HAL
open science

Managing and Organizing Concurrent Processes According to the CMM Levels

Michael David, Zahra Idelmerfaa, Jacques Richard

To cite this version:

Michael David, Zahra Idelmerfaa, Jacques Richard. Managing and Organizing Concurrent Processes According to the CMM Levels. Concurrent Engineering: Research and Applications, 2005, 13 (3), pp.241-251. 10.1177/1063293X05057113 . hal-00571183

HAL Id: hal-00571183

https://hal.science/hal-00571183v1

Submitted on 1 Mar 2011

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.

CONCURRENT ENGINEERING: Research and Applications

Managing and Organizing Concurrent Processes According to the CMM Levels

Michael David,* Zahra Idelmerfaa and Jacques Richard

Centre de Recherche en Automatique de Nancy (CRAN - CNRS UMR 7039)
Université Henri Poincaré, Nancy I, BP 239, 54506 Vandoeuvre Lès Nancy cedex, France

Abstract: Research for reduced lead time and cost in concurrent process development (CPD) becomes crucial in new work organizations. Today, the concurrent processes involve more and more activities and work groups which must be efficiently coordinated. In this article, the authors propose a framework for structuring concurrent activities in order to improve the global performance (lead time, workload, risks, etc.). Many research works aim at proposing methods for management and control of concurrent development activities, but these methods are often 'isolated' and specific to an activity. There is no global solution that covers the entire development process and the set of related management and organizational activities. In this article, the authors used the SEI's (Software Engineering Institute) CMM (Capability Maturity Model) as a guideline to define an approach for the management and the control of a set of concurrent development activities. According to the CMM levels, the proposed approach is decomposed into four improvement steps that direct the organization's focus to manage its organizational development process. The first step applies basic project management methods to decompose and to schedule a process into several work groups. The second step addresses activities such as controlling the size of the obtained work groups, and characterizing each actor in a group. In the third step, a model based on the work transformation matrix (WTM) is proposed for the performance estimation of an organization. The last level aims at continually optimizing the organizational processes. For this purpose, models for sequential and parallel execution of processes are proposed and compared.

Key Words: concurrent process development, work group management, CMM, DSM.

1. Introduction

New communication and information technologies made exchanges (e.g. voices, documents, specifications of design) can become dematerialized and distant. To improve processes, companies lean on concurrent and cooperative work organizations, which enable them to collect a great number of skills and thus of actors within the development process. But, these organizations require definition, planning, and coordination of the work groups and activities in the most effective way. Consequently, it is important to provide methods that enable them to structure and to control concurrent process development (CPD). In this article, the objective is to develop an approach based on a sequence of improvements in order to manage, organize, and control the concurrent activities. The approach imposes an acquisition and an effective division of information. This study is not interested in ways of creating information or knowledge, but on the manner of treating and using it to optimize the concurrent process.

Numerous approaches have been developed for the organization and scheduling of concurrent engineering activities [1,2]. These approaches are based on project management techniques that allow one to analyze the activity dependencies and also to partition and schedule a work organization into a set of work groups.

The approach developed in this article is based on the guidance provided by the capability maturity model (CMM). CMM seems to be a suitable model to guide the improvement of the development process organizations [3]. Section 2 aims at positioning the context of the research work. Some related work are presented and the CMM model is introduced. Section 3 presents a set of complementary techniques adapted to the scheduling of concurrent activities. The application of these techniques allow the transition from the maturity level 1 to the maturity level 2. Section 4 addresses activities such as controlling the size of the work groups and characterizing each actor in a work group. The purpose of these activities is to form integrated teams for the development of the work and thus to reach the maturity level 3. Section 5 presents a set of methods to introduce quantifiable measurements of the process and thus to pass to the maturity level 4. In Section 6, models for evaluating broken up processes are introduced.

*Author to whom correspondence should be addressed.
E-mail: michael.david@cran.uhp-nancy.fr

1063-293X/05/03 0241–11 $10.00/0 DOI: 10.1177/1063293X05057113
© 2005 Sage Publications

These models allow continuous optimization of the development process, necessary to cross the last stage of maturity and to reach the maturity level 5. The last section deals with discussion and conclusion.

2. Context and Positioning

2.1 Frame of Application and Related Work

To improve a concurrent process like a design project, the new work organizations tend to take into account, as soon as possible, the whole of the relevant information not only on the final goal but also on the way to achieve it. These organizations need work structures which enable them to collect a great number of skills or actors, but more importantly they require strategies for making these actors cooperate in the most effective way possible (the term 'actor' is generic and indicates all the individuals implied in an activity. Thus the words actor and activity are used interchangeably). However, this evolution does not simplify the management and control of the development process, rather it adds a tremendous amount of inter-task coupling, which makes the overall work considerably more difficult [4]. It is the ambition of the authors through this study to propose an integrated approach, which can progressively define a work organization and manage its inherent complexity. The approach is founded on the study of the activity dependencies. In this context, there are several studies that are based on the analysis of the information dependencies between activities. The design structure matrix (DSM) developed by Steward [10] is one of the principal works in this field. DSM is detailed in Section 3. Several studies used the DSM model to manage concurrencies of activities during a development process. Krishnan et al. [18] presented a framework to manage the overlapping of CPD activities. This model is founded on sensitivity and evolution flows between upstream and downstream activities. Hu et al. [19] have proposed a constraint-driven execution plan in order to maximize concurrency in CPD. 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 organize complex and collaborative CPD, i.e., a set of strongly interconnected activities for which an execution plan is difficult to define.

Indeed, the particularity of concurrent engineering is the introduction of iterations in the process [5]. Iterations are cycles or return back in the process. Safoutin and Smith have proposed in their study that the iteration problem be placed in a central position [6], especially in design problems. An important mission of a manager is to find the role, the importance, the scale, and the origin of each iteration. Iteration is the main risk in the scheduling activity. It can be said that a good overview of the iterative phenomenon leads to a good control of the process. The presence of iterations in a development process implies to rework or to supplement the primary work related to the precedent activities and to report the changes. The consequence is a cycle of continual improvement.

2.2 The Capability Maturity Model

The capability maturity model (CMM) was initiated by the US Department of Defense (DoD) with the goal of obtaining control of the quality of their software suppliers [7,8]. CMM establishes a set of areas that should be considered by the organizations to determine the level of maturity of their practices and to improve their processes. This model defines five maturity levels indicating how structured the development process being performed is (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 result in an increase in the process capability of the organization.

CMM provides models or standards that 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 [9]. In this article, the authors have defined a set of techniques to implement an integrated approach to manage and to organize concurrent development processes through the predefined path of the five CMM levels.

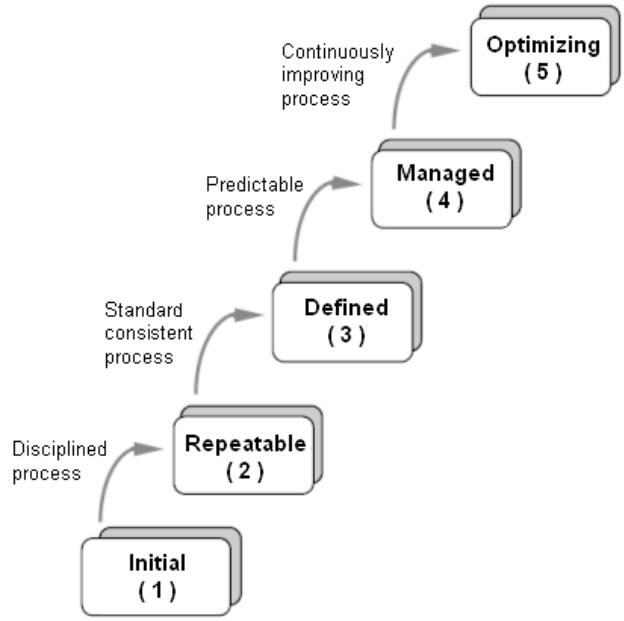

Figure 1. The five maturity levels of CMM.

2.3 Methodology

This section defines the methodology for applying the CMM practices to a CPD. Continuous process improvement is based on many small, evolutionary steps rather than revolutionary innovations. The CMM provides a framework for organizing 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 development process and for improving its capability (Figure 2). At level 1 (*Initial*), a process is usually disorganized 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 subprocesses and to reach the maturity level 2 (*Repeatable*). The actions to be implemented are typical of planning or project management.

The use of the design structure matrix (DSM) and a partitioning methodology is proposed 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 the 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. The authors propose to use an iterative estimation method based on the work transformation matrix (WTM). How this tool allows one to predict the criteria of a concurrent process is discussed.

The focus of the level 5 (*Optimizing*) is to continually and incrementally improve the process performance. The estimation models for the parallel and serial decompositions of a process are proposed. These models are used to define different organizational strategies. The organizational structure is flexible to satisfy constraints and to react to the intermediary results.

Sections 3–6 detail each maturity level and the proposed methods to pass from the present level to the next one.

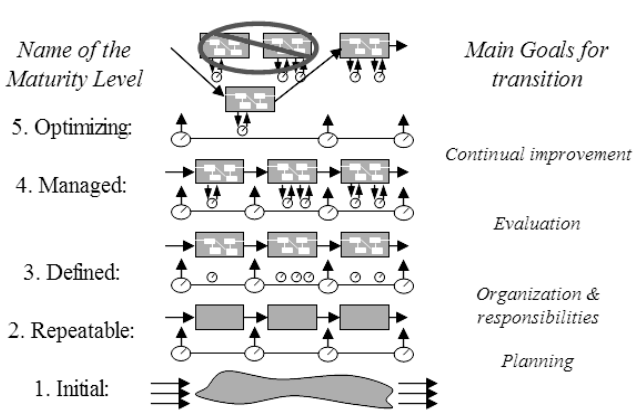

Figure 2. The five maturity levels of a concurrent process development (CPD).

3. From the Initial to the Repeatable Maturity Level: Process Planning

This phase is adapted for new or badly controlled processes. It consists in analyzing the needs and the relationship between activities. The ambition is to find constant or predicable relations. To reduce the complexity of the problem, a matrician tool was used to analyze the flow of information between activities. This tool allows one to decompose a process into steps and to define control points between steps.

3.1 Identifying the Interdependencies between the Activities

The first phase aims at determining an initial organization of the activities. The decomposition is based on the analysis of information flow. It enables one to identify and schedule the activities:

- that can be sequenced so that each one can be executed only after it receives all the information it requires from its predecessors (serial activities),
- that do not depend on others' activities (parallel activities),
- that are interdependent and that could be executed simultaneously (connected activities).

3.2 Managing the Complexity: The Design Structure Matrix

This approach comes from works relative to the management of complex systems [10]. A process is modeled by means of a design structure matrix (DSM). In a DSM, an identically labeled row and column represent an activity. The elements '1' within each row identify which activities must contribute toward the information for the proper achievement of the development process [11]. The order of the rows indicates the chronology, i.e., the sequence to start the activities. The initial DSM depicted in Figure 3 represents the exchanges among eleven actors of a process. The upper triangle visualizes unknown information and the lower, known information. Thus, elements '1' in row a_3 indicate that the actor a_3 requires information produced by the actors a_1 and a_6. The information to be

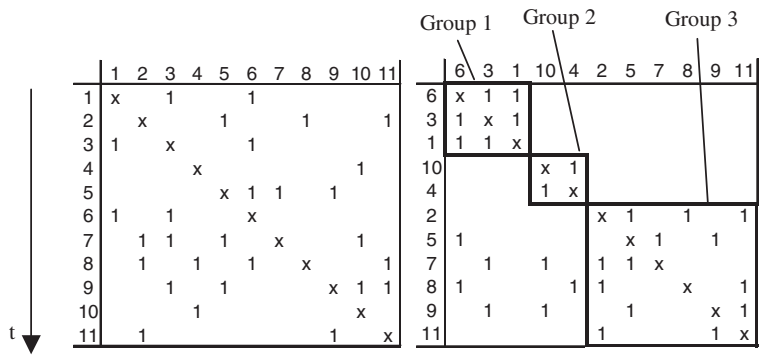

Figure 3. Initial to partitioned DSM of the example.

transferred from the actor a_1 is known because he has finished his work. But the information transferred from the actor a_6 is unknown and must be estimated by the actor a_3 because a_6 has not yet started his activity.

3.3 Partitioning the Process

The matrix is initially not structured (unspecified chronology) and does not show any visible connected activity. The objective is to find a sequence of development activities that enables the matrix to become lower triangular. It consists in rearranging the initial matrix by interchanging rows and swapping the corresponding columns in order to achieve a more organized work sequence that allows the different actors to work on valid information. The method aims at identifying the strongly related components inside the matrix. The actors of the same strongly related component are then collected within a coupled task, i.e., a workgroup where the actors must closely cooperate.

In the example, three strongly related components constituted of the actors $\{a_1, a_3, a_6\}$, $\{a_4, a_{10}\}$, and $\{a_2, a_5, a_7, a_8, a_9, a_{11}\}$ are identified. They are gathered 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. The partitioned matrix indicates that the two coupled tasks implying the actors $\{a_1, a_3, a_6\}$ and the actors $\{a_4, a_{10}\}$ can be executed in parallel because they do not have the interaction. Then, as soon as these groups have performed, the third group implying the actors $\{a_2, a_5, a_7, a_8, a_9, a_{11}\}$ will be able to work, since it will have all the information it requires, transferred from the first two groups. The major interest in defining a work organization by the research of strongly related components in a DSM is to decompose and schedule the process. However, the decomposition does not enable one to control the size of the groups, i.e., the number of actors within a group. Thus, the application of this method to a complex process can lead to the constitution of groups of various sizes. If all the activities are strongly connected, this phase can lead to the constitution of only one group (no decomposition).

The authors are then interested in the study of large groups. In the example, the focus is mainly on the third group (or coupled task 3). Indeed, to collaborate at the same time, in the same work session, a too great a number of people can strongly slow down the performance of this workgroup [12]. Thus it is important to be able to manage this aspect. For that, a second decomposition phase is performed to reorganize a group containing too many participants.

4. From the Repeatable to the Defined Maturity Level: Work Group Organization

In the case of processes better known or already used, more information about the relationship between entities is known. Previous experiences give references for choosing the indicators of performance. At this level, the major steps of the process are planned. To improve the process capacity, the authors studied the cooperations between the various entities more precisely. The objective is to study the information exchanges and to reduce the complexity of a workgroup by separating it into subgroups. A spectral algorithm is proposed for defining the way of separation. The roles and responsibilities of every actor are also defined.

4.1 The Cooperation Graph (CG)

Each matrix is associated with a directed graph or cooperation graph (CG) [13], where the actors are represented by nodes and a dependence is symbolized by an oriented edge (Figure 4).

Aj process can read values x, y owned by Ai. Edge from agent Aj to Ai means that Aj needs information from Ai. Ai cooperates with Aj if Ai gives or shares some of its information with Aj. The concept of CG is used in this article to study the interactions between the actors involved in a CPD. It provides a formal representation that enables the analysis of the kind of relationship between the actors.

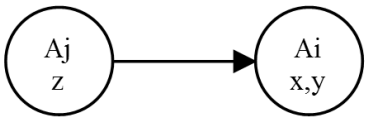

Figure 4. Cooperation graph.

4.2 Separation of the Working Groups: A Spectral Algorithm

To minimize reviews and work repetitions, the authors wanted the identified subgroups to be relatively independent. That is the first criterion that will be applied for a partitioning. 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 follows. Given a graph $G=(V,E)$ with V being the set of nodes, E, the set of edges, and $|V|=n$, the partitioning of this graph corresponds to the decomposition of the set of nodes V in k subgroups $V_1, V_2, \ldots, V_k$ 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 genetic [14], spectral [15], or multi-levels [16]. The authors chose to apply a spectral algorithm. The interest of this algorithm is to be able to arrange nodes on a scale characterizing the force of dependencies of nodes two for two. The first decomposition criterion is satisfied. Then it remains to choose which place the cut will be applied on. If all the nodes have identical weights, the median is taken as caesura to respect the criterion of balancing the workload. The spectral algorithm uses the following procedure to partition a graph into two subsets:

- Building the Laplacian matrix, L of the graph, $L = D - A$, where A is the matrix such as: $A = [a_{ij}]$ and,

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

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

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

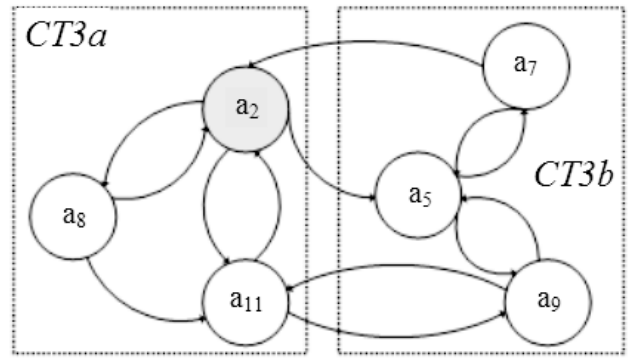

Figure 5. Decomposition of the coupled task 3 (CT3) with the spectral algorithm.

- Calculating the eigenvalues of the Laplacian matrix L.
- Identifying the second smaller eigenvalue λ_2 and finding the corresponding eigenvector, y.
- Calculating the median M for the values y_i.
- Dividing the set of nodes V into two subsets P_1 and P_2 with the following criteria:

if $y_i \geq M$, then $V_i \in P_1$ and if $y_i < M$, then $V_i \in P_2$.

For example, consider the third coupled task obtained after the preliminary partitioning phase {a2, a5, a7, a8, a9, a11} (Figure 5). The application of the previous algorithm allows to decompose this group into two subgroups CT3a: {a2, a8, a11} and CT3b: {a5, a7, a9}.

This decomposition shows that the obtained organization 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 interactions inside the groups are homogeneous (5 exchanges in CT3a and 4 in CT3b).

The first phase based on the DSM method, coupled to the application of a spectral algorithm allows one to decompose a complex cooperative process into several load balanced workgroups and to schedule them onto several development process steps. The review or validation activities are defined with a central position in the global process. In the example, the authors proposed to place this verification activity after the parallel realization of CT1 and 2. In the next section a method that allows the identification of the responsibilities for every subprocess was proposed.

4.3 Responsibilities for the Concurrent Process

To determine the main responsibilities of connected processes, two kinds of actors can be identified by

means of the CG analysis: the *validating members* and the *task managers*.

Validating members (VM) are actors who are authorized to validate or to reject propositions on models, components, and values. These actors are able to react on a proposition expressed by another one. In a general manner, the following definition is proposed to identify a VM inside a workgroup:

Definition 1: The actors implied in strongly related components of a CG constitute the set of VM.

Task managers (TM) are actors able to observe the whole process and to immediately react by sending new orders or updating information to the other actors. For this, a TM must have a central position in a workgroup. The survey to specify this actor is to analyze the distances between each actor. To achieve this, the concepts of eccentricity and anti-eccentricity defined in the graph theory were used.

Definition of eccentricity: 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 $e^{+}(v) = \max\{d(v, w): w \text{ in } V(G)\}$.

Definition of anti-eccentricity: Let G be a graph and v a vertex of G, the anti eccentricity of 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 work organization, eccentricity represents the ability of an actor to capture information inside the group with a minimum of mediators and anti-eccentricity represents the ability of an actor to propagate information with a minimum of mediators. Thus, to identify the actor who has a central position, the sum of its eccentricity and its anti-eccentricity, is calculated for each actor. 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 element corresponding to the $\max d(a_j, a_i)$ and the $\max d(a_i, a_j)$ is the TM.

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

$$\min |\{\max\{d(a_i, a_j)\} \cup \{\max\{d(a_j, a_i)\}: a_i, a_j \text{ in } V(CG)\}|$$

These two definitions enable the determination of roles and to control if the group can efficiently work in a cooperative way, i.e., it collects the right skills, otherwise to restructure it. CT1 and CT2 activities are respectively engaged in total cooperation. Each actor has the same distance with another one. So these workgroups have no TM and the actors implied have the same level of responsibilities (VM).

Table 1. Eccentricity and anti-eccentricity of activities in coupled task 3 (CT3).

	a_2	a_5	a_7	a_8	a_9	a_{11}	$e^{+}(a_i)$	$e^{+}(a_i) + e^{-}(a_i)$
a_2	0	1	2	1	2	1	2	4
a_5	2	0	1	4	1	2	4	6
a_7	1	1	0	2	2	2	2	5
a_8	1	2	3	0	2	1	3	7
a_9	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	–	–

The two definitions for CT3 were applied. Table 1 identifies that this group collects:

- a set of validating members: all the actors involved in the coupled task;
- a task manager: the actor 2 (that minimizes the expression $e^{+}(a_i) + e^{-}(a_i)$).

On this principle, couples (key members) responsible for transfer of information between groups can also be determined [20]. At this maturity level, the organization of CPD is totally defined: the groups are determined and planned (structural organization); roles and responsibilities are distributed (strategical organization). The next objective is to estimate the performance of a work organization.

5. From the Defined to the Managed Maturity Level: Quantitative Measurement

This stage is adapted to the processes for which the organizational structure exists, but has not yet been evaluated. It enables the determination of relevant control criteria about the overall process and its subprocesses. At this stage, experts can give quantitative information about the activities (lead time, costs, risks) or about the sensitivity (impact) of one activity compared to another. So a model for the performance estimation of a set of interconnected activities is proposed.

5.1 Characterization of the Dependencies between Activities – Modeling Elements

The process can be modeled by tracking the information that is exchanged between activities. This modeling approach includes two types of elements, namely activities and reviews probabilities. The characteristics of the model's elements evolve with the iteration count.

With every activity, the characteristics of execution time and activity cost per time unit are associated, as described in Figure 6. To create a flexible and

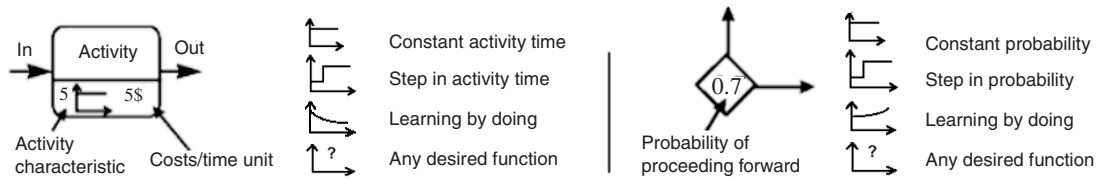

Figure 6. Model elements: activity and review probability.

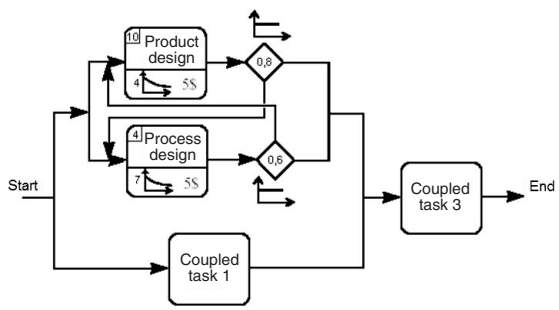

Figure 7. Model focused on the second coupled task.

	a_6	a_3	a_1	a_{10}	a_4	a_2	a_5	a_7	a_8	a_9	a_{11}
a_6	**4**	.45	.3								
a_3	.65	**1.5**	.15								
a_1	.25	.5	**3**								
a_{10}				**4**	.2						
a_4				.4	**7**						
a_2						**1**	.4		.15		.25
a_5	x_1						**7**	.1		.3	
a_7		x_2		x_3		.7	.3	**4**			
a_8	x_4				x_5	.35			**5.5**		.2
a_9		x_6		x_7			.5			**4**	.3
a_{11}						.2				.6	**1.5**

Figure 8. WTM model of the example.

accurate model, the activity characteristics vary with the number of iterations done. The review element models the probability of proceeding forward to the next activity, otherwise the process flows back to an earlier activity. The review is evaluated with the help of a random function. The characteristics of the review are also a function of the number of iterations done.

The global process is also depicted in Figure 7. The authors focused on activities 4 and 10, which comprise CT2 in the example. Work generated by the 'product design' activity is transferred to 80% to the next step of the process (CT3) without modifying the work of the 'product design' activity; i.e., 20% of activity 4 must be reworked with each iteration of activity 10. The observations also show that 40% of activity 10 must be revised after each realization of activity 4.

For the continuation of this study, the authors will make the following hypothesis as an example. Activity time is considered with the characteristic 'learning by doing' and it is supposed that the activities have equivalent costs. In the same way, the probability of proceeding forward is considered to be a constant function. Note that these assumptions can be differently formulated in order to be adapted to specific cases.

5.2 Performance Estimation of an Organization: The Work Transformation Matrix (WTM)

In this direction, [17] developed the WTM model to estimate the lead time and the workload of a development process. The WTM is constituted by two types of information. The diagonal elements represent the time to complete each activity during the first iteration. The off-diagonal elements represent the strength of dependence between activities, giving rise to the transfer of work, or rework, involved in the iterations.

It is assumed that each activity creates a deterministic amount of rework for other activities. While reconsidering the second group in the Figure 8, diagonal elements mean that activity 10 has an estimated time of 4 units and activity 4 lasts 7 units of time. The element '0.2' means that 20% of activity 10 must be reworked after the realization of activity 4. Finally, element '0.4' means that 40% of activity 4 must be resumed after the realization of activity 10. This model is in conformity with the model described in Section 5.1. Now, the probability of proceeding forward is not considered any more, but the probability of returning back in the process. This approach allows the additional working times to be taken directly into account. WTM model uses two matrices: a time matrix W and a rework matrix A. 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 $u_0 = (1, 1, \ldots, 1)^T$ with T, number of activities. After iteration, the working vector is multiplied by the rework matrix.

$$\text{So} \quad u_{t+1} = A \cdot u_t \quad \text{or} \quad u_t = A^t \cdot u_0$$

The total working vector can 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 men per week) and is expressed by:

$$R = WU = W \cdot \left(\sum_{t=0}^{\infty} A^t \right) \cdot u_0 \tag{2}$$

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 the matrix M is lower than 1 (I is the unit matrix). It is always true for A, so:

$$R = W(I - A)^{-1} \cdot u_0 \tag{3}$$

E is the total workload of the set of CT. E is the sum of all the elements of R:

$$E = \sum_{i=0}^{n} R^{(i)} \tag{4}$$

n represents the number of activities in a CT and notation $R^{(i)}$ means the ith element of vector, V.

If all the activities have the same cost per unit of time, the total cost $C = E$. In the general case, the activities have different costs C_i and:

$$C = \sum_{i=0}^{n} C_i \cdot R^{(i)} \tag{5}$$

T is the duration of a CT. T is obtained by adding the duration of the longest activity for each iteration:

$$T = \sum_{t=0}^{\infty} \max[W \cdot u_t]^{(i)} \tag{6}$$

This technique allows for the comparison between different organizations according to criteria like duration, workload, or cost. These equations are applied to estimate the workload and the lead time of the second coupled task: At the first iteration, the execution of activities 10 and 4 start. So $u_0 = [1; 1]$.

$$\begin{aligned}
R_2 &= W \cdot (I - A)^{-1} \cdot u_0 = [5.22; 10.65]; \\
E_2 &= 15.87 \text{ resources} \cdot \text{week} \\
T_2 &= t(\text{iter. } 1) + t(\text{iter. } 2) + \cdots + t(\text{iter. } x) \\
&= \max(W \cdot u_0) + \max(W \cdot u_1) + \cdots + \max(W \cdot u_{x-1}) \\
T_2 &= 7 + 2.8 + 0.56 + 0.22 + 0.05 + 0.02 + \cdots \\
&= 10.65 \text{ weeks}
\end{aligned}$$

The duration and the workload for the two other CT can be calculated. The results are collected in Table 2.

The authors remark that the optimization criteria depend only on R. Indeed the working vectors R allow to obtain an estimation about the costs and lead time of every subprocesses of a concurrent work.

Table 2. Estimated workload and lead time of the three coupled tasks.

	CT1	CT2	CT3
Workload	36.54	15.87	70.48
Lead time	17.08	10.65	19.08

6. From the Managed to the Optimizing Level of Maturity: Continuous Optimization

When the process is evaluated and mastered, the phase of maturity is called *managed*. In the last level (*optimizing*), it remains to continuously improve management. Specific models for the decomposition of a group, i.e., the realization in series and in parallel (Sections 6.1 and 6.2) are then introduced. These models are the first proposition to optimize a complex CPD. The authors chose to model processes with CG and to arbitrarily study a group containing four strongly connected activities. By identification, the authors can define a model for the sequential execution of a group after bipartitioning, i.e., a dichotomy of the set of activities.

6.1 Sequential Decomposition

Based on the example in Figure 9, the sequential decomposition of the coupled task initializes a first subgroup composed of activities 1 and 2, and then in a second phase, a second subgroup (activities 3 and 4). The first phase of the sequential decomposition consisted in activities that go to their term. Then in the second phase, the authors initialize the remaining activities. According to their results, the first activities are resumed with the preliminary hypothesis that there are no variations of parameters between the two iterations.

For the calculation of the responsibility R, a factor K, called cutting matrix, is introduced. It allows only activities executed during the first phase (R_1) to be taken into account. I represents the unit matrix. During the second phase, there's initial work to be completed only on activities that were not done in the first one. Iterative rework may need to be done on any activity, whether it is a first or second phase task (R_2).

This method allows for the estimation of the performance of a sequential decomposition. However two problems are noticed: on one hand, there is no successful method to find optimal decomposition in series (only the various organizations can be simulated exhaustively). On the other hand, a complete serial decomposition does not optimize the development time. It can also lead 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 organization which proposes an overlapping of the subgroups.

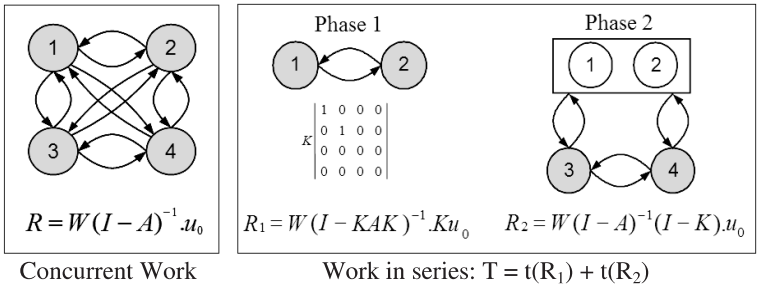

Figure 9. Sequential decomposition of a coupled task.

6.2 Parallel Decomposition

In Figure 10, the realization in parallel for the coupled task depicted in Figure 9 is represented.

In an identical way, the values of vectors R corresponding in a parallel decomposition need to be determined from the parameters of the WTM. For these calculations, the following are defined as:

$$K_1 = K$$

$$K_2 = I - K$$

$$u_0' = (K_1AK_2 + K_2AK_1)u_0$$

K_1 and K_2 allows for the definition of the composition of each group and are additional. u_0' defines the flow to be taken into account in the second phase, i.e., only intergroup exchanges. Phase 2 is called reconnection phase. The duration of this stage will highly influence the total delay. So it is important to limit the length of this phase. Instinctively, this approach has to bring a decrease in the total duration of process because all the activities are initialized from the first phase. Then it remains in a second phase only to collaborate the working subgroups together.

For example, in Table 3 the estimated results for the decomposition of CT3 according to the spectral algorithm are collected. Note that the serial decomposition allows 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 2) allows for the reduction of the workload and so to improve the resource 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 in the total workload, while increasing the lead time. In this case, main efforts are concentrated on the first phase. In this case, there is no optimal configuration. The choice must be guided by the availabilities of the resources and the urgent character and not of the finalization of the project.

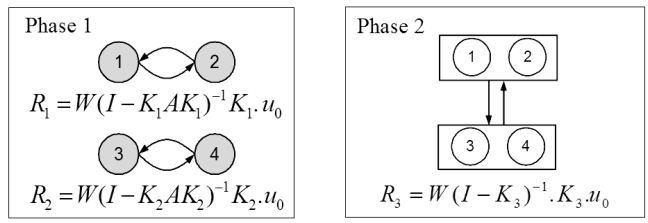

Figure 10. Parallel decomposition of a coupled task.

Table 3. Estimated workload and lead time for the decomposition of CT3.

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

7. Conclusion

In this article, the authors propose an integrated approach for management and organization of complex concurrent development process. The approach describes a set of techniques to apply according to a predefined path based on the five CMM levels. Progressing through the CMM levels allows the authors to ensure the management and the control of a set of concurrent development activities beginning with basic project management techniques until the application of different organizational strategies. The proposed approach is founded on the study of the activity information dependencies. According to the CMM levels, the approach is decomposed into four improvement steps that direct the organization's focus to manage its organizational development process. The first step is based on the information flow analysis within the initial process. This step allows one to identify several groups of activities called coupled tasks. If these coupled tasks

consist in an important number of activities, they can be very difficult to manage. Therefore a second decomposition can be applied to reduce the size of the groups. 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 focus of the second step is to characterize a development process and thus to identify the roles and responsibilities of the actors inside a work group. The objective is to facilitate the information flow and to avoid the information semantic losses. In the third step, the objective is to evaluate the duration of each decomposition in order to determine the optimal organization. For that, this paper proposes, on the basis of the works of [17], a model to analyze complex and interconnected activities. This model allows for the estimation of several criteria like lead time and costs for a chosen decomposition. When all criteria are evaluated, the process is quantitatively controlled. The last step of the approach concerns the optimization according to different criteria. This paper discusses means to define different organizational strategies through the estimation of duration and the cost for a sequential decomposition and a parallel decomposition. The automated calculation of criteria can help decide to decompose a work group or not. The organizational strategies allow for the comparison of different choices in the organization of a development process, resulting in a cycle of continual improvement.

References

1. Kusiak, A., Wang and Larson, J. (1994). Reengineering of Design and Manufacturing Processes, *Journal of Computers and Industrial Engineering*, **26**(3): 521–536.
2. Prasad, B. (1996). *Concurrent Engineering Fundamentals, Volume 1: Integrated Product and Process Organization*, New Jersey: Prentice Hall.
3. Jiang, J.J., Klein, G., Hwang, H.G., Huang, J. and Hung, S.Y. (2004). An Exploration of the Relationship between Software Development Process Maturity and Project Performance, *Information & Management*, **41**: 279–288.
4. Browning, T.R. (1998). Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development, PhD Thesis, M.I.T., Cambridge.
5. Luh, P.B., Liu, F. and Moser, B. (1999). Scheduling of Design Projects with Uncertain Number of Iterations, *European Journal of Operational Research*, **113**: 575–592.
6. Safoutin, M.J. and Smith, R.P. (1998). Classification of Iteration in Engineering Design Processes, In: *ASME Design Engineering Technical Conference*, Atlanta.
7. Carnegie Mellon University (1999). *The System Security Engineering – Capability Maturity Model (SSE-CMM)*, Version 2.
8. Christie, A.M. (1999). Simulation in Support of CMM-based Process Improvement, *The Journal of Systems and Software*, **46**: 107–112.
9. Niazi, M., Wilson, D. and Zowghi, D. (2003). A Maturity Model for the Implementation of Software Process Improvement: An Empirical Study, *The Journal of Systems and Software* (in press).
10. Steward, D.V. (1981). The Design Structure System: A Method for Managing the Design of Complex Systems, *IEEE Transactions on Engineering Management*, **28**(3): 71–74.
11. Malmström, J., Pikosz, P. and Malmqvist, J. (1999). Complementary Roles of IDEF0 and DSM for the Modeling of Information Management Processes, *Concurrent Engineering Research and Applications*, **7**(2): 95–103.
12. Zhao, J.M. and Liu, Z.J. (2003). Team size Constraint-based Moderate Decoupling Algorithm for Large Coupled Task Sets, In: *Conf. on Concurrent Engineering Research and Applications*, pp. 123–128, ISBN 90-5809-622-X.
13. Diaz, M., Vernadat, F. and Villemur, T. (1996). Spécification et réalisation formelles de systèmes coopératifs, Ingénierie des Protocoles, CFIP 96, pp. 357–376.
14. Todd, D. and Sen, P. (1999). Distributed Task Scheduling and Allocation using Genetic Algorithms, *Computers & Industrial Engineering*, **37**: 47–50.
15. Pothen, A. (1990). Partitioning Sparse Matrices with Eigenvectors of Graphs, *SIAM Journal of MAA*, **11**(3): 430–452.
16. Karypis, G. and Kumar, V. (1998). A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs, *SIAM Journal on Scientific Computing*, **20**: 359–392.
17. Smith, R.P. and Eppinger, S.D. (1997). A Predictive Model of Sequential Iteration in Engineering Design, *Management Science*, **43**(8): 1104–1120.
18. Krishnan, V., Eppinger, S.D. and Whitney, D.E. (1997). A Model-Based Framework to Overlap Product Development Activities, *Management Science*, **43**(4): 437–451.
19. Hu, J., Liu, J. and Prasad, B. (2003). A Constraint-driven Execution Plan for Maximizing Concurrency in Product Development, *Concurrent Engineering Research and Applications*, **11**(4): 301–311.
20. David, M., Idelmerfaa, Z. and Divoux, T. (2004). Organizing the Cooperative Work for the Product Development Process, In: *IFAC Symposium on Information Control Problem in Manufacturing*, Salvador Bahia, Brazil.

Dr Michael David

Michael David is a post-doctoral researcher at the Research Center in Automatic Control of Nancy (CRAN), France. He received his Masters degree in mechanical engineering and manufacturing systems and his PhD degree in automatic control at University Henri Poincare of Nancy in 2004. His current research interests include concurrent engineering, complex manufacturing systems, and quality management. His E-mail address is michael.david@cran.uhp-nancy.fr.

Dr Zahra Idelmerfaa

Zahra Idelmerfaa is an associate professor of automatic control at the University Henri Poincare of Nancy, France since 1994. She received her Masters degree in mechanical engineering and manufacturing systems and her PhD degree in automatic control at University Henri Poincare of Nancy in 1989 and 1994, respectively. She undertakes her research at CRAN (Research Center in Automatic Control of Nancy). Her research interests are in collaborative work and quality management. Her E-mail address is zahra.idelmerfaa@cran.uhp-nancy.fr.

Pr Jacques Richard

Jacques Richard is a Professor of automatic control in the Faculty of Sciences of Nancy since 1988 and the director of the University Professional Institute of Mechanical Engineering and Manufacturing Systems at the University Henri Poincare of Nancy. He undertakes his research at CRAN (Research Center in Automatic Control of Nancy) in the field of integrated quality in manufacturing systems.