Maintenance Modularity Optimization using Clustering Algorithm: Application

Ngnassi Djami Aslain Brisco[#1], Nzié Wolfgang[2], DokaYamigno Serge[3]

[1]*PhD Candidate, National Advanced School of Agro-Industrial Sciences, University of Ngaoundere, Ngaoundere, Cameroon*

[2]*Department of Basic Sciences and Techniques for the Engineer, EGCIM, University of Ngaoundere, Ngaoundere, Cameroon*

[3]*Department of Physics, Faculty of Science, University of Ngaoundere, Ngaoundere, Cameroon*

Abstract

For a given product, knowledge of its modular architecture has the advantage of facilitating maintenance. Indeed, during a maintenance problem, we will not act on all the product except on the module concerned and we would also gain time to detect the defect observed. Due to the major difficulties of maintenance of complex topology systems, modular design is therefore an asset for companies.On the other hand, the modular design although with the advantages of standardization and reconfigurability is enormously expensive, which could require more resources, more time and more work designers. This has consequences for product maintenance because the more robust or dense product modules, the more expensive the maintenance. This paper will therefore be used to deploy a clustering algorithm that will be subdivided into several steps in order to evaluate the coupling cost of the product modules in order to identify the most constraining modules for the design and consequently for maintenance.

Keywords — *Maintenance, Modularity, Clustering, Design Structure Matrix, Module Strength Indicator.*

I. INTRODUCTION

Product modeling is fundamental to the identification of a modular architecture. Indeed, this modeling leads to a global relationship matrix between all the components of a product. Thanks to this relationship matrix, which is strongly correlated with the first step of the clustering algorithm, which consists of deploying a grouping chronology of all the components in modules, this results in an optimal modular architecture. The modular architecture is thus obtained and a good topology of the modules being defined, by means of one of the metrics (Module Strength Indicator) proposed by Ghassen [1] which is the second step of the clustering algorithm, the modularity will be measured in order to identify the module that provides greater internal cohesion and easier maintenance. At the end of this measure of modularity, the third step of the clustering algorithm will consist in evaluating the coupling costs between the product modules in order to identify the modules that attract the most attention from the designers.

II. METHODOLOGY

A. Product modeling

The product components and their relationships are represented by a link graph, G = (C, R) where each vertex represents a component, and each pair of two components called "arc" represents the information related to the relationships.

Let C= $(C_1, C_2, C_3, \ldots, C_n)$ be the set of all the components of the cardinality product n.

Let R= $(R_1, R_2, R_3, \ldots, R_m)$ be the set of all the relationships between the components of the cardinality product m.

For each pair of components of the graph, we assign it a value which is the sum of the parameters representing the disassembly. So the result will be presented in the form of a relationship matrix. The relationship matrix is defined by an element $c_{i,j}$ such that:

$$c_{i,j} = \begin{cases} \sum_k \left(TC + TCO + OD + NDD + QO + ER\right), \\ if\ c_i\ and\ c_j\ have\ a\ relationship \\ 0, \qquad\qquad else \end{cases} \quad (1)$$

with :

- **TC** : Types of Contacts ;
- **TCO** : Types of Combinations ;
- **OD** : Disassembly tools ;
- **NDD**: Number of disassembly directions;
- **QO** : Operator qualifications ;
- **ER** : Required equipment ;
- **k** : index of the connection between the componentsc_i and c_j.

B. Clustering algorithm and choice of a good topology

a) First step of algorithm

Clustering is a process that partitions a set of data into meaningful subclasses (clusters).

This step makes it possible to rearrange consecutively the rows and columns of the matrix modeling the relationships between the components of a product, according to a similarity index until the diagonal blocks are obtained. Thus all components that are in maximal interaction are grouped together to form modules.

Optimizing clustering therefore consists of maximizing intra-class (components) similarity and minimizing interclass (modules) similarity.

To find the number of modules, we use the formula proposed by Ericsson and Erixon[2]:

$$0,5\sqrt{NCP} \leq NM \leq \sqrt{NCP} \quad (2)$$

with:

- **NCP**: Number of Product Components;
- **NM**: Number of Product Modules.

This grouping is done in several stages:

1.Generate a number of possible modules according to equation 2.
2.Add the entries for each column in the relationship matrix.
3.Choose components with the highest link strengths and assign them to a module.
4.Choose a column from those that correspond to the components of the previous step and draw a vertical line.
5. For each input (value) other than 0 cut by the vertical line, draw a horizontal line.
6. Assign the components to the module to which the selected column component belongs so as to maximize the component relationships of the same module and to minimize the relationships between the different module components.
7. Transform the array by removing the components assigned to a module.
8. Repeat steps 6-7 until each component is assigned to a module.

At the end of the grouping obtained, we will have an optimal grouping of the modules if each module checks the relation 3 (Ericsson and Erixon) [2].

$$\frac{NCP}{NMOYCM} \leq NCM \leq \frac{NCP}{NMICM} \quad (3)$$

with:

- **NMOYCM**: Average Number of Components in a Module;
- **NMICM**: Minimum number of Components in a Module.

b) Choosing a good module topology

The choice of a good topology of the modules is strongly correlated with the sequences or ranges of disassembly of the product. Indeed, with the disassembly process, we can identify components that can be retrieved in serie. It can be seen from this moment that during a maintenance operation using disassembly, one can end up dismantling several components one after one. These components therefore have a serie-type operating mode (the malfunction of one of the components systematically affects the others). The other components not coming from this mode of operation (parallel type) can therefore be disassembled independently of one another. Thus, at the end of the disassembly ranges, a module topology is defined.

To obtain the optimal disassembly sequence (leading to complete disassembly), we are inspired by the methods developed by Bourjault [3], Subramani and Dewhurst [4], Lee and Kumana [5], Srinivasan and Gadh [6], Jabbour and Mascle [7], Kuo et al.[8], Srinivasan and Gadh [9], Gungor and Gupta [10], Moore et al.[11], Kongar and Gupta [12], Failli and Dini [13], Srinivasan and Gadh, 2002 [14], Torres et al.[15], Wang et al.[16], Lambert and Gupta [17], Kongar and Gupta, 2006 [18], Giudice and Fargione [19], Tseng et al.[20], Tripathi et al.[21], Zhang and Zhang [22], Xue et al. [23].

The disassembly sequence generated by the algorithm that we propose highlights the order of disassembly of the product modules, then the components of a module. For the generation of the disassembly sequence, three input data are needed:

- the matrix of precedence between the components;
- the disassembly directions of the modules;
- the matrix of precedence between the modules.

The matrix of precedence between the components is defined from the graph of the connections and the modular architecture of the product. An element p_{ij} of the matrix is such that:

$$p_{ij} = \begin{cases} 1, if \ \ the\ component\ i\ is\ disassembled\ before\ the\ component\ \ j \\ 0, if\ \ the\ component\ i\ is\ not\ deassembed\ before\ the\ component\ \ j \\ x, if\ \ no\ \ precedence\ constraint\ exists\ between\ components\ i\ and\ \ j \end{cases}$$

$$this\ notaion\ can\ take\ the\ value\ 0\ or\ 1 \quad (4)$$

The entries of the main diagonal of the precedence matrix are equal to 0.

Finally, the resulting disassembly range can be represented in different ways (link graph, adjacency graph, ...). We propose to represent it as a disassembly tree for a more complete view of the modules and components to disassemble and taking into account the order of disassembly.

c) Second step of algorithm

In this step, we will define a measure of modularity.

When we talk about modularity, we are entitled to ask the question: how modular is a product? In order to quantify modularity, several measures and metrics

have been developed (Newcomb et al. [24]), (Blackenfelt [25]), (Guo and Gershenson [26]), (Holtta et al. [27]). The developed measures are then used to compare different architectures of the same product. Whietfield et al. [28] use a metric similar to that of Guo with more relevant weights (the number of interactions available). This measure of modularity is the MSI indicator for Module Strength Indicator. The MSI applies to a module and consists of two parts (Equations 5 and 6).

$$MSI_i = \frac{\sum_{i=n_1}^{n_2}\sum_{j=n_1}^{n_2} DSM(i,j)}{(n_2-n_1+1)^2-(n_2-n_1+1)} \quad (5)$$

$$MSI_e = \frac{\sum_{i=1}^{n_1-1}\sum_{j=n_1}^{n_2}\left(DSM(i,j)+DSM(j,i)\right)}{2\times\left((n_1-1)\times(n_2-n_1+1)\right)} + \frac{\sum_{i=n_2+1}^{N}\sum_{j=n_1}^{n_2}\left(DSM(i,j)+DSM(j,i)\right)}{2\times\left((N-n_2)\times(n_2-n_1+1)\right)} \quad (6)$$

$$MSI = MSI_i - MSI_e \quad (7)$$

With:

- ***DSM (i, j)***: Value of the interaction between the element *i* and *j* in the *DSM*;
- n_1:Index of the first element of the module;
- n_2:Index of the last element of the module;
- N :The total number of elements;
- MSI_i and MSI_e represent respectively the density of the interactions inside and outside a module.

The MSI indicator then makes it possible to measure the degree of modularity of each module by comparing the interactions inside and outside the module. This indicator allows the system architect to evaluate the modularity of each module and to simulate several configurations. This last indicator seems to be the most comprehensive for evaluating the modularity of an architecture (Ghassen [1]). A last remark concerning this metric: the authors propose indices n_1 and n_2 and affirm that the number of elements is $n_2 - n_1 + 1$, for this affirmation to be true it is necessary that $n_1$is the index of the last element in the previous module (Ghassen[1] and Eric [29]).

d) Third step of algorithm

The idea behind this step is to evaluate an objective function that identifies the modules that require the most attention from the designer for a given product. We will focus here on a more generic function, which is the "Total Coupling Cost", because this function is independent of the domain being studied.

The algorithm is built around an "objective function" called the total cost of coupling. This function aims to match our vision of the architecture of a system to the mathematical formulation it uses. Thus, the goal is to transcribe the following observations into a mathematical formulation:

• When an interaction links two elements belonging to two different modules, then the cost of specifying / defining this interaction depends on the size of the two modules that it brings together. This remark is at the origin of the evolution of the coupling cost proposed later by Ghassen[1] and by Eric [29] (see equation 9)

• To obtain the final expression of the total cost of coupling: like Idicula's proposal [30], Ghassen [1] and Eric [29] constructed two complementary functions of the coupling cost: one to characterize the interactions internal to one module and the other to characterize the external interactions to this module.

For each interaction in the DSM matrix, the Idicula algorithm calculates a coupling cost. Then, the sum of all the coupling costs gives the total cost of coupling. Equations 8 and 9 show the coupling costs for an interaction.

If the two elements i and j belong to the same module k (M_k), then:

$$Coupling\ Cost(i,j) = \left(DSM(i,j)+DSM(j,i)\right)\times\left(size\left(M_k\right)\right)^2 \quad (8)$$

Otherwise, if no module contains the elements i and j, then there are two modules $M_{i'}$ and $M_{j'}$ .which contain respectively the elements i and j. The cost of the interaction between the elements i and j is then written:

$$Coupling\ Cost(i,j) = \left(DSM(i,j)+DSM(j,i)\right)\times\left(size\left(M_k\right)+size\left(M_{i'}\right)+size\left(M_{j'}\right)\right)^2 \quad (9)$$

Equation 10 summarizes the expression of the total cost of coupling.

$$Total\ Coupling\ Cost = \sum_i\sum_j Coupling\ Cost(i,j) \quad (10)$$

The previous three formulations of the coupling costs can be rewritten by adopting the modules as a reference. We can then formulate a coupling cost function, either internal or external to a module k, M_k(Equations 11 and 12).

$$Internal\ Coupling\ Cost\left(M_k\right) = \sum_{i\in M_k}\sum_{j\in M_k, j\neq i}\left(DSM(i,j)+DSM(j,i)\right)\times\left(size\left(M_k\right)\right)^2 \quad (11)$$

$$External\ Coupling\ Cost\left(M_k\right) = \sum_{i\in M_k}\sum_{j\notin M_k, j\in M_n}\left[\begin{array}{l}\left(DSM\left(i,j\right)+DSM\left(j,i\right)\right)\times \\ \left(size\left(DSM\right)+size\left(M_k\right)+size\left(M_n\right)\right)^2\end{array}\right] \quad (12)$$

Then the Total Cost of Coupling can be written in the form presented in Equation 13.

$$Total\ Coupling\ Cost = \sum_{M_k}^{Modules}\left(\begin{array}{l}Internal\ Coupling\ Cost\left(M_k\right)+ \\ External\ Coupling\ Cost\left(M_k\right)\end{array}\right) \quad (13)$$

III. RESULTS

This part illustrates the implementation of the approach presented above which consists in defining a good modular topology, evaluating the degree of modularity (the *MSI*) and evaluating the coupling cost of the modules, on the soy roaster.

Remember, however, that the exploded view, nomenclature, and linkage of the soybean roaster are given in Figure **1**, Table **1**, and Table **2**.

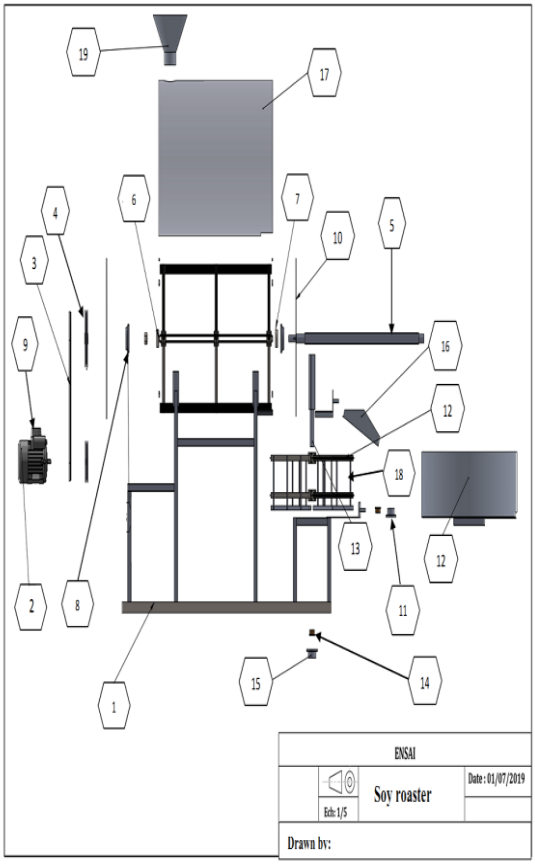

Figure 1: Exploded view of the solution

.

Table 1 : Component Legend.

Reference of components	Component names
1	Frame
2	Engine
3	Belt
4	Disc 2 (pulley)
5	Main axis
6	Main shaft bearing 1
7	Main shaft bearing 2
8	Engine side bearing bracket
9	Motor disk
10	hood
11	bowl-sidebearing support
12	Bowl
13	Secondary axis
14	Secondary axis bearing
15	secondary axle bearing support
16	output hopper
17	Main cylinder
18	Scraper
19	Inlet hopper

Table 2: Evaluation of the bonds between the components.

Connexion	TC	OD	NDD	TCO	QO	ER	Value
1--2	6	3	5	4	3	2	23
2--9	4	1	4	3	1	1	14
3--9	4	3	5	2	2	2	18
3--4	4	1	4	3	1	1	14
4--5	4	3	5	4	3	2	21
5--6	4	3	5	2	3	2	19
5--7	4	3	5	2	3	2	19
6--8	6	1	5	2	2	2	18
7--11	6	1	5	2	2	2	18
11--10	4	3	5	4	3	2	21
17--10	6	3	5	2	2	2	20
17--8	4	3	5	4	3	2	21
17--1	6	3	5	4	3	2	23
17--16	6	3	5	4	3	2	23
1--12	6	3	5	4	3	2	23
15--14	4	3	4	2	2	2	17
12--15	4	3	5	4	3	2	21
14--13	4	3	5	4	3	2	21
18--13	4	3	5	4	3	2	21
19--17	4	3	5	4	3	2	21

A. Product modeling

By executing the formula given by equation (2) which consists in generating all the cells of the relationship matrix of the system components and the function of the disassembly parameters selected for the soy roaster, we obtain the relationship matrix given by the figure **2**.

Components	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0	23										23					23		
2	23	0							14										
3			0	14					18										
4			14	0	21														
5				21	0	19	19												
6					19	0		18											
7					19		0				18								
8						18		0									21		
9		14	18						0										
10										0	21						20		
11							18			21	0								
12	23											0			21				
13													0	21				21	
14													21	0	17				
15												21		17	0				
16																0	23		
17	23							21		20						23	0		21
18													21					0	
19																	21		0
TOTAL	69	37	32	35	59	37	37	39	32	41	39	44	42	38	38	23	108	21	21

Figure 2: Relationship Matrix of the Soy Roaster

B. First step of clustering algorithm and module topology

a) First step of clustering

The application of the Ericsson and Erixon formula (1999) to calculate the number of possible modules allowed us to note that the number of modules that can be formed is 4 (see equation 2).

$$NM \leq \sqrt{19} \tag{14}$$

So according to equation 2 developed by Ericsson and Erixon [4], the definition of a good architecture of the modules must verify equation 15.

$$3 \leq NCM \leq 19 \tag{15}$$

with:

- NMCM = 1;
- NMOYCM =5.

The components of the soy roaster are grouped into modules, applying the component grouping algorithm in modules proposed above. The addition of the entries of each column of the relationship matrix (previously obtained), found that the components 1, 5, 12 and 17 have the highest link intensities. We assign them to a module each. Then we assign the rest of the components to a module, so as to maximize the interactions between the components of the same module and minimize the interactions between the components of different modules.

The modular decomposition of the soy roaster is then obtained in figure **3**.

Thus the module $\boldsymbol{M_1}$is formed of the components 8; 10; 16; 17 and 19. The module $\boldsymbol{M_2}$ is formed of the components 12; 13; 14; 15 and 18. The module $\boldsymbol{M_3}$ is formed of the components 3; 4; 5; 6; 7 and 11. The module $\boldsymbol{M_4}$is formed of the components 1; 2 and 9.

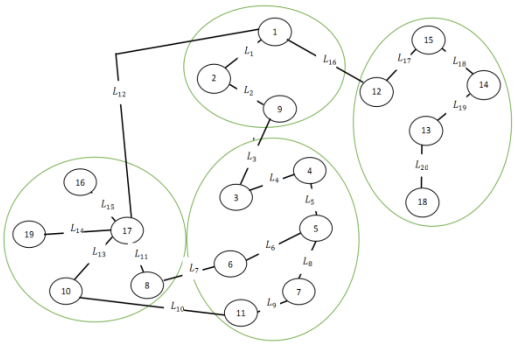

Figure 3: Grouping into modules of the soy roaster

It can be seen that the components of each module verify equation 3, so the grouping of the modules is optimal.

b) Choosing a good module topology

We will generate the disassembly sequence (range) of the modules in order to deduce the possible couplings between the different modules of the soy roaster.

1) Matrix of precedence of components

Figure **4** illustrates the precedence matrix for the soy roaster.

		M_1					M_2						M_3					M_4		
		8	10	16	17	19	11	12	13	14	15	18	3	4	5	6	7	1	2	9
M_1	8	0	0	x	x	x	x	x	x	x	x	x	0	x	x	x	x	0	x	x
	10	1	0	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	16	x	x	0	1	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	17	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	19	x	x	x	1	0	x	x	0	x	x	x	x	x	x	x	x	x	x	x
M_2	11	x	1	1	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	12	x	x	x	x	x	x	0	x	x	x	0	x	x	x	x	x	x	x	x
	13	x	x	x	x	1	x	x	0	x	x	x	x	x	x	x	x	1	x	x
	14	x	x	x	x	x	0	x	x	0	x	x	x	x	x	x	x	x	x	x
	15	x	x	x	x	x	x	1	x	x	0	0	x	x	x	x	x	1	x	x
	18	x	x	x	x	x	x	x	x	x	1	0	x	x	x	x	x	x	x	x
M_3	3	1	x	x	x	x	x	x	x	x	x	x	0	x	x	0	x	x	x	x
	4	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	0	x
	5	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	x	x	x	x
	6	x	x	x	x	x	x	x	x	x	x	x	1	0	1	0	x	x	x	x
	7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	x	x
M_4	1	0	x	x	x	x	x	x	0	x	0	x	x	x	x	x	0	0	x	0
	2	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	0	x
	9	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	x	0

Figure 4: Matrix of precedence of components of the soy roaster.

2) Changing disassembly direction of the modules

The component precedence matrix is used to derive the disassembly levels (in Figure 6) to construct the disassembly tree. Since components belong to the same module, the value 0 is assigned instead of 1 (Figure 5).

		M_1					M_2						M_3					M_4		
		8	10	16	17	19	11	12	13	14	15	18	3	4	5	6	7	1	2	9
M_1	8	0	0	x	x	x	x	x	x	x	x	x	0	x	x	x	x	0	x	x
	10	0	0	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	16	x	x	0	0	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	17	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	19	x	x	x	0	0	x	x	0	x	x	x	x	x	x	x	x	x	x	x
M_2	11	x	1	1	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
	12	x	x	x	x	x	x	0	x	x	x	0	x	x	x	x	x	x	x	x
	13	x	x	x	x	1	x	x	0	x	x	x	x	x	x	x	x	1	x	x
	14	x	x	x	x	x	0	x	x	0	x	x	x	x	x	x	x	x	x	x
	15	x	x	x	x	x	x	0	x	x	0	0	x	x	x	x	x	1	x	x
	18	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x
M_3	3	1	x	x	x	x	x	x	x	x	x	x	0	x	x	0	x	x	x	x
	4	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	0	x
	5	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x
	6	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x
	7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	x	x
M_4	1	0	x	x	x	x	x	x	0	x	0	x	x	x	x	x	0	0	x	0
	2	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	0	x
	9	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	0

Figure 5: Modified precedence matrix of soy roaster.

3) Module precedence matrix and disassembly sequence

$$M_1\ M_2\ M_3\ M_4 \text{ Level}$$

$$M_{ij} = \begin{matrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} 0 \\ 2 \\ 2 \\ 1 \end{matrix}$$

Figure 6: Module precedence matrix

The combination of the results of the hierarchical analysis of the modules constituting the soy roaster and the matrix of precedence of the constituent components makes it possible to generate the disassembly range (Figure 7).

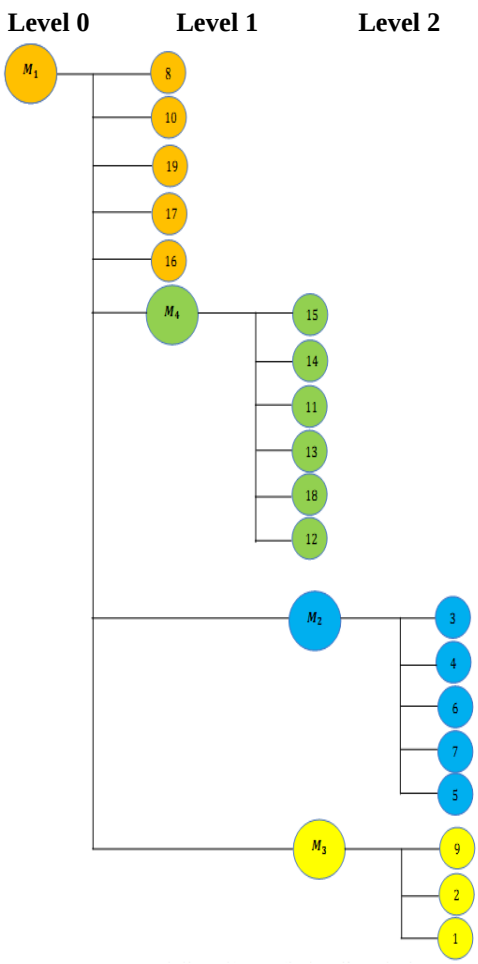

Figure 7: Complete disassembly range of the soy roaster.

This tree presents the steps to follow for the disassembly of the product. The decomposition of the product in modules makes it possible to disassemble the components or modules in parallel, as shown for level 1, the components 8, 10, 19, 17, 16 and the module 4 are recovered in parallel. At level 2, the components 15, 14, 11, 13, 18, 12 and the modules 2 and 3 are recovered in parallel.

4) Topology of soybean roaster modules

According to the disassembly range obtained previously, there are three disassembly levels coupled in parallel (because by levels one intervenes on a category of modules).

At level 0, the module 1 is dismounted. At level 1, the module 4 is dismounted. At level 2, both modules 2 and 3 are dismounted, the two modules therefore have a series-type operating mode. It is strong from this observation that the topology of the roaster's modules is given in Figure **8**.

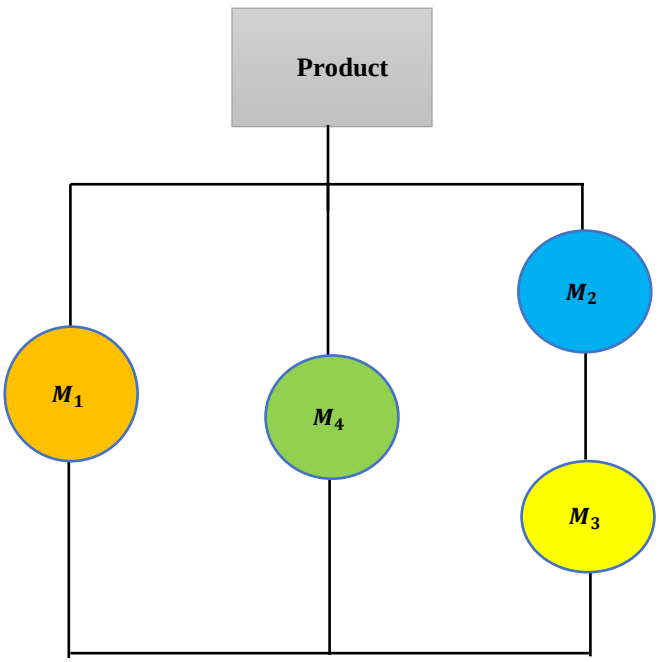

Figure 8: Topology of soybean roaster modules

C. Second step of algorithm

We will quantify the modularity of our soy roaster using the MSI (Module Strength Indicator).

This modularity indicator being a function of the DSM (Design Structure Matrix), we will first draw up the structural design matrix that derives directly from the matrix of relationships and the modular decomposition of the components of the soy roaster.

a) Structural design matrix

From Figure **2** (relationship matrix) and clustering of components into modules, the DSM of the soy roaster is given in Figure **9**.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}
C_1	0			21											18				
C_2		0		20		21													
C_3			0	23															
C_4	21	20	23	0	21												23		
C_5				21	0	19													
C_6		21			19	0										18			
C_7							0			21							23		
C_8								0	21		21								
C_9								21	0	17									
C_{10}							21		17	0									
C_{11}								21			0								
C_{12}												0	14						18
C_{13}												14	0	21					
C_{14}													21	0	19	19			
C_{15}	18													19	0				
C_{16}			18			18								19		0			
C_{17}				23			23										0	23	
C_{18}																	23	0	14
C_{19}												18						14	0

Figure 9: DSM from the soy roaster

This new codification proposed components of the soy roaster by the $C_i(1 \le i \le 19)$, is to better apply the measurement formula of the modularity.

B. Quantification of modularity

Our product is composed of four modules, we go for each module to measure the degree of modularity by comparing the interactions inside and outside the module.

Remember, however, that the measures of the interactions inside and outside the module are given by the relations 5 and 6.

For reasons of symmetry of our DSM, the relation (6) thus becomes:

$$MSI_e = \frac{\sum_{i=1}^{n_1-1}\sum_{j=n_1}^{n_2} 2DSM(i,j)}{2\times\left((n_1-1)\times(n_2-n_1+1)\right)} + \frac{\sum_{i=n_2+1}^{N}\sum_{j=n_1}^{n_2} 2DSM(i,j)}{2\times\left((N-n_2)\times(n_2-n_1+1)\right)} \quad (16)$$

The DSM of the considered module is thus given by the relation (17)

$$MSI = MSI_i - MSI_e \quad (17)$$

For our DSM, we have: **N = 19** components. The degree of modularity of the modules of the soy roaster is given by Table **3**.

Table 3: Degree of modularity of the modules of the soy roaster.

Modules		M_1	M_2	M_3	M_4
MSI	MSI_i	8,5	5,3333	7,3	12,3333
	MSI_e	1,1571	2,1875	2,1818	1,3333
	MSI	7,3429	3,1458	5,1182	**11**

In view of these results, it can be seen that the degree of modularity increases as the number of internal interactions in the module increases and the number of external interactions decreases. Module 4 has the highest interaction, so it provides good internal cohesion because the interactions within the module are all the more important as decreasing external interactions to the module. According to Hwai-En et al. [12], the higher the intensity of connection is strong inside a module, means that it is weak outside the module, which indicates from a maintenance point of view that it is easy to assembling and disassembling components in a module. This observation results in a high degree of close connection between the components of a module. As a result, Module 4 is the easiest module to maintain, and this observation made by Hwai-En et al. [12] is therefore relevant because it is clear that the module 4 is less complex with a smaller size than those of other modules. However, Module 2 has the weakest interaction, it provides a weak internal cohesion and therefore more difficult to maintain. It is clear that it is more complex with a larger size of 6 components. So the interactions of a module are a decreasing function of the size of the module (Figure **10**). On the other hand, it is clear that modules 1 and 3 are the same size. By just trusting this parameter, it would be difficult to know during a maintenance problem the module that can easily maintain. Thanks to the knowledge of the density of each module (MSI), it is found that the module 1 is denser than the module 3, therefore it is easier to operate on the module 1 than on the module 3.

Figure **10** reflects the evolution of the MSI modules.

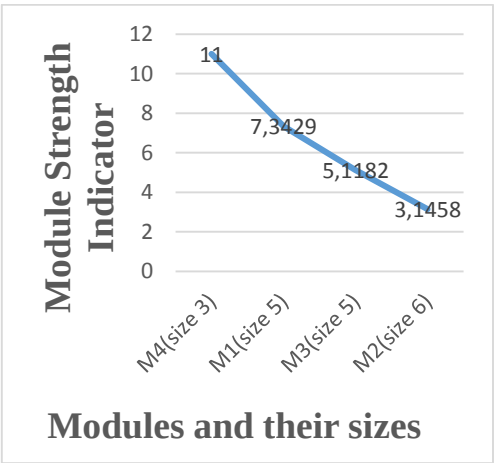

Figure 10: Evolution of the MSI modules

D. Third step of algorithm

Here we will evaluate the coupling costs of each module of the soy roaster and deduce the overall coupling cost (Table **4**).

We have:

Table 4: Coupling cost of each module of the soy roaster.

Modules		M_1	M_2	M_3	M_4
Coupling Cost	*Internal*	4250	5760	3650	0
	External	135810	68464	26244	666
	Module	**140060**	74224	29894	666

In view of these results, it is found that the module 1 has the highest coupling cost, so this module requires more attention, more resources, or more work designers. On the other hand, Module 4 has the lowest coupling cost, so this module requires less attention, less resources, or less designer work. The overall coupling cost of the soy roaster modules is therefore obtained. as following (Figure 11):

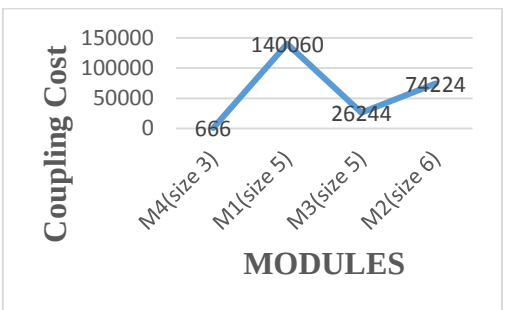

Figure 11: Evolution of the cost of the modules.

With regard to this graph, the coupling cost is hardly an increasing function of the size of the modules. In fact, the module 2 which is larger has a lower coupling cost than module 1 of smaller size than that of module 2. This means that the interactions of module 1 are stronger than those of module 1. module 2. On the other hand, the coupling cost is an increasing function of the size of the modules 4, 3 and 2 respectively. It can be seen from this moment that the coupling cost is at the same time a function of the size of the values of the interactions of the modules.

IV. CONCLUSIONS

This part was devoted to the implementation of a clustering algorithm in order to highlight the topology of the modules using the complete disassembly tree, to evaluate the degree of modularity of the modules by means of the MSI and to evaluate the coupling cost of the modules of the soy roaster. It turns out that knowledge of the topology of a system and its range of disassembly, allows us to save in time of intervention during a maintenance problem. Indeed, because each module performs a specific function, we now know which module (s) to intervene first and chronologically which component must be disassembled before the other. On the other hand, the evaluation of the degree of modularity of the various modules of the product allowed us to note that the interactions of a module increases when the size of the modules decreases, which constitutes an asset for the maintenance. Regarding the coupling cost of the modules, it turns out that the latter is not always an increasing function of the size of the modules, but also depends on the interactions of the module. However, when the size of a module is large or when the interactions of the module are strong, it becomes difficult and expensive to design because of its complexity. In addition, it is less easy to standardize. It is therefore important to reduce these coupling costs for easier design and maintenance.

DATA AVAILABILITY

The data needed to support these results are available in the text, specifically in Tables 1 and 2 and Figure 1.

CONFLICTS OF INTEREST

All authors declare that there are no conflicts of interest regarding the publication of this paper.

ACKNOWLEDGEMENTS

The authors of this document all residing in Cameroon, from the University of Ngaoundere, wish to thank the editorial team of SSRG International Journal of Industrial Engineering for the opportunity they gave him to publish the research paper. Thank you also to the Department of Mechanical Engineering of the National School of Agro-Industrial Sciences of the University of Ngaoundere for its contribution, in one way or another, to the final rendering of this paper.

REFERENCES

[1] H. Ghassen, "Towards a joint conception of the architectures of the product and the organization of the project within the framework of the System Engineering ", Ph.D. thesis, University of Franche-Comté, UFR Sciences and Techniques, Doctoral School Physical Sciences for the Engineer and Microtechniques, 249 pages, 2007.

[2] A. Ericsson and G. Erixon, "Controlling design variants: Modular product platforms ", (p. pp145). New York: ASME press, 1999.

[3] A. Bourjault, State Thesis, "Contribution of a methodological approach to automated assembly: automatic development of operating sequences ", University of Franche-Comté, 1984.

[4] A. K. Subramani, and P. Dewhurst, "Automatic Generation of Product Disassembly Sequences ", CIRP Annals- Manufacturing Tehnology, 40 (1), 115-118, 1991.

[5] Y. Lee, and S. Kumana," Individual and group disassembly sequence generation through freedom and interference spaces", Journal of Design and Manufacturing, 2, 143-154, 1992.

[6] H. Srinivasan, and R. Gadh, "A geometric algorithm for single selective disassembly using wave propagation abstraction ", Computer-Aided Design, 30 (8), 603-613, 1998.

[7] T. Jabbour and C. Mascle, A Database for the representation of assembly features in mechanical products, International Journal of Computational Geometry & Applications, 8 (5-6), 1998.

[8] T. Kuo, H. C. Zhang, and S. Huang," Disassembly analysis for electromechanical products: a graph-based heuristic approach",

International Journal of Production Research, 38 (5), 993-1007, 2000.

[9] H. Srinivasan and R. Gadh, "Efficient geometric disassembly of multiple components from an assembly using wave propagation ", Journal of Mechanical Design, 122 (2), 179-184, 2000.

[10] A. Gungor and S. Gupta, "Disassembly sequence plan generation using a branch-and-bound algorithm ", International Journal of Production Research, 39 (3), 481-509, 2001.

[11] E. Moore, A. Gungor, and S. Gupta," Petri net approach to disassembly process planning for products with complex AND/OR precedence relationships", European Journal of Operational Research, 135 (2), 428-449, 2001.

[12] E. Kongar and S. Gupta," Genetic algorithm for disassembly process planning", in Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing II (pp. 54-62). Newton, Massachusetts, 2001.

[13] F. Failli and G. Dini, "Optimization of disassembly sequences for recycling of end-of-life products by using a colony of ant-like agents ", Engineering of Intelligent Systems, 2070, 632-639, 2001.

[14] H. Srinivasan, and R. Gadh, A non-interfering selective disassembly sequence for components with geometric constraints, IIE Transactions, 34, 349-361, 2002.

[15] F. Torres, S. T. Puente, and R. Aracil, Disassembly Planning Based on Precedence Relations among Assemblies, International Journal Advanced Manufacturing Technology, 21 (5), 317-327, 2003.

[16] J. F. Wang, J. H. Liu, S. Li, Y. F. Zhong, "Intelligent selective disassembly using the ant colony algorithm ", Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, 17 (4-5), 325-333, 2003.

[17] A. Lambert, and S. Gupta," Disassembly modelling for assembly, maintenance, reuse, and recycling (éd. CRC Press)", 2005.

[18] E. Kongar, and S. Gupta," Disassembly sequencing using genetic algorithm, International Journal Advanced Manufacturing Technology", 30 (5-6), 497-506, 2006.

[19] F. Giudice and G. Fargione, "Disassembly planning of mechanical systems for service and recovery: A genetic algorithm based approach ", Journal of Intelligent Manufacturing, 18 (3), 313-329, 2007.

[20] Y. Tseng, H.T. Kao, and F.Y, Huang, "Integrated assembly and disassembly sequence planning using a GA approach ", International Journal of Production Research, 48 (20), 1-23, 2009.

[21] M. Tripathi, S. Agrawal, M. Pandeyc, Shankar, and M. T iwari, "Real world disassembly modeling and sequencing problem: Optimization by Algorithm of Self-Guided Ants (ASGA) ", Robotics and Computer-Integrated Manufacturing, 25 (3), 483-496, 2009.

[22] X. Zhang, and S. Y. Zhang, "Product cooperative disassembly sequence planning based on branch-and-bound algorithm ", International Journal Advanced Manufacturing Technology, 51 (9-12), 1139-1147, 2010.

[23] J. Xue, S. Qian, and Y. Zhang, "Disassembly sequence planning based on ant colony optimization algorithm ", IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications, 1125-1129, 2010.

[24] P. J. Newcomb, B. Bras, and D. W. Rosen, "Implications of modularity on product design for the life cycle", Journal of Mechanical Design, Volume 120, 1998, pp. 483-490.

[25] M. Blackenfelt, "Modularization by Relational Matrices - a Method for the Consideration of Strategic and Functional Aspects", Proceedings of the 5th WDK Workshop on Product Structuring. January 2000, Edt. Riithahuhta A., Pulkkinen A., Springer- Verlag

[26] F. Guo, and J. K. Gershenson, "A comparison of modular product design methods on improvement and iteration", In Proc of ASME Design Engineering Technical Conferences., Salt Lake City, UT. 2004.

[27] K. Hölttä, E. Suh, and De O. Weck, "Tradeoff Between Modularity and Performance for Engineering Systems and Products", International Conference On Engineering Design Iced 05 Melbourne, August 15 – 18, 2005.

[28] R.I. Whitfield, J.S. Smith, and A.H.B. Duffy, "Identifying Component Modules ", In Proceedings of Seventh International Conference on Artificial Intelligence in Design (AID'02), Cambridge, United Kingdom, pp. 571-592, 2002

[29] B. Eric, "Contributions to the instrumentation of the architectonic system: from the modular architecture of the product to the organization of the design system ", Habilitation to Direct Research, Université de Franche-Comté, Faculty of Science and Technology, FEMTO-ST Institute / Department AS2M .206 pages, 2008.

[30] J. Idicula," Planning for Concurrent Engineering", Thesis draft, Nanyang Technology University, Mars, 1995.