Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia CIRP 7 (2013) 169 – 174

www.elsevier.com/locate/procedia

Forty Sixth CIRP Conference on Manufacturing Systems 2013

# Matching bills of materials using tree reconciliation

Mohamed Kashkoush, Hoda ElMaraghy*

*Intelligent Manufacturing Systems (IMS) Centre, University of Windsor, 401 Sunset Avenue, Windsor N9B 3P4, ON, Canada*

* Corresponding author. Tel.: +1-519-253-3000 (Ex. 5034) .*E-mail address*: hae@uwindsor.ca.

**Abstract**

A product Bill of Materials (BOM) is a structured tree which represents its components and their hierarchal relationships. The BOMs are traditionally used for Material Requirement Planning (MRP). However, they do have other useful applications in product modeling and variety management. Recent research used graph difference operations, linear algebra and integer programming to match trees of BOM and find pairwise similarity measures for applications such as clustering product variants into families and retrieval of design and manufacturing data. Matching phylogenetic trees has been utilized in biological science for decades and is referred to as "tree reconciliation". A new application of this approach in manufacturing to match pairs of BOM trees and retrieve the most similar design is presented. This novel method can help speeding-up other downstream planning activities such as process planning, hence, improving productivity and shortening time to market. Assembly of chemical processing centrifugal pumps is used as a case study for demonstration. This novel matching of Bills of Materials uses linear time algorithms, compared to state-of-the-art algorithms which use integer programming and matrix approximation, hence, leading to more computational efficiency.

*Keywords: Bill of Materials (BOM); Tree Reconciliation; Design retrieval; Assembly Process Planning*

## 1. Introduction

The Bill of Materials (BOM) was first introduced by Orlicky [1, 2] as a product data structuring form for Material Requirements Planning (MRP) systems commonly used in production planning and inventory control. Bills of Materials have other useful applications in product modeling and variety management [3]. A BOM includes a list of sub-assemblies, components, parts, and raw materials and the quantities required of each to produce an end product. Unordered rooted trees are used to represent BOMs. Fig. 1 shows a simple three-level BOM tree for the major components of a vibration motor used in cell phones. BOM of complex products would have more hierarchical levels down to fasteners and small parts. In today's manufacturing environment, responsive and cost-effective design and manufacturing activities has become a pre-requisite for survival in a global economy. Automatic retrieval of relevant past designs and manufacturing data is one of the tools used to support rapid product development.

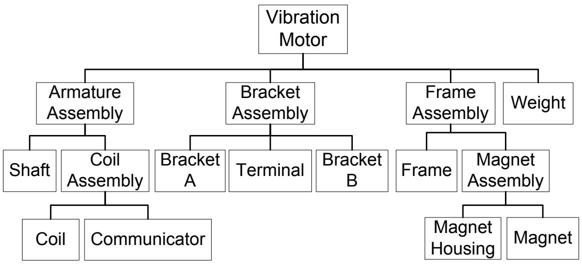

Fig. 1. BOM tree for a vibration motor (Adapted from **[4]**)

Matching BOM trees and finding pairwise similarity or dissimilarity measures between trees of BOM has recently emerged as a product clustering technique for grouping similar products into families; as well as a data mining approach for retrieving design and manufacturing data. It could be seen as a new alternative to conventional Group Technology (GT) classification and coding systems [5] used for clustering and design retrieval [6, 7]. However, application of classification and coding systems to design retrieval, and most other design retrieval methods, such as 3D modeling [8] and

doi:10.1016/j.procir.2013.05.029

image processing based methods [9], are mainly concerned with individual parts not products based on similarity of their geometrical features. Bill of Materials matching addresses the assembly level, where similarity between whole products is sought based on the product structure and components.

Finding BOM of an existing product that is most similar to that of a new product helps in subsequent applications. For instance, process plan of a retrieved product could be used as a starting process plan that is be modified to fit the new product. Hence, the time required to develop the new process plan would be significantly reduced. Furthermore, other useful data such as supply chain data, sourcing, procurement, and logistics management can be also obtained as a result of design retrieval. All this contributes to increasing the efficiency and productivity of design and process and production planning.

Tree reconciliation has been well-studied in biological research fields. Typical applications include gene function prediction, gene annotation, and identifying drug targets [10-12]. This paper introduces a novel method, based on tree reconciliation, for matching products Bills of Materials (BOM).

## 2. Related Work

Limited research work is found in the literature regarding implementations of tree reconciliation techniques in engineering. The most recognized of them is the work of AlGeddawy and ElMaraghy on modeling co-evolution of products and manufacturing systems for applications such as future planning of manufacturing systems and knowledge discovery from manufacturing data [13-15].

Bills of Materials (BOM) trees are categorized as unordered rooted trees. At any given hierarchical tree level, the order of nodes belonging to the same node has no significance and the first level nodes all belong to a single node which is the root. An obvious approach for matching trees of BOM is classical tree-matching algorithms commonly used in mathematics and computer science [16-18]. Most of these techniques are based on performing limited sets of editing operations to transform two trees into isomorphism (i.e. congruency). Romanowski and Nagi [19] presented a critical evaluation of editing-based trees matching and showed, using examples, that they may yield incorrect results when used to match trees of BOM. Romanowski and Nagi [19] introduced the first tree-matching algorithm specific to BOM trees and named it Decomposition-Reduction or DeRe algorithm, which is a modified weighted symmetric difference operation from set theory in which partial matching between edges is allowed. Romanowski et al. [20] reported that while the DeRe algorithm is more accurate than conventional editing-based techniques, it may in some cases overstate the dissimilarity measure. Accordingly, they formulated a novel Integer Programming model that maximizes, for a given pair of BOM trees, the matching score between every possible sub-tree of a given tree, and those of the other tree. The model has been proven to be NP-hard, which means that heuristic algorithms would be required to solve real-sized models. Some suggestions were made to deal with this problem but none of them was actually tested.

Shih [21] solved the BOM tree matching problem as an Orthogonal Procrustes - a matrix approximation problem in linear algebra. In the classical form of Orthogonal Procrustes problem, given two matrices $X$ and $Y$, it is required to find an orthogonal transformation matrix $T$ that most closely maps $X$ to $Y$. The simplest algebraic statement of a Procrustes problem is finding matrix $T$ that minimizes $\|X - YT\|$. The author represents any two BOM trees by adjacency matrices, and subsequently finds that transformation matrix $T$ to be then used to calculate a similarity measure between the two adjacency matrices.

The algorithm involves several matrix multiplication operations that make it computationally intensive, especially with large size matrices.

In conclusion, a few techniques exist for matching trees of BOM; however, they take into consideration the intermediate nodes which are believed to be of no significance. A BOM intermediate node (sub-assembly) is equivalence to its sub-nodes (components). The difference in topologies and contents of end nodes are the important factors in matching BOMs. Well-developed tree reconciliation algorithms used in biological science do satisfy this objective and are more computationally efficient because they are based on linear time algorithms as will be shown in section 3.

## 3. Tree Reconciliation

### 3.1. Background

Tree reconciliation is the problem concerned with the proper mapping between two phylogenetic trees (known as cladograms): an associate tree (usually a gene family tree) and a host tree (usually species tree) with the aim of properly reconstructing the evolutionary history for the associate tree to match that of the host tree. Such a reconstruction process is based on the assumption that some biological events; such as *gene duplication* and *loss* events, are not properly recorded on the existing form of the associate tree and, hence, they need to be algorithmically postulated.

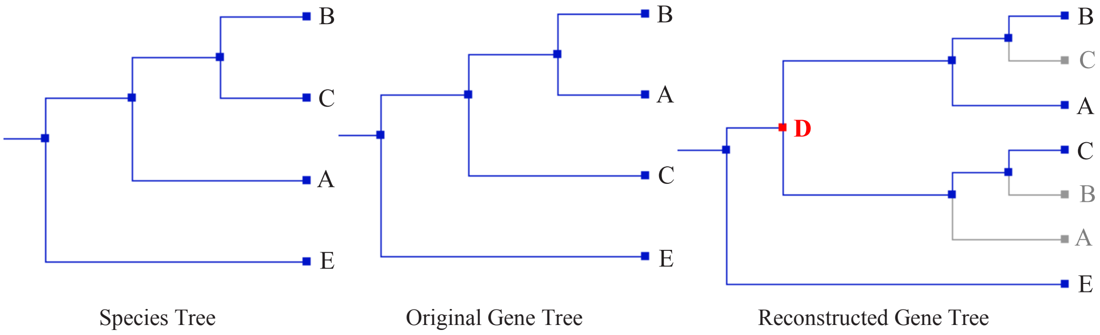

Fig. 2. An example for tree reconciliation

Fig. 2 shows a simple example of what is meant by tree reconciliation. The first tree in this figure is the cladogram of the 4 species *A*, *B*, *C* and *E*; while the second tree is the corresponding cladogram of genes collected from those four species. Note that the labels on the gene tree are the names of the species from which each gene is obtained, and not the names of the genes themselves.

It is clear that both trees do not match; hence, a third cladogram is reconstructed to rectify that. The reconstructed gene tree is a new cladogram that contains the original gene tree as a homomorphic (structure-preserving) sub-tree and represents the evolutionary history of the species tree [22].

In this example, one duplication event is postulated at the point where a capital letter D is shown, as well as three gene losses events (pale branches). Gene duplication event means that two copies of a given gene start changing independently of each other and both being inherited by subsequent generations. A gene loss event refers to genes that are either missing or not yet discovered. Hence, a tree reconciliation algorithm searches for the reconstructed gene tree with the least postulated number of duplication and loss events (most parsimonious cladogram [23]).

In the context of BOM matching, duplication and loss events refer to the least number of modifications needed in an existing product structure to match the new one.

### 3.2. Tree reconciliation algorithm

A significant part of a typical tree reconciliation algorithm is to find the Least Common Ancestor (LCA), which means finding the shared ancestor that is located farthest from the root, for any pair of nodes in a given tree [24-26]. Many tree reconciliation algorithms and software packages are now available such as GeneTree [27], Softparsmap [28] and NOTUNG [29]. The last version of the later one, NOTUNG 2.6, is the software package used in this research.

The reconciliation algorithm implemented in NOTUNG is based on the work reported in [29-31]. Given a gene family tree $T_G$ and a binary species tree $T_S$ the algorithm identifies all gene duplication and loss events. The total cost of reconciliation (matching cost) of two trees $T_G$ and $T_S$ is given by Equation 1 [29], where $N_D$ is the number of duplication events and $N_L$ is the total number of loss events, while $C_D$ and $C_L$ are unit costs for duplication and loss events respectively.

$$C(T_G\ ,T_S) = C_D\ N_D + C_L\ N_L \tag{1}$$

The kernel of this algorithm is the Least Common Multiple (LCA) algorithm and the algorithm of inferring duplication events, which are both linear time algorithms [26, 29].

## 4. BOM Tree Reconciliation

In the following subsections, the proposed BOM trees matching method based on tree reconciliation is illustrated.

### 4.1. Handling non-binary trees

Typical tree reconciliation algorithms are capable of considering binary trees only for both gene and species trees, where each node has at most two sub-nodes.

However, some algorithms, including the one implemented in NOTUNG [32], are capable of reconciling binary gene tree to a non-binary species tree, or a non-binary gene tree to a binary species tree. It is very common for BOM trees to exist as non-binary trees; however, converting a non-binary tree into a binary tree is a straightforward operation. Thus, before reconciling any two non-binary BOM trees, one of them should be first converted into a binary tree. Those algorithms that are capable of reconciling two trees, one of them is non-binary, they do actually internally convert the non-binary tree first into a binary while maintains minimum reconciliation cost.

The application targeted by this work, which is the retrieval of the most similar historical BOM tree to a new one, requires pairwise tree reconciliation on a new

BOM tree on one side and several existing BOM trees on the other side. Hence, it would be more practical to only convert the new BOM tree into binary instead of converting several existing ones. In addition, if this is done in reverse, misleading results would be obtained as it would be very difficult to maintain consistency in the non-binary to binary conversion process throughout the entire BOM database.

### *4.2. Extra components in the existing BOM trees*

For any tree reconciliation algorithm, all items of the gene tree must also exist in the species tree; however, the species tree may include extra items. From a BOM trees matching perspective, those extra components are indeed a source of dissimilarity to be accounted for. Therefore, before reconciliation, any existing BOM tree is first searched for any items that do not exist in the new BOM tree. This set of items is added as pseudo items to the new BOM tree, thus, creating a modified version of the original new BOM for each existing BOM tree. The example in Fig. 3 shows three trees; the first is a new BOM tree, the second is an existing BOM tree, while the third is the modified new BOM tree after adding a new branch of pseudo items (outgroup). The new branch includes the two items *K* and *L* which exist in the existing BOM tree but not the new one.

### *4.3. Extra components in the new BOM tree*

Another source of dissimilarity that needs to be considered is the extra items in the new BOM tree that do not exist in the existing ones. In order to account for this type of dissimilarity without doing further alterations to the BOM trees, the number of extra items in the new BOM tree would be considered as an additional term in the total reconciliation cost (Equation 1). Accordingly, the total reconciliation cost is now given by Equation 2, where $N_C$ is the number of extra items in the new BOM tree, and $C_C$ is its corresponding unit cost.

$$C(T_G\ ,T_S) = C_D\, N_D + C_L\, N_L + C_C\, N_C \qquad (2)$$

### *4.4. Difference in number of components*

The last source of dissimilarity, that has not been considered yet, is the difference in the number (or quantity) of components in each of the existing and new BOM trees. With regard to process planning type of applications, that sort of dissimilarity would not be of a great impact as having duplicates of the same component or sub-assembly in a given product would duplicate the corresponding assembly operation. This is not of a significant effect on the product process plan which is basically concerned with the type and sequence of assembly operations. Thus, such an aspect of dissimilarity is to be taken into consideration if for a certain two pairwise matchings, between a new BOM tree and two existing ones, the total reconciliation cost given by Equation 2 was equal.

In that case, difference in number of components is applied as an additional criterion to decide which of the given two existing trees is more similar to the new one. The difference in number of components is given by the sum of absolute differences in numbers of each component that exist in both trees. The number to be actually considered for a certain component in a certain tree is given by the multiplication of numbers needed from every ancestor node to that component and the number needed from the component itself.

For instance, the difference in quantities of components of the two trees, *T1* and *T2*, shown in Fig. 4 is given by the sum of absolute differences in quantities of components *B*, *D* and *E*. For tree *T1*, we need a quantity 2 from *B*, 2x3 from *D* and 2x2 from *E*. As for the tree *T2*, we need quantity 3 from *B*, 1X3 from *D* and 1x3 from *E*. Thus, the total difference in quantities should be 1 for difference in B plus 2 for difference in *D* plus 1 for difference in *E*, which gives a total difference of 4.

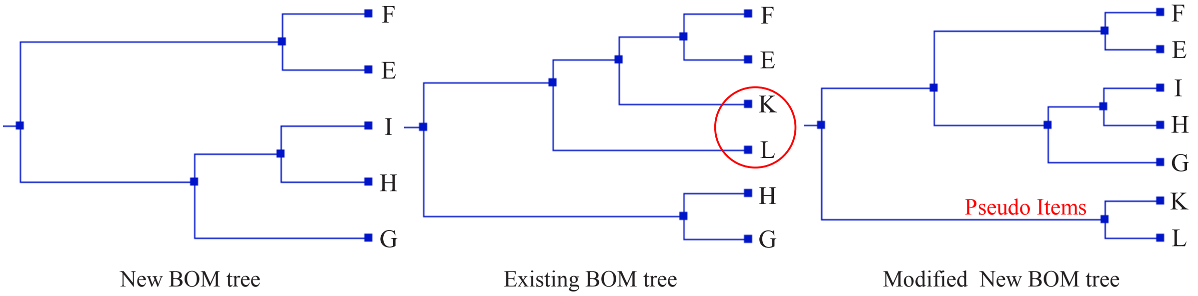

Fig. 3. Adding pseudo items to the new BOM tree

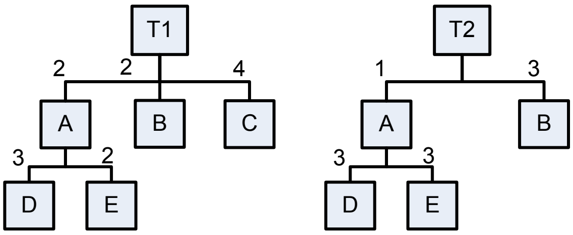

Fig. 4. Calculating difference in number of components

### *4.5. Summary of proposed BOM trees matching method*

The proposed BOM trees matching method, based on tree reconciliation, is summarized in the following steps:

| | |
|---|---|
| **Step 1:** | Build a cladogram for the existing BOM tree |
| **Step 2:** | Build binary cladogram for the new BOM with extra items in the existing tree as pseudo items |
| **Step 3:** | Calculate duplication and loss events using NOTUNG |
| **Step 4:** | Find extra items in the new BOM tree |
| **Step 5:** | Calculate total reconciliation cost using Equation 2 |
| **Step 6:** | If more than one existing BOM has the same total reconciliation cost with the new BOM, the best matching tree is the one with the least difference in numbers of components |

## 5. Case Study

A case study that involves six different designs for chemical processing centrifugal pumps adopted from Liquiflo™ catalogues (http://www.liquiflo.com) is used to demonstrate the proposed BOM matching method. It is assumed that one of these six pumps is a new design. Thus, it is required to find the existing design that is most similar to the new one. The ultimate objective would be to use the assembly data corresponding to the best matching pump design in planning for the assembly of the new pump. A BOM tree for each pump design was constructed from labeled catalogue drawings. Main components only were considered and standard components such as bolts, nuts, screws, washers, plugs, pins, and keys were not included. This is to keep the case study at a manageable size for demonstration purposes. Accordingly, the number of involved components, for each pump design, ranges between 8 and 12. Fig. 5 shows a schematic drawing for one of the pump designs (pump 5). BOM tree of pump 1, Centry 620 (Single Mechanical Seal), is assumed to be BOM tree of the new pump, while the remaining five BOM trees are existing trees in the manufacturer database.

As mentioned in Section 4, the BOM tree of the new pump should be converted first into a binary tree. A modified version of the binary tree is then generated for each of the existing five trees. Each modified version includes an additional branch for pseudo components that exist in a given existing tree but not in the original new tree. For instance, for the modified version corresponding to pump 5, the “impeller-2” is added as a pseudo component.

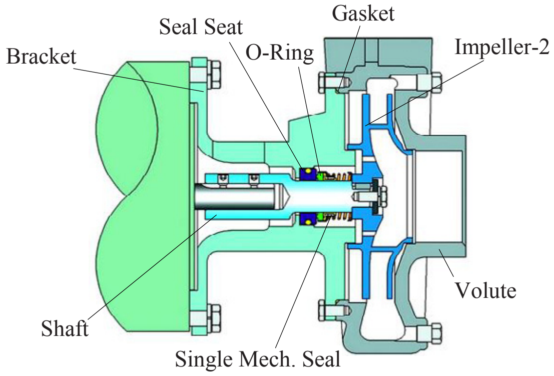

Fig. 5. Pump 5, Centry 621 - Single Mechanical Seal (http://www.liquiflo.com)

Reconciliation costs (Equation 2) obtained for all pairwise matchings between BOM tree of pump 1 and the other five trees, as well as the corresponding matching rank of each existing BOM tree are given in Table 1. The default unit costs used in NOTUNG software for duplication and loss events are applied ($C_D$ = 1.5 and $C_L$ = 1). A unit cost of 1 is assumed to be a reasonable estimate for the components term ($C_C$ = 1). Results of test runs have shown that the proposed reconciliation method is not sensitive to the values assigned to these unit costs. Accordingly, BOM tree of pump 5 is the one that best matches the BOM tree of pump 1; it is the one corresponding to the least total reconciliation cost.

It took less than 1 second to perform the reconciliation step, which is the core part of the proposed algorithm (step 3). For a small data set, such as the one considered in this example, the results could have been obtained without a computerized algorithm. However, for cases where the number of existing BOM trees and the average number of components per tree is in the tens or hundreds, then using an efficient computerized tool becomes a must.

The proposed tree reconciliation method rapidly and efficiently identifies similar current or existing designs that best matches a new one in terms of contents and assembly structure. Matching BOMs of new and existing products has many benefits including automating and speeding up retrieval of design information as well as associated plans for assembly and tooling, NC programs for robots and assembly machines, supply chain and logistics information. Capturing this knowledge and modifying and improving such plans and information as needed, known in literature as variant design or process planning [33], can significantly improve design and planning efficiency and manufacturing productivity.

Table 1. BOM trees matching results

| BOM Tree | Dup. Events | Loss Events | Extra Comp. | Total Recon. Cost | Matching Rank |
|---|---|---|---|---|---|
| **Pump 2** | 2 | 10 | 4 | 17 | 2 |
| **Pump 3** | 3 | 16 | 3 | 23.5 | 3 |
| **Pump 4** | 3 | 18 | 4 | 26.5 | 4 |
| **Pump 5** | 2 | 6 | 3 | 12 | 1 |
| **Pump 6** | 3 | 20 | 5 | 29.5 | 5 |

## 6. Conclusion

In this research, the problem of matching trees of Bill of Materials (BOM) was addressed. One of the very useful applications for this problem would be to use the process plans of retrieved designs as starting process plans for new designs. Well-developed tree reconciliation techniques studied and applied in biological science fields for decades were proposed in this paper as a new method for BOMs matching. A case study of six chemical processing centrifugal pumps was used for demonstration.

This novel BOM matching method, compared to existing algorithms which use integer programming and matrix approximation, is more computationally efficient as it is based on linear time algorithms. Future work would be to study alternatives for adding pseudo items to the new BOM tree. Other applications would also be studied such as clustering of product families.

## References

[1] Orlicky, J. A., 1971. Material Requirements Planning. New York: McGraw-Hill.

[2] Orlicky, J. A., Plossl, G. W., Wight, O. W., 1972. Structuring the Bill of Material for MRP. Production and Inventory Management 13, p. 19-42.

[3] Jiao, J., Tseng, M., Ma, Q., Yi, Z., 2000. Generic Bill-of-Materials-and-Operations for High-Variety Production Management. Concurrent Engineering: Research and Applications 8 4, p. 297-321.

[4] Jiao, J., Zhang, L., Zhang, Y., Pokharel, S., 2008. Association Rule Mining for Product and Process Variety Mapping. International Journal of Computer Integrated Manufacturing 21 1, p. 111-124.

[5] Opitz, H., 1970. A Classification System to Describe Workpieces. Oxford: Pergamon Press.

[6] Lee-Post, A., 2000. Part Family Identification Using a Simple Genetic Algorithm. International Journal of Production Research 38 4, p. 793-810.

[7] Barton, J., Love, D., 2005. Retrieving Designs from a Sketch Using an Automated GT Coding and Classification System. Production Planning and Control 16 8, p. 763-773.

[8] Kunpeng, Z., Yoke San, W., Han Tong, L., Wen Feng, L., 2012. 3D Cad Model Retrieval with Perturbed Laplacian Spectra. Computers in industry 63 1, p. 1-11.

[9] Lim, C., Lee, Y., Choi, J., 2006. Database Construction and Data Retrieval for Optimal Casting Design. International Journal of Cast Metals Research 19 4, p. 259-264.

[10] Bourgon, R., Delorenzi, M., Sargeant, T., Hodder, A. N., Crabb, B. S., Speed, T. P., 2004. The Serine Repeat Antigen (SERA) Gene Family Phylogeny in Plasmodium: The Impact of GC Content and Reconciliation of Gene and Species Trees. Molecular biology and evolution 21 11, p. 2161-2171.

[11] Gabaldon, T., 2006. Computational Approaches for the Prediction of Protein Function in the Mitochondrion. American Journal of Physiology-Cell Physiology 291 6, p. C1121-C1128.

[12] Searls, D. B., 2003. Pharmacophylogenomics: Genes, Evolution and Drug Targets. Nature Reviews Drug Discovery 2 8, p. 613-623.

[13] AlGeddawy, T., ElMaraghy, H., 2011. Manufacturing Systems Synthesis Using Knowledge Discovery. CIRP Annals - Manufacturing Technology 60 1, p. 437-440.

[14] AlGeddawy, T., ElMaraghy, H., 2011. A Model for Co-Evolution in Manufacturing Based on Biological Analogy. International Journal of Production Research 49 15, p. 4415-4435.

[15] AlGeddawy, T., ElMaraghy, H., 2012. A Co-Evolution Model for Prediction and Synthesis of New Products and Manufacturing Systems. Journal of Mechanical Design, Transactions of the ASME 134 5, p. 051008-1 - 051008-12.

[16] Shasha, D., Wang, J. T.-L., Zhang, K., Shih, F. Y., 1994. Exact and Approximate Algorithms for Unordered Tree Matching. IEEE Transactions on Systems, Man and Cybernetics 24 4, p. 668-678.

[17] Jiang, T., Lusheng, W., Kaizhong, Z., 1995. Alignment of Trees-an Alternative to Tree Edit. Theoretical Computer Science 143 1, p. 137-48.

[18] Chawathe, S. S., Garcia-Molina, H., 1997. "Meaningful Change Detection in Structured Data," in ACM SIGMOD International Conference on Management of Data, USA, p. 26-37.

[19] Romanowski, C. J., Nagi, R., 2005. On Comparing Bills of Materials: A Similarity/Distance Measure for Unordered Trees. IEEE Transactions on Systems, Man & Cybernetics, Part A (Systems & Humans) 35 2, p. 249-260.

[20] Romanowski, C., Nagi, R., Sudit, M., 2006. Data Mining in an Engineering Design Environment: Or Applications from Graph Matching. Computers & operations research 33 11, p. 3150-3160.

[21] Shih, H. M., 2011. Product Structure (BOM)-Based Product Similarity Measures Using Orthogonal Procrustes Approach. Computers & Industrial Engineering 61 3, p. 608-628.

[22] Bonizzoni, P., Vedova, G. D., Dondi, R., 2005. Reconciling a Gene Tree to a Species Tree under the Duplication Cost Model. Theoretical Computer Science 347 1, p. 36-53.

[23] Page, R. D. M., 1994. Maps between Trees and Cladistic Analysis of Historical Associations among Genes, Organisms, and Areas. Systematic Biology 43 1, p. 58-77.

[24] Harel, D., Tarjan, R. E., 1984. Fast Algorithms for Finding Nearest Common Ancestors. SIAM Journal on Computing 13 2, p. 338-355.

[25] Berkman, O., Galil, Z., Schieber, B., Vishkin, U., 1989. "Highly Parallelizable Problems," in Proceedings of the twenty-first annual ACM symposium on Theory of computing, p. 309-319.

[26] Bender, M., Farach-Colton, M., 2000. The Lca Problem Revisited. LATIN 2000: Theoretical Informatics, p. 88-94.

[27] Page, R. D. M., Charleston, M. A., 1997. From Gene to Organismal Phylogeny: Reconciled Trees and the Gene Tree/Species Tree Problem. Molecular phylogenetics and evolution 7 2, p. 231-240.

[28] Berglund-Sonnhammer, A. C., Steffansson, P., Betts, M. J., Liberles, D. A., 2006. Optimal Gene Trees from Sequences and Species Trees Using a Soft Interpretation of Parsimony. Journal of molecular evolution 63 2, p. 240-250.

[29] Chen, K., Durand, D., Farach-Colton, M., 2000. Notung: A Program for Dating Gene Duplications and Optimizing Gene Family Trees. Journal of Computational Biology 7 3-4, p. 429-447.

[30] Durand, D., Halldórsson, B. V., Vernot, B., 2006. A Hybrid Micro-Macroevolutionary Approach to Gene Tree Reconstruction. Journal of Computational Biology 13 2, p. 320-335.

[31] Benjamin, V., Stolzer, M., Goldman, A., Durand, D., 2008. Reconciliation with Non-Binary Species Trees. Journal of Computational Biology 15 8, p. 981-1006.

[32] Danicic, D., Durand, D., Goldman, A., Stolzer, M., Benjamin, V., 2008. Notung 2.6: A Manual: Durand Lab.

[33] ElMaraghy, H. A., 1993. Evolution and Future Perspectives of CAPP. CIRP Annals - Manufacturing Technology 42 2, p. 739-751.