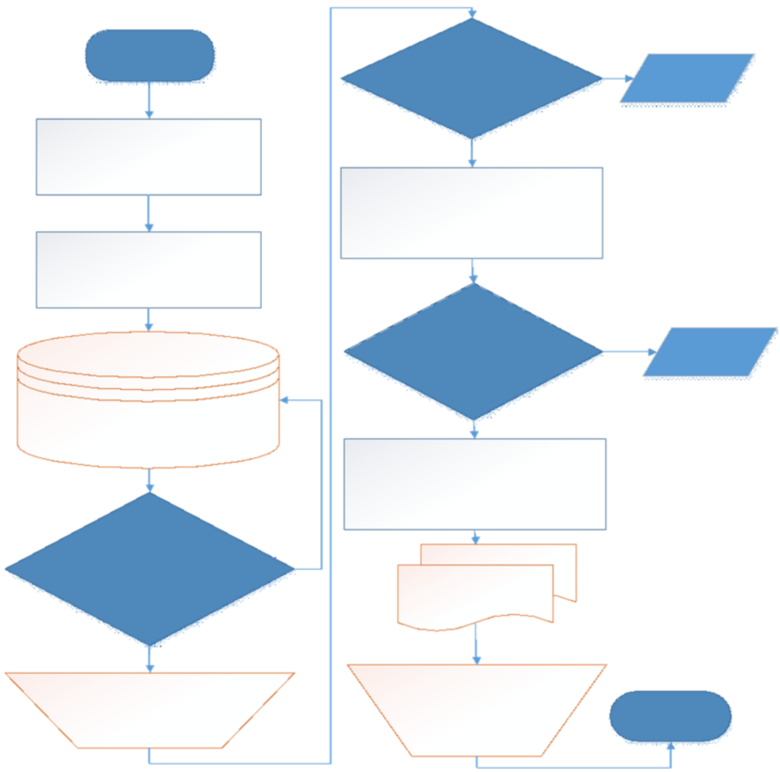


Figure 2. Determination of Feature Associations algorithm

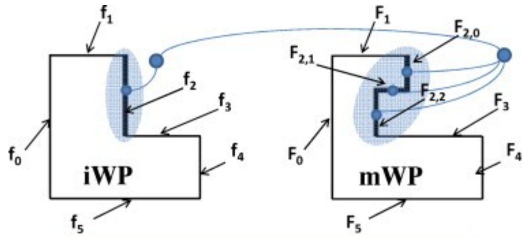
2.2. *Correspondence Implementation*

As it was assumed from the beginning that in the context of this work one of the parts of the assembly will be extracted, modified, and then reinserted back into the assembly. Consequently, some Features will be affected. Indeed, there are features that will be affected or modified while there are others that will be deleted and others that are newly created. Therefore, the lists of recreationally created features must also be modified. To do this, matches between the original and modified features of the room that has been modified must be made. Because the newly created features do not have features associated to them in the lists of the previous part. On the other hand, there are others that have been deleted and that must be deleted or replaced by their equivalent.

Therefore, to solve this problem, correspondences must be made. The starting point in this section is the work of Louhichi et al. [17].who perform mappings between the initial Work Package "iWP" and the modified Work Package "mWP" by associating,

In general, m entities of the iWP to n entities of the mWP. The entities in question are topological entities and generally faces (Fig.3).

The same principle will be applied but for Features. For the recently created Features, the deleted Features and also those that have been modified, the determination of possible matches that can be implemented is necessary (Fig.4).



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

Figure 3. The correspondence principle used by Louhichi et al.[17]

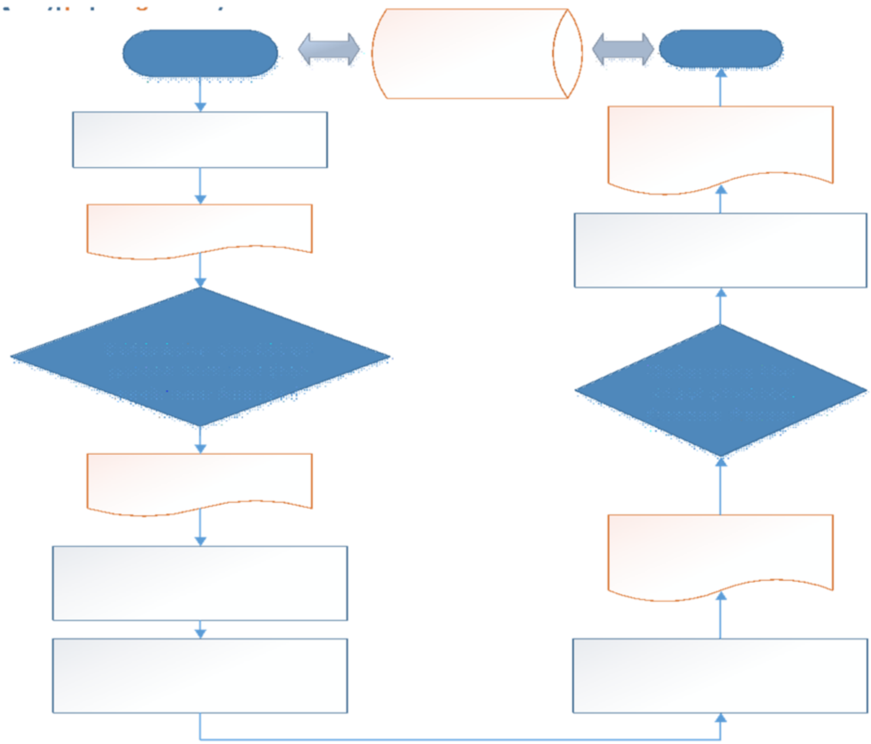


Figure 4. Algorithm for determining correspondences between features.

2.3. Associations Reconciliation

A model for implementing associations within the CAD model based on assembly constraints have been already designed. This is done by creating the links that will be later used in the industrial change management process. The model highlights the information that allows us to evaluate a proposed change. This information allowed us to create correspondences between the Features of the initial model and those of the modified model. Thus, the lists of associated Features as well as lists of corresponding Features are obtained.

Reconciliation is the Action of Restoring the links that existed before. In this case, the modification made to a part of the assembly may damage some assembly constraints and links (which initially exist) and therefore they need to be restored them. Subsequently, the Lists of associated Features that exist at the beginning will be updated (or reconciled), while considering the corresponding Feature lists. If a Feature called i belongs to a list of associated Features called list x, and at the same time this same Feature (Feature i) has corresponding Features, then all the list of corresponding Features will be inserted in the list x and the new list of associated Features called list x' is obtained.

3. Conclusions and perspectives

The algorithms developed in this paper are only sub-algorithms of a general decision support algorithm which aims at facilitating the task of the designer and his different partners in the context of collaborative design. Some illustrative examples of the different steps are also presented. The objective of the general algorithm developed is to be able to measure the impact of a modification and the determination of a propagation tree of the modifications. The determination of the propagation tree is mainly based on the algorithm of graph traversal in width which showed good performances. The presented approach allows to properly judge the feasibility and the impact of a proposed modification. These algorithms will be translated and implemented in a further phase to obtain an efficient tool that facilitates the management of modifications and is easy to use.

References

[1] F. Giguère, L. Rivest, and A. Desrochers, “Improving Design Productivity and Product Data Consistency,” in *Feature Based Product Life-Cycle Modelling*, Boston, MA: Springer US, 2003, pp. 77–91.

[2] J. Tan, “Self-organizing assembly modeling based on relational constraints,” *Chinese J. Mech. Eng. (English Ed.*, vol. 13, no. 02, p. 145, 2000, doi: 10.3901/CJME.2000.02.145.

[3] Thomas Guillaume Tremblay, “La Propagation Du Changement Dans Un Contexte Plm : Vers La Maîtrise Du Rôle Des Associations Dans Le Développement De Produits Et Des Processus Associés,” École De Technologie Supérieure Université Du Québec, 2006.

[4] O. W. Salomons, F. van Slooten, G. W. F. de Koning, F. J. A. M. van Houten, and H. J. J. Kals, “Conceptual Graphs in CAD,” *CIRP Ann.*, vol. 43, no. 1, pp. 125–128, Jan. 1994, doi: 10.1016/S0007-8506(07)62179-9.

[5] M. Masmoudi, “Contribution au développement d’une méthode analytique pour l’identification des dépendances et la propagation des changements d’ingénierie en re-conception de produits [thèse de doctorat],” L’UNIVERSITÉ PARIS-SACLAY, 2016.

[6] X. Zhang, “Contribution to engineering change management in product development projects : reference models and multi-agent-based simulation,” L’Université Bordeaux 1, 2013.

[7] K. Rahmani and V. Thomson, “Ontology based interface design and control methodology for collaborative product development,” *Comput. Des.*, vol. 44, pp. 432–444, 2012, doi: 10.1016/j.cad.2011.12.002.

[8] B. Hamraz, N. H. M. Caldwell, and P. John Clarkson, "A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design," *J. Mech. Des.*, vol. 134, no. 10, p. 100905, 2012, doi: 10.1115/1.4007397.

[9] L. Yin, D. Tang, I. Ullah, Q. Wang, H. Zhang, and H. Zhu, "Analyzing engineering change of aircraft assembly tooling considering both duration and resource consumption," *Adv. Eng. Informatics*, vol. 33, pp. 44–59, 2017, doi: 10.1016/j.aei.2017.04.006.

[10] Y. G. Ma A Britton S B Tor L Y Jin, "Associative assembly design features: concept, implementation and application," *Int J Adv Manuf Technol*, vol. 32, pp. 434–444, 2007, doi: 10.1007/s00170-005-0371-8.

[11] T. G. Tremblay, L. Rivest, O. Msaaf, and R. Maranzana, "The role of associations in CAD and PLM for handling change propagation during product development," in *13th ISPE International Conference on Concurrent Engineering: Research And Applications*, pp. 507–514, [Online]. Available: http://portal.acm.org/citation.cfm?id=1566652.1566726.

[12] B. Louhichi and L. Rivest, "Management of associations within digital mock-ups for improved collaboration," *IFIP Adv. Inf. Commun. Technol.*, vol. 388 AICT, pp. 411–423, 2012, doi: 10.1007/978-3-642-35758-9_37.

[13] Y. M. Chen, W. S. Shir, and C. Y. Shen, "Distributed engineering change management for allied concurrent engineering," *Int. J. Comput. Integr. Manuf.*, vol. 15, no. 2, pp. 127–151, 2002, doi: 10.1080/09511920110047181.

[14] B. Hamraz, "Engineering Change Modelling Using a Function-Behaviour-Structure Scheme," the University of Cambridge, 2013.

[15] D. T.- Gilles Glemarec, *Le guide de la CAO* - , vol. 192. 2008.

[16] G.. Huang, W. . Yee, and K. Mak, "Current practice of engineering change management in Hong Kong manufacturing industries," *J. Mater. Process. Technol.*, vol. 139, no. 1–3, pp. 481–487, Aug. 2003, doi: 10.1016/S0924-0136(03)00524-7.

[16] G.. Huang, W. . Yee, and K.. Mak, "Current practice of engineering change management in Hong Kong manufacturing industries," J. Mater. Process. Technol., vol. 139, no. 1–3, pp. 481–487, Aug. 2003, doi: 10.1016/S0924-0136(03)00524-7.

[17] F. Demoly, L. Toussaint, B. Eynard, D. Kiritsis, S. Gomes, Geometric skeleton computation enabling concurrent product engineering and assembly sequence planning, CAD Comput. Aided Des. 43(12) (2011) 1654–1673

[18] A. Eltaief, B. Louhichi, S. Remy, Associations management and change propagation in the CAD assembly, Computers in Industry 98, 134-144, 2018.

[19] A. Eltaief, S. Remy, B. Louhichi, G. Ducellier, B. Eynard, Comparison between CAD models using modification ratio calculation, International Journal of Computer Integrated Manufacturing, 996-1008, Vol. 32 2019.

[20] A. Eltaief, S. Remy, B. Louhichi, G. Ducellier, B. Eynard, Engineering Change Risk Assessment: Quantitative and qualitative change characterization, Computers in Industry 140, 103656, 2022.

[21] B. Louhichi and L. Rivest, "Maintaining consistency between CAD elements in collaborative design using association management and propagation," Comput. Ind., vol. 65, no. 1, pp. 124–135, 2014, doi: 10.1016/j.compind.2013.08.003.