MODELING MANUFACTURING CELLS USING PRINCIPLES OF REENGINEERING AND COMPONENT CLUSTERS

Rafael d'Ávila
Federal University of Rio Grande do Sul (UFRGS), Brazil
rafaeld@ieee.org

This paper presents a framework to support the analysis of manufacturing processes, providing the correct understanding concerning information and material flows. In addition, this analysis is combined with design structure matrixes based on components to consider better arrangements to the manufacturing process in cells, intending to provide high performance capabilities. To demonstrate the framework an example has been carried out in modeling the manufacturing of a control pack unit, which is a part of an airplane climate control system.

1. INTRODUCTION

Considering the benefits of modular products, many manufacturing companies are undertaking critical analysis and redesign of their productive process and manufacturing organization.

Understanding a product architecture as the arrangement of functional elements into physical chunks that compose the building blocks for a product, studies have shown that innovations in this subject represent a source of competitive advantage for product research and development enterprises (Ulrich; Eppinger, 2000 and Henderson; Clark, 1990).

The modularity of product architecture is inversely proportional to the modules interaction, and can be adopted with strategically interests. The development of modular products requires full identification of highly interactive groups of elements and clustering them into product modules or productive cells.

This paper presents a robust framework to identify, model, optimize and analyze component clusters, intending to provide correct understanding about relationships between product components and optimized arrangements to the manufacturing process in cells based in component clusters. This is demonstrated with an applied study analyzing a control pack unit, a subsystem of an airplane air conditioning system.

2. PROCESS MODELING USING IDEF0

The IDEF (Integration Definition for Function Modeling) pattern its provided to design, to model and to understand the set of activities that compose a specific process, as well as data that support connectivity between these activities (Colquhoun, 1993). The IDEF was derived from a well-established graphical language know as the Structured Analysis and Design Technique (SADT). The literature about IDEF reports sixteen methods, which provide a high visibility of manufacturing and business process (Aguilar-Savén, 2003).

In this paper the IDEF0 is used, which represents hierarchically process decomposed in activities or functions, through ICOM (Input, Control, Output, Mechanism) diagrams using a simple and efficient notation, as shown in figure 1.

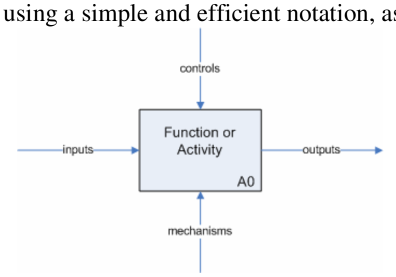

Figure 1 – ICOM Model (Aguilar-Savén, 2003)

As can be observed in figure 1, each function or activity is described by a rectangular box that can be decomposed in many levels of hierarchy. The arrows defining “Controls” refer to all necessary conditions to the activity or function operation, generating expected outputs after processing the inputs. On the other hand, the arrows defining “Mechanisms” indicate the executor of the activity, which can be persons, machines, equipments, or even other organizations.

Thus, IDEF0 provides powerful means of analysis and development for the manufacturing enterprise, establishing the scope of analysis either for a particular functional analysis or for future analyses from another system's perspective. As an analytical tool, IDEF0 assists the modeler in identifying the performed functions and what is needed to perform them.

3. DESIGN STRUCTURE MATRIX (DSM)

3.1 General

The design structure matrix (DSM) is a popular analysis tool for system modeling and representation, especially for purposes of integration and decomposition (Browning, 2001). A DSM displays the relationships established between components of a system in a compact and visual format.

Based on information extracted from IDEF0 modeling, it’s possible to decompose data in a graph that represents precedence and connectivity relations between different mapped activities, through information flows. An information flow (t_{ij}) is represented by an arc and an activity by a node on graph, as shown in figure 2.

Figure 2 – Information flow

Thus, the activity t_i represents the operation time required by activity i, what indicates that each information flow t_{ij} is defined by an activity pair (i,j). The graph originated from relations between activities can be expressed as $P(V, A)$, where V is the set of activities and A the set of information flows in the graph.

Considering K as a graph with a vertex-set $\{v_1, v_2, v_3, v_4,...,v_n\}$, then the adjacency matrix of K is the n x n matrix $G(K) = (g_{ij})$, where g_{ij} is the number of edges joining g_i and g_j. There are various other matrixes associated with K. For example, if the edges of K are $\{e_1, e_2, e_3, e_4,...,e_m\}$, then the binary incidence matrix of K is the n x m matrix $A(K) = (a_{ij})$ (GOULD, 1988).

Given a binary activity-to-activity matrix $A(K) = \lfloor a_{ij} \rfloor_{n \times m}$, where

$a_{ij} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$, adopting 1 if activity i is used to process activity j and 0 otherwise, (1)

the most desired decomposition of matrix A would be into mutually separable matrices $A_1, A_2, A_3, A_4,..., A_t$. Some algorithms are used to transform an initial matrix (1) into a more structured form by the permutation of rows and columns, aiming to rearrange the rows and columns of the matrix in order to place the non-zero elements in sub-matrices, associated with a cluster. An example of a DSM is showed in figure 3.

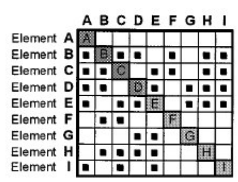

Figure 3 – Example DSM (Browning, 2001)

Summarizing, an off-diagonal mark in a DSM signifies the dependency of one element on another. Reading across a row reveals what other elements the element in that row provides to (output sinks); otherwise, scanning down a column reveals what other elements the element in that column depends on (input sources). (Browning, 2001).

3.2 Component-based DSM

The component-based DSM is a static representation of system elements existing simultaneously, such as components of a given product architecture or groups in an organization. Product architecture is the arrangement of functional elements into physical chunks that compose the building blocks for a product or a family of products (Ulrich; Eppinger, 2000). Many studies have shown that innovative product architectures represent a source of competitive advantage for product research and development enterprises (Henderson; Clark, 1990).

An essential prerequisite for innovation is understanding, which can be developed and increased through the use of models that highlight the interfaces or interactions between system elements. A DSM can represent a product architecture in terms of the relationships between its constituent components. In general, the system engineering exercise to obtain a correct understanding involves the following three steps:

a) decompose the system into elements;
b) document and understand all interactions or integrations between the elements;
c) analyze potential reintegration of the elements into clusters;

Integration analysis by clustering off-diagonal elements and reordering rows and columns of the DSM can provide new insights into system integration and decomposition. However, clustering requires some considerations. The foremost objective of clustering is to maximize interactions between elements within clusters while minimizing interactions between different clusters (Sanchez; Mahoney, 1997). It has also been suggested to minimize the size of the clusters (Altus; Kroo; Gages, 1995). Also it may be useful to allow for some overlapping of clusters, for example, recognizing certain elements in more than one cluster.

Clusters algorithms are very helpful in integration analysis. By reordering rows and columns, a clustering algorithm seeks a DSM configuration that optimizes an objective function. Several algorithms and heuristics have been offered (Fernandez, 1998). After clustering analysis, any exogenous interactions to the cluster should be noticed as interfaces, requiring special attention and verification. No single clustering approach is the best solution, but visual inspection and manipulation are often adequate for small or sparse matrices.

4. PURPOSED MODEL

The purposed model and its tools provides a complete framework to identify, model, optimize and analyze component clusters, originated after a process reengineering project team has identified and modeled the manufacturing process. This paper is concerned about the use of IDEF0 methods to identify components, the DSM to model, clustering algorithms to optimize data and qualitative analysis to evaluate results obtained using the framework. The purposed model is shown in figure 4.

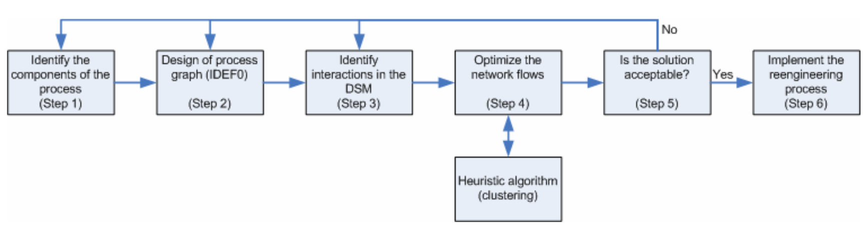

Figure 4 – Purposed Model

In the first step the team must to identify core components of the process in a manufacturing enterprise. This information can be frequently revised and involves data about time, cost and information flows. In the step two is designed all the activities using an IDEF0 model and linked one to each other that have some relationship. After designed the IDEF0 model, in step three this model is transposed to a DSM and all relationships between the elements are marked in the matrix. After this, in step four a heuristic clustering algorithm is applied in the DSM to optimize the arrangement of information flows between elements previously marked in the structure. Finally, is necessary to evaluate if the obtained solution is considered acceptable, in step five. If the solution is not acceptable, is recommended to avail the enter parameters since the first step and to execute various simulations. Else, if solution is considered usable, the team may implement the reengineering process analyzed.

The clustering algorithm used in step four of this purposed model is based on the work of Fernandez (1998), where a coordination cost function can be developed to evaluate different clustering arrangements within the DSM. For each team in the DSM the algorithm calculates a coordination cost. Then, the sum of all coordination costs for each team gives a total coordination cost. Equations 2 and 3 show the coordination cost from a team *i*.

If both teams *i* and *j* are in any cluster *k*, then

$$\text{Coordination Cost}(team_i) = \sum_{j=1}^{size} \left(DSM(i,j) + DSM(j,i)\right) * \sum_{k=1}^{Cl} cl\_size(k)^{pow\_cc}, \quad (2)$$

else, (if no *k* cluster contains both *i* and *j*, the entire DSM acts as a cluster)

$$\text{Coordination Cost}(team_i) = \sum_{j=1}^{size} \left(DSM(i,j) + DSM(j,i)\right) * size^{pow\_cc}, \quad (3)$$

where

size is the size (or the number of teams) of the DSM

DSM(i,j) is the value of the interaction or dependency between teams *i* and *j*, noting that when i=j, DSM(i,j)=0 and DSM(j,i)=0, because the diagonal entries in the DSM don't list interactions.

Cl is the maximum number of clusters or system teams that the algorithm explores. In the algorithm Cl is equal to *size*.

cl_size(k) is the number of teams contained in cluster *k*.

pow_cc is a parameter that controls the type of penalty assigned to the size of the cluster in the coordination cost.

Equation 4 is the expression for the total coordination cost and is the objective function that the algorithm attempts to minimize.

$$\text{Total Coordination Cost} = \sum_{i=1}^{size} \text{Coordination Cost}(team_i). \quad (4)$$

The algorithm proceeds as follows. Initially, there are as many clusters as there are DSM elements. Then, the algorithm randomly selects an element and calculates a bid from clusters. The highest bid is chosen, and if there is an improvement in the total coordination cost, the task is added to the bidding cluster. This process continues until, after several attempts, there is no further improvement in the coordination cost (Fernandez, 1998).

5. APPLYING THE PURPOSED MODEL: A STUDY CASE

5.1 General

Intending to demonstrate the operation and to validate the purposed model was used for analysis a control pack unit, a subsystem of an airplane conditioning system (ACS). This control pack unit is composed by many different components that were enumerated, to attend the first step of the framework. After this, as recommended in second step, were identified the relationship flows by each one with others and designed an IDEF0 model, as shown in figure 5.

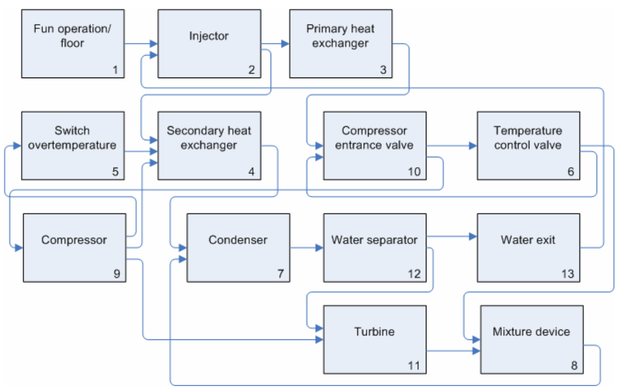

Figure 5 – IDEF0 model for a control pack unit of an airplane ACS.

This IDEF0 model was used to transpose all the identified relationships between components into a DSM matrix, as described in the third step of the purposed model. This transcription of IDEF0 into a DSM is shown in figure 6.

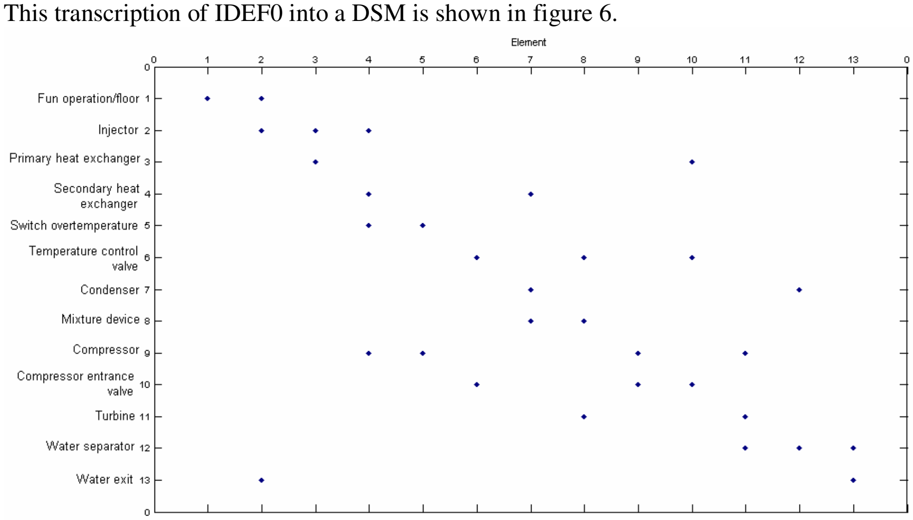

Figure 6 – DSM Matrix identifying dependencies between elements

The clustering algorithm purposed by Fernandez (1998) and described in section 4 was applied to the generated DSM, intending to optimize and rearrange components through the measurement of coordination costs. The algorithm was applied using a MATLAB toolbox written by Thebeau (2001) and the clustered version of the DSM is shown in figure 7.

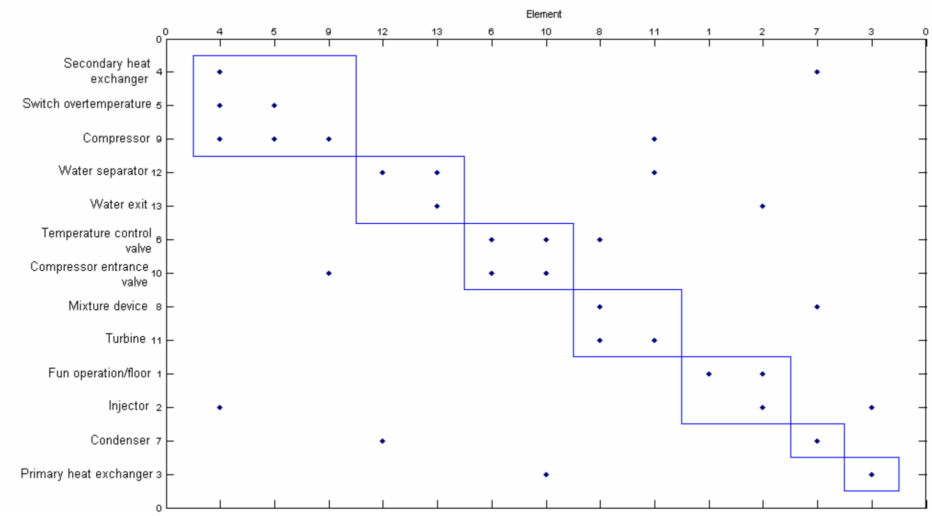

Figure 7 – Clustered DSM Matrix

The obtained results were acceptable and reasonable, considering that elements with strong interactions, demonstrated by relationships designed in IDEF0 and after identified in DSM, were put together in the same clusters. To reach this solution, a total coordination cost was calculated and when, after several attempts, there was no further improvement in the coordination cost, the solution was considered optimal, as shown in figure 8.

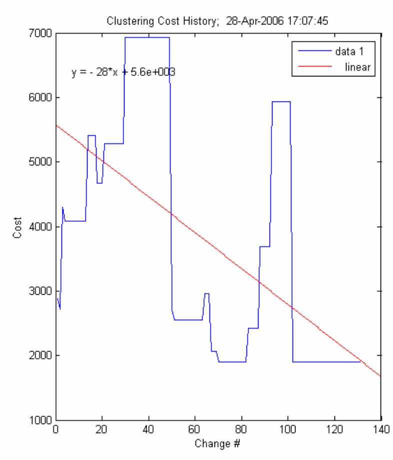

Figure 8 – Clustering cost history

6. CONCLUSIONS

This paper started by reviewing the IDEF0 modeling tool and DSM literature related to clustering and product modularity. Then, was presented a framework to support the identifying, modeling and optimization of manufacturing processes. Providing the correct understanding about information and material flows, the identified process is converted in a component design structure matrix, intending to obtain better arrangements to the manufacturing process in cells.

To demonstrate the framework an example using a real-world problem has been carried out in modeling the manufacturing of a control pack unit, a subsystem of an airplane climate control system. The DSM is a powerful tool for representing product architectures, allowing for the analysis and development of modular products, by clustering the DSM.

As can be observed, decomposition of a product manufacturing into clusters of components can reduce the complexity of the process, decrease the size of the design product team and may have a real impact on its performance. As a result of decomposition, the design product cycle can be reduced and there is a simplification of scheduling and management of manufacturing process.

7. REFERENCES

1. Aguilar-Savén, R. "Business process modeling: review and framework", International Journal of Production Economics, vol. 90, pp. 129-149, 2003.
2. Altus SS, Kroo IM, Gage PJ, "A genetic algorithm for scheduling and decomposition of multidisciplinary design problems," in Proc.21st ASME Design Automation Conf., Boston, MA, 1995.
3. Fernandez CIG, "Integration analysis of product architecture to support effective team co-location", Master Thesis, Massachusetts Institute of Technology, 1998.
4. Henderson RM, Clark KB, "Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms," Administ. Sci. Quart., vol. 35, pp. 9–30, 1990.
5. Sanchez R, Mahoney JT, "Modularity, flexibility, and knowledge management in product and organization design," IEEE Eng. Manage. Rev., pp. 50–61, 1997.
6. Thebeau, RE, "Knowledge Management of System Interfaces and Interactions for Product Development Processes", MIT Masters Thesis, System Design & Management Program, Feb. 2001.
7. Ulrich KT, Eppinger SD, Product Design and Development, 2nd ed. New York: McGraw-Hill, 2000.