

Organising, valuing and improving the engineering design process

Michael David

► To cite this version:

Michael David. Organising, valuing and improving the engineering design process. Journal of Engineering Design, 2013, 24 (7), pp.524-545. 10.1080/09544828.2013.776214. hal-00874962

HAL Id: hal-00874962 https://hal.science/hal-00874962

Submitted on 22 Oct 2013

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.

Organising, valuing and improving engineering design process

Research for improving the design process has become crucial in new work organisations. Today, projects involve more and more activities and work groups which must be efficiently coordinated. In this paper, a method for structuring cooperative activities in order to improve its global performance is proposed. This methodological approach is based on matrix modelling and graph theory to represent and to structure information dependencies and can be used to specify collaborative frameworks like Web technology environments. Knowledge of a specific design process depends on acquired experience in relatively similar projects. That is why we refer in this work to standards like ISO 9001 requirements and Capability Maturity Model Integration (CMMI). CMMI models are collections of best practices that help organisations to improve their processes. Classical methods decompose a project into several work groups which are then scheduled. This first decomposition level is supplemented with a more detailed structuring level in order to control the size of the working teams. The steps of the design process being defined and planned, the responsibilities inside the working groups are specified. Finally, the method proposes to identify the coupled pairs of actors who generate the best interface for the effectiveness of the cooperative work.

Keywords: Design Project Management, Work Organisation, Design Structure Matrix, Concurrent Engineering, Graph Theory.

1. Introduction

To improve the design process (DP), companies lean on cooperative work organisations, which enable collection of a great number of skills and thus of actors for the development project. But, these organisations require defining, planning and coordinating of the work groups and activities in the most effective possible way. Consequently, it is important to provide methods that enable to efficiently structure and control the DP.

With the emergence of Concurrent Engineering, numerous research works and publications (Gebala and Eppinger, 1991; Kusiak, and al., 1994; Krishnan, and al.,

1997) were developed for organisation and scheduling of cooperative activities. The different approaches are based on modelling tools to describe the design activities and their dependencies (Steward, 1981; Yassine, and al., 2003; Luh, and al., 2011). They use partitioning methods to decompose a work organisation into a set of work groups and to organise them according to their dependencies (Harary, 1962).

A refinement of the groups obtained by partitioning methods is proposed. The objective is to improve lead time and workload of these groups. The application of a spectral algorithm allows identifying independent task groups capable of working in parallel. This decomposition allows to minimize reworking and repetition between tasks. The method proposes a set of decomposition criteria relative to the balancing of workload and a model for the estimation of development time and cost. The model allows to estimate and to compare various work organisations.

The method is derived from the analysis of a complex DP in order to deduce its optimal organisation. For this, the recommendations of the ISO 9001 standard (ISO, 2008) are applied, which define different requirements to control and to plan the design and/or development of a product and/or service:

- the steps of the DP are defined and planned,
- the required activities for review, verification and validation are well identified,
- the responsibilities and authorities for the DP are well identified,
- the interfaces between groups involved in the DP are managed.

For the analysis of the information flow between the actors of a DP, different techniques based on the graph theory are proposed in order to help a quality manager to determine the fulfilment of the requirements. Methodological utilities are applied to the same design example along this paper. Section 2 presents the related work and the integrative framework for the proposed methods. In section 3, a method for

decomposing and planning the steps of the DP is defined. The responsibilities inside the obtained groups and the interfaces between groups to ensure an efficient communication are specified in section 4. Section 5 synthetises all the propositions regarding with the Capability Maturity Model and presents the COOP'R software.

2. Context and Positioning

2.1 Frame of application: interactions and iterations

To improve a concurrent process like a design project, the new work organisations tend to take into account all of the relevant information as soon as possible not only on the final goal but also on the way to achieve it. These organisations need work structures which enable to collect a great number of skills or actors, but especially they require making these actors cooperate in the most effective possible way (the term "actor" is generic and indicates the whole of the individuals implied in an activity. Thus we indifferently use the words actor and activity). However, this evolution does not simplify the management and the control of the design process; rather it adds a tremendous amount of inter-tasks coupling which makes the overall work considerably more difficult (Browning, 1998). Our ambition through this study is to propose an integrated approach which allows to progressively define a work organisation and to manage its inherent complexity. The approach is founded on the study of the activity dependencies. In this context, there are several works that are based on the analysis of the information dependencies between activities. The Design Structure Matrix (DSM) developed by Steward (1981) is one of the fundamental works in this field. DSM is detailed in section 3. We use DSM to analyse the DP. Several works used the DSM model in order to manage concurrencies of activities during a DP. Krishnan (Krishnan, and al., 1997) presented a framework to manage the overlapping of DP activities. This model is founded on sensitivity and evolution flows between upstream and downstream

activities. The concept that we have called "review" in this paper is the discretisation of sensitivity and evolution flows. Hu and al. (2003) have proposed a constraint-driven execution plan in order to maximize concurrency in DP. This execution plan is based on a perfect knowledge of the precedence constraints between activities so that a possible sequence of the activities could be defined. The objective of this work is to organise complex and collaborative DP, i.e. a set of strongly interconnected activities for which an execution plan is impossible to define.

Indeed, the particularity of the design engineering is the introduction of iterations in the process (Luh, and al, 1999). Iterations are cycles or return back in the process. Safoutin (2003) has proposed to place the iteration problem in a central position especially in design problem. We think iterations are necessary to develop innovative design solutions but, as noticed in the literature, it is crucial for manager first to find the role and the origin of all iterations and to efficiently control its scale. Because it is the main risk in the scheduling activity, we can say that a good overview of the iterative phenomenon leads to a good control of the process (Unger and Eppinger, 2010). Presence of iterations in a DP imply to supplement the primary work related to the precedent activities and to report the changes. The consequent is a cycle of continual improvement which can considerably increase the timeout of a project.

2.2 The Capability Maturity Model (CMM)

CMM was initiated in 1991 by the U.S. Department of Defence with the goal to obtain control of the quality of their software suppliers (Software Engineering Institute of Carnegie Melon University, 2010). CMMI was transposed to engineering systems development. This guidance establishes a set of areas which should be faced by the organisations in order to determine the level of maturity of their practices and to improve their processes. The model defines five maturity levels indicating how structured the development process is being performed (Figure 1). A *maturity level* is a well-defined evolutionary stage toward achieving a mature process. Each level comprises a set of process goals that, when satisfied, stabilize an important component of the development process and results in an increase of the process capability of the organisation.



Fig. 1. Five Maturity Levels of the CMM

CMM provides models or standards which determine the actions making it possible to cross the maturity levels. But it does not really define a strategy to effectively implement these models i.e. how to carry out the actions. In this paper, we define a set of techniques to implement an integrated approach to manage and to organise DP through the predefined path of the five CMM levels.

2.3 Methodology

This section defines the methodology for applying the CMM practices to a DP. Continuous process improvement is based on many small, evolutionary steps rather than revolutionary innovations. The CMM provides a framework for organising these evolutionary steps into five maturity levels that lay successive foundations for continuous process improvement. These five maturity levels define an ordinal scale for measuring the maturity of a design process and for improving its capability. At the first level (*Initial*), a process is usually disorganised and not sufficiently defined. Only inputs and outputs of the process are established. The first step is to discipline the process in order to decompose it in sub processes and to reach the maturity level 2 (*Repeatable*). The actions to be implemented are typical of planning or project management.



Fig. 2. Five Maturity Levels of a Development Process

We propose to use the Design Structure Matrix (DSM) and a partitioning methodology in order to define intermediate states or milestones for the global process. The evolution from the second to the third maturity level (*Defined*) is to characterize the process so that team members understand their roles and responsibilities within the process. Several algorithms are proposed to decompose the interconnected activities with guarantee of the workload balancing, and to identify roles and responsibilities inside the process.

To reach the level 4 (*Managed*), the process must be predictable, i.e. the quantitative objectives for quality and process performance are established and used as criteria in managing the process. We propose to use an iterative estimation method based on the Work Transformation Matrix (WTM). We discuss how this tool allows to predict criteria of a design process.

The focus of the level 5 (*Optimizing*) is to continually and incrementally improve the process performance. We propose estimation models for the parallel and serial decompositions of a process. These models are used to define different organisational

strategies. The organisational structure is flexible to satisfy constraints and to react to the intermediary results.

The next sections detail the proposed methods to pass from a level to the next one.

3. Structural Organisation of the Design Process

3.1 General Decomposition of the DP

The first step of the proposed method aims at determining an initial organisation of the cooperative activities of a DP. It enables to schedule the activities such as those:

- can be executed only after they receive all the information required from their predecessors (Serial activities),
- do not depend on others tasks (Parallel activities),
- are interdependent and must be executed simultaneously (Coupled activities).

This step is based on the work relative to the management of complex systems (Steward, 1981). A product design project is modelled by means of a matrix called Design Structure Matrix (DSM).

	aı	a_2	a₃	a_4	a_5	a_6	a7	a ₈	a ₉	a 10	a 11
a ₁	х		1			1					
a_2		х			1			1			1
a_3	1		х			1					
a_4				х						1	
a_5					х	1	1		1		
a_6	1		1			х					
a ₇		1	1		1		х			1	
a_8		1		1		1		х			1
a_9			1		1				х	1	1
a_{10}				1						х	
a ₁₁		1							1		х

Fig. 3. Initial Design Structure Matrix of the design example

In a DSM, an identically labelled row and column represent a design activity (Fig.3). The elements "1" within each row identify which activities must contribute to information for the proper achievement of the DP. The initial DSM depicted on figure 3 represents the exchanges between eleven activities of a design project. The upper

triangle visualises unknown information and the lower, known information. Thus, elements "1" in row a_3 indicate that the activity a_3 requires information produced by the activities a_1 and a_6 . The information to be transferred from the activity a_1 is known because it has finished its work. But the information transferred from the activity a_6 is unknown and must be estimated by a_3 because a_6 has not yet started its activity. The order of the rows (or columns) indicates the chronology of activities, i.e. the sequence to start design activities. The matrix is initially not structured (unspecified chronology) and doesn't show any visible connected activity.

The objective is to find a sequence of activities that enable the matrix to become lower triangle. The partitioning process consists in rearranging the initial matrix by interchanging rows and swapping the corresponding columns in order to achieve a more organised work sequence i.e. to enable the different actors to work on valid information. The method aims at identifying the strongly related components inside the matrix (Harary, 1962). The actors are then collected within a coupled task, i.e. a work group where the actors must closely cooperate.

		a ₆	a ₃	a ₁	a ₁₀	a ₄	a ₂	a ₅	a ₇	a ₈	a ₉	a ₁₁
6	a ₆	Х	1	1								
ć	a 3	1	Х	1	_	СТ	1					
6	a ₁	1	1	х			/	CT	2			
a	10				Х	1			/	СТ	3	
6	a ₄				1	Х						
â	a ₂						х	1		1		1
6	a ₅	1						Х	1		1	
6	a ₇		1		1		1	1	Х			
6	a ₈	1				1	1			Х		1
6	39		1		1			1			х	1
a	11						1				1	Х

t

Fig. 4. Partitioned DSM. Three strongly related components called Coupled Tasks (CT) They are gathered in the matrix in three coupled tasks which are then scheduled in order to remove the "1" on the upper triangle of the matrix and to obtain a partial order (Fig. 4). The two-coupled tasks implying the activities {a1, a3, a6} and the activities {a4, a10} can be executed in parallel because they do not have interaction.

Then, as soon as these two tasks are performed, the third coupled task implying the activities $\{a_2, a_5, a_7, a_8, a_9, a_{11}\}$ will be able to work since it will have all information it requires, transferred from the first two tasks.

The major interest to define a work organisation by partitioning a DSM is to decompose and to schedule the design project. However, the method does not enable to control the size of the groups, i.e. the number of activities or actors within a team. Thus, the application of this method to a complex design project can lead to the constitution of groups with various sizes. In the next level, the interest is related to large groups. We note that a lot of algorithms have been proposed since Harary for the identification of coupled tasks. But in strongly interdependent projects, this phase can give only one coupled task and so, be ineffectual.

3.2 Detailed Decomposition of the DP

The second step is concerned with the reduction of the number of participants inside groups identified in the first decomposition level. Indeed, collaboration in the same work session with too many people can slow down the work performance of this group significantly (Zhao and Liu, 2003).

3.2.1 Performance Estimation of an Organisation

3.2.1.1 Work Transformation Matrix. The Work Transformation Matrix (WTM) model is used to estimate time and workload of a coupled task and so global duration and cost of a design project (Smith and Eppinger, 1997). It is an extension of the DSM, which integrates an assessment of activities in term of estimated time and rework rate (see Fig 5). The diagonal elements represent the time to complete each activity during the first iteration. The off-diagonal elements represent the strengths of dependence between activities, giving rise to the transfer of work, or rework, involved in the iterations. It is allowed that each activity creates a deterministic amount of rework for other activities.



Fig. 5. WTM model (applied on the second Coupled Task of the example) <u>Time matrix *W*</u>: Diagonal elements mean that the activity 10 has an estimated initial time of 4 units and the activity 4 lasts 7 units of time.

<u>Rework matrix A</u>: The element ".2" on the first line means that 20 % of the activity 10 must be reworked after the realisation of the activity 4. Finally the element ".4" on the second line means that 40 % of the activity 4 must be resumed with the results of the activity 10.

3.2.1.2 Identifying performance indicators. The working vector u_t represents the remaining work for every activity to the tth iteration. At first, it remains the totality of the work to be executed for each activity, so the initial vector is $u_0 = (1,1,...,1)^n$ with n, number of activities. After iteration, the working vector is multiplied by the rework matrix: $u_{t+1} = A.u_t$ or $u_t = A^t.u_0$

A total working vector can also be defined:

$$U = \left(\sum_{t=0}^{\infty} A^t\right) \cdot u_0 \tag{1}$$

R is the vector of work times representing the workload of each activity by unit of time (for example people per week) and is expressed by:

$$R = WU = W.(\sum_{t=0}^{\infty} A^{t}). u_{0}$$
(2)

R integrates the time factor. This expression can be reduced. Indeed $\lim_{t \to \infty} (M^t) = (I - M)^{-1}$ if the maximum eigenvalue of a matrix M is lower than 1 (I is the unit matrix). This condition is always true for A, so: $R = W(I - A)^{-1} . u_0$ (3)

E is the total workload representing the cost of the coupled tasks set. E is the sum of all

elements of R:
$$E = \sum_{i=0}^{n} R^{(i)}$$
(4)

n represents the number of activities in the project and the notation $V^{(i)}$ means the ith element of V.

T is duration of a coupled task. *T* is obtained in adding the duration of the longest activity for each iteration: $T = \sum_{t=0}^{\infty} \max[W.u_t]^{(i)}$ (5)

To illustrate, we apply the equations (3), (4), (5) to estimate the workload and the lead time of the second coupled task or CT2 (Fig. 6): At the first iteration, the execution of activities 10 and 4 start. So $u_0 = [1; 1]$

$$R_{2} = W . (I-A)^{-1} . u_{0} = [5.22 ; 10.65]$$

$$E_{2} = R^{(10)} + R^{(4)} = 15.87 \text{ weeks}$$

$$T_{2} = t(\text{iter. 1}) + t(\text{iter. 2}) + ... + t(\text{iter. z}) = \max(W.u_{0}) + \max(W.u_{1}) + ... + \max(W.u_{z-1})$$

$$T_{2} = 7 + 2.8 + 0.56 + 0.22 + 0.05 + 0.02 + ... = 10.65 \text{ weeks}$$

We note that the critical time delay for CT2 comes from the activity 10 during all iterations. Same manner we calculate the durations and workload for the two other coupled tasks with the WTM model of the global process (Fig. 6). We collected the results in Table 1. On figure 6, (x_1 to x_7) express CT1 and CT2 result's that can be directly used by CT3. We do not need to determine these parameters because there is no questioning of the results of these activities by CT3.

	a_6	a_{s}	a_{i}	a_{10}	a4	a_2	as	a_7	as	a_{9}	<i>a</i> ₁₁
a_6	4	.45	.3								
<i>a</i> 3	.65	1.5	.15								
a_{i}	.25	.5	3								
a_{10}				4	.2						
a4				.4	7						
a_2						1	.4		.15		.25
a _s	x,						7	.1		.3	
a_7		x_2		<i>x</i> ₃		.7	.3	4			
a _s	<i>x</i> ₄				x _s	.35			5.5		.2
a,		<i>x</i> ₆		X ₇			.5			4	.3
<i>a</i> ₁₁						.2				.6	1.5

Fig. 6. WTM model of the design example

Criteria (week)	CT 1	CT 2	CT 3
Workload E	36.54	15.87	70.48
Lead Time T	15.85	10.65	18.73

Table 1. Estimated workload and lead time of the three coupled tasks

The WTM model allows estimating the performance of a cooperative work. The performance only depends on working vector R. Indeed R allows to obtain cost E and duration T for a cooperative process. Then specific models will be introduced for the coupled task decomposition, i.e. the realisation in series or in parallel for a set of workgroups.

3.2.2 The Decomposition of a Coupled Task

3.2.2.1 Cooperation graph. The objective is to reduce the complexity of a work task by separating it in sub-tasks. The goal is to find a compromise for structuring and to make collaborating these sub-tasks. The concept of cooperation graphs (CG) is used to study the information flow between actors and then to deduce the positions of each one (Fig. 7). Edge from agent A_j to A_i means that A_j needs information transfer from A_i. Agent A_i cooperates with agent A_j if A_i gives or shares some of its information with A_j.



Aj process can read values x, y owned by Ai

Fig. 7. Cooperation graph.

The concept of CG has been defined by (Diaz, and al., 1996) who applied it to specify network services and protocols for managing agents in team work. A cooperative group is composed of agents who are organised according to the relationships between the actors. Vertices represent the agents; edges represent the relations between them. CG provides a formal representation that enables to analyse the kind of cooperation between the actors according to their relationships (Rondeau, and al., 1999):

- if a relation is symmetrical then the cooperation is bilateral,
- if a relation is transitive then the cooperation is hierarchical,
- if a relation is both symmetrical and transitive then the cooperation is total.

3.2.2.2 Model for the Sequential Decomposition of a Coupled Task. We introduce specific models for the decomposition of a group, i.e. the realisation in series and in parallel for a set of workgroups. We arbitrarily choose to study a coupled task containing 4 strongly connected activities. By identification, we define a model for the sequential execution of a group after a bi-partitioning, i.e. a dichotomy of the set of activities. On the example of the figure 8, the sequential decomposition of the coupled task initializes a first subgroup composed by activities 1 and 2, then in a second phase, a second subgroup (activities 3 and 4). The first phase of the sequential decomposition of a consensus. Then in the second phase, we initialize remaining activities.



Fig. 8. Sequential decomposition of a coupled task

According to their results, we resume the first activities with the preliminary hypothesis that there are no variations of parameters between two iterations. For the calculation of the responsibility R, a factor K, called cutting matrix, is introduced. It allows to take into account only activities executed during the first phase (R_I) . I represents the unit matrix. During the second phase, there is initial work to be completed only on the activities that were not done in the first phase. Iterative rework may need to be done on any activity, whether it is a first or second phase task (R_2) . Model for the sequential decomposition is described by the following equations:

Model for the sequential decomposition							
In the first phase:	$R_1 = W(I - KAK)^{-1} K.u_0$						
In the second phase:	$R_2 = W(I - A)^{-1}(I - K).u_0$						
Total Workload:	$E = \sum_{i=0}^{n} \left(R_1^{(i)} + R_2^{(i)} \right)$						
Total lead-time: $T = \sum_{t=0}^{\infty} \max_{t=0}^{\infty} \max_{t=0}$	$[W.KA^{t}Ku_{0}]^{(i)} + \sum_{t=0}^{\infty} \max[W.A^{t}(I-K)u_{0}]^{(i)}$						

This method allows estimating the performance of a sequential decomposition. However we notice two problems: on the one hand, there is no successful method to find an optimal decomposition in the series (we can only simulate exhaustively the various organisations). On the other hand, a complete serial decomposition does not optimise the development time. It can drive to a repetition phase (phase 2) as long and complex as the process before its decomposition. In the next step, the objective is to define an organisation, which proposes an overlapping of the subgroups.

3.2.2.3 Model for the Parallel Decomposition of a Coupled Task. By identification with equation (3), a model for the parallel execution of a coupled task after a bi-partitioning is defined, i.e. a dichotomy of the set of activities (Fig. 9).



Fig. 9. Decomposition and realisation in parallel of a coupled task.

Phase 1 consists in the independent working of two subgroups. The factors K_1 and K_2 allow defining the composition of each subgroup and are additional. The u'₀ term defines the information flows to be taken into account in the second phase, i.e. only inter-group exchanges. The total lead-time T is the sum of the delay's phases. So T= max [t(R₁); t(R₂)] + t(R₃).

The parameters of the WTM determine the values of vectors R corresponding in a parallel decomposition. Phase 2 is called re-connexion phase. The duration of this stage is going to highly influence the total delay. Instinctively, this approach has to bring a decrease of the total duration of cooperative work because all activities are initialized from the first phase. In the second one, it has to make the working subgroups collaborate efficiently together.



This model enables the comparison of different choices in the project organisation. The approach gives the means to estimate in a quantitative and relevant way the quality of the organisation of DP during decomposition when the border line (or caesura) is chosen. So it remains to define the best way of decomposing coupled tasks.

2.2.2.4 Separation of the Working Groups. In order to minimize reviews and work repetitions in the second phase of the parallel decomposition, the identified subgroups have to be relatively independent (determination of the cutting matrix K). That is the first criterion for a partitioning, which will be applied. The second criterion of decomposition is to balance the workload of each group. The analysis of these criteria conforms to the problems defined in the researches relative to the graph partitioning. The general problem of the group decomposition can be expressed as follow.

Given is a graph G = (V, E) with V the set of nodes, E the set of edges and |V| = n. Decomposition or partitioning of this graph corresponds to the decomposition of the set of nodes V in k sub-groups $V_1, V_2..., V_k$ such as:

- $\sum_{i=1}^{k} V_i = V$ and $V_i \cap V_j = \phi$ with $i \neq j$
- The number of edges connecting its groups is minimal (minimizing inter-group streams)
- The weight of the nodes of every group is appreciably equal (balancing workload)

To solve this problem, several types of algorithms were developed such as spectral algorithms (Pothen et al., 1990), multi-levels (Karypis and Kumar, 1998) or genetic (Todd and Sen, 1999). A spectral algorithm is applied in this context. The interest of this algorithm is to be able to arrange nodes on a scale characterizing the force of dependencies of nodes couples. So the first decomposition criterion is satisfied. Then it remains to choose at which place the cut will be applied. If all nodes have identical weights, the median cut shall be taken to respect the criterion of balancing workload.

The spectral algorithm allows obtaining the cutting matrix K, which gives the optimal decomposition of a coupled task.

It uses the following procedure to partition a graph into two subsets:

• Step 1: Building the Laplacian matrix L of the graph, L = D - A

where A is the matrix such as: $A = [a_{ij}]$ and,

$$a_{ij} = \begin{bmatrix} w(v_i, v_j) & \text{if } (v_i, v_j) \in E_m \\ 0 & \text{otherwise} \end{bmatrix}$$

(w represents the weight of the edges) and D is a diagonal matrix such as $D = [d_{ij}]$ and

$$d_{ij} = \begin{bmatrix} \sum w(v_i, v_j) & \text{if } i = j \\ 0 & \text{otherwise} \end{bmatrix}$$

- Step 2: Calculating the eigenvalues of the Laplacian matrix L;
- Step 3: Identifying the second smaller eigenvalue λ_i and finding his eigenvector y_i (the first smaller eigenvalue is zero and cannot be used to separate activities);
- Step 4: Calculating the median M for the values y_i;
- Step 5: Dividing the set of nodes V into two subsets P₁ and P₂ with the following criteria:
 if y_i ≥ M then V_i ∈ P₁ and if y_i < M then V_i ∈ P₂.

For example, the third coupled task (CT3) obtained after the preliminary partitioning phase (Fig. 4) is considered. The spectral algorithm is applied to the set of activities: $\{a2, a5, a7, a8, a9, a11\}$. The first step consists in building the Laplacian matrix. For that purpose, the cooperation graph of CT3 is symmetrized (Fig. 10). Note that we used here the binary parameters (activities dependencies) instead of the review rates *w* of WTM. Nevertheless the resulting partition is the same one.





Cooperation graph

Symmetrized graph



				6	-1	-1	-2	0	-2
				-1	5	-2	0	-2	0
Lonlogian matrix of the su	-1	-2	3	0	0	0			
Lapracian matrix of the sy	is. $L =$	-2	0	0	3	0	-1		
				0	-2	0	0	4	-2
				2	0	0	-1	-2	5
Its eigenvalues λ are:	λ_2	λ_5	λ_7		λ_8	λ_{g}	Ð	λ_{11}	
M_1 ergenvarues M_1 are.	-0.000	1.9619	2.8985	5 5.	5910	6.87	67	8.6718	3

The second smaller eigenvalue of the Laplacian matrix L is $\lambda_{5.}$ To define the subgroups, we use the corresponding eigenvector, which is:



Fig. 11. Spectral representation of activities interactions for CT3

The activities are gathered according to values of the eigenvector 5 components. Fig. 11 shows the representation of the spectral analysis of CT3: if manager want to balance the team sizes, the 3 largest values in a group and the 3 smallest in the other. So the

application of the spectral algorithm enables to decompose this group into two subgroups: {a2, a8, a11} and {a5, a7, a9} (Fig. 12).



Fig. 12. Decomposition of the Coupled Task 3 with the Spectral Algorithm The analysis of this decomposition shows that the obtained organisation leads to an optimal solution:

- the number of actors in the two groups is balanced (3 in each group),
- the interactions between the two groups are minimized (4 exchanges)
- the number of interactions inside the groups are homogeneous (5 exchanges in the first group and 4 in the second)

We collected in Table 2 the estimated results for the decomposition of CT3 according to the spectral algorithm.

Exe	cution	In series	In series	Parallel
CT3a / CT3b		(a < b)	(b < a)	
	Phase 1	13.62	25.41	39.04
Ε	Phase 2	43.50	26.99	8.09
	Total	57.12	52.40	47.13
	Phase 1	11.16	14.92	10.74
Т	Phase 2	14.92	11.63	10.96
	Total	26.08	26.56	21.70

Table 2. Estimated workload and lead-time (in weeks) for the decomposition of CT3

Note that the serial decomposition leads to a choice between two possibilities according to the order of execution for the subgroups. The decomposition compared to a concurrent realization of CT3 (see Table 1) allows to reduce the workload and so to improve the resources management. For the serial decomposition, the execution of CT3b before CT3a leads to the equilibration of the workload during the two phases and increases the total duration of the process. The parallel decomposition still brings a reduction to the total workload while increasing the lead-time. In this case, main efforts are concentrated in the first phase. There is not an optimal configuration here. The choice must be guided by the availabilities of the resources and the urgent character and not by the finalization of the project. These models allow comparing different choices in the organisation of a process. The automated calculation of criteria gives good help to decide if managers have to decompose a task or not. This approach gives the means to estimate in a quantitative and relevant way the quality of the organisation of cooperative processes during decomposition as far as we choose the border line.

The proposed approach to decompose a complex cooperative process into several load balanced workgroups and to schedule them onto several development process steps is based on two phases. First, the analysis of information flows with the DSM allows a general decomposition of the process in coupled tasks. In a second phase, these tasks are separated with a spectral algorithm in relatively independent groups to be able to work in parallel. The benefit of the spectral algorithm compared with another algorithms is that we can choose the number (and so the size) of the sub-groups obtained from the decomposition of a coupled task. These two levels define the steps of the DP and a basic structure for the organisation.

4. Defining the Responsibilities in the Design Process

To determine the main responsibilities of the design process, three kinds of actors can be identified by means of the Cooperation Graph analysis:

- the validating members,
- the task managers,
- the key members.

4.1 Identification of the validating members

<u>Validating members</u> are actors who are authorised to validate or to reject propositions on models, components, values ... These actors are able to react on a proposition expressed by another actor.

That means the kind of interactions in a coupled task is either:

- a bilateral cooperation (interdependency between two actors),
- a total cooperation (all the links between actors are used and the matrix is full),
- or actors implied in a circuit and not in a hierarchical cooperation.

In a general manner, the following definition (definition 1) is proposed to identify a validating member inside a work group.

<u>Definition 1</u>: The actor is employed in strongly related components of a cooperation graph, which constitutes the set of validating members.

4.2 Identification of the Task managers

<u>Task managers</u> are actors who are able to observe the whole work process and to immediately react by sending new orders or updating information to the other actors. For this, a task manager must have a central position in a work group. This position and the information exchange with the other actors result from the expected skills of a task manager. Thus, the survey to specify this actor is to analyse the different distances between each activities. Concepts of eccentricity and anti-eccentricity defined in the graph theory are used.

<u>Definition of the eccentricity</u>: let G be a graph and v be a vertex of G, the eccentricity of the vertex v is the maximum distance from v to any vertex.

That is $e^+(v) = max \{ d(v,w) : w \text{ in } V(G) \};$

<u>Definition of the anti-eccentricity</u>: let G be a graph and v a vertex of G, the antieccentricity of the vertex v is the maximum distance from any vertex to v.

That is $e^{-}(v) = max \{ d(w,v) : w \text{ in } V(G) \}.$

In the context of a cooperative work organisation, the eccentricity represents the ability of an actor to capture information inside the group with a minimum of mediators and the anti-eccentricity represents the ability of an actor to propagate information inside the group with a minimum of mediators. Thus, to identify the actor who has a central position in a DP, i.e. a task manager, the sum of eccentricity and anti-eccentricity for each actor is calculated. The actor with the minimum value is the task manager. When two or more actors have the same result, the actor who has the least number of elements corresponding to the max $d(a_j, a_i)$ and the max $d(a_i, a_j)$ is the task manager. Thus, the definition that is applied to identify the task manager is:

<u>Definition 2</u>: The task manager in a CG is identified by the actor a_i with: $\min\{e^+(a_i)+e^-(a_i): a_i \text{ in } V(CG)\}$. When two or more actors have the same value, the task manager is the actor a_i with: $\min\{\max\{d(a_i, a_j)\}\cup\{\max\{d(a_j, a_i)\}: a_i, a_j \text{ in } V(CG\}\}$

The groups 1 and 2 (TC1 and TC2) in the example are composed of activities engaged in a total cooperation. Inside these groups, each actor has the same distance with any other one. So these workgroups have no task manager and the actors employed in this cooperation have the same level of responsibilities (validating members).

a2	0	1	2	1	2	1	2	4
a5	2	0	1	4	1	2	4	6
a ₇	1	1	0	2	2	2	2	5
ag	1	2	3	0	2	1	3	7
ag	2	1	2	3	0	1	3	5
^a 11	1	2	3	2	1	0	3	5
e ⁻ (a _i)	2	2	3	4	2	2		

Table 3. Eccentricity and anti-eccentricity of activities in Coupled Task 3

 Applied to the third coupled task depicted in the Fig. 4, we identify that this group collects (Table 3):

- a set of validating members: the whole actors involved in the coupled task;
- a task manager: the actor 2, who minimizes eccentricity and anti-eccentricity.

The application of these two definitions used to analyse dependencies in a work group enables to check if the group can efficiently work in a cooperative way, i.e. it collects the right skills; otherwise there is a need to restructure it.

4.3 Interfaces between work groups

4.3.1 Objectives

One of the major difficulties to manage a DP is to ensure the information consistency between the different work groups. The objective required in the DP as defined in the ISO 9001 is to both identify:

- a way for communication, which ensures an efficient transfer between groups,
- the responsibilities for information consistency within each work group.

Firstly, optimising the information transfer needs to globally minimize the distance between the producer and the consumer of information. That means it is important to avoid mediators for transfer of information from an actor to another one. The objective is to reduce semantic losses on the information. Secondly, the information consistency must be the responsibility of specific actors named key-members. The key-members are coupled pairs of actors who have to control the interface between groups. The key-members between two groups (for example group α and group β) involve two actors:

- one actor is responsible of the *information consistency* inside its group;
- one actor is responsible of the *information exchange* with the other group.

4.3.2 Identification of the key-members

To identify these actors, the definition 2 for a set of two groups, which must be interfaced, is applied. Four cases can be analysed (Fig. 13):

<u>Case 1:</u> There is only a hierarchical cooperation between the two groups. In this case, the objective is to determine only one pair of actors to interface the groups. Let's consider two groups α and β . For each coupled pair of actors (a_i, a_j) , with a_i belongs to the group α and a_j belongs to the group β , expression $e^+(a_i) + e^-(a_j)$ is calculated. $e^+(a_i)$ represents the ability of the actor a_i to capture information in the group α with a minimum of mediators and $e^-(a_j)$ represents the ability of the actor a_j to propagate information in the group β with a minimum of mediators. The key-members of this cooperation are the coupled pair of actors (a_i, a_j) with: **Min** { $(e^+(a_i)+e^-(a_j))$ }



Fig. 13. Cases for two group's interfaces

<u>Case 2:</u> The coupled pairs of actors competent to interface two groups are involved in a bilateral cooperation. That means that the abilities of the actors to transfer information in two directions must be taken into consideration. In this case, one pair of actors

responsible for managing the communication involving two groups is identified. The same properties defined in the case 1 are used but applied in the two directions. Thus, the key-members of this cooperation are the pair of actors (a_i, a_j) with a_i belongs to the group α and a_j belongs to the group β and: Min { $(e^+(a_i) + e^-(a_i) + e^+(a_j) + e^-(a_j))$ }

<u>Case 3:</u> The communication between the two groups is managed by pairs of actors in hierarchical cooperation. But, at least one pair of actors transfers information only in one direction and at least one pair transfers information in the other direction. That means that two coupled pairs of actors responsible for managing the group interface must be selected. To identify the key-members, the properties defined in the case 1 are applied, firstly to identify the pair of actors responsible to transfer information from the group β , and secondly to identify the pair of actors responsible to transfer information from the group β to group α .

<u>Case 4:</u> This case is a hybrid solution between the case 2 and the case 3. Some coupled pairs of actors are involved in a bilateral cooperation and others in a hierarchical cooperation. In this case, the pairs involved in a hierarchical cooperation are not taken into account and the properties defined in the case 2 are consequently applied. The reason of this choice is that it is better to have only one pair of actors in a bilateral cooperation, who manages all the communication between two groups than two coupled pairs of actors in a hierarchical cooperation.

<u>Remark:</u> When pairs of actors have the same position in cooperation, the definition 2 described in the previous section is applied. The objective is to decide between the set of all possible pairs.

Let a_i , a_{α} belong to group α and a_j , a_{β} belong to the group β , the expression to be evaluated for each pair of actors is: $Min|\{max\{d(a_i, a_{\alpha})\}\}\cup\{max\{d(a_{\beta}, a_j)\}\}|$ for cases 1&3

 $Min |\{max \{d(a_i, a_{\alpha})\}\} \cup \{max \{d(a_{\alpha}, a_i)\}\} \cup \{max \{d(a_j, a_{\beta},)\}\} \cup \{max \{d(a_{\beta}, a_j)\}\}| \text{ for cases } 2\&4$

To illustrate this study, the example of the interface analysis connecting the second and the third coupled task (group 2 and 3) depicted in the Fig. 4 is used. The analysis is applied to three couples: (a_{10}, a_9) ; (a_{10}, a_7) ; (a_4, a_8) . The cooperation is hierarchical (case 1), thus only one pair have to be choose. The results are collected in Table 4. Activities in group 2 are engaged in a total cooperation, so $e^+(a_i)$ is equal to 1 for activities 4 and 10. $e^-(a_i)$ for activities in group 3 result from Table 3.

Couple (a _i , a _j)	$e^+(a_i)$	e (a _j)	$e^+(a_i) + e^-(a_j)$
(a ₁₀ , a ₉)	1	2	3
(a ₁₀ , a ₇)	1	3	4
(a ₄ , a ₈)	1	4	5

Table 4. Key members for interface between second and third group

Thus, the activities 9 and 10 have the smallest value and constitute the pair, which will be responsible for the interface between these two groups. Same manner key members for the interface between the groups 1 and 3 are leaders of activities 5 and 6.



Major control point

Fig. 14. Structural and strategic organisation of the design example

The figure 14 resumes the main results of the proposed method related to the structural organisation applied on the design example. In order to balance the workload and the

role repartition, task manager for group 1 and 2 are chosen in minimizing the extragroup exchanges number. A principal control point is defined for review and validation of the project after realisation of group 1 and 2. Other control points can be defined after work iteration of group 3.



5. Integrating the methods and discussion

Fig. 15. Overall CMM improvement framework for a design process The figure 15 synthetises the proposed approach and their associated techniques. The proposals allow to structure a design process in homogeneous and coherent work groups

(level 1), to schedule those groups in time (level 2) and to identify major roles in and between groups (level 3). If the process is mature and the designers have enough knowledge about organisational practices, they can evaluate the DP performances (level 4). Once the process was assessed and had been controlled throughout development, its optimization can be treated as part of a step of uninterrupted improvement (level 5).

The set of the techniques can be applied by algorithms mainly based on the Graph Theory. Considering the CMM levels (see Fig. 1), DSM technique associated with a partitioning algorithm allow a macroscopic organisation of the DP and to achieve a *structured process* (level 2). The spectral algorithm allows to reduce the complexity of Coupled Tasks by decomposition with a workload balancing. We can notice that, contrary to other matrix partitioning algorithms, the spectral algorithm lets the designer able to choose the number and the size of the workgroups when he decomposes a coupled task. After decompositions, the different work groups can be scheduled. In order to set the DP as a *standard and coherent process* (level 3), Cooperation Graphs are used to identify intra- and inter-groups roles and responsibilities. In order to set the DP as a *predictable and evaluate process* (level 4), a method based on WTM is proposed to evaluate the performances of a coupled task.

Models for serialization or parallelization of the coupled tasks allow to optimize the DP. These different solutions for organising the design lead to push the DP towards a *continual improvement process* (level 5).

Those techniques are easily programmable and can so be inserted into a tool to constitute helping functionalities for managing cooperative work. A software named COOP'R, which integrates and automates the set of methods described in this paper, can be used to organise and estimate cooperative processes like design projects (see Fig. 16). COOP'R was developed with Matlab® due to their capacities of using and treating

matrix. Indeed the most part of methods used in this work is based on matrices, from the analysis and definition of cooperative design processes to the determination of performance indicators. This framework gives the possibility to independently develop functions and to create a toolbox with reusable components for other developers. COOP'R can be used whatever the maturity of the process is, because you can used only the specific methods regarding the amount of knowledge about the process. Therefore developed features have been structured by sections that comprise each specific action relative to a level of maturity (or knowledge) about the organisation of a collaborative process. This structure allows to apply the complete approach or just to use only specific groups of features.



Fig. 16. Overview of the COOP'R software *Decomposition of the Design Process (see section 3)*

The first class of functionality is to use COOP'R essentially as a representation and cooperative process partitioning tool. The main developments concern functions like:

- direct or Wizard publishing, modification, backup of a design process
- representation of a process under different formalisms (DSM, WTM, CG, ...),
- decomposition of the process in working groups,
- partitioning with workload balancing groups.

Responsibilities and evaluation of the Design Process (see section 4 and 5)

The second class of features of COOP'R is focused on the characterization of cooperative process with regard to the internal organisation of the process and the definition of performance indicators. Developments concern functions of:

- determination of the manager of a working group,
- determination of the key (interface between two working groups),
- specification of the characteristics of the design activities,
- evaluation of the performances of a process (duration, workload and cost).

The software allows understanding in advance the performances of a new work organisation. However the decomposition can be systematically applied, even if it seems promising. Indeed the management of the cooperative work is related to uncertainty and risk. Sequential decomposition strategy is more risky than the concurrent strategy, because the results of the first subgroup can be inoperable for the second if it is not at all involved in the first phase.

Parallel decomposition strategy is even more risky, because you have to be more familiar with the process to separate two subgroups, whose works are related. Therefore, the choice of decomposition will be always related to the maturity of the process. More the maturity is high and more the manager can have certainty on the veracity and validity of the model. The risk is lower in the case of advanced knowledge about the cooperative process, so improvement strategies can be more effective.

No extensive experience is required to determine whether two design activities are dependent or not. However to characterize activities (durations) and dependencies (review rates), interviews with specialists of the DP or the use of similar previous projects will be required. Indeed, small changes in the WTM model input parameters can cause large variations in the assessment of performance. Therefore, this step can be done only with an advanced level of maturity. For example, a first interview about the strength of dependency between activities can lead to identify if it is low, medium or high. The corresponding rework rates should also be respectively 0.1, 0.3 and 0.6. In this case, we can study and compare work organisations in order to define the best. However, there may be a substantial gap between expected and actual performance. If our knowledge on a DP increases the rates can be refined to be closer to reality.

6. Conclusion

In this article, an organising method for complex cooperative design processes is proposed. The approach offers modelling and structuring elements to methodologically improve the collaborative work with the respect to ISO 9001 requirements and CMM guidance. The first point to study is the specification of the process steps. After a first decomposition based on the information flows analysis, several coupled tasks are obtained. If these tasks consist of to many activities, a second decomposition level, which allows reducing the size of the groups, is defined. This decomposition is based on a spectral algorithm, which takes into account the workload balancing and the minimization of the number of exchanges between groups. The method needs to evaluate the duration of each step in order to determine the optimal organisation. For that purpose, on the basis of the works of Smith and Eppinger, (1997) relative to concurrent design systems, a model for parallel approach is used. This model enables to compare different organisations according to criteria like duration and workload. The proposed decomposition allows finding a compromise which optimises both delays and costs on the basis of the volume of exchange information.

The definition of the responsibilities among the work groups is also required. In a complex process involving a lot of exchanges, it is important to clearly define the actors who must ensure the consistency of information produced and consumed by the different groups. The approach proposes the identification of these actors. This technique takes into account criteria that minimize the distance of the information linking the producer and the consumer. The objective is to minimize the semantic losses during the information routing.

This method can give the inputs for defining network needs and for example using the Web technology as well as possible. The main objective is to link the methodological and the technological parts of the cooperative work design. The proposed algorithms and the software COOP'R can be used to both configure collaborative environment (software tools for cooperation like workflows or web services, network architecture ...) and to define the work organisation management (structural and temporal decomposition, responsibilities ...).

REFERENCES

- Browning T.R. (1998). Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development, *Ph.D. Thesis*, M.I.T., Cambridge.
- Diaz M., F. Vernadat, Villemur (1996). Spécification et Réalisation Formelles de Systèmes Coopératifs. *Ingénierie des protocoles, CFIP 96*, pp. 357-376.
- Gebala D.A. and S.D. Eppinger (1991). Methods for Analyzing Design Procedures. *ASME International Conference on Design Theory and Methodology*, pp. 227-233.

- Harary F. (1962). A Graph Theoric Approach to Matrix Inversion by Partitioning. *Journal of numerical mathematics*.
- Hu J., J. Liu and B. Prasad (2003). A Constraint-driven Execution Plan for Maximizing Concurrency in Product Development, *Concurrent Engineering Research and Applications*, Vol. 11(4), pp. 301-311.
- ISO/CD2 9001 (2008). Quality management systems. Requirements. ISO Standard.
- Karypis G. and V. Kumar (1998). A Parallel Algorithm for Multilevel Graph Partitioning and Sparse Matrix Ordering. *Journal of Parallel and Distributed Computing*, Vol. 48, pp. 71-95.
- Krishnan V., Eppinger S.D., Whitney D.E.(1997). A model-based framework to overlap product development activities, *Management Science*, Vol. 43, n°4: pp. 437-451.
- Kusiak A., Larson and J. Wang (1994). Reengineering of Design and Manufacturing Processes. *Computers and Industrial Engineering*, Vol. 26, n°3, pp. 521-536.
- Luh P.B, F. Liu and B. Moser (1999). Scheduling of design projects with uncertain number of iterations, *European Journal of Operational Research*, Vol. 113, pp. 575-592.
- Luh D., Y. Ko, C. Ma (2011). A structural matrix-based modelling for designing product variety. *Journal* of Engineering Design, Vol. 22, Iss. 1, pp. 1-29.
- Pothen A., H.D. Simon and K.P. Liou (1990). Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM Journal of Matrix Analysis and Applications*, Vol. 11, n°3, pp. 430-452.
- Rondeau E., Z. Idelmerfaa and J. Richard (1999). Identification of Group Organization during a Design Process by Using Cooperation Graphs. *Concurrent Engineering Research Application*, ISSN 1063-293 X, Vol. 7, pp. 191-199.
- Safoutin M.J. (2003). A Methodology for Empirical Measurement of Iteration in Engineering Design Processes, *Ph.D. Thesis*, University of Washington.
- Smith R.P. and S.D. Eppinger (1997). A Predictive Model of Sequential Iteration in Engineering Design. *Management Science*, Vol. 43, n°8, pp. 1104-1120.
- Software Engineering Institute of Carnegie Mellon University. 2010. Improving processes for developing better products and services (CMMI-DEV), Technical report CMU/SEI Version 1.3. http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm
- Steward D.V. (1981). The Design Structure System: A Method for Managing the Design of Complex Systems. *IEEE Transactions on Engineering Management*, Vol. EM-28, n°3, pp. 71-74.
- Todd D. and P. Sen (1999). Distributed Task Scheduling and Allocation using Genetic Algorithms. *Computers and Industrial Engineering* n°37, pp.47-50.
- Unger D. and S. Eppinger (2010). Improving product development process design: a method for managing information flows, risks, and iterations. *Journal of Engineering Design*, Vol. 22, Iss. 10, 2010, pp. 689-699.
- Yassine A., D. Whitney, S. Daleiden, J. Lavine (2003). Connectivity maps: Modeling and analysing relationships in product development processes. *Journal of Engineering Design*, Vol. 14, Iss. 3, pp. 377-394.
- Zhao J.M. and Z.J. Liu (2003). Team-size Constraint based moderate Decoupling Algorithm for large Coupled Task Sets. 10th International Conference on Concurrent Engineering Research and Applications, ISBN 90-5809-622-X, pp. 123-128.