



**HAL**  
open science

# Improving Collaborative Decision-Making in New Product Development Projects using Clustering Algorithms

Hadi Jaber, Franck Marle, Marija Jankovic

► **To cite this version:**

Hadi Jaber, Franck Marle, Marija Jankovic. Improving Collaborative Decision-Making in New Product Development Projects using Clustering Algorithms. *IEEE Transactions on Engineering Management*, 2015, 62 (4), pp.475-483. 10.1109/tem.2015.2458332 . hal-01207087

**HAL Id: hal-01207087**

**<https://hal.science/hal-01207087>**

Submitted on 30 Sep 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# **Improving Collaborative Decision-Making in New Product Development Projects using Clustering Algorithms**

*Hadi JABER<sup>1</sup>, Franck MARLE<sup>1,\*</sup> and Marija JANKOVIC<sup>1</sup>*

*Laboratoire Genie Industriel, Ecole Centrale Paris*

*Grande voie des Vignes, Chatenay-Malabry, FRANCE*

*franck.marle@centralesupelec.fr*

**Keywords:** Organizational decision processes, project teams, organization design, technical organizations, project and R&D management, New Product Development.

## **Abstract**

Numerous decisions have to be made in early design processes. Often times they involve many actors, with the difficulty that they are shared across numerous parallel collaborative groups, for coordination and meeting scheduling reasons. This paper aims at facilitating collaborative decision-making process by grouping actors according to the relationships they have due to their assignment to decisions. Clusters of actors are proposed in order to provide decision-makers with a temporary and complementary organization designed for making efficiently simultaneous collaborative decisions. This approach has been illustrated through actual data in new product development project in the automotive industry.

## 1. Introduction

Importance of early design stages decisions has been underlined for many years [1]–[3]. In complex system design, management of collaborative decision-making is characterized by many decisions impacting numerous product- and project-related parameters. Multi-domain nature of these processes needs involvement of a wide range of actors, like project manager, system engineer, technical engineers, purchasers, architect engineers, product planners, supply chain managers and quality engineers. During early complex system design stage, decision owners need to manage decision-making process and establish temporary decision teams, identifying relevant experts in the project. In many cases, these teams are not properly established; hence, many actors participate in a large number of meetings but fail to be efficiently related to the decision-making outcome and impact. This may involve loss of efficiency and additional risks in communication and coordination between actors [4], [5].

We propose in this paper complementary temporary teams mutualizing the maximum number of collaborative decisions to be managed in each cluster. The aim of this approach is thus to support the management of NPD projects by providing more consistent and fewer collaborative teams, to make the process more adaptive and less risky [6]. It permits to enhance communication and coordination in complex system design, underlined as important in [7], by increasing within-cluster interaction values while reducing the time load due to meetings. With increase of within-cluster value, actors can better follow-up on decision impacts and manage design results. Section 2 introduces how data are gathered using matrix-based approach. Section 3 describes the clustering strategy. Section 4 presents results of an analysis based upon one actual complex system design project in automotive industry. Discussion about managerial implications is introduced in Section 5. Conclusions and perspectives are drawn in Section 6.

## 2. Modeling the relationships between actors due to the decisions they contribute to

### 2.1 The matrix-based modeling

The Dependency and Structure Modeling approach has proven to be a practical tool for representing and analyzing relations and dependencies among system components. This methodology has been created and developed by Steward, Eppinger and Browning [8]–[10]. A DSM (Design Structure Matrix) is a square matrix, representing interactions between its elements. A DSM is a square matrix, with the rows and columns identically labeled and ordered, and where the off-diagonal elements indicate relationships between the elements. Depending on the number and location of identified relationships, elements may be [10], [11]:

- Dependent (e.g. sequential if temporality is a parameter of the relationship),
- Independent (or parallel),
- Coupled,
- Conditionally connected (contingent relationship).

A DSM can be binary or numerical, with qualitative or quantitative assessment.

A DMM (Domain Mapping Matrix) is a rectangular matrix, displaying relationships between two types of elements of different domains [12], [13]. For instance, the Actor-Decision matrix (**AD**) is built by modeling affiliation relationships between actors and decisions. This data is an input of the approach.

Actor relationships or dependencies are important because they affect the efficiency of team communication, thus directly influencing design process outcomes. Hepperle et al. use DSM principles to analyze communication dependencies between actors in product development [14]. They propose a Communication Grid Method based on the identification of the network

structure underlying the communication relationships between these actors. Finally, in this paper, one DSM and 4 DMM are defined and used:

- The Actor-Decision Matrix, called **AD**, usually known as Responsibility Assignment or Affiliation Matrix (RAM), defined as a DMM.
- The Decision-Group Matrix, called **DG**. **DG** is a way of grouping decisions introduced in [15], as a first attempt to take into account collaborative decision-making processes. It is also a DMM.
- The Actor-Group Matrix, called **AG**, is based on the previous collaborative decision-making processes. **AG** is a DMM obtained by combining **AD** and **DG** matrices.
- The Actor-Actor Matrix, called **AA**. It represents the relationships between actors, on which clustering will be applied. It is an organization-related DSM, which has been the object of several works [16]–[18]. **AA** is a matrix obtained from **AD**, as introduced in Sect. 2.2.
- The Actor-Cluster Matrix (called **AC**) is resulting organization using our approach. It is the improved version of collaborative decision-making groups **AG**. **AC** is a DMM obtained by clustering Actor-Actor **AA** matrix.

## 2.2 Global approach

The initial organization is made of 13 groups, called  $G_k$ . The 70 decisions  $D_j$  are affiliated to one and only group  $G_k$ , forming the **DG** matrix. The 64 actors  $A_i$  are assigned to decisions, forming the **AD** matrix, with the relation  $AD_{ij}=1$  iff actor  $A_i$  is assigned to decision  $D_j$ . As a consequence, the **AG** matrix is built as the product of **AD** and **DG**. This means that  $AG_{ik}=0$  iff the actor  $A_i$  does not belong to the group  $G_k$ . Otherwise,  $AG_{ik}$  represents the number of decisions included in group  $G_k$  owned by actor  $A_i$ . The **AG** matrix is obtained as follows using Eq. (1) and (2):

$$\mathbf{AG} = \mathbf{AD} * \mathbf{DG} \quad (1)$$

$$\forall i \in [1..NA], j \in [1..NG], AG_{ik} = \sum_{k=1}^{ND} AD_{ij} * DG_{jk} \quad (2)$$

Where NA=number of actors,

ND = number of decisions,

NG = number of groups.

The existing organization **AG** serves as a comparison point with proposed clusters **AC**. **AC** is the result of the clustering of the **AA** matrix, itself obtained by multiplying the matrix **AD** by its transpose  $\mathbf{DA}=\mathbf{AD}^T$ , using the Eq. (3):

$$\mathbf{AA} = \mathbf{AD} * \mathbf{DA} \quad (3)$$

This product is obtained as follows (Eq. (4)):

$$\forall i, j \in [1..NA], AA_{ij} = \sum_{k=1}^{ND} AD_{ik} * DA_{kj} \quad (4)$$

This matrix enables direct interactions between actors to be analyzed. For instance, direct interactions in the US Senate have been analyzed by Bartolomeiet al[19].

Several proposals are obtained for **AC**, running simultaneously several algorithms with several configurations. The final recommendation is made considering the relevance of the clusters (within-clusters total value, cluster size, cluster density, number of clusters), in order to keep the algorithmic solution applicable to real-life project.

This generation of several alternatives enables comparisons and sensitivity analysis. Finally, the most relevant complementary organizational configuration **AC** is compared to the existing one **AG**, and implemented if judged better and applicable. Details of the clustering approach will be given in Section 3.

### **2.3 The automotive project case study**

Vehicle development projects are very long and complex, with the participation of between 1500 and 2000 project members. Usually, this type of project can take between two to four years when concurrent engineering is used as a basic organizational hypothesis. Early design stages can be long as 8 to 10 months. The overall early design stage integrates 70 decisions organized into 13 collaborative decision-making processes. The data gathering process represents a result of several working groups integrating 30 cross-domain project members. Some of these processes are: innovation integration process, manufacturing and supply chain feasibility and scheduling, design style, economic optimization, and purchasing.

Collaborative decisions integrate members from different domains. There are in total 64 different actors participating in the 13 collaborative decision-making processes. Fig. 1 shows affiliations of actors to decisions (**AD**) and of decisions to groups (**DG**). Some of these actors contribute to the process, even if not assigned to collaborative decisions. Fig. 1 permits to simultaneously visualize the three matrices used as inputs in our approach, respectively **AD**, **DG** and **AG**. Several members, like the project manager ( $A_{51}$ ), participate in almost all groups (their rows in **AG** being almost filled), whilst some other actors participate to fewer groups

like  $A_1$  (the sum of non-null cells in **AG** first row being equal to 1). Maximum number of decision-makers in one group is 21 (the sum of non-null cells in **AG** columns). Defining the groups can be difficult to decide and to implement. There are two main parameters that need to be discussed: 1) the size of the group, i.e. the number of actors one wants to put in one group, and 2) the number of groups, i.e. the total number of groups that one wants to coordinate in one project. Indeed, it is very time-consuming for people, with intertwined meetings and decisions and potential issues like meeting sequence.

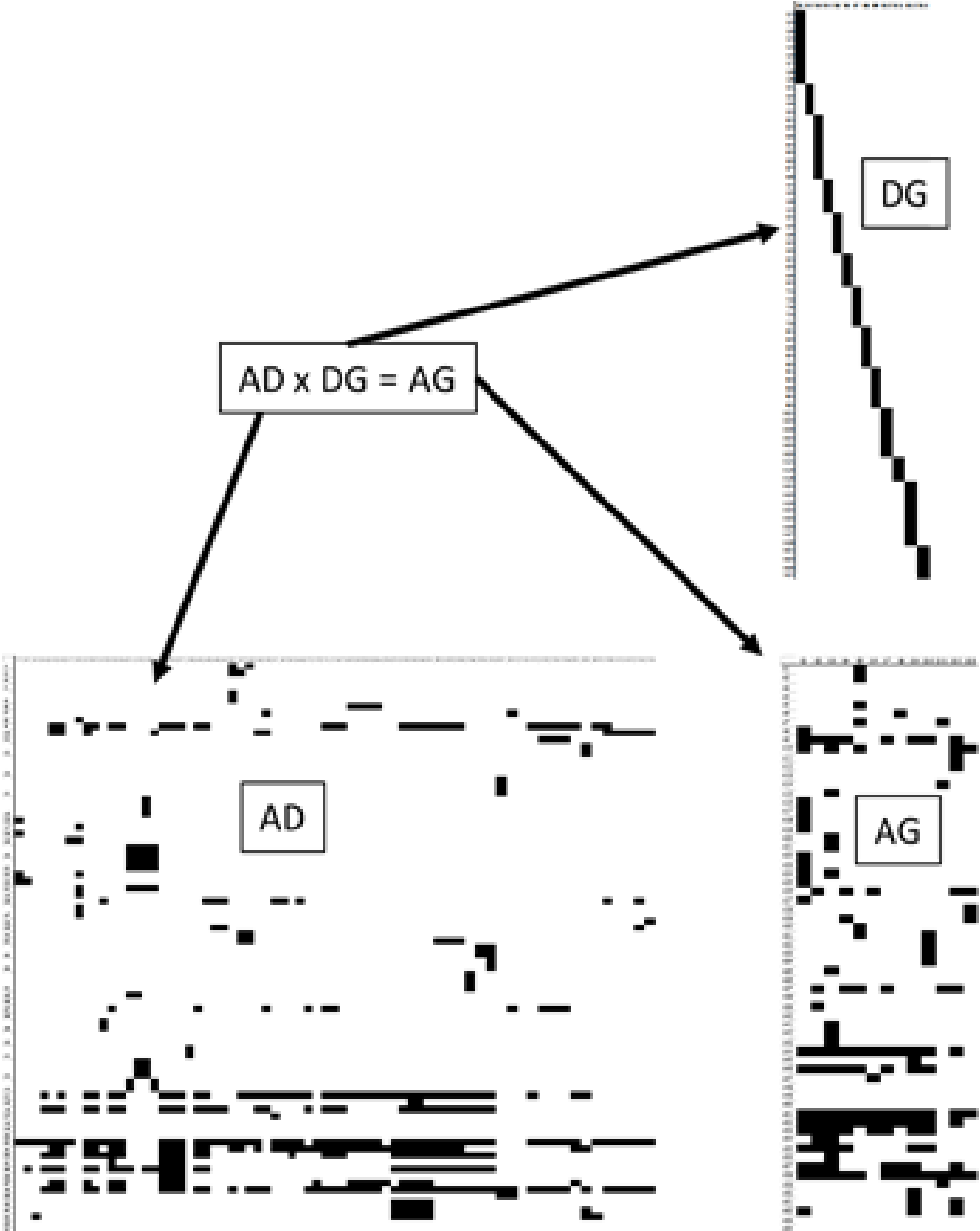


Fig. 1. The affiliation matrices (**AD**, **DG** and **AG**).



The network of direct connections between actors due to their assignment to their respective decisions, is shown hereafter in Fig. 2.

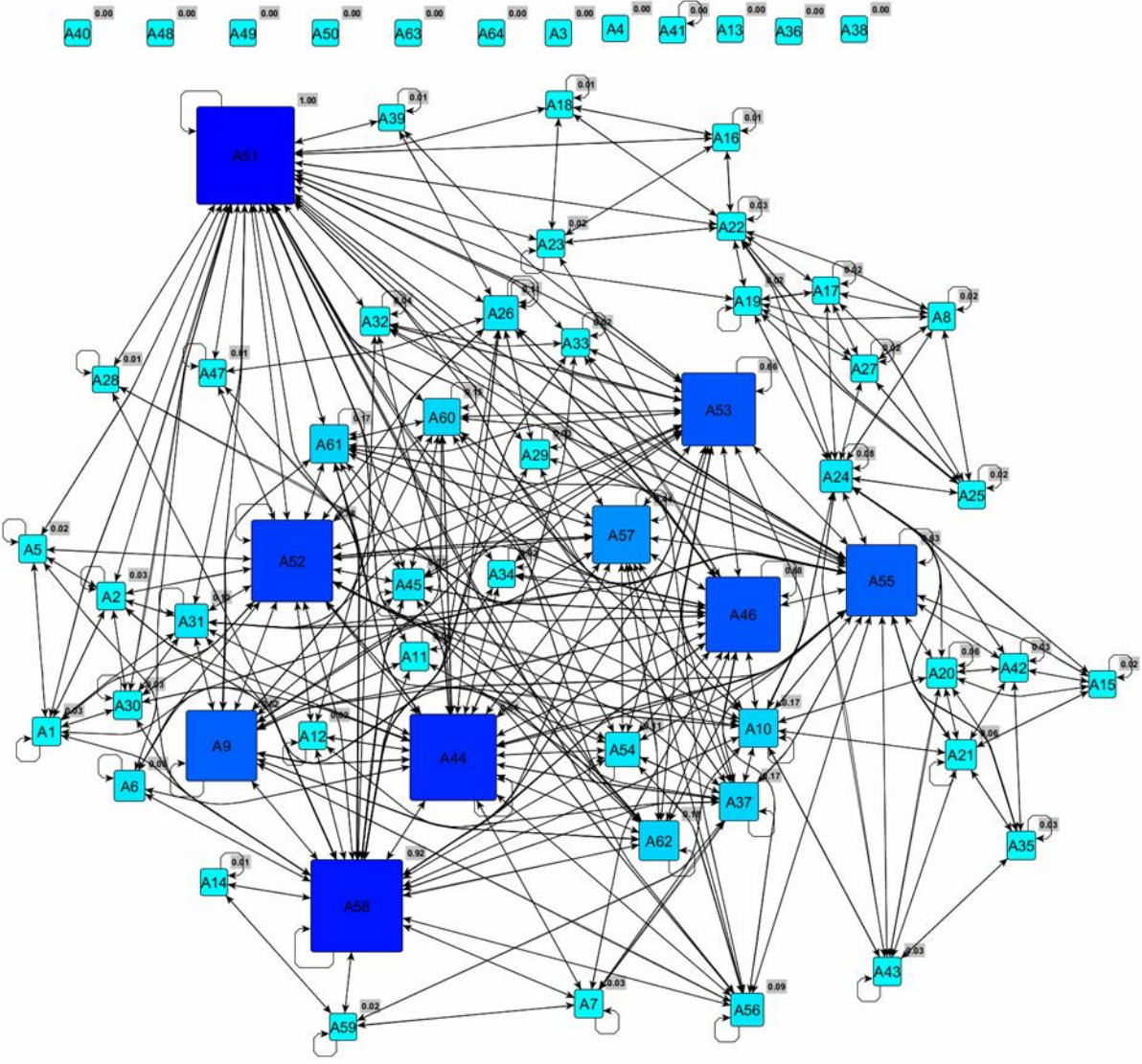


Fig. 2. The network of direct relationships between actors (**AA**)

Fig. 2 is a graph representation of the matrix **AA** calculated using Eq. (4). A graph consists of nodes and edges, respectively actors and relationships between pairs of actors. The weight on the edge between two actors represents the number of common decisions assigned to these two actors. The size of the node (and its color) is proportional to the number and the weight of its direct edges, the darkest and the biggest node corresponding to the actor who has the

highest value of connected weighted edges. The network is composed of very interrelated parts, difficult to cut into disjunctive clusters. This is the object of the following Section to define an adequate algorithm to propose clusters tailored to decision-makers requirements and constraints.

### **3. Clustering approach**

#### **3.1. General terms**

In this section, we note  $G(V, E)$  a graph where  $V$  is the list of nodes, and  $E$  is the list of edges in the graph. Lines that connect two nodes and thus define a relationship between them are called edges. We also note partition of the graph  $P=(C_1, \dots, C_k)$ . Actors  $A_i$  will be assigned to clusters  $C_j$ , forming the matrix  $\mathbf{AC}$ . The most common approach to this problem in the literature has been to ignore edge direction and apply methods developed for community discovery in undirected networks, but they discard potentially useful information contained in the edge directions. In this paper, we selected algorithms while extending clustering objective function and methodology to directed graph. Measures are extended by considering edge directionality as inherent network characteristics, like the directed version of modularity (clustering objective function) used by Leicht and Newman[20].

#### **3.2 Resolution strategy**

A clustering approach has been developed in a similar context forming clusters of heterogeneous elements (project risks). The Risk-Risk Matrix (RR) had been modeled to capture risk interactions. Then, a risk clustering approach had been applied in order to enhance efficiency in risk management by capturing possible risk propagation paths [21]–[23].

In this case, actors groups were formed indirectly due to the fact that they own risks within the same cluster. In our case, actors are directly grouped together because of the decisions they are contributing to.

A cluster may contain similar elements, with a particular element called centroid [24]. On the opposite, some works focus on edges that are least central or most “between” clusters, and remove them from the original graph in order to build the strongest clusters with the remaining edges [25]–[27]. Newman et al. are co-authors of numerous works in the field of finding community structures in complex networks [20], [28]–[31]. Specific DSM-related clustering techniques have been developed and implemented in industrial applications like IGTA (Idicula-Gutierrez-Thebeau Algorithm) for clustering Component-DSM [32]–[34], or the DSM-based algorithm of Borjesson and Holtta-Otto [35]. In the field of product architecture modularity, Hsuan and Gassmann proposed a mathematical model to analyze, then optimize this clustering objective[36]. Moreover, several works focus on clustering rectangular matrices [37], [38]. An example of the applications of clustering DSM is the clustering of organizational units performing overlapped activities in order to reduce complexity of coordination in a product development project [39].

Instead of selecting a single algorithm and optimizing in the space of possibilities, our resolution strategy will be based on 3 well-known algorithms, developed in different contexts [20], [27], [35]. This provides the benefits of each of these algorithms, which may offer either large clusters or dense clusters or balanced clusters, etc.

An innovation of this work is thus to assembly a solution from pieces of solutions obtained in different ways and using different problem configurations. There is no universally optimal configuration of clusters, but it depends on the judgment of the decision maker. Clustering

aims then at defining the best data set partitioning for given parameters. The solution is strongly dependent upon the decision-maker.

We have tested a wide spectrum of clustering algorithms, and we decided to use the three algorithms which are most adapted to our case study. These algorithms can be either parameterized or unsupervised, which is useful when no prior knowledge is provided. Such parameters can be the number of desired clusters, the maximal size of clusters, and allowance for clusters to overlap (to produce non-disjoint clusters). Furthermore, we have created a framework that allows to enter clustering parameters (forming what we call a configuration), then calculate and operate efficiently and ergonomically the input data. The solutions provided by these algorithms are automatically processed, to give local and global quality indicators of clusters. Moreover, it helps building the final solution from pieces of one or more proposed solutions to assemble the best solution corresponding to the expectations of the decision maker.

We propose a 3 stage clustering process for clustering numerical DSMs. The first stage is information gathering, about input data and parameters definition. The second stage consists in running each algorithm several times with several problem configurations. Afterwards, one obtains a number of clustered solutions, with quality indicators for each solution and for each cluster in the solution. In addition, a frequency analysis is done to indicate the number of times that each couple of elements (actors in our case study) were put together in a clustered solution. The idea is that the more often pairs of actors are proposed together in the different configurations, then the more robust the decision of putting them together in the final solution is. The third stage is the post processing of the obtained results. This is done by combining extractions of particular clusters or pieces of clusters from different solutions. This combination is based on the quality indicators and the frequency analysis of the results (the

number of times the couple of actors were put together). A hybrid solution, that meets best the needs of the decision maker, is built using a mix of clusters from all configurations.

### 3.3 Cluster validity

Cluster validation is a major issue in cluster analysis; in fact, much more attention has to be paid to cluster validity issues (checking the quality of clustering results). However, it must be emphasized that the results obtained by these methods are only tools at the disposal of the expert in order to evaluate the resulting clustering.

We define  $INTRA(C_i)$  in Eq. (5) as the sum of edges included in cluster  $C_i$  (noted  $W_i$ ), divided by the total sum of edges in the matrix  $\mathbf{AA}$ , denoted  $TW$  (for Total Weight).

$$INTRA(C_i) = W_i / TW \quad (5)$$

The term INTRA has been chosen to reflect the notion of intra-cluster interdependencies, obtained as the sum of intra-cluster edges. To obtain the  $W_i$ , we create the matrix  $\mathbf{CC}$  as the following product using Eq. (6):

$$\mathbf{CC} = \mathbf{CA} * \mathbf{AA} * \mathbf{AC} \quad (6)$$

This is obtained in two steps following Eq. (2) as the product of  $\mathbf{CA}$  by the product of  $\mathbf{AA}$  and  $\mathbf{AC}$ :

$$\mathbf{CC} = \mathbf{CA} * (\mathbf{AA} * \mathbf{AC}) \quad (7)$$

The  $W_i$  are the diagonal cells of  $\mathbf{CC}$ .

However, the implementation of the  $i^{\text{th}}$  cluster  $C_i$  requires the use of a certain number of actors. This is why we moderate the raw performance of the clustering algorithm by the

managerial efficiency, counting the Number of Actors involved in  $C_i$ , called  $NA(C_i)$ , as described in Eq. (8):

$$P(C_i) = \text{INTRA}(C_i) / \text{NA}(C_i) \quad (8)$$

Moreover, we also consider the interdependency value between two clusters  $C_i$  and  $C_j$ , called  $\text{INTER}(C_i, C_j)$ . It is defined as the sum of edges for the couples of nodes where one belongs to  $C_i$  and the other one belongs to  $C_j$ . This represents the amount of inter-clusters interactions.

It corresponds to the non-diagonal cells of the matrix  $\mathbf{CA} * \mathbf{AA} * \mathbf{AC}$  previously introduced in Eq. (6). For a given  $C_i$ , we define the  $\text{INTER}(C_i)$  as the total  $\text{INTER}(C_i, C_j)$  values for all the  $C_j$ . The meaning of  $\text{INTER}$  is to compare relatively  $\text{INTER}$  and  $\text{INTRA}$  in order to determine whether actors in the cluster should be leaders (if  $\text{INTRA} \gg \text{INTER}$ ) or guests (if  $\text{INTRA} \ll \text{INTER}$ ).

The final performance index is then calculated as in Eq. (9):

$$P'(C_i) = 1000 * \text{INTRA}(C_i) / (\text{NA}(C_i) * \text{INTER}(C_i)) \quad (9)$$

The factor 1000 is added to give results in an easily readable range (to estimate and to compare). These indicators permit comparison between proposed clustered configurations against each other; and afterwards against initial configuration  $\mathbf{AG}$ , both in terms of organizational efficiency ( $P$  index) and in terms of role given to the actors ( $P'$  index). For instance, if a cluster is always proposed whatever the algorithm and whatever the configuration, then one can be confident to put it in the final proposed configuration.

#### 4. Results: proposing new groups of interrelated actors

Initial organization into collaborative decision-making processes had been a proposition in order to enhance project management and gain time. Several previous works have accounted

for this organization proposition. In the follow-up of three projects the overall gain for the early design stages was estimated up to 20% which is an interesting improvement with regard to duration of this phase. This organization was based upon collaborative decision-making model and in particular information modelling. However, one of the aspect that was important and that has been discussed with design team is to take into account information propagation. Due to the time, these configurations have not been implemented in projects but have been considered as important. In view to this observation, authors have started to look into different clustering techniques and related organizations that are discussed henceforth.

In this case, no constraint was initially applied on cluster size for first runs, then several maximal sizes have been introduced to test the sensitivity of the solutions (in a range from 7 to 13). It showed a great stability in the proposed configurations.

Fig. 3 shows the final clustering results for **AA**. Firstly, whatever the algorithm and the configuration, there is systematically a 9-actor cluster which is proposed. This cluster  $C_1$  is not only strongly connected (high *INTRA* within-cluster interactions) but also connected to the rest of the network.

Secondly, two clusters are proposed as a discussion group with the actors of  $C_1$ , for *INTER*-connection reasons. Indeed, the number and strength of interdependencies are far higher between  $C_2$  and  $C_1$  than within  $C_2$  for instance (idem for  $C_3$ ). Fig. 3 shows that 21 actors are in this situation, respectively 11 actors in  $C_2$  and 10 in  $C_3$ . These clusters  $C_2$  and  $C_3$  could also be implemented as working groups, but their density *INTRA* is judged too low.

Thirdly, a 16-actor cluster  $C_4$  is proposed where interactions are mainly within the cluster, and not outside. This is a cluster built for *INTRA*-connection reason (called Intra-only on Fig. 3).

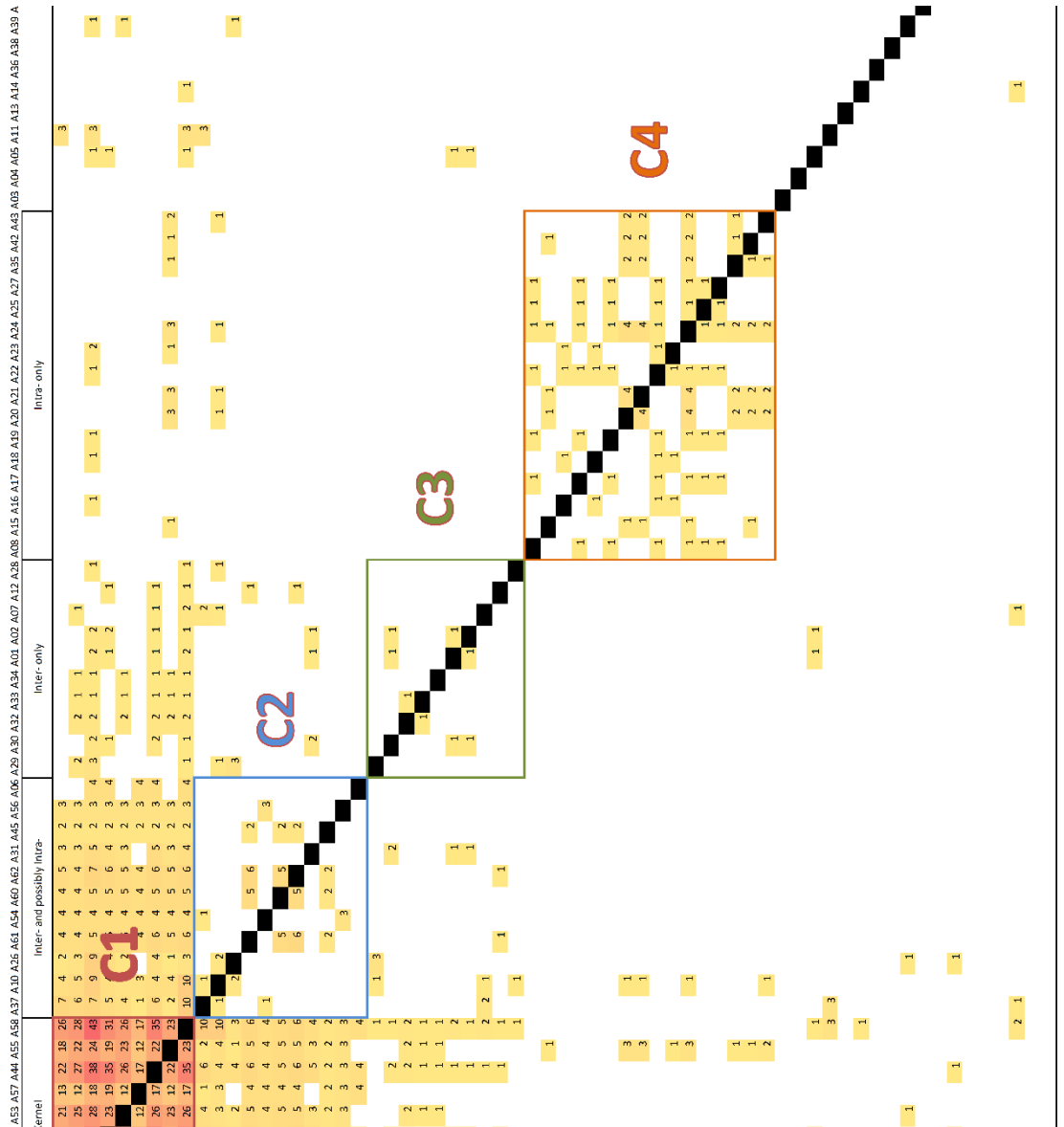


Fig. 3. The clusters for AA

Eighteen actors are not included in any group. Of course, this does not mean that they won't continue to contribute to the decisions they are assigned to, but not as a specific collaborative working group. Section 5 will discuss the implication of both the implementation of clusters and the management of the non-clustered actors.

The performance indicators  $P$  and  $P'$  are calculated using Eq. (8) and (9) for the clusters  $C_i$ ,  $i=1$  to 4 and the groups  $G_i$ ,  $i=1$  to 13, as summarized in Table 1. For memory, the groups  $G_i$



form the existing configuration **AG** and the aim of the clustering is to propose an improved version **AC** composed of clusters  $C_i$ .

Table 1. Comparison between proposed clusters  $C_i$  and initial groups  $G_i$  in terms of performance indicators

Id	INTRA	NA	P	INTER	P'
$C_1$	1724	9	191.5	486	394.1
$C_2$	58	11	5.3	412	12.8
$C_3$	8	10	0.8	74	10.8
$C_4$	126	16	7.9	25	315
$G_1$	1988	21	94.7	16908	5.6
$G_2$	1956	13	150.4	16976	8.8
$G_3$	2056	20	102.9	16979	6.1
$G_4$	1652	11	150.2	15752	9.5
$G_5$	896	13	68.9	11839	5.8
$G_6$	416	6	69.3	8667	8
$G_7$	1490	9	165.6	15053	11
$G_8$	616	6	102.7	10041	10.2
$G_9$	2074	13	159.5	17212	9.3
$G_{10}$	1594	12	132.8	15450	8.6
$G_{11}$	40	5	8	2909	2.7
$G_{12}$	1164	13	89.5	13442	6.7
$G_{13}$	174	6	29	5566	5.2

Two elements may be highlighted with these results. First, the initial organization is far from efficient due to multiple assignments of actors to multiple groups. They are involved in numerous meetings, but with many connections outside the groups, which is a factor of loss of time and potential loss of coordination.

Second, clusters  $C_1$  and  $C_4$  are dense enough in terms of *INTRA* to justify their implementation.  $C_4$  is less dense than  $C_1$  in terms of *P* indicator, but is also quite independent from other clusters, meaning that  $P'(C_4)$  index is almost as good as  $P'(C_1)$ .  $C_2$  and  $C_3$  are not dense enough to be created as a collaborative group. However, their *INTER* value with  $C_1$  is high enough to justify the integration of their actors in some  $C_1$  meetings as guests. The question is now how to implement clusters  $C_1$  and  $C_4$ , and particularly what to do with the rest of the decisions and actors.

The clustering of actors implies grouping decisions, due to the assignments of these actors. For  $C_1$ , there are two types of decisions: the ones when they are leaders, inviting other actors as guests; and the ones when they are guests. In the first type, we propose to split the corresponding 51 decisions associated to  $C_1$  into two subgroups. This means two series of meetings, but subgroups contain respectively 32 and 19 decisions; so a single group could also have been judged as practically intractable in view to number of decisions and number of participants.  $C_4$  involves a 6-decision group involving 16 actors.

The 13 remaining decisions may be considered using three strategies:

- A single group, with less consistence and the only advantage to propose less meetings to people, but knowing that their connections are lower than in the other groups,

- 13 singletons, meaning that each decision is managed independently with its affiliated actors. This increases the number of assignments but may also increase the efficiency of decision-making for each decision,
- The current organizational groups, knowing that decisions of seven groups 1, 3, 4, 5, 11, 12 and 13 are present. This means running 7 series of meetings.

With regard to these strategies, we have the choice for the 13 remaining decisions between 1, 7, and 13 groups. There are slight differences, but the performance of the overall configuration is always far better than the initial organization; hence the result is not sensitive to this choice.

The most important resulting fact is that with a 9-actor group plus occasional guests, 51 decisions (on a total of 70) can be managed in a coordinated and collaborative way; and with an extra 16-actor group, 6 additional decisions are grouped consistently. The managerial implication of implementation of one of these scenarios is discussed in Section 5.

## **5. Implication of these groups on early stage new product development management**

Main challenge observed in industry is to manage these collaborative decision-making processes with regard to efficiency (especially time loss that actors have been experiencing) and coordination. The proposed approach highlights several strategies with the possibility of implementing one or more clusters, involving multiple decisions to be made with coordination. However, whatever the chosen scenario, it is always far more performant than the initial configuration **AG**, since the latter had the major disadvantage to assign many actors to almost every group. The constraint of disjunction between clusters really permits to focus

for a given actor on less groups, reducing drastically the number of assignments and meetings compared to **AG**.

Comparing clustering results for **AA** with the initial organization **AG** enables interesting remarks to be done. First, cluster  $C_1$  is very important in terms of *INTRA* and *INTER* values. It is constituted of important members of the project (e.g. project manager, prototype design manager, technical project manager, design process manager). However, this 9-actor group was not currently formed in the existing project organization.

Second, the other implemented cluster  $C_4$  permits to group 16 actors with less assignments, which is a gain compared to **AG**. For the rest of the actors of  $C_2$  and  $C_3$ , it is not mandatory to make them participate to groups which would be not dense enough to deserve the effort. They are simply guests when some specific meetings require their presence, which can be more efficient and easier to coordinate.

Third, the proposed groups are composed of permanent members and temporary guests. Members of  $C_1$  need to participate to other decision groups (but far less than in **AG**) and actors outside of  $C_1$  need to participate punctually to some  $C_1$  meetings. The proposed decision groups corresponding to actors' clusters are as follows: a 32-decision  $DG_1$  and a 19-decision  $DG_2$  managed by  $C_1$ , a 6-decision  $DG_3$  managed by people of  $C_4$ , and a last 13-decision  $DG_4$  whose leadership has to be defined. By counting the total number of actors involved in each proposed decision group, we find the following results showed in Table 2.

Table 2. Composition of decision groups within actors' clusters

Decision group	Nb of decisions	Members	Guests
DG1	32	9 (C1)	14
DG2	19	9 (C1)	13
DG3	6	16 (C4)	3

We believe that this approach can effectively support management of NPD projects and related teams. The proposed modeling approach can also be discussed in relation to the group organization. Initial organization stemmed from the cultural and historical development of the company. That organization is built around 13 collaborative decision-making processes that are domain oriented. However, possible matrix-based visualizations and modeling can be a support to constructively decide possible organizations. Moreover, it brings flexibility and agility to the organization with regards to NPD objectives (for example if there is a new technology integration that might be extremely risky).

Another important point is clustering and proposition of different organizations in view to different NPD objectives, like enhancing communication, or geographical proximity, etc. As the approach proposes different clustering techniques with regard to different criteria, possible team organizations can be examined at the very beginning of the project in order to adapt the organization with the objective of maximizing the possibility for project success.

## **6. Conclusion**

In this paper, a methodology is introduced to propose groups of actors involved in numerous collaborative decisions. These groups are formed using an original approach combining several classical clustering algorithms. The first results show different reasons to group actors and different roles of these actors in the network structure and behavior. Particularly, a strong cluster of 9 actors has been identified. Two strategies of recommendations can be done after this analysis. Groups can be formed, for two main reasons, *INTRA*- and *INTER*-connections. Moreover, recommendations for simplifying assignments can also be proposed, with two main strategies: reassigning people in order to have less actors involved in decision chains, and reducing redundancy (the same couple of actors assigned to the same couple of

decisions). Future works will be done to test such reassignment strategies and their impact on the structure complexity. The aim is to understand the impact of a mitigation action against the risk of non-coordination and non-communication due to this complexity level.

## 7. References

- [1] S. Bellut, *Les processus de la décision*. AFNOR, 2002.
- [2] P. Badke-Schaub and A. Gehrlicher, “Patterns of Decisions in Design: Leaps, Loops, Cycles, Sequences and Meta- processes.” in *International Conference on Engineering Design - ICED 03*, 2003.
- [3] C.T. Hansen and M. M. Andreasen, “A Mapping of Design Decision Making.” in *International Design Conference - Design*, 2004.
- [4] T. R. Browning, J. J. Deyst, S. D. Eppinger, and D. E. Whitney, “Adding Value in Product Development by Creating Information and Reducing Risk,” *IEEE Trans. Eng. Manag.*, vol. 49, no. 4, pp. 443–458, 2002.
- [5] T. R. Browning and S. D. Eppinger, “Modeling impacts of process architecture on cost and schedule risk in product development,” *IEEE Trans. Eng. Manag.*, vol. 49, no. 4, pp. 428–442, 2002.
- [6] V. Lévardy and T. R. Browning, “An adaptive process model to support product development project management,” *IEEE Trans. Eng. Manag.*, vol. 56, no. 4, pp. 600–620, 2009.
- [7] M. D. Morelli, S. D. Eppinger, and R. K. Gulati, “Predicting Technical Communication in Product Development Organizations,” *IEEE Trans. Eng. Manag.*, vol. 42, no. 3, pp. 215–222., 1995.
- [8] D. Steward, “The Design Structure System: A Method for Managing the Design of Complex Systems.” *IEEE Trans. Eng. Manag.*, vol. 28, pp. 79–83, 1981.
- [9] S. Eppinger, D. Whitney, R. Smith, and D. Gebala, “A Model-Based Method for Organizing Tasks in Product Development,” *Res. Eng. Des.*, vol. 6, no. 1, pp. 1–13, 1994.
- [10] T. R. Browning, “Applying the design structure matrix to system decomposition and integration problems: a review and new directions.” *IEEE Trans. Eng. Manag.*, vol. 48, no. 3, pp. 292–306, 2001.

- [11] J. Thompson, *Organizations in action*. McGraw-Hill, New York, 1967.
- [12] M. Danilovic and T. R. T. R. Browning, “Managing complex product development projects with Design structure matrices and domain mapping matrices,” *Int. J. Proj. Manag.*, vol. 25, pp. 300–314, 2007.
- [13] Y. Akao, *Quality Function Deployment*. Cambridge, MA: Productivity Press, 1990.
- [14] C. Hepperle, A. M. Maier, M. Kreimeyer, U. Lindemann, and J. Clarkson, “Analyzing communication dependencies in product development using the design structure matrix,” in *9th International Design Structure Matrix Conference, DSM’07.*, 2007.
- [15] M. Jankovic, J. Stal-Le Cardinal, and J.-C. Bocquet, “Collaborative Decision in design project management : a particular focus on automotive industry,” *J. Decis. Syst.*, vol. 9, no. 1, pp. 93–117, 2010.
- [16] J. Lorsch and P. Lawrence, *Managing Group and Intergroup Relations*. Homewood, IL.: Richard D. Irwin, 1972.
- [17] K. McCord and S. Eppinger, “Managing the Integration Problem in Concurrent Engineering,” 1993.
- [18] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, “The Misalignment of Product and Organizational Structures in Complex Product Development,” *Manage. Sci.*, vol. 50, no. 12, pp. 1674–1689., 2004.
- [19] J. Bartolomei, D. Hastings, R. de Neufville, and D. Rhodes, “Engineering Systems Multiple Domain Matrix: an Organizing Framework for Modeling Large-Scale Complex Systems,” *Syst. Eng.*, vol. 15, no. 1, pp. 41–61, 2012.
- [20] E. A. Leicht and M. E. Newman, “Community structure in directed networks,” *Phys. Rev. Lett.*, vol. 100, no. 11, p. 118703, 2008.
- [21] F. Marle and L.-A. Vidal, “Project Risk Management Processes : Improving Coordination Using a Clustering Approach,” *Res. Eng. Des.*, vol. 22, pp. 189–206, 2011.
- [22] F. Marle, L. Vidal, and J. Bocquet, “Interactions-Based Risk Clustering Methodologies and Algorithms for Complex Project Management,” *Intern. J. Prod. Econ.*, vol. 142, no. 2, pp. 225–234, 2013.
- [23] F. Marle and L. Vidal, “Forming risk clusters in projects to improve coordination between risk owners,” *J. Manag. Eng.*, 2014.
- [24] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, “A survey of kernel and spectral methods for clustering,” *Pattern Recognit.*, vol. 41, 2008.

- [25] M. Girvan and M. Newman, "Community structure in social and biological networks.," *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [26] L. Freeman, "Set of measures of centrality based on betweenness," *Sociometry*, vol. 40, pp. 35–41, 1977.
- [27] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. P10008, 2008.
- [28] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, pp. 98–101, 2008.
- [29] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Proc. Natl. Acad. Sci. USA*, vol. 103, pp. 8577–8582, 2006.
- [30] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, no. 066111, pp. 1–6, 2004.
- [31] A. Clauset, C. Moore, and M. Newman, "Structural Inference of Hierarchies in Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006.
- [32] J. Idicula, "Planning for Concurrent Engineering," Gintic Institute, Singapore, 1995.
- [33] C. I. Gutierrez-Fernandez, "Integration Analysis of Product Architecture to Support Effective Team Co-Location," Massachusetts Institute of Technology, 1998.
- [34] R. E. Thebeau, "Knowledge Management of System Interfaces and Interactions for Product Development Process," Massachusetts Institute of Technology, 2001.
- [35] F. Borjesson and K. Holtta-Otto, "A Module Generation Algorithm for Product Architecture based on Component Interactions and Strategic Drivers," *Res. Eng. Des.*, vol. 25, pp. 31–51, 2014.
- [36] M. Hsuan and O. Gassmann, "Managing modularity of product architectures: toward an integrated theory," *IEEE Trans. Eng. Manag.*, vol. 50, no. 2, pp. 204–218, 2003.
- [37] S. Li, "A matrix-based clustering approach for the decomposition of design problems," *Res. Eng. Des.*, vol. 22, no. 4, pp. 263–278, Apr. 2011.
- [38] S. Li, "Methodical Extensions for Decomposition of Matrix-Based Design Problems," *J. Mech. Des.*, vol. 132, no. 6, p. 061003, 2010.
- [39] Q. Yang, T. Yao, T. Lu, and B. Zhang, "An Overlapping-based Design Structure Matrix for Measuring Interaction Strength and Clustering Analysis in Product Development Project," *IEEE Trans. Eng. Manag.*, vol. 61, no. 1, pp. 159–170, 2014.



