



HAL
open science

System design and project planning: Model and rules to manage their interactions

Élise Vareilles, Thierry Coudert, Michel Aldanondo, Laurent Geneste, Joël Abeille

► To cite this version:

Élise Vareilles, Thierry Coudert, Michel Aldanondo, Laurent Geneste, Joël Abeille. System design and project planning: Model and rules to manage their interactions. *Integrated Computer-Aided Engineering*, 2015, 22, p. 327-342. 10.3233/ICA-150494 . hal-01384985

HAL Id: hal-01384985

<https://imt-mines-albi.hal.science/hal-01384985>

Submitted on 20 Oct 2016

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.

System Design and Project Planning: model and rules to manage their interactions

Elise Vareilles^{a,*}, Thierry Coudert^b, Michel Aldanondo^a, Laurent Geneste^b and Joel Abeille^c

^a *University of Toulouse - Mines Albi, route de Teillet Campus Jarlard, 81000 Albi cedex 09 France*

E-mail: firstname.lastname@mines-albi.fr

^b *University of Toulouse - ENIT, 47 avenue d'Azereix, 65000 Tarbes France,*

E-mail: firstname.lastname@enit.fr

^c *Pulsar Innovation, Colomiers France,*

E-mail: firstname.lastname@pulsarinnovation.fr

Abstract. This article proposes a model and rules dealing with the management of the interaction between system design processes and project planning ones. An industrial benchmark analysis has reinforced our belief that the interaction between the two processes has to be supported by models, processes and relevant tools. Firstly, after presenting the results of the analysis, the different entities are defined and the one-to-one relationship or bijection between the structure of the system and the structure of the project is made. Then, a model, taking into account design activities and planning activities as well as management of interactions, is proposed in compliance with existing project and design standards. A process of interaction is presented to carry out design and project management. Two interaction modes have been proposed. On the one hand, the structural interaction establishes links between entities of the two domains. On the other hand, the behavioral interaction (subject of this paper) is based on the definition of states for each entity following feasibility and verification criteria, and can thus manage the changes between states. Some rules are defined (precedence and synchronous rules) to forbid certain changes when they are inconsistent and to synchronize them.

Keywords: System Design, Project Planning, Aiding Decisions, Knowledge based Systems

1. Introduction

Because of the increasing complexity of products, the considerable reduction of products time-to-market and de facto, the dwindling of design time without affecting product quality and innovation, today's product and system design processes have to interact more and more closely with all other business processes in a company, such as procurement process, production process, assembly process, maintenance process, etc ([27] and [28]).

In such a context, the building of design activities, from requirement expression to solution design, and the planning and control of these activities, are important tasks ([5] and [25]). Therefore, the interactions between

the design of a system and the building of its design project are crucial and have to be formalized and monitored in order to avoid inconsistencies between these two processes.

Recent project failures (A380 Program, Olkiluoto Nuclear Power Plant) have highlighted the fact that the management of system design, project planning and their interactions, is critical. Much interdisciplinary and concurrent work as well as better communication between the design side and the project side are therefore required ([21] and [33]). Co-operation and interactions across engineering teams and project managers are vital to the planning process and the success of the project in terms of quality, delay and cost ([7] and [43]). This statement is true whatever the complexity of the designed system and the stage of its development.

* Corresponding author. E-mail: elise.vareilles@mines-albi.fr

However, few studies have focused on the management of interactions between design process and planning process or have proposed models and tools that can assist engineers and managers in the task of building design projects in accordance with the design of systems, in planning the design activities, and especially in controlling their execution ([5] and [26]).

In [1], the structural interaction linking system design and project planning and the meta-model supporting this kind of interaction have been described. This structural interaction relies on a one-to-one or bijective link between system and project structures.

In this article, a focus is made on the *behavioral interaction* that allows these two processes to be synchronized, through two specific attributes and nine precedence and coupling rules. Behavioral coupling ensures monitoring and the control of the system design and project planning by imposing synchronous milestones during project development. This synchronization forces engineers and managers to be aware of the situation of the other side, and to take it into account in their own process. Consequently, they can decide together, considering the overall situation, whether to modify some or all of their requirements to reach the project goals under suitable conditions. Behavioral interaction has to be directed regardless of the stage of the design and planning processes, regardless of the system and project complexities, and regardless of the type of companies.

We have to stress that our proposal is rather pragmatic and easily conceivable and may belong to everyone's experience and common sense. However, to the best of our knowledge, only a couple of scientific papers have made explicit this kind of interaction between design and planning ([20], [5]), but neither has formalized them. We introduce the term of *coupling* to denote the identified interactions (structural and behavioral) between the two processes as well as their management. By formalizing such aided interactions we make it possible to integrate them into a PLM (Product Lifecycle Management) software.

The remainder of the article is organized as follows: in Section 2, the context of the study is described. In Section 3, the background is defined and the structural coupling proposed in [1] is synthesized with regard to existing approaches. In Section 4, we focus on synchronous coupling and we introduce and define the specific attributes used in behavioral interaction. In Section 5, the rules, based on the specific attributes, are defined in order to forbid certain changes when they are inconsistent and to synchronize the two processes.

In Section 6, a real-life engineering example, designed with a sample of companies, illustrates our proposals.

2. Context of the Study

The need for interaction between engineers and managers has been consolidated by interviewing fifteen experienced project managers from different companies belonging to the world competitiveness cluster, Aerospace Valley¹. Out of the 10 sample companies, 11% were very small enterprises, 22 % were small or medium enterprises, and 67% were large companies.

The most important results can be summarized as follows:

- All the companies interviewed are confronted with this coupling problem but they have not implemented specific tools to support this coupling process. Only 18% of companies use software or collaborative tools.
- Half of the companies make decisions taking into account design and planning as well as their interactions, during meetings involving the different stakeholders.
- Most of the time (66%), the coupling is performed by means of non-formalized human interactions. Even if companies use software, procedures or standards (45%), their decisions are based on human experience and meetings.

Meanwhile, the complexity of systems and projects is increasing. In a multi-national context, the design of a system is often carried out in several sites involving several partners. In such a context, human-based procedure is no longer sufficient enough to detect problems as early as possible, to analyse information and data, or to react correctly to the situation. Design cycles are more and more reduced and the slightest error in the design or planning sides can jeopardize the success of the whole project. Clearly, the use of adapted and interconnected tools, supporting multi-responsibility projects, for managing these complex design projects is becoming a requirement in such contexts ([21] and [27]). PLM software is designed to help organizations in coping with the increasing complexity and engineering challenges of developing new products [18]. The proposed structural and behavioral couplings easily fit PLM functionalities. Furthermore, they can be added,

¹<http://www.aerospace-valley.com/en/>

without any difficulty, to PLM software without modifying users' habits or the PLM software process. This can extend the scope of their functionalities by imposing a specific structure for systems and projects, and regular synchronous milestones for engineers and managers.

To illustrate the global context of this study, a project (associated with a system that needs to be designed) is considered to be under the responsibility of a program manager, i.e. the highest person in the hierarchy. The program manager interacts with (i) a design manager who works within a design system environment and (ii) a planning manager who works within a project planning environment. The difficulty involved in designing the system, as well as the complexity of the associated project, leads to them being hierarchically decomposed according to the axiomatic design approach [42], [35], [2].

In such cases, systems can be decomposed into sub-systems, leading to the decomposition of associated development projects into associated sub-projects. The corollary is that complex design projects can be decomposed into sub-projects leading to the decomposition of the system in the same manner. Therefore, at each level of the hierarchy, the interactions can be observed. In this context, the program manager, at his/her level, can be seen as a "coupling manager" who fixes orientations and objectives, makes decisions and defines decision parameters for the two other parts. (S)he is also in charge of resolving conflicts.

Within this framework, considering two hierarchical levels, either the design manager of the upper level becomes the program manager of the lower level, or it is the planning manager who takes on this role. This outcome varies from company to company. It is also possible, in some cases at the lowest levels, to have only one person in charge of the three responsibilities.

3. Background and Proposals

In this section, the background of our work is defined and the structural coupling is synthesized with regard to existing approaches. In Subsection 3.1, after reviewing design process definitions, we define what we mean by system design in this paper. In Subsection 3.2, after reviewing planning process definitions, we define what we mean by project planning. Then in Subsection 3.3, after a literature review of interactions between design and project processes, structural coupling is synthesized.

3.1. System Design Process

H. A. Simon [30] first characterized design as a search process. Design can be seen as a project that aims to create a new object or to transform an existing one [16]. Design is also considered to be a knowledge discovery process in which information and knowledge of diverse sources are shared and processed simultaneously by a team of designers involved in the life phases of a product [37], [41]. There are many existing design methodologies described in the literature (see, for instance, the methodologies described in [6], [24], [35], [38], or for a wide panorama, [4]).

Among the widely used methodologies, Axiomatic Design (AD) proposed by Suh [35] is a top-down and iterative approach that makes links between requirements or functions (functional requirements) to be fulfilled and technical solutions (design parameters and process variables). The design process zigzags between the four following domains: needs, solutions, tasks and resources.

From a system engineering viewpoint, the works of the International Council on Systems Engineering (INCOSE) have been considered in detail. Among them, the EIA-632 standard [23], [8] provides some structuring processes for system design broadly used by companies in the electronics domain. It defines a global engineering process that makes it possible to transform customer requirements into technical solutions.

Given these previous studies, the design process proposed in this article is structured as follows:

1. The definition and/or the specification of the requirements,
2. The identification of the technological solutions which can fulfill these requirements,
3. The matching of requirements and solutions, and
4. According to the complexity, the decomposition of the design process up to a certain level of abstraction.

3.2. Project Planning Process

The project planning domain of our work concerns the Project Time Management (PTM) process as defined by the Project Management Institute [25]. In the proposed approach, the project planning definition is a top-down approach, where some kind of global planning is achieved at a high level and is progressively detailed at lower levels by means of sub-projects. This multi-level and multi-project approach makes it pos-

sible to perform adequate multi-level planning by simultaneously considering, at all planning levels, different objectives, constraints, degrees of aggregation, and capacity flexibility (see for instance [14] for a study on hierarchical multi-project planning). In order to define a design project, we consider that project planning involves:

1. Project activities definition,
2. Resource and duration identification,
3. Scheduling activities and resources, and
4. If needed recursive decomposition at the lower level of some activities.

Scheduling of activities and resources is based on several techniques (see, for instance, [15] or [19]) that are not detailed in this article.

3.3. Design and Planning Processes Interactions

In this subsection, firstly, a literature review of interactions between design and project processes is carried out. Secondly, structural coupling, the basis of behavioral interaction, is synthesized.

3.3.1. State of the art

The axiomatic design and the above-mentioned standards allow four interacting domains to be identified:

1. The requirements or specifications,
2. The design solutions,
3. The tasks or activities, and
4. The resources.

The first two domains relate to the system design process and the last two domains to the project planning process. Of the few studies that address this coupling problem, one can mention:

- The study of [20] explicated a link between PBS (Product Breakdown Structure) and WBS (Work Breakdown Structure). The author stated that the WBS tree (or structure) is derived from that of the PBS. The isomorphism of PBS and WBS trees is suggested. Other trees are introduced at the end of the paper, such as a specification tree and a drawing tree.
- The studies initialized at M.I.T. [9] on the use of methods and techniques used in product design in order to facilitate project design. These studies are the source of scientific developments around DSM (Design Structure Matrix), such as those of Lindemann [22] where the interactions between the four identified domains are defined.

- Simultaneously, axiomatic design identifies various domains (Customer Needs, Functional Requirements, Design Parameters and Process Variables) and sees them as interacting [36]. An example of implementation is presented in [13]. The interactions between domains are clearly defined: not only design towards planning but also planning towards design.
- Another approach, introduced by Gero [10], proposes models based on three domains: Function, Behavior and Structure (FBS). The aim of this study is to take into account product behavior (expected and effective) and to inventory in a formal way eight sub-processes of design. However, tools for interactions between processes are not explicitly considered.
- A study that is very close to the problem addressed was undertaken by Stewart and Tate [34] who were interested in the coupling of axiomatic design with project planning in the case of software engineering. Their idea was to associate design variables with the tasks of the development process. This approach was implemented with an ad hoc development coupled with the Microsoft Project® software package and tested in a software engineering context.
- The work of R. Lu [28] describes an approach coupling task management and design. The structure of projects is represented by means of a Working Breakdown Structure (WBS) and is related to a Product Breakdown Structure (PBS). A matrix represents the relationships between both domains.
- In [31] and [32], the authors proposed a Project Product Lifecycle Management approach (PPLM). The aim of their work is to develop a methodology and a software environment for integrating the product that is being developed with the project as undertaken by the company.
- In [5], the authors proposed a model to enable concurrent product design and assembly sequence planning. The aim of their work is to manage vital yet complex and inherent product relationship information to allow such concurrent product design and assembly sequence planning. This paper gives the detailed description of the background and models which highlight the need for a more efficient PLM approach.
- In [11], the authors consider that a design project, in mechanical system engineering, is a network of various interacting design domains such as project, product and process design. The paper presents

an object-oriented design methodology integrated into a human-based and co-operative design life cycle. This methodology was implemented in an educational web-based environment and applied to student design projects.

All these studies indeed confirm the four domains (requirements, solutions, tasks and resources) and the existence of causal links that engender interactions between these four domains. However, except in [34], [11], [28] and [32], no tools are provided to support or aid interactions between both the design and planning processes.

3.3.2. Structural Coupling Synthesis

Structural coupling is based on a structural and hierarchical decomposition of systems into sub-systems and/or projects into sub-projects (guided by their intrinsic complexity). Two coupled entities are then defined:

- A system entity is composed of one set of requirements and one or more solutions (or system alternatives), as shown in the upper part of Fig. 1.
- A project entity is composed of one task of requirement definition and one or more tasks of solution design (or an alternative development task), as shown in the lower part of Fig. 1.

It is important to preserve the semantic of the entities of both domains without mixing the information. Indeed, designers and managers have to keep their respective autonomy of decision-making. Once a decision is made on one side, they have to be informed of any negative impacts (or potential problems) on the other side. In such a case, the current situation should be resolved in a style of mediation: engineers and managers have to share their requirements, constraints and goals in order to react appropriately and ensure the success of the project. This mediation can be supported by the AHP method [29] in order to evaluate different solutions and select the one that best suits system and project requirements at the same time.

Furthermore, in this article we favor a progressive top-down design approach where sub-projects and sub-systems are defined only when required and when their context is well known. For instance, the decision to subcontract the design of a particular sub-system can be taken only after a suitable requirement analysis and when these requirements have been compared with available resources. At this time, the choice of a subcontractor and the definition of the required sub-project, as well as its characteristics and constraints, can be established. Therefore, a bijective link between the BOM

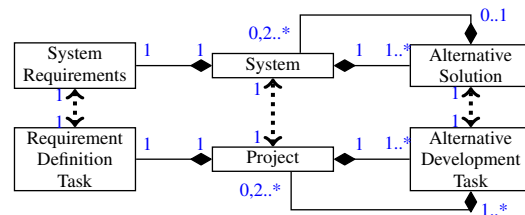


Fig. 1. Integrated Model of Structural Coupling

(Bill of Materials) of a system and the WBS (Work Breakdown Structure) of a project appears to be the only comprehensive solution.

This bijective link is fully compliant with the Axiomatic Design of Suh [35] and its zigzagging methodology, with the work of R. Lu [28] which links PBS and WBS and their relationship with a matrix, and with the work of [31] and [32] who have proposed a Project Product Lifecycle Management approach (PPLM). In our proposal and as shown in Fig. 1, we link also in a bijective way and at any level in the hierarchy:

- A system S and a project P,
- The set of system requirements SR and the requirement definition task PR,
- A system alternative SA and an alternative development task PA.

These entities are the foundations of structural and behavioral couplings (see [1], [3], [39] and [40] for an overview of the meta-model supporting structural coupling).

Regarding industrial benchmark results, the coupling between system design and project planning processes does not only draw bijective links between system and project entities. Structural coupling ensures an isomorphism between system and project structures, but this is not sufficient to manage all their interactions. Managers also need to synchronize the two processes in order to plan and monitor the complete system design project. This synchronization is the main goal of the paper. Our proposal is quite original:

- Firstly, the monitoring is done through two specific attributes which qualify the feasibility and the verification of system and project entities,
- Secondly, these two attributes foster the control of both processes as well as their synchronization thanks to precedence and coupling rules, and
- Thirdly, the synchronization is recursive: it is undertaken at each level and between two consecutive levels of the system and project hierarchies.

Our proposal is fully compliant with the Systems Engineering V-cycle. On one side, the feasibility attribute characterizes the left side of the V-cycle. It corresponds to the ability of being able *a priori* to reach the goals regarding the current context (requirements, constraints, risks and uncertainty). On the other side, the verification attribute characterizes the right side of the V-cycle. It corresponds to the achievement of a solution leading to the success of the project (in terms of quality, schedule and cost). Each time a system is decomposed into sub-systems and a project into sub-projects, a new V-cycle has to be conducted for each of them: the feasibility and verification attributes have to be evaluated for each sub-system and sub-project in order to have an overview of the progress of the whole project.

4. Synchronous Coupling Attributes

Synchronous coupling attributes characterize the requirements SR and alternatives SA for the system S, and on those of the requirement definition PR and alternative development PA tasks for the project P. These coupling attributes help the program manager to better identify potential problems, as soon as possible and respond appropriately to deviations and changes.

In the following sub-sections, we start with the definition of two specific attributes which qualify the feasibility and the verification of systems and projects. We illustrate their meaning for the system and for the project entities at any level of the decomposition. Secondly, we set up the precedence relation that links them, before proposing a synthesis.

In the rest of the paper, we note $X_{hl}^{xRi_xAj}$ where:

- X can be any entity: SR, SA, PR or PA,
- hl corresponds to the level of the entity in the hierarchy of systems or projects,
- xRi_xAj corresponds:
 - * for xRi to the id of requirement entities, such as SR or PR,
 - * for xAj to the id of alternative entities, such as SA or PA.

For example, let us consider a landing gear system S_1^1 composed of two sub-systems noted as S_2^{111} for the wheel and S_2^{112} for the brake, and the associated design project P_1^1 composed of two sub-projects noted as P_2^{111} for the wheel project and P_2^{112} for the brake project. The structure of the system S_1^1 and of the project P_1^1 are illustrated in Fig. 2.

We have to precise that the attribute of feasibility or that of verification for the whole system S or for the whole project P levels are not considered in our proposals. Indeed, these attributes aggregate the information on the feasibility and verification and are closely dependent on those of SR and SA for the system S, and on those of PR and PA tasks for the project P.

4.1. Feasibility Attribute Definition

In preliminary design, the feasibility characterizes the ability of a system, a product or a service to be developed technically and economically. During the feasibility evaluation phase, the design manager and the planning manager have to determine if they can reach a feasible solution with respect to their requirements, for instance performance and technical characteristics for a system or cost and duration for a project.

In order to qualify the feasibility of each system and each project, we propose adding a specific attribute to each entity, noted as Fa . This attribute has three states with different semantics:

- UD, for undetermined and default value, meaning that the design manager or the planning manager has not yet determined if, with respect to the requirements, a solution seems achievable or not,
- OK, for feasible, meaning that it seems possible to achieve a solution with respect to the requirements for the design or for the planning,
- KO, for unfeasible, meaning that it seems impossible to achieve a solution with respect to the requirements for the design or for the planning.

In order to value the feasibility attribute $xR_n^i.Fa$, a design manager or a planning manager firstly have to collect information on their own context and secondly to analyze it considering the degree of uncertainty and the risk of failure. It is important to note that the evaluation of the feasibility attribute relies on the skill and expertise of the managers and their own subjective point of view on the whole situation. For the design side, when the needs and requirements have been gathered, the design manager has to analyze them, pre-design a solution and evaluate the risk of reaching an unfeasible solution. For the project side, when the plan has been scheduled and the resources allocated, the planning manager has to analyze the plan and evaluate the risk of reaching an unfeasible project. It is only after this phase that the managers can switch the value of their attribute to feasible, $xR_n^i.Fa = OK$ or unfeasible, $xR_n^i.Fa = KO$.

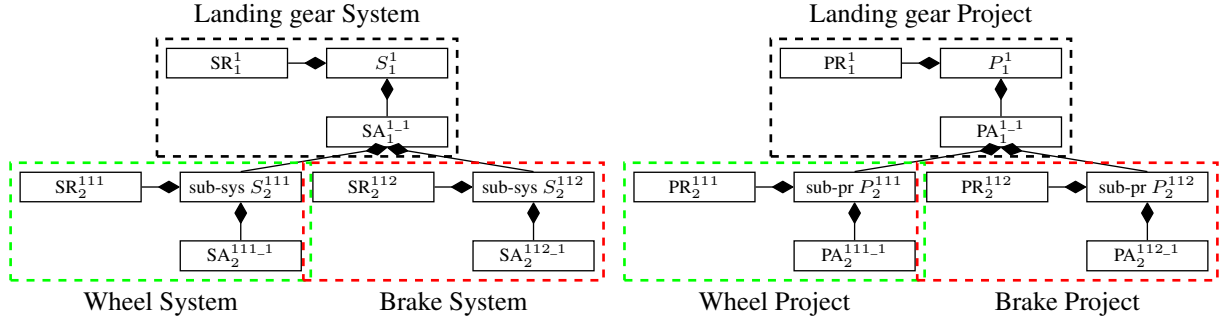


Fig. 2. Example of the Decomposition of a System into two Sub-systems and of a Project into two Sub-projects

For the design side, when it seems that a feasible solution is reachable for the requirements, the designer can start working on the solution itself $SA_n^{i,j}$. While (s)he has not described the functioning principles and possibly decomposed his/her system into sub-systems and identified their requirements, (s)he cannot determine the feasibility of the solution. For the project side, when it seems that a feasible project is reachable for the plan and resource allocation, the planning manager can start working on the alternative development task $PA_n^{i,j}$. When (s)he has scheduled and possibly decomposed it into sub-projects, (s)he can switch the value of this attribute to feasible, $PA_n^{i,j}.Fa = OK$ or unfeasible, $PA_n^{i,j}.Fa = KO$.

We can see that a relation of precedence exists between the attributes of feasibility of xR_n^i and of each potential solution $xA_n^{i,j}$. Therefore, we set up the rule $r1$:

$$r1 : xR_n^i.Fa \neq OK \\ \Rightarrow \forall j \in [1, m], xA_n^{i,j}.Fa = UD \quad (1)$$

In Fig. 3 and 4, the relation described by the rule $r1$ (eq. 1) between the feasibility attributes is synthesized. With regard to the association of these attributes, and without considering any rules or relations, there are $3 * 3 = 9$ possible combinations for each pair of xR_n^i and $xA_n^{i,j}$. Considering the rule $r1$ (eq. 1), only 5 combinations are permissible.

Let us consider now the case of a decomposed solution into l sub-systems and l sub-projects. For the design side, while the designer has not identified each sub-system requirement and is not convinced of the feasibility of his/her decomposed solution $SA_n^{i,j}.Fa \neq OK$, the feasibility of each sub-system requirement cannot be determined: the feasibility attribute of each of the l sub-systems requirements is therefore undeter-

mined $\forall k \in [2, l], SR_{n+1}^{ijk}.Fa = UD$. In the same way, for the project side, as long as the planning manager has not identified each sub-project requirement (estimated resource workload, duration, budget, ...), has not scheduled it and is not convinced of the feasibility of his/her decomposed alternative development task $PA_n^{i,j}.Fa \neq OK$, the feasibility of each sub-project requirement definition task cannot be determined: the feasibility attribute of each of the l sub-project requirement definition tasks is therefore undetermined: $\forall k \in [2, l], PR_{n+1}^{ijk}.Fa = UD$.

We can see that a relation of precedence exists between the attributes of feasibility of a decomposed solution $xA_n^{i,j}$ and of the requirements of each sub-system xR_{n+1}^{ijk} . Therefore, we set up the rule $dr1$:

$$dr1 : xA_n^{i,j}.Fa \neq OK \\ \Rightarrow \forall k \in [2, l], xR_{n+1}^{ijk}.Fa = UD \quad (2)$$

In Fig. 3 and 4, the relation described by the rule $dr1$ (eq. 2) between the feasibility attributes of a decomposed entity and two sub-entities is synthesized.

With regard to the association of these attributes with the decomposed entity xA_1^{i-1} and its two sub-entities X_2^{i11} and X_2^{i12} , and without considering any rules or relations, there are $3 * 3 * 3 = 27$ possible combinations. Considering the rule $dr1$ (eq. 2), only 11 combinations are permissible.

4.2. Verification Attribute Definition

The definition of the verification comes from the ISO 9000:2000 standard [17]. After a complete design, the verification confirms through the provision of objective evidence that the specified requirements have been fulfilled. During the verification phase, the design man-

ager and the planning manager have to justify that the solution matches their requirements to the letter.

In order to qualify the verification of each system and each project, we propose adding a specific attribute to each entity, noted as Ve . This attribute has three states with different semantics:

- UD, for undetermined and default value, meaning that the design manager or the planning manager has not yet proven that the solution exactly matches the requirements to the letter or not,
- OK, for verified, meaning that the solution fulfills the requirements to the letter for the design or for the planning,
- KO, for unverified, meaning that the solution does not match at least one requirement for the design or for the planning.

The system requirements entity is composed of needs (expression of the stakeholders' requirements or the specifications stemming from the upper level if it exists) formalized by means of text or expressed as specifications (both functional and technical) that are declined into technical requirements by the designers. As regards the definition of the verification, we do not associate a verification attribute with the system requirements. Indeed, in this case, it implies verifying that the requirements match the needs expressed by the client, the stakeholders or the upper level. We assume that this is the case and that a large majority of the specifications have already been formalized and tested against the upper-level needs.

For the project side, as regards verification, until a task ends, a planning manager cannot verify that it has run well: the verification attribute of all the tasks is therefore undetermined: $PR_n^i.Ve = UD \wedge PA_n^{i-j}.Ve = UD$. After a task has been carried out, the planning manager has to verify its validity compared to the project requirements: resource workload, real-time consumption, budget, ... When (s)he has analyzed and if necessary, integrated all the information coming from the sub-projects, (s)he can verify the task and switch the value of its attribute to verified or unverified. A relation of precedence exists between the attributes of verification of these two tasks: we cannot verify an alternative development task before a requirement definition task. We set up the rule $r2$:

$$r2 : PR_n^i.Ve \neq OK \\ \Rightarrow \forall j \in [1, m], PA_n^{i-j}.Ve = UD \quad (3)$$

We can remark that a task PR_n^i that has not been properly completed with respect to its requirements $PR_n^i.Ve = KO$ does not compromise the rest of the complete project: the project can still continue but its requirements and objectives (estimated resource workload, duration, budget, ...) have to be reassessed in order to achieve the design.

In Fig. 4, the relation described by the rule $r2$ (eq. 3) between the verification attributes is synthesized. With regard to the association of these attributes, and without considering any rules or relations, there are $3 * 3 = 9$ possible combinations for each pair of PR_n^i and PA_n^{i-j} . Considering the rule $r2$ (eq. 3), only 4 combinations are permissible.

Let us consider now the case of a decomposed solution into l sub-systems and l sub-projects. For the design side, when the design of an alternative has been completed, the design manager has to verify its validity according to the requirements, $SA_n^{i-j}.Ve$. As long as (s)he has not tested the functioning principles and potentially integrated all the sub-systems into the system, (s)he cannot determine the verification of the solution. The design manager has to wait for the verification of each sub-system and for its integration in order to determine the verification of the decomposed solution. If one of the sub-systems is not verified $\exists SA_{n+1}^{ijk-p}.Ve = KO$, de facto the complete solution cannot be either. But the fact that all the sub-systems have been verified $\forall k \in [2, l], \forall p \in [1, m], SA_{n+1}^{ijk-p}.Ve = OK$, is not enough to determine if the decomposed solution matches the requirements to the letter or not because a problem can still occur during the integrations and tests.

In the same way, for a decomposed alternative development task, the project manager has to wait for the verification of each sub-project and for the integration of its information, such as resource workload, real time-consumption, or budget in order to determine the verification of the decomposed alternative development task $PA_n^{i-j}.Ve$. If one of the sub-projects is not verified $\exists PA_{n+1}^{ijk-p}.Ve = KO$, de facto the complete alternative development task cannot be either. But the fact that all the sub-projects are verified $\forall k \in [2, l], \forall p \in [1, m], PA_{n+1}^{ijk-p}.Ve = OK$, is not enough to determine if the decomposed alternative development task matches its requirements to the letter or not because a problem can still occur before its end.

We can see that a logical and precedence relation exists between the attributes of verification of the l lower-level entities SA_{n+1}^{ijk-p} and of the decomposed entity

itself $xA_n^{i,j}$. Therefore, the rule $dr2$ can be applied:

$$dr2 : \forall k \in [2, l], \forall p \in [1, m],$$

$$\left\{ \begin{array}{l} \exists xA_{n+1}^{ijk-p}.Ve = KO \Rightarrow xA_n^{i,j}.Ve = KO \\ \text{or} \\ \forall xA_{n+1}^{ijk-p}.Ve = OK \Rightarrow xA_n^{i,j}.Ve \neq UD \end{array} \right. \quad (4)$$

In Fig. 3 and 4, the relation described by rule $dr2$ (eq. 4) between the verification attributes of a decomposed entity and of two lower-level entities is synthesized. With regard to the association of these attributes with the decomposed entity xA_1^{i-1} and its two sub-entities X_2^{i11} and X_2^{i12} , and without considering any rules or relations, there are $3 * 3 * 3 = 27$ possible combinations. Considering rule $dr2$ (eq. 4), only 11 combinations are permissible.

4.3. Feasibility and Verification Links

The possible states of this attribute Ve depends on the previous value Fa : an entity X cannot be verified if it has not been previously feasible. Therefore, we set up the rule $r3$:

$$r3 : X.Fa \neq OK \Rightarrow X.Ve = UD \quad (5)$$

Each of these attributes has three possible values (UD, OK or KO). With regard to the association of these attributes with the entities ($X.Fa$, $X.Ve$) and without considering any rules or relations, there are $3 * 3 = 9$ possible combinations. Considering the rule $r3$ (eq. 5), only 5 combinations are permissible.

For these two attributes, the modification of requirements at any stage in the design process by anyone (stakeholders, client, upper level) [12] implies a return to the initial state ($X.Fa = UD \wedge X.Ve = UD$) and the processes of feasibility and verification must be carried out again.

It is extremely important to point out that these two attributes can be switched to KO at any time in the processes. As soon as a problem with a negative impact on the global project is detected, these attributes have to be switched to KO, without waiting for the end of the current process. This scheme allows the managers to be immediately alerted that a problem has occurred. They will then have to find a common solution to solve it without threatening the overall success of the project.

4.4. Feasibility and Verification Attributes Synthesis

Seven synchronous attributes $SR_n^i.Fa$, $SA_n^{i,j}.Fa$, $SA_n^{i,j}.Ve$, $PR_n^i.Fa$, $PA_n^{i,j}.Fa$, $PR_n^i.Ve$ and $PA_n^{i,j}.Ve$ ensure consistency in the design process by imposing precedence relations, through three precedence rules $r1$ (eq. 1), $r2$ (eq. 3) and $r3$ (eq. 5), one top-down precedence rule $dr1$ (eq. 2) and one bottom-up logical and precedence rule $dr2$ (eq. 4), as synthesized in Fig. 3 and 4.

For the design side, without considering any rules or relations, there are $3 * 3 * 3 = 27$ possible combinations. Considering the precedence relations $r1$ and $r3$, and placing ourselves at a single decomposition level, there are 7 possible combinations.

For the project side, without considering any rules or relations, there are $3 * 3 * 3 * 3 = 81$ possible combinations. Considering the precedence relations $r1$, $r2$ and $r3$, and placing ourselves at a single decomposition level, there are 15 possible combinations.

Taking into account both sides and without considering any coupling rules, there are $27 * 81 = 2187$ possible combinations. If only the precedence relations $r1$ (eq. 1), $r2$ (eq. 3) and $r3$ (eq. 5) are considered, there are $7 * 15 = 105$ possible combinations.

5. Synchronous Coupling Rules

The goal of synchronous coupling is to enforce synchronous milestones between the two processes. Two types of synchronous milestones have been identified: the first one relies on a relation of precedence between project planning and system design, rules cO_1 (eq. 6) and cO_2 (eq. 7); the second one synchronizes them without any dominance, rules $c1$, (eq. 8) and $c2$ (eq. 9), as shown in Fig. 5.

Firstly, a relation of precedence exists between the project planning entities and the system design entities. A designer cannot start working without a feasible project plan to which (s)he has been allocated and planned. This first coupling milestone is supported by the feasibility attributes and can be formalized as cO_1 for the requirement definition task and the system requirements entity

$$cO_1 : PR_n^i.Fa \neq OK \Rightarrow SR_n^i.Fa = UD \quad (6)$$

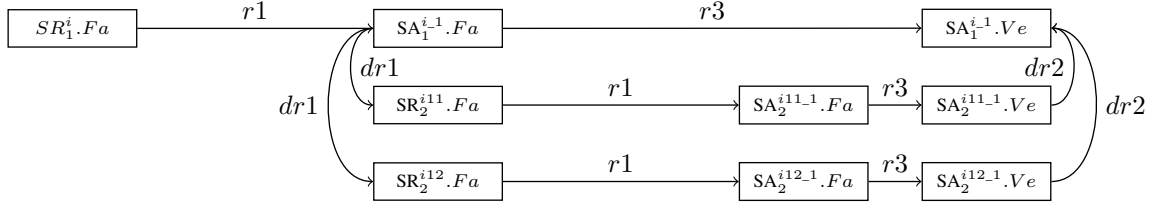


Fig. 3. System Design Attributes and Rules

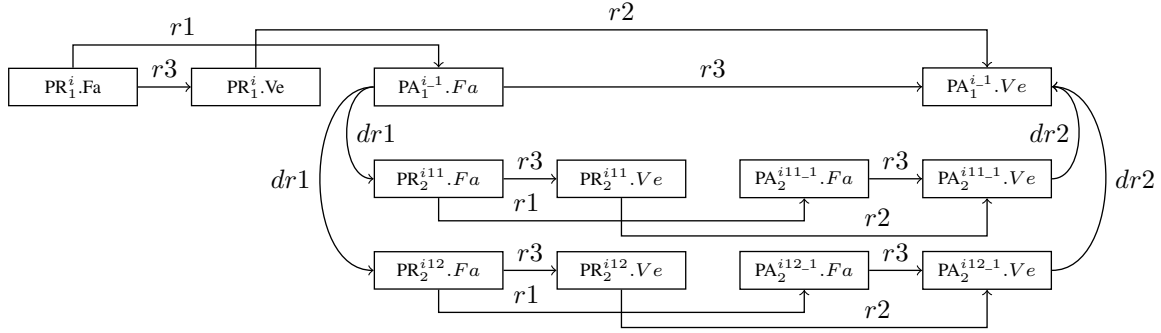


Fig. 4. Project Planning Attributes and Rules

and cO_2 for the alternative development task and the system alternative entity

$$cO_2 : PA_n^{i-j}.Fa \neq OK \Rightarrow SA_n^{i-j}.Fa = UD \quad (7)$$

Secondly, synchronization of the two processes is needed when one of them has reached a solution. Let us consider the pair (system requirements SR_n^i , requirement definition task PR_n^i). Let us start with the system design side. When all the system requirements have been gathered and analyzed, the design manager has achieved his/her job and can deliver his/her conclusions on the risk of reaching an unfeasible solution. If the risk is weak $SR_n^i.Fa = OK$, the corresponding task PR_n^i has to end $PR_n^i.Ve \neq UD$. On the other hand, where $SR_n^i.Fa = KO$, the requirements have to be modified in order to continue the design. In this case, the system design comes back to its initial state: $SR_n^i.Fa = UD$ but the project side is not necessarily impacted by this modification. A renegotiation of the requirements with regard to the work already done can be supported by the AHP method [29]. Let us now look at the planning project side. When the task PR_n^i is nearing its end, all the time has been consumed ($PR_n^i.Ve \neq UD$), so a decision has to be made on the feasibility of the system regarding the requirements. These two analyses allow us to deduce that a relation exists between the feasibility of the system requirements $SR_n^i.Fa$ and the verification

of the $PR_n^i.Ve$ task. This relation is formalized as:

$$c1 : SR_n^i.Fa \neq UD \Leftrightarrow PR_n^i.Ve \neq UD \quad (8)$$

Let us consider the pair (system alternative SA_n^{i-j} , alternative development task PA_n^{i-j}), starting with the system design side. When the whole solution has been completely designed, the design manager has achieved his/her job and can verify the consistency of the solution with regard to the requirements. If the solution matches the requirements $SA_n^{i-j}.Ve = OK$, the corresponding task PA_n^{i-j} has to end $PA_n^{i-j}.Ve \neq UD$. On the other hand, if $SA_n^{i-j}.Ve = KO$, it will be necessary to modify the requirements in order to finish the design. In this case, the system design comes back to its initial state: $SR_n^{i-j}.Fa = UD \wedge SA_n^{i-j}.Fa = UD \wedge SA_n^{i-j}.Ve = UD$ and the project side is necessarily impacted by this modification: tasks PR_n^i and PA_n^{i-j} have to be restarted and lengthened. The project keeps its time consumption but the complete process has to be carried out once more: $PR_n^i.Fa = UD \wedge PR_n^i.Ve = UD \wedge PA_n^{i-j}.Fa = UD \wedge PA_n^{i-j}.Ve = UD$. Let us now start with the planning project side. When task PA_n^{i-j} is nearing its end and all the time has been consumed ($PA_n^{i-j}.Ve \neq UD$), the system alternative has to be verified with regard to the requirements and then delivered. If a solution has not been designed, the complete project can start once

more. These two analyses allow us to deduce that a relation exists between the verification of the system alternative $SA_n^{i,j}.Ve$ and of the verification of the $PA_n^{i,j}.Ve$ task. This relation is formalized as:

$$c2 : SA_n^{i,j}.Ve \neq UD \Leftrightarrow PA_n^{i,j}.Ve \neq UD \quad (9)$$

The goal of ensuring a better consistency between system design and project planning has now been reached through two specific attributes and nine rules. With regard to the seven entities ($SR_n^i.Fa$, $SA_n^{i,j}.Fa$, $SA_n^{i,j}.Ve$, $PR_n^i.Fa$, $PR_n^i.Ve$, $PA_n^{i,j}.Fa$ and $PA_n^{i,j}.Ve$), there are 2187 possible combinations (cf. 4.4). Considering now only the precedence rules $r1$ (eq. 1), $r2$ (eq. 3) and $r3$ (eq. 5), there are 105 out of 2187 possible combinations (cf. 4.4). Considering now all the precedence rules, and the coupling rules cO_1 (eq. 6), cO_2 (eq. 7), $c1$ (eq. 8) and $c2$ (eq. 9), the number of possible combinations is reduced from 105 down to 27, as shown in Fig. 6. Only these 27 combinations are allowed by the proposed coupling in order to monitor and synchronize these two processes.

In all the dead-end states (states 3, 6, 7, 9, 11, 13,14, 15, 17 and 18), a problem that can jeopardize the success of the project, had occurred. The design manager and the project manager have to find a common solution that best suits system and project requirements at the same time to solve it. When the solution has been found, the process comes back to the initial state (state 1) in Fig. 6, while keeping time, budget and resource consumption, for the project planning side, in order to consolidate these information at the project end.

6. Coupling Process Experimentation

This section highlights our proposals for coupling (structural and synchronous) with the simplified but realistic example of a landing gear system (cf. section 4). This landing gear system is composed of one potential solution for level 1 decomposed into two sub-systems (a wheel and a brake) and two sub-projects (a wheel project and a brake project) for level 2, as shown in Fig. 2. In order to value the feasibility and the verification attributes, only two requirements are used: the weight for the system side and the duration for the project side.

Four typical coupling situations, belonging to a scenario, are described and show:

- The coupling process between entities on the same decomposition level (paragraphs [A], [B], [C] and [D]);
- The coupling process between entities between two successive levels (paragraphs [E] and [F]);
- The fact that a project entity which has not been properly completed does not compromise the design project (paragraph [G]);
- That this is not the case for a system entity (paragraph [H]).

In the proposed scenarios, we follow the succession of states 1, 2, 8, 12, 19 and finally 24 for the first scenario and 27 for the second one in Fig. 6.

[A] First of all, a program manager is appointed. (S)he starts creating a project entity and via the structural coupling, six entities are automatically created P_1^1 , PR_1^1 and PA_1^{1-1} for the project side and S_1^1 , SR_1^1 and SA_1^{1-1} for the system side, and matched $\{(S_1^1, P_1^1), (SR_1^1, PR_1^1), (SA_1^{1-1}, PA_1^{1-1})\}$. All of these entities have their attributes of feasibility and of verification to UD, state 1 in Fig. 6.

Then the program manager appoints the design manager and the planning manager. Once this is done, (s)he gives them their orientations (objectives and requirements) and defines their decision frames; for instance, the global budgets (for design and project), the time allotted to conduct the global project and the quantity of resources available. In our example, a 50-kilogram landing gear system has to be designed in 22 time units.

[B] The project side must be considered first. After collecting the information relevant to his/her context and exchanging views with all the people involved in his/her project (design and planning sides), the planning manager starts estimating the duration, allocating the resources and scheduling the requirement definition task PR_1^1 . In our example, (s)he allocates 2 time units to task PR_1^1 and 16 units to task PA_1^{1-1} . Two units are needed to verify project P_1^1 , and in order to reduce the risk of unfeasibility, a margin of 2 units is taken.

When the plan has been scheduled and the resources allocated, it appears to the planning manager that this task PR_1^1 is viable: (s)he switches its feasibility attribute to OK, $PR_1^1.Fa = OK$, state 2 in Fig. 6. During state 2:

- Rule $r3$ (eq. 5) is applied and the requirement definition task can be now analyzed and verified.
- Rule $r1$ (eq. 1) is applied and in our case, the planning manager concentrates on the alternative development task PA_1^{1-1} . (S)he does exactly the

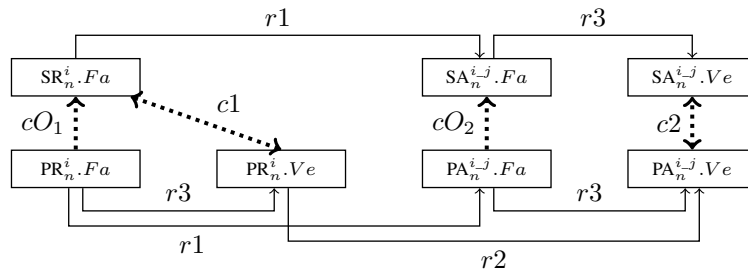


Fig. 5. Coupling Attributes and Rules

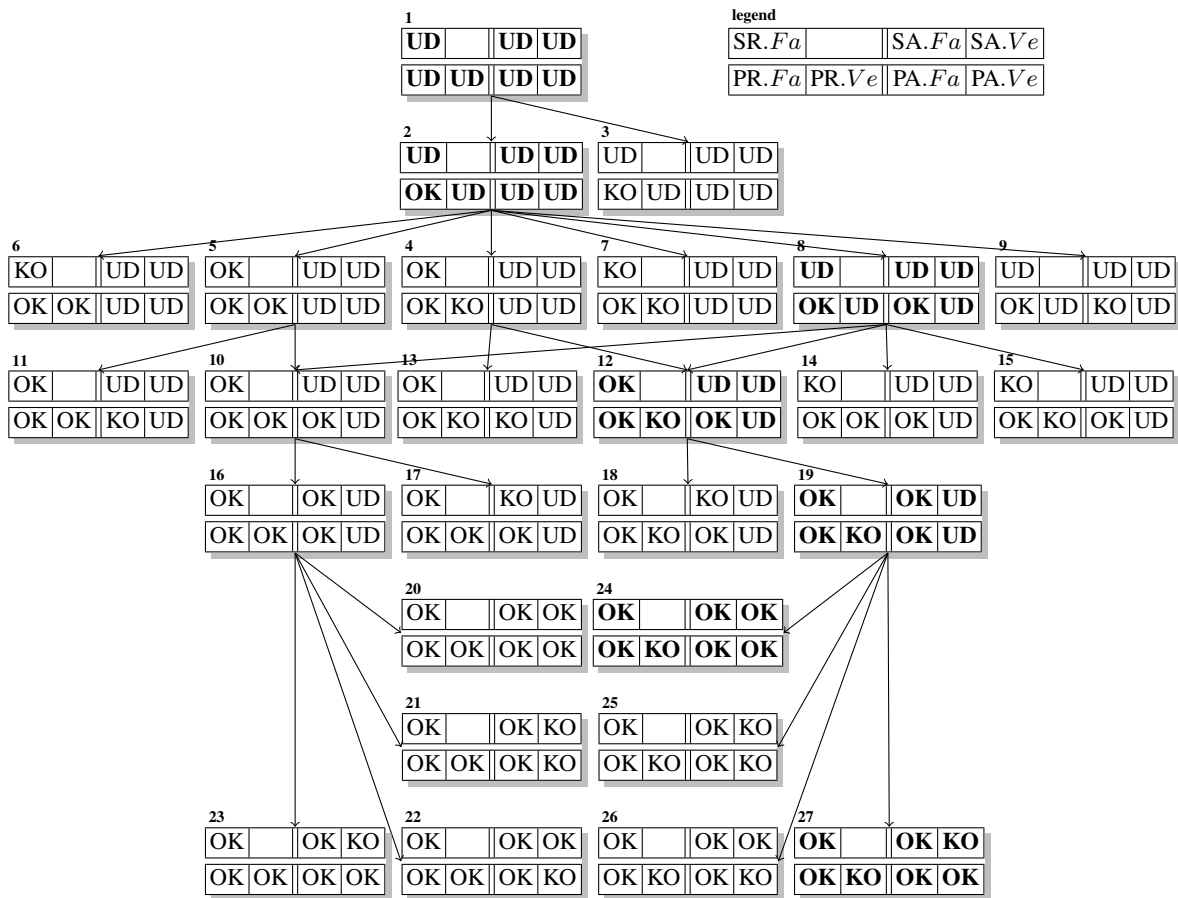


Fig. 6. Synchronous Coupling State Diagram

same job: duration estimation, resource allocation and scheduling.

- Rule cO_1 (eq. 6) is applied, so the designers are now allowed to start working and collecting the system needs and requirements.

[C] It appears to the planning manager that the alternative development task PA_1^{1-1} is also viable: (s)he switches its feasible attribute to OK, $PA_1^{1-1}.Fa = OK$,

state 8 in Fig. 6. At this moment, this means that the whole project planning (requirement definition task PR_1^1 and alternative development task PA_1^{1-1}) are both feasible with no conclusion on the feasibility of the system requirements SR_1^1 . During state 8:

- Rule $r3$ (eq. 5) is applied and only after the verification of the requirement definition task PR_1^1 (rule $r1$ (eq. 1)), can task PA_1^{1-1} be verified.

- The coupling rule cO_2 (eq. 7) is applied. But as the design manager has not yet determined the feasibility of the system requirements, the designers cannot start working on a solution (rule $r1$, eq. 1 has not yet been applied to the system side).

[D] When the needs and requirements have been collected and analyzed (a 50-kilogram landing gear system designed in 22 time units), it seems to the design manager that a solution is reachable: (s)he switches the feasibility attribute of the system requirements entity to feasible $SR_1^1.Fa = OK$, state 12 in Fig. 6. Simultaneously, the coupling rule $c1$ (eq. 8) is applied and forces the corresponding task to end, $PR_1^1.Ve \neq UD$. In our scenario, as this task does not respect its requirements (for instance, the design team has spent too much time collecting and analyzing the system requirements, 3 units instead of 2), this attribute is switched to unverified, $PR_1^1.Ve = KO$, by the planning manager, state 12 in Fig. 6. But the fact that this task has consumed more than expected is not critical for the whole project. The planning manager has to negotiate new objectives and requirements with the program manager. This negotiation can involve the design manager in order to take into account design problems and constraints. If the negotiation is a success, the project continues, as in our scenario ; otherwise, it is stopped. In our example, the negotiation has led to splitting the margin as follows: 1 unit for task PR_1^1 and 1 unit for task PA_1^{1-1} . During state 12:

- Rule $r1$ (eq. 1) can be applied: as rule cO_2 (eq. 7) has already been applied, the design team can now start designing the system alternative SA_1^{1-1} .
- Rule $r2$ (eq. 3) is applied and the alternative development task PA_1^{1-1} can now be verified.

[E] In our first scenario, we consider that the system alternative SA_1^{1-1} is too complex to be designed in a simple way, so the design manager has to split it into several sub-systems. In our example, the landing gear system is decomposed into a wheel and a brake. In this case, firstly, the design manager decomposes his/her system alternative into x sub-systems (in our case, SA_1^{1-1} is split into two sub-systems $sub-sys S_2^{111}$ and $sub-sys S_2^{112}$). Secondly, via the structural coupling, the project is decomposed in the same way into x sub-projects (in our case, PA_1^{1-1} is split into two sub-projects $sub-pr P_2^{111}$ and $sub-pr P_2^{112}$) that the planning manager has to analyze with regard to her/his project requirements (time and availability of resources). A complete coupling process has to be restarted for this new level and

for each pair of (sub-system S_{hl}^{xRi} , sub-project P_{hl}^{xRi} : $\{(S_2^{111}, P_2^{111}), (S_2^{112}, P_2^{112})\}$).

[F] When the design team has described the functioning principles of solution SA_1^{1-1} and given the requirements of each of the sub-systems SR_2^{111} and SR_2^{112} , the feasibility of the solution can be determined. The design manager specifies the weight of the sub-systems: 35 kg for the axle beam assembly (not considered in our example), 10 kg for the wheel and 5 kg for the brake. In our scenario, we consider that, regarding the requirements and the decomposition into two sub-systems, a solution is reachable: the feasibility attribute of the alternative is switched to feasible $SA_1^{1-1}.Fa = OK$, state 19 in Fig. 6. Rule $r3$ (eq. 5) is applied but the design manager has to wait for the verification of each of the sub-systems in order to verify the system alternative validity.

[G] When the design of each sub-system, $sub-sys S_2^{111}$ and $sub-sys S_2^{112}$, has been completed, the design manager has to integrate all the sub-systems, test the functioning of the system alternative SA_1^{1-1} , and switch its verification attribute to OK or KO depending on the results. In our first scenario, all sub-systems are verified $SA_1^{111-1}.Ve = OK$ and $SA_1^{112-1}.Ve = OK$ and their integration matches the requirements to the letter (9.9 kg for the wheel and 5 kg for the brake): the verification attribute of the system alternative is switched by the design manager to verified $SA_1.Ve = OK$, state 24 in Fig. 6. The coupling rule $c2$ (eq. 9) is then applied and forces the corresponding task to end $PA_1^{1-1}.Ve \neq UD$. In our scenario, the task respects its requirements to the letter (17 units); its verification attribute is switched to verified $PA_1^{1-1}.Ve = OK$, state 24 in Fig. 6. At this stage, all the feasibility and verification attributes on both sides have been determined, and the system design project is a success.

[H] In our second scenario, the verification attribute of the system alternative is switched to unverified $SA_1.Ve = KO$, state 27 in Fig. 6. This switch can come:

- From an integration problem of verified sub-systems $SA_1^{111-1}.Ve = OK$ and $SA_1^{112-1}.Ve = OK$ leading to a functioning problem. For instance, the wheel weighs more than required: its weight has not been verified properly by the sub-system design manager in charge of its design and it has been qualified as verified $SA_1^{111-1}.Ve = OK$. In this case, it is the system design manager who has to switch this attribute to unverified $SA_1.Ve = KO$;

- Or from a design problem, when it is impossible for the sub-system design manager to reach a feasible solution regarding the requirements. For instance, it is not possible to design a wheel with such a low weight. The sub-system design manager has switched the verification attribute $SA_1^{11-1}.Ve$ to KO. In such a case, the switch to unverified is done automatically through rule $dr2_1$ (eq. 4).

The coupling rule $c2$ (eq. 9) is then applied and forces the corresponding task to end $PA_1^{1-1}.Ve \neq UD$. The task respects its requirements to the letter (less than 17 units), so its verification attribute is switched to verified $PA_1^{1-1}.Ve = OK$, state 27 in Fig. 6. At this stage, all the feasibility and verification attributes on both sides have been determined. The project is interrupted because there is a problem on the system design side. Two options are possible:

- Either the complete project ends;
- Or a discussion, involving the design manager, the project manager, the program manager and if necessary, all the stakeholders, has to be conducted in order to redefine the objectives, requirements and constraints. This discussion can lead, for instance, to enlarging the budget of the project, to fixing a new deadline, to allocating more resources, or to changing some of the requirements, etc. If the discussion comes to a compromise, the process of feasibility and verification must be carried out again: the process comes back to the initial state (state 1 in Fig. 6 where all the attributes are valued to UD). But in this case, the time, budget and resource consumption has to be kept (and not reset) in order to consolidate the relevant information at the end of the project. For instance, in our example, a new weight distribution between sub-systems can be decided on.

7. Conclusion

The aim of this article is to propose a model and rules capable of supporting a coupling between the system design process and the project planning process. The context and the background of the study has been presented in section 2 and 3. The multi-level design process and planning process have been described in accordance with academic standards and with standards used in companies. The natural and logical bijective relationships between system entities and project en-

tities (discussed in section 3) is the base of the proposed structural coupling. The model that supports coupling groups together project entities (PR_n^i and PA_n^{i-j}), systems entities (SR_n^i and SA_n^{i-j}) and coupling rules.

In section 4, we have proposed and defined specific attributes which are used to develop synchronous coupling: feasibility and verification attributes. These attributes allow states to be defined as entities during their life cycle and make it possible to control their consistency throughout the design of the whole system. Three precedence rules $r1$ (eq. 1), $r2$ (eq. 3) and $r3$ (eq. 5) on a single level of decomposition, as well as two hierarchic rules $dr1$ (eq. 2) and $dr2$ (eq. 4) have been set up between system attributes or project attributes. In section 5, synchronous coupling is described by identifying different states and transitions between these states for entities, and by defining some rules which guarantee the consistency of the transitions. In other words, the proposed coupling synchronizes the project planning process with the system design process while preserving consistency between the changing of states. It authorizes certain attribute changes and forbids others, leading to better planned and controlled design tasks. Firstly, two precedence coupling rules $cO1$ (eq. 6) and $cO2$ (eq. 7) order the two processes: the feasibility of project entities must be qualified before those of the system. Secondly, two synchronized coupling rules $c1$ (eq. 8) and $c2$ (eq. 9) synchronize them: none of the processes prevails on the other but a change in one or the other environments has a strong impact on the other (for instance, the delivery of a system implies the end of the associated task and vice-versa). Finally, in section 6, a coupling scenario is described in order to illustrate the proposal.

The proposed synchronous coupling is quite simple but very powerful. The formalization of the behavioral interaction, which no previous scientific paper has made explicit, is unquestionably original. The nine identified rules have been set up thanks to the fifteen company interviews. In some specific situations, some of them can be needless. Of course, the less the coupling rules are applied, the more the resulting synchronous coupling state diagram has states compared to the one presented in Fig. 6.

These rules have been integrated in the ATLAS IT platform which has been developed in order to test and validate our proposals. Some tutorials and the link to the prototype can be found on the following webpage <http://perso.mines-albi.fr/~vareille/>, heading "Projects and industrial partners" and project "ATLAS".

It must be clear that according to the industrial development context, other rules could be added. This opens interesting issues leading to a kind of customization of the coupling process. Structural and synchronous couplings can be added, without any difficulty, to PLM software. In this case, engineers and managers would be forced to discuss together and to find a compromise acceptable to both sides, before a project failure occurs. Structural and synchronous couplings could help them to identify problems as early as possible and thus make better decisions by considering the overall situation.

Perspectives of this work concern the integration of local decoupling mechanisms between both domains when required at lower levels of decomposition. In this case, the coupling could be carried out manually (the program manager explicitly constructs the links between entities) and all the identified couplings continue to work. Alternatively, no coupling is performed and is thus no longer supported by the integrated model. Clearly, this method of operation needs to be checked and reserved for specific situations where designers of low levels want to decompose a very simple system to develop it without a framework given by the planning environment.

References

- [1] J. Abeille, T. Coudert, E. Vareilles, L. Geneste, M. Aldanondo, T. Roux, *Formalization of an integrated system/project design framework : first models and processes*, SPRINGER , pp.207-217, ISBN 978-3-642-15653-3, 1st International Conference on Complex Systems Design and Management CSDM2010, Paris, FRANCE, 2010.
- [2] L. Albano, N. Suh, *Axiomatic approach to structural design*, Research in Engineering Design, Vol. 4, pp. 171-183, 1992.
- [3] M. Aldanondo, E. Vareilles, M. Djefel, *Towards an association of product configuration with production planning*, International Journal of Mass Customisation 2010 - Vol. 3, No. 4 pp. 316 - 332, 2010.
- [4] L. T. M. Blessing, *Comparison of Design Models Proposed in Prescriptive Literature*, Proceedings of the COST A3 / COST A4 International Research Workshop on the role of design in the shaping of technology, Lyon, 1996.
- [5] F. Demoly, X.-T. Yan, B. Eynard, S. Gomes and D. Kiritsis, *Integrated product relationships management: a model to enable concurrent product design and assembly sequence planning*, In Journal of Engineering Design, Vol. 23, pp. 544-561, 2012.
- [6] G. E. Dieter, *Engineering design - A materials and processing approach*, Mc Graw-Hill International Editions, 2000.
- [7] I. Dörfler and O. Baumann, *Changing Organizational Routines in Response to a Drastic Failure: The Case of Airbus A380 Program*, Available at SSRN 2123297, 2012.
- [8] EIA Standard, *Processes for Engineering a System*, Electronic Industries Alliance, 1999.
- [9] S. D. Eppinger, D. E. Whitney, R. P. Smith D. A. Gebala, *Organizing the tasks in complex design projects.*, MIT Workshop on CAOPD, Springer-Verlag New York, Inc. 1991.
- [10] J. S. Gero, *Design prototypes: a knowledge representation schema for design*, Artificial Intelligence magazine. Vol. 11 No. 4, 1990.
- [11] S. Gomes and J-C. Sagot, *A concurrent engineering experience based on a cooperative and object oriented design methodology*, Best papers book, 3rd International Conference on Integrated Design and Manufacturing Engineering, Kluwer Academics Publisher, pp. 11-18, 2002.
- [12] E. Gómez de Silva Garza, M. L. Maher, *Design by interactive exploration using memory-based techniques*, Knowledge-Based System Journal, Vol. 9, No. 3, pp. 151-161, [http://dx.doi.org/10.1016/0950-7051\(95\)01016-5](http://dx.doi.org/10.1016/0950-7051(95)01016-5), 1996.
- [13] A. Goncalves-Coelho, *Axiomatic Design and the Concurrent Engineering Paradigm*, Proceedings of COSME, Brasov, Romania, 2004.
- [14] H. Hans, W. Herroelenb, R. Leusb, G. Wullink, *A hierarchical approach to multi-project planning under uncertainty*, The International Journal of Management Science, Vol. 35, pp. 563 - 577, 2005.
- [15] W. Herroelen, B. De Reyck, E. Demeulemeester, *Resource-constrained project scheduling: a survey of recent developments*, Computers and Operations Research, Vol. 25, No. 4, pp. 279-302, 1998.
- [16] J. Huysentruyt, D. Chen, *Contribution to the development of a general theory of design*, Proceedings of the 8th International Conference of Modeling and Simulation - MOSIM'10 - May 10-12, 2010 - Hammamet - Tunisia, 2010.
- [17] ISO 9000:2000, *Quality management systems - Fundamentals and vocabulary*, International Organisation of Standardization, http://www.iso.org/iso/catalogue_detail?csnumber=29280, 2005.
- [18] A. Karnie and Y. Reich, *Managing the Dynamic of New Product Development Processes. A new Product Lifecycle Management Paradigm*, Springer, P. 13, 2012.
- [19] T. Kis, *Project scheduling: a review of recent books*, Operations Research Letters, Vol. 33, Issue 1, pp. 105-110, 2005.
- [20] M. Lamers, *Do you manage a project, or what? A reply to "Do you manage work, deliverables or resources"*, International Journal of Project Management, Vol. 20, Issue 4, pp. 325-329, 2002.
- [21] P. Lawrence and J. Scanlan, *Planning in the Dark: Why Major Engineering Projects Fail to Achieve Key Goals*, Journal of Technology Assessment and Strategic Management, Vol. 19, Issue 4, pp. 509-525, 2012.
- [22] U. Lindemann, *A vision to overcome "chaotic" design for X processes in early phases*, Int. Proc. of Conference on Engineering Design (ICED), Paris, France, 2007.
- [23] J. N. Martin, *Evolution of EIA-632 from an interim standard to a full standard*, Proceedings of INCOSE 1998 Symposium, 1998.
- [24] G. Pahl, W. Beitz, *Engineering Design: a Systematic Approach*, Springer-Verlag, 1996.
- [25] PMBOK Guiden *A Guide to the Project Management Body of Knowledge*, Fifth Edition, Project Management Institute, 2013.
- [26] G. Pol, C. Merlo, J. Legardeur, G. Jared, *Analysing collaborative practices in design to support project managers*, International Journal of Computer Integrated Manufacturing, Taylor & Francis, Vol. 20, No. 7, pp. 654-668, November 2007.

- [27] H. Li, Y. Fan, C. Dunne, P. Pedrazzoli, *Integration of business processes in Web-based collaborative product development*, International Journal of Computer Integrated Manufacturing, Taylor & Francis, Vol. 18, No. 6, pp. 452-462, September 2005.
- [28] R. Lu, W. Peng, C. Wang, *Integration of Product Design Process and Task Management for Product Data Management Systems*, in proceedings of IFIP International Federation for Information Processing, Volume 254, Research and Practical Issues of Enterprise Information Systems II Volume 1, eds. L. Xu, Tjoo A., Chaudhry S. (Boston: Springer), pp. 207-218, 2007.
- [29] T. L. Saati, *Engineering Design: a Systematic Approach*, RWS Publications, 3rd Revised edition, 2012.
- [30] H. A. Simon, *The science of the artificial*, MIT Press, 1969.
- [31] A. Sharon, V. Perelman, D. Dori, *A Project-Product Lifecycle Management Approach for Improved Systems Engineering Practices*, Proc. Eighteenth Annual International Symposium of the International Council on Systems Engineering (INCOSE), Utrecht, the Netherlands, 2008.
- [32] A. Sharon, O. de Weck, D. Dori, *Is there a Complete Project Plan? A Model-Based Project Planning Approach*, Proceedings of the Nineteenth Annual International Symposium of the International Council on Systems Engineering (INCOSE), Singapore, 2009.
- [33] B. Shore, *Systematic biases and culture in project failures*, Project Management Journal, Vol. 39, Issue 4, pp. 5-16, 2008.
- [34] D. Stewart, D. Tate, *Integration of Axiomatic design and project planning*, Proc. of first Int Conference on Axiomatic Design, Cambridge USA, 2000.
- [35] N. Suh, *The principles of Design. Oxford series on Advance Manufacturing*, Oxford University Press, New York, 1990.
- [36] N. Suh, *Axiomatic Design: Advances and Applications*, Oxford Series, 2001.
- [37] M. X. Tang, *A knowledge-based architecture for Intelligent Design Support*, International Journal of Knowledge Engineering Review, vol. 12 (4), pp.387-460, 1997.
- [38] D. G. Ullman, *The mechanical design process*, third edition, McGraw-Hill, Higher Education, New York, 2003.
- [39] É. Vareilles, M. Aldanondo, M. Djefel, P. Gaborit, *Coupling interactively Product and Project Configuration: a Proposal using Constraints Programming*, International Mass Customization Meeting (IMCM) and International Conference on Economic, Technical and Organizational Aspects of Product Configuration Systems (PETO), 2008.
- [40] É. Vareilles, T. Coudert, M. Aldanondo, L. Geneste, J. Abeille, *Coupling system design and project planning: discussion on a bijective link between system and project structures*, forthteenth edition of IFAC's triennial symposium on Information Control Problems in Manufacturing (INCOM), 2012.
- [41] J. X. Wang, M. X. Tang, B. Gabrys, *An Agent-Based System Supporting Collaborative Product Design*, R.J. Howlett, L.C. Jain (Eds.): KES 2006, Part II, LNAI 4252, pp. 670 - 677, Springer-Verlag Berlin Heidelberg, 2006.
- [42] H. Yoshikawa, *Design philosophy: the state of the art*, CIRP annals manufacturing technology, 38(2) : pages 579-586, 1989.
- [43] X. Zhang, Y. Li, S. Zhang, C. M. Schlick, *Modelling and simulation of the task scheduling behavior in collaborative product development process*, Integrated Computer-Aided Engineering, vol. 20 (1), pp. 31-44, 2013.