HAL open science

Organization and Coordination of Project Actors to Manage Impact Propagation between Deliverables

Hadi Jaber, Franck Marle, Ludovic-Alexandre Vidal, Yassine Ali, Lionel Didiez

To cite this version:

Hadi Jaber, Franck Marle, Ludovic-Alexandre Vidal, Yassine Ali, Lionel Didiez. Organization and Coordination of Project Actors to Manage Impact Propagation between Deliverables. 19th Dependency and Structure Modelling Conference (DSM), Sep 2017, Helsinki, Finland. hal-01589610

HAL Id: hal-01589610

https://hal.science/hal-01589610v1

Submitted on 4 Oct 2020

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.

19TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2017
ESPOO, FINLAND, SEPTEMBER 11 – 13, 2017

Organization and Coordination of Project Actors to Manage Impact Propagation between Deliverables

Hadi Jaber[1], Franck Marle[2], Ludovic-Alexandre Vidal[2], Ali Yassine[1], Lionel Didiez[3]

[1] American University of Beirut, Faculty of Engineering and Architecture, Department of Industrial Engineering and Management, Lebanon
[2] Université Paris-Saclay, CentraleSupélec, Laboratoire Genie Industriel, France
[3] Renault S.A., Strategy of Quality Management Department, France

Abstract: The article deals with managing the deliverables shared between activities of a product development (PD) project. In particular, we are interested in understanding the impact of incomplete deliverables (e.g., missing or poor quality) from one activity on subsequent activities, and how do these impacts propagate throughout the entire project. To capture the complexity of exchanged and shared deliverables within a PD project, we propose a modeling framework based on the Design Structure Matrix (DSM). By focusing on actors who are involved in many deliverable-related interdependencies, we can propose strategies to mitigate risks of poor communications and coordination due to the complex project structure. Our approach is to form complementary teams (or working groups) according to the relationships they have due to their deliverable exchanges. This improved teaming arrangements allows for increased coordination between actors who are interdependent, albeit not always formally connected via the hierarchical structure of the project organization

Keywords: DSM, Impacts Propagation, Project Organization, Clustering.

1 Introduction

Management by deliverables, is a newer method of project management and can have a positive influence on the project (Dickau and King 1999), (Richtnér and Åhlström, 2010). Rather than viewing project management as a time-driven support function, project management is factored into the deliverables budget and quality, allowing project management performance to be monitored against outputs (deliverables) rather than inputs i.e. time spent (Indelicato,2009), (Miller, 2009). This is an alternative to traditional project management techniques which traditionally focused on resource management (Bryde and Joby 2007). For instance, consider the purpose of Earned Value curves which compare the budget to the work performed. Some researchers stressed that "we could not execute the project by following only the schedule and budget. These are two fundamental concerns, but we should ensure the compliance of delivered features such as quality of technical implementation. Management by deliverables focuses on operational monitoring of the project; it focuses on results and allows for anticipation" (Fernandez, 2011).

An activity is performing an action to achieve a result. Each outlined activity must involve: (a) a precise and measurable objective; (b) appropriate human, material and financial resources; (c) a workload expressed in the number of resources / day; and (d) a specified period with clear start and end dates. In a schedule, activities are interconnected

by dependency relationships. Project milestones are defined as the key events within the project, showing important progress in significant dates with concrete realizations (deliverables production). A project consists of deliverables that meet objectives that are realized through activities. These deliverables are themselves broken down into sub-deliverables and activities. A deliverable is a term used in project management to describe a tangible object produced as a result of the project that is intended to be delivered to a customer (either internal or external). For example, requirements' specification and feasibility study are two deliverables within a project. A deliverable could be a report, a document, a permit or any other building block of an overall project. A deliverable may be composed of multiple smaller deliverables. It may be either an outcome to be achieved or a product to be provided (Browning and Ramasesh, 2009). These deliverables are updated according to the changes and developments that occur throughout the project life cycle. They are archived at the end of the project and provide a practical basis for future projects within the company. For example, the final deliverables of PD projects in the automotive industry are documents for manufacturing vehicles in factories. Due to the large number of interactions between deliverables produced by various actors across the company, the danger is that the communication and coordination between actors may not be correctly done (Barry et al. 2015). Despite the events that disrupt the project progress, the propagation of impacts should be managed in order to ensure meeting of targets in terms of cost, lead-time, quality, and technical performance. The problems of impacts' propagation encountered in projects are usually due to inadequate anticipation of the adverse consequences of incomplete (or poor quality) information on dependent downstream tasks.

In the automotive industry, a typical car contains about 2000 functional components, 30000 parts, and 10 million lines of software code. Similarly, vehicle development projects are very long and complex, with the participation of 1500 to 2000 project members, 320 milestones during 26 months of development, and the release of about 4000 deliverables. This growing complexity is one of the greatest challenges of project management and one of the causes for project failure in terms of cost overruns and time delays. This increasing number of project elements and their dependencies increase risks since a problem in one element can propagate to other elements directly and indirectly. When faced with such complex situations, the way that project members are organized is crucial to determine how they will be able to collectively cope with nontrivial problems and risks. Current project organizations are generally based on single-criterion decompositions, whether product- or process- or organizational entity-based. Project can be decomposed into either Product- (or System)-related elements, phases or organizational entities, but there will always be numerous interdependencies between actors who do not belong to the same part. This implies risk of bad communication, bad coordination or locally optimal decisions. Due to the number of interactions outside the official project structures, the danger is that the communication and coordination between actors may not be correctly done. Our objective is to propose a complementary project organization to be practically closer to the real network structure of project actors in order to manage and control efficiently and collectively the impacts propagation between project deliverables. This permits to reduce rework and increase performance and productivity. This organizational reshuffling will be done using clustering methodology, based on actor-actor interdependency matrices.

2 Modeling of Product Development Projects

This section proposes a modeling approach of complex projects using the Design Structure Matrix (DSM) methodology, which takes into account the elements of a system and their interactions. Modeling and analyzing the interactions between risks, processes, product elements, and actors contribute to understanding the project complexity aspects in order to reduce them when making decisions. A framework is created which allows the user to enter, calculate and operate efficiently the input data. The input data are analyzed in a simple and non-matrix format in Excel, and an automated process creates the corresponding graph and associated DSM. This framework allows the assembly of the global network of project elements interactions from local data.

In this section, we describe the steps of modeling complex projects by modeling interdependencies between theirs actors via the exchanged deliverables, with the purpose of anticipating impacts propagation between project deliverables through actors.

To manage better the interfaces between project actors, we need to identify and communicate the deliverables produced and received by each actor. The Dependency Structure Modeling (DSM) approach has proven to be a practical tool for representing and analyzing relations and dependencies among system elements. The DSM approach has several advantages, such as the calculations inherent to the matrix format to get the benefits of different types of analyses. It avoids issues associated with the visual display of complex networks, especially in the case of structures including lots of interactions and even loops (Steward, 1981), (Eppinger et al., 1994), (Eppinger and Browning, 2012). It is a highly compact, easily scalable, and intuitively readable representation to navigate across dependencies between elements.

A DSM is a square matrix, representing interactions between its elements, with the rows and columns identically labeled and ordered, and where the off-diagonal elements indicate relationships between the elements. In this paper we use the following convention for DSM orientation: An element's inputs appear in its matrix row and its outputs appear in its column. Domain Mapping Matrix (DMM) is a rectangular matrix mapping elements of a certain domain to elements of another domain (Danilovic and Browning, 2007), (Akao, 1990).
Our DSM in this research paper is called the **Actor-Actor** matrix, **AA**. It represents the relationships between project actors, on which several improvements and analyses will be applied in order to understand and control the project behavior, more precisely the impacts' propagation analysis between the deliverables exchanged between these actors.

Moreover, we define two matrices as Domain Mapping Matrices (**DMM**s): the first one is the $\mathbf{Actor_{Transmitter}}$**-Deliverable** matrix**,** which is built by modeling affiliation relationships between actors (transmitters) and deliverables. The second one is the **Deliverable-**$\mathbf{Actor_{Receiver}}$ matrix, which is built by modeling affiliation relationships between deliverables and actors (receivers). Both matrices can be obtained using an algorithm presented in the following section applied on project plans to extract global interactions data from local interactions data. **AA** can be obtained by the following formula, when we have the same list of actors receivers and actors transmitters in both matrices:

$$(5): AA = Actor_{Transmitter} Deliverable * Deliverable\ Actor_{Receiver}$$

After obtaining a first version of AA, we can conduct interviews to enrich and verify the deliverables produced and received by each actor.

3 A Three-Stage Clustering Process for Project Actors Network

Our contribution is grouping people to increase organizational capacity in terms of communication and coordination and reduce propagation of risks. This is done using an original three-stage process for clustering of project actors. This permit to maximize interactions inside a group and minimize interactions inter-groups. This Section introduces the clustering strategy used to group actors taking into account the number, direction and strength of their interdependencies. The solving approach consists in running in parallel several complementary algorithms with several parameters configurations. This three-stage approach may propose the best possible solution adaptable to the needs of the decision maker.

3.1 First Stage: Parameters Definition

Section 2 proposed an approach to model the deliverables exchanged between project actors. Here we consider that the network of project actors is an input data but we need to define parameters of the desired clustering solution. Furthermore, in this research work we created an interface that allows to enter clustering parameters, calculate and operate efficiently and ergonomically the input data with a given clustering configuration. We achieved automatic processing to the solutions provided by these algorithms, which will give quality indicators: local and global, but also helps to build the final solution from part of one or more proposed solutions to assemble the best solution corresponding to the expectations of the decision maker. Clustering algorithms can be either parameterized or unsupervised, if no prior knowledge is provided. Such parameters can be:

- The number of the desired clusters (groups)
- The maximal size of the clusters
- The number of project deliverables interchanged between actors within a cluster
- Allowing clusters to overlap (to produce non-disjoint clusters)
- Actors who need to be put together or actors who are not to be put conjointly

3.2 Second Stage: Execution

Second step consists in running multiple algorithms many times with several configurations. First, we did a benchmark for assessing the performance of a wide spectrum of graph-based clustering algorithms with regards to two different problems: cluster detection and parametric clustering (the capacity to tailor the algorithm to more specific parameters asked by the decision-maker). The performance of the considered algorithms has also been tested in a case of a past real case study, in order to calibrate the models (in terms of performance and outcomes). We did a benchmarking activity for assessing eight graph-based clustering algorithms with regards to the quality of detected clusters and parameters' selection. Then we decided to use the most adequate four algorithms in order to apply them to our case study:

- The first one is "community structure in directed networks", which takes into account edge direction, by generalization of the widely used benefit function known as modularity. This method is shown to give demonstrably better results than previous methods on a variety of test networks, both real and computer-generated (Leicht and Newman, 2008).

- The second one is "fast unfolding of community hierarchies in large networks", which takes into account edge direction, additionally to parameterize if we need small or big size of clusters (Blondel et al., 2008).
- The third one is Idicula-Gutierrez-Thebeau Algorithm for clustering Component-DSM, which takes into account the maximal size of the desired clusters, and represents a significant improvement in speed and quality of solution obtained (Thebeau, 2001), (Borjesson and Holtta-Otto, 2012).
- The fourth one is "1-Spectal Clustering", which takes into account the number of desired clusters, additionally to constraints where nodes, which need to be put together or nodes, which are not to be put conjointly (Bühler and Hein, 2010).

The experiments show that the results found by these four algorithms are at least as good as the other clustering algorithms but often lead to significantly better results. This provides the benefits of each of these algorithms, which may offer either large, dense or balanced clusters, etc.

3.3 Third Stage: Post-processing using cluster validity & frequency analysis

This section introduce the third stage of the clustering process. First, it presents the global and quality indicators to validate clusters and compare solutions; second it presents the frequency analysis, and finally the methodology to assembly the final solution.

3.3.1 Cluster validity: Global and Local Indicators

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. For these reasons we define two types of indicators. The first type is global and permit to compare the quality of two clustering solutions; the second one is local and permit to compare two clusters either within the same solution or from different solution.

The existing organization, called AG, represents the assignment of actors A to organizational groups G. It always serves as a comparison point with proposed clusters. The aim is to propose an improved version of AG, called AC. CA is the transpose matrix of AC.

We define INTRA (Ci) in Eq. (2) as the sum of deliverables exchanged in cluster Ci (noted Wi), divided by the total sum of deliverables exchanged in the matrix AA, denoted TW (for Total Weight).

(6): INTRA(Ci) = Wi / TW

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 Wi, we create the matrix CC as the product of CA by the product of AA and AC using Eq. (3):

*(3): CC = CA * (AA * AC)*

The Wi are the diagonal cells of CC.

However, the implementation of the i-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 N_A involved in Ci, called NA(Ci), as described in Eq. (5):

(7): P (Ci) = INTRA (Ci) / $\boldsymbol{N_A}$ *(Ci)*

3.3.2 Frequency analysis

We define $\mathbf{N_{Config}}$ as the number of different tested problem configurations. We introduce a new index which calculates the percentage of times where two actors are put in the same cluster (Common Cluster Frequency Index: CCFI), and we introduce the variable **L** as the number of the tested configuration (**L** varies between 1 and $\mathbf{N_{Config}}$). An associated complementary index gives the percentage of times where an actor is included in a cluster (Clustered Frequency Index: CFI). For different configurations $\mathbf{C_L}$, we have different Clustered Organization matrices $\mathbf{CO_L}$, and we define the Frequency Matrix as the sum of the $\mathbf{CO_L}$ matrices. The non-diagonal terms of the Frequency Matrix give the Common Cluster Frequency Index **(CCFI)** for a couple of actors (Equation 6), and the diagonal terms give the Clustered Frequency Index **(CFI)** for an actor (Equation 7):

(8): $\mathbf{CCFI(i,j)} = \frac{\sum_{L}^{\mathbf{NConfig}} \boldsymbol{CO}\,(i,j)}{\mathbf{NConfig}}$

(9): $\mathbf{CFI(i)} = \frac{\sum_{L}^{\mathbf{NConfig}} \boldsymbol{CO}\,(i,i)}{\mathbf{NConfig}}$

The interesting values are 0% and 100%.CCFI = 0 means that the actors are never clustered together and 100% means that they are always in the same cluster.

We introduced a frequency matrix which indicates, for its non-diagonal elements the percentage of times where two actors $\mathbf{A_i}$ and $\mathbf{A_j}$ are assigned to the same cluster, and for its diagonal elements the percentage of times where one actor is assigned to a cluster. These information give an indication for pre-assigning some variables to 0 or 1, expressing that two actors cannot be together or must be together. Moreover, it gives an idea of the robustness of the final clustering decision, since we are more confident with an index of 1 (or close to 1) than an index of 0.5. To conclude, it should be noted that the frequency indicator is a decision aid, not an automatic assignment rule.

3.3.3 Assembly of the final solution

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

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 actors 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. We created a decision support system that report the clustering results with their corresponding quality indicators. This decision support system helps in the selection of the best possible clusters based on rigorous comparisons. Then it helps in proposing solutions composed of best clusters from all configuration, and it compares the proposed solution with existing organization based on the global quality indicator. To conclude, a hybrid solution, that meets best the needs of the decision maker, is built using a mix of clusters from all configurations.

4 Improving Coordination between Actors

This section explains the application of the three-stage clustering process to propose groups of actors in new PD projects. The focus is on actors who are involved in many deliverable-related and inter-phase interdependencies. We propose an approach to form complementary teams or working groups according to the relationships they have due to their deliverable exchanges. This permits to increase coordination between actors who are interdependent, albeit not always formally connected via the hierarchical structure of the project organization. This enables potential issues due to complexity, like bad communication and coordination, to be dealt by actors who are not initially put together. In this research work, we are interested in understanding the impact of incomplete deliverables (e.g., missing or poor quality) from one activity on subsequent activities, and how do these impacts propagate throughout the entire project. For instance, if we have bad quality of input data for a simulation model, these will trigger errors in the output results of the simulation, these results will trouble the receiver, and so on. Our contribution is to suggest the ideal communication groups of actors who make project deliverables meet their intended timeline, cost, and quality. Ideal clusters of interlarded actors via deliverables help in prevention and reducing errors propagation between project deliverables.

4.1 Analysis of existing vehicle development organization

Vehicle development projects are very long and complex, with the participation of various actors from different departments. In this section, we analyze the early design stage of these projects which can be as long as 8 to 10 months. The data gathering process represents a result of several working groups integrating cross-domain project members. Some of these domains are: innovation integration, manufacturing and supply chain feasibility and scheduling, design style, economic optimization, and purchasing.

Numerous deliverables exchanges take place during the upstream phase of vehicle project. They often involve many actors, with the difficulty that they are shared across numerous parallel collaborative groups, for coordination and meeting scheduling reasons. The initial organization is made of 93 types of actors, called G_k. The deliverables D_j are affiliated to one or multiple actors.

- Where N_A= number of actors = 93,
- N_D = number of deliverables = 564
- N_G = number of groups = 7 departments

The existing organization **AG** serves as a comparison point with proposed clusters **AC**. **AC** is the result of the clustering of the **AA** matrix.

The existing organization is composed of 7 departments. More than 40 % of the project deliverables are exchanged between departments (between actors who don't belong to the same group). Our aim is to improve error predication and accelerate error correction by maximizing the deliverables exchanged inside groups and minimize the deliverables exchanged inter-groups.

We will employ the proposed DSM modeling framework in order to analyze and improve the development logic of new vehicles, which its initial data are centered on project deliverables. Every deliverable is produced by a transmitter activity and received by one or more activities (receivers), and produced by one actor (transmitter) and received by one or more actors (receivers).

For the upstream phase, the network of project actors "AA" is obtained using Equation 1.

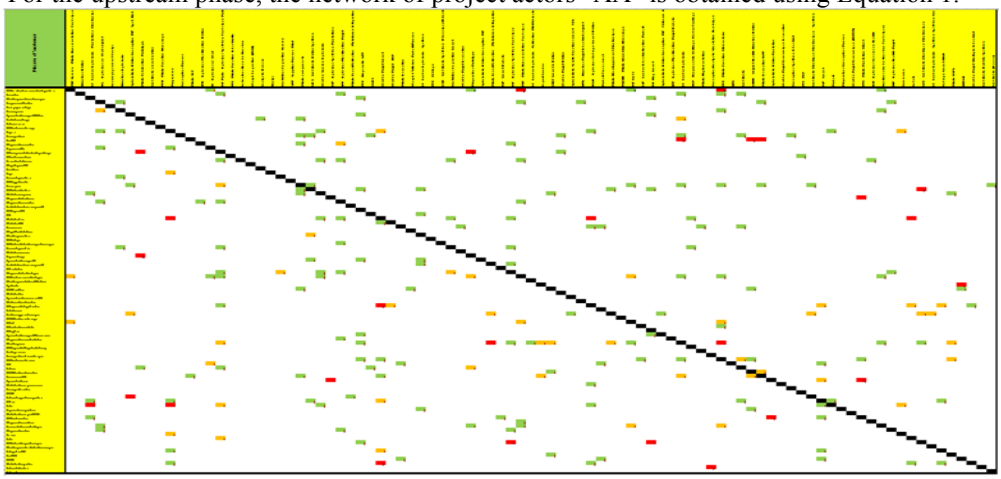

Figure 21. Initial AA matrix in the upstream phase

Figure 21 represents AA which is a weighted directed network of 93 actors within the vehicle development project. Entries in this matrix represent numbers of deliverables exchanged between actors (emitted deliverables in rows and received deliverables in columns). This matrix AA enables direct interactions between actors to be analyzed. If 0< AA (i,j) <4, then the cell is represented in green. If 4<= AA(i,j)<7, then cell is represented in orange. If AA(i,j) >=7, then cell is represented in red.

Gathering information in a global network of exchanged deliverables between types of actors, provide an updated and exhaustive description for local interaction. A local vision on each actor was reported, for example, the Project Planning Engineer receives two deliverables from Module Planning Engineer, six deliverables from Functional Planning Engineer and produces two deliverable to prototype engineer and one deliverable to Technical Documentation Leader, etc. Additionally, in order to reduce ambiguity between project actors and manage their interfaces better, the names of deliverables exchanged (not only the number but also the names of these deliverables) were reported explicitly.

4.2 Results: Aligning the project organization

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.

The network is composed of very interrelated parts, difficult to cut into disjunctive clusters. This requires the application of our proposed strategy to define an adequate process to propose clusters tailored to decision-makers' requirements and constraints.

Several proposals are obtained for **AC,** running simultaneously several algorithms with 15 configurations: by imposing groups of 14, and smaller groups (down to 8). The final

recommendation is made considering the relevance of clusters (within-clusters total value, cluster size, cluster density, number of clusters), in order to keep the algorithmic solution applicable to real-life projects. In addition we did a frequency analysis to indicate the number of times that each couple of actors were put together in a clustered solution. For example Actor A9 and A67 were put together 15 times. A9 receives 13 deliverable from A67.

It seemed interesting, in the exploitation of proposed configurations, to allow some actors to be straddling two clusters, because the algorithms proposed both opportunities (an actor within a cluster or another). A few actors in high interactivity with the overall organization as "Systems Engineering Leader" or "Integration responsible," were assigned as transverse actors. They are out of the clusters but in interface with (almost) everyone. Besides, there are some actors who do not interact with the rest.

This generation of several alternatives enables comparisons and sensitivity analysis. The final solution is a hybrid solution that meets best the needs of the decision maker. It is built using a mix of best clusters from all configurations.

Finally, the most relevant complementary organizational configuration **AC** is compared to the existing one **AG**, and implemented if judged better and applicable. This analysis is done for the three project phases: Upstream Framing, Development and Industrialization. **Error! Reference source not found.** shows the final clustering results for AA by proposing seven new groups (because we had seven existing departments in the initial organization) of interrelated actors. As we can see, all red cells are put within the proposed clusters.

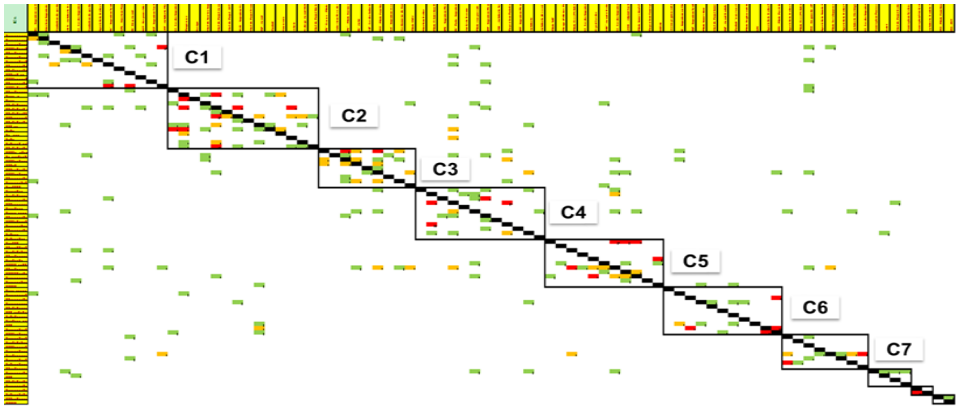

Figure 22. Proposing seven new groups of interrelated actors

The percentage of interactions put within the seven clusters is **81.56%.** This value is by far higher than the value of the initial organization (between existing departments) **59.77%.** This increased percentage permits: 1) to improve communication between connected actors and afterwards decrease project ambiguity; 2) to promote management of interfaces and subsequently reduce risks of propagation; 3) to diminish project uncertainty by increasing ability to pre-evaluate characteristics of the project deliverables as well as the impact of actions and decisions. Particularly, a strong cluster ***C2*** of 14

actors has been identified. ***C2*** contains 147 deliverables exchanged between 14 types of actors during the upstream phase of the project.

In this section, we applied our three-stage clustering process to propose groups of actors involved in numerous deliverables exchanges. These groups are formed using and combining results of several clustering algorithms (See section 3.2) with different parameters (See section 3.1). The first results show different reasons to group actors and different roles of these actors in the network structure and behavior.

The density of proposed clusters **(Number of deliverables exchanged within a cluster /cluster size)** vary 4.25 to 10.5. In the existing organization the maximum density of a department is 4.1.

C4 has 69 deliverables exchanged between its actors. Its density number is by far greater than the best density of departments in the current organization. The implementation of this communication group allows a better control and monitoring of the deliverables. This can improve the error prediction and the quality of produced deliverables. The most central actor in this group is the Safety & Reliability Pilot who produces and receives deliverable from all actors in C4.

5. Summary & Conclusion

Our contribution is a modeling and analysis process to manage/control the impacts propagation between deliverables through actors. We proposed a three-stage process for clustering a network of project actors. The first stage is information gathering, about input data and parameters definition. The second stage consists of running each algorithm many times with several problem configurations. Afterwards, we obtain 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 actors 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 on the results (the number of times the couple of actors were put together). A hybrid solution, that meets at best the needs of the decision maker, is built using a mix of best clusters from all configurations. This approach has been illustrated through actual data in a new product development project in the automotive industry. The industrial application has shown promising results by grouping people according to interdependencies (Jaber et al., 2015), changing more or less the way that actors were initially organized. Forming alternative teams based on exchanged deliverables between project actors, which is complementary to the classical project breakdown structure organization, is an emerging and vital topic to the performance of projects (Jaber et al., 2017). We argue that the approach presented here has a theoretical and practical importance, albeit some insights remain to be improved or discovered.

References

Akao Y., Quality Function Deployment. Cambridge, MA: Productivity Press, 1990.

Blondel V. D., Guillaume J.-L, Lambiotte R., and Lefebvre E. , “Fast unfolding of communities in large networks,” J. Stat. Mech. Theory Exp., vol. 2008, no. 10, p. P10008, 2008.

Borjesson Fredrik and Holtta-Otto Katja, "IMPROVED CLUSTERING ALGORITHM FOR DESIGN STRUCTURE MATRIX," in ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering ConferencePhysical Review, 2012.

Bühler T. and Hein M. Supplementary Material for" Spectral Clustering based on the graph p-Laplacian". 2010.

Miller D. P., Building A Project Work Breakdown Structure Visualizing Objectives, Deliverables, Activities, And Schedules. 2009.

Barry W. , Leite F., and O'Brien W. J., "Late Deliverable Risk Catalog: Evaluating the Impacts and Risks of Late Deliverables to Construction Sites," J. Constr. Eng. Manag., vol. 141, no. 4, p. 04014087, Apr. 2015.

Browning T. R. and Ramasesh R. V., "A Survey of Activity Network-Based Process Models for Managing Product Development Projects," Prod. Oper. Manag., vol. 16, no. 2, pp. 217–240, Jan. 2009.

Bryde D. J. and Joby R., "Product-based planning: the importance of project and project management deliverables in the management of clinical trials," RD Manag., vol. 37, no. 4, pp. 363–377, 2007.

Danilovic M. and Browning T. R. , "Managing complex product development projects with Design structure matrices and domain mapping matrices," Int. J. Proj. Manag., vol. 25, pp. 300–314, 2007.

Dickau M. and King B., "Managing with the delivera-ble in mind," IIE Solut., vol. 31, no. 9, pp. 31–37, 1999.

Eppinger S. D. and Browning T. R. , Design structure matrix methods and applications. Cambridge, Mass.: MIT Press, 2012.

Eppinger S. D. , Whitney D. E., Smith R. P., and Gebala D. A. , "A model-based method for organizing tasks in product development," Res. Eng. Des., vol. 6, no. 1, pp. 1–13, 1994.

Fernandez A., Les nouveaux tableaux de bords des managers, Editions d'organisation. 2011.

Indelicato G. , "Building a project work breakdown structure: Visualizing objectives, deliverables, activities, and schedules," Proj. Manag. J., vol. 40, no. 3, pp. 85–85, Sep. 2009.

Jaber H., Marle F., Jankovic M., "Improving collaborative decision making in new product development projects using clustering algorithms", IEEE Transactions on Engineering Management, 2015.

Jaber H., Marle F., Vidal L.A., Didiez L., "Criticality and propagation analysis of impacts between project deliverables", Research in Engineering Design, 2017.

Leicht E. A. and Newman M. E., "Community structure in directed networks," Phys. Rev. Lett., vol. 100, no. 11, p. 118703, 2008.

Richtnér A. and Åhlström P. , "Organizational Slack and Knowledge Creation in Product Development Projects: The Role of Project Deliverables: ORGANIZATIONAL SLACK AND KNOWLEDGE CREATION," Creat. Innov. Manag., vol. 19, no. 4, pp. 428–437, Dec. 2010.

Steward D. V. , "The design structure system: a method for managing the design of complex systems," Eng. Manag. IEEE Trans. On, no. 3, pp. 71–74, 1981.

Thebeau R. E. , "Knowledge Management of System Interfaces and Interactions for Product Development Process," Massachusetts Institute of Technology, System Design & Management Program, 2001.

Contact: Hadi Jaber, American University of Beirut, Faculty of Engineering and Architecture, Department of Industrial Engineering and Management, 1107 2020 Beirut, Lebanon. +96170241671, Hadi.jaber@aub.edu.lb.