

## SYSTEM DESIGN AND DESIGN PLANNING : AN INTERACTION IDENTIFICATION

M. ALDANONDO, E. VAREILLES

University of Toulouse  
Mines Albi - CGI  
81013 Albi - France  
Name@mines-albi.fr

J. ABEILLE, T. COUDERT, L. GENESTE

University of Toulouse  
ENI Tarbes - LGP  
65000 Tarbes - France  
Name@enit.fr

**ABSTRACT:** *Within a project dealing with the development of a software mockup that should support interactions between system design and project planning processes, a study aiming to identify and to characterize these interactions is the object of this communication. First main issues and definitions of the study and related works are summarized. Then an industrial survey of a dozen of companies shows the interests and the main trends of these interactions. In the next sections, five different types of interactions will be described and discussed. Three of them directly depends on the availability of knowledge relevant to design and planning activities and are rather autonomous. The two others are associated with process control, consistency and human decisions. It will be shown that these mechanisms reduce on one side errors and process iterations and, on the other side, increase process control and consistency between design and planning.*

**KEYWORDS:** *System design, project planning, processes interaction, decision support system, intelligent system.*

### 1 INTRODUCTION

The goal of this short communication is to present investigations relevant to the various aspects that can take interactions between the design of a product or system and the planning of its design project.

If many works have been achieved singly on system design and project management, very few consider the interactions existing between them. Nevertheless, it seems rather logical that:

- on the one hand, product design decisions can have strong consequences on the planning of its design (for example: this technology requires at least one month more in order to be integrated in the system),
- and that, on the other hand, planning decisions can provide hard constraints to system design (for example: this due date forbids the use of this technical solution).

Considering and aiding interactions between these two processes should reduce or avoid

- planning impossibilities due to design
- and design infeasibilities resulting from planning.

The rest of this section: considers related scientific literature, details what we mean by design and planning and introduces our propositions gathering five kinds of interactions. The second section presents a short industrial survey about the problem. The third one is concerned with three interactions that are rather autonomous, while the last section deals with follow up issues and human decisions.

#### 1.1 System design

The design process we consider conforms the ideas of the Axiomatic Design proposed by (Suh 1990). We consider therefore a top down approach with some kind of "zigzagging" cycle between requirements or functions that should be fulfilled and technical solutions. According to the level of abstraction of this top down approach, we comply with the approach of (Pahl and Beitz 1996) and relevant conceptual, embodiment and detailed design. Therefore, we consider that system design gathers:

- requirements definition,
- solutions identification,
- requirement/solution matching,
- and if needed, recursive decomposition at the lower level of some solutions.

#### 1.2 Project planning

When we speak of project planning, we restrict our considerations to the "project time management" (PTM) as defined by the Project Management Institute (PMI 2004). The PTM is one of the nine processes proposed by the PMI that gathers the six activities: identification and sequencing of activities, estimation of resources and durations, scheduling and follow-up. We also consider that planning is a top down approach, where some kind of global planning is achieved at a high level and is progressively detailed at lower levels. In order to make

some kind of parallel point of view with design, we consider that project planning gathers:

- project activities definition,
- resource and duration identification,
- scheduling activities and resources,
- and if needed, recursive decomposition at the lower level of some activities.

### 1.3 Links between design and planning

Given these first elements, our two processes gather four domains: requirements, solutions, activities and resources. Even if very few works consider simultaneously these two processes and relevant domains, some of them must be recalled. The “Design Structure Matrix” community (DSM), has proposed various matrixes that can represent the links between these domains. A recent paper of (Lindemann 2007) proposed a mapping of the four domains: functions, components, process and resources very close to our problem. An interesting extension of axiomatic design proposed by (Goncalves-Coelho 2004) has shown the possibility of extending the top down “zigzagging” process towards project planning. Other authors like (Stewart and Tate 2000) have proposed, for a specific application, to link axiomatic design with a project management software. Works of the International Council on Systems Engineering (INCOSE <http://www.incose.org/>) dealing with system engineering have been considered in detail. Among them, the standard EIA632 that recommends different processes for system design and also a top down approach that processes building blocks. It can be also noticed that our approach conforms the two entities, Product Breakdown Structure (PBS) and Work Breakdown Structure (WBS), frequently used in project management.

According to the previous survey, we consider that the validity of the four domains (requirements, solutions, activities and resources) is confirmed. We also consider the top down and zigzagging approach for system design extended towards project planning. Most of the previous works are mainly descriptive in terms of causal links or processes influences. None of them, except (Stewart and Tate 2000) as far as we know, has tried to analyse or to describe interactions between design and planning process. We consider now these interactions and the global process of figure 1. Our goal is to show how design and planning can interact and how these interactions can be aided with some methods or software techniques in order

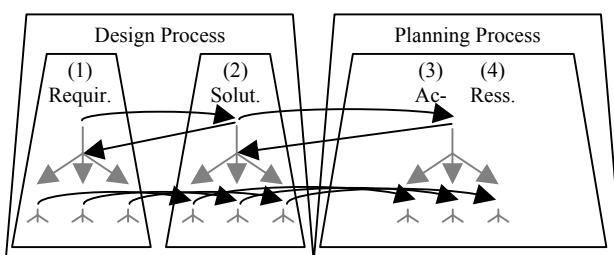


Figure 1. Four domains, and zigzagging

to help people in charge of their control.

## 2 INDUSTRIAL SURVEY

In order to initialize the investigations, a preliminary survey dealing with industrial practices has been conducted. Twelve companies, belonging to the “World Competitiveness Cluster Aerospace Valley” (<http://www.aerospace-valley.com/en>), of various sizes and sectors (industry, engineering and service) have been interviewed. The goal was to establish if companies manage design and planning interactions, when they do so and if some methods or aiding tools could be identified. More details about this survey can be found in (Abeille *et al* 2009).

### 2.1 Design, planning and interaction balance

The first trend characterizes the fact that one of the two processes tends to “dominate” the other. Globally, half of the situations show that the design process leads the main interaction flows. Remaining cases spread equally between a balance situation (no clear lead of one of the two processes) and a planning process with ascendancy. It has also been pointed out, that a design ascendancy is more often associated with a company working in industry rather than in engineering or service.

### 2.2 Interaction occurrences

The second trend is relevant to the occurrences of interactions or interaction needs. Of course, interactions have been detected after contracting, but also and with a very significant impact before contracting for preparing a response for an ordering party. This logically results from the fact that concurrence is always increasing. Consequently, relevant margins (on cost or performances) are reduced and require from suppliers to be much more accurate in their technical forecast and cost estimation. Preliminary studies and/or pre-design activities (before contracting) are therefore more and more sophisticated and systematically require compromises between performance, cost and due dates. As a result, relevant interactions between design and planning process become compulsory and critical.

### 2.3 Interaction tools

The third tendency concerns methods and tools that can support interactions. Around a third of the situations limit their interactions to informal talks and/or get-togethers, therefore interaction quality relies mainly on people skill, good relationship and passed lesson learned. A little less than half of the situations show companies

that have reached a first level in their interaction formalization. They operate with procedures, reference score-cards or some business process models that clearly define milestones and synchronicity points that should be respected. Remaining cases, around 15%, have gone much farther thanks to the set up of some computer tool supporting collaborative work and co-design. It is clear that these three different levels can be directly associated with some kind of maturity in the global organization of the company. However, all the companies reported that they consider any project meeting (project reviews, validation reviews, verification reviews...) as key moments to boost interactions between design and planning that should be preserved and extended.

## 2.4 Industrial survey conclusion

As a conclusion of this survey, it can be said that interaction existence between design and planning (the word coupling is often used) is a fact inside companies. Nevertheless, the level of formalism and the use of aiding tool is much more fuzzy and varies a lot according to each situation. In the next section, three types of interactions that correspond with some kind of decision propagation in the top down design and planning process are presented and discussed. Then in section four, two types of interaction more associated with follow-up activities and human compromises are presented and discussed in a same way.

## 3 INTERACTIONS AS DECISION PROPAGATION

Three mechanisms are presented and are directly dependant of the nature, more or less creative/routine, of the design activity according to the typology proposed by (Brown and Chandrasekaran 2005) and relevant availability of design knowledge.

### 3.1 A. Interaction with lowest knowledge level

When design is rather creative or innovative, the knowledge relevant to the system that must be designed and planned, or domain knowledge (relevant for example to: possible functions, various performances, available components or solutions and consequences on due date...) is always new and cannot therefore be re-used. Each system design and associated project planning is a new "adventure" where all domain knowledge (most of the time system specific, for example relevant to: plane, drone, crane, boat...) must be identified, gathered, validated before being used for designing and planning. Designing and planning actors cannot rely on their previous design experiences and don't have any knowledge asso-

ciating the four domains: requirements, solutions, activities and resources.

The single available knowledge concerns some rules that logically link the existence of the main entities of the design and planning processes. In order to derive these rules we assume that any system object of a design process is associated with one project, and that any project is associated with one system. The first type of interactions (dotted line on figure 2) expresses that on a same abstraction level:

- each new system design initialization requires a new project initialization and in a symmetrical way,
- each new project planning initialization requires a new system initialization.

These two initialization events are of course associated within each process with the creation of entities relevant to design (requirements and solutions) and planning (activities and resources). These mechanisms are also used recursively when a decomposition step is launched, or when a system is broken down in sub-systems and associated project in sub-projects as shown with the plain lines in figure 2. On this example it is possible to see that sub-system 3 is not decomposed anymore at a lower level as sub-system 1 and sub-system 2.

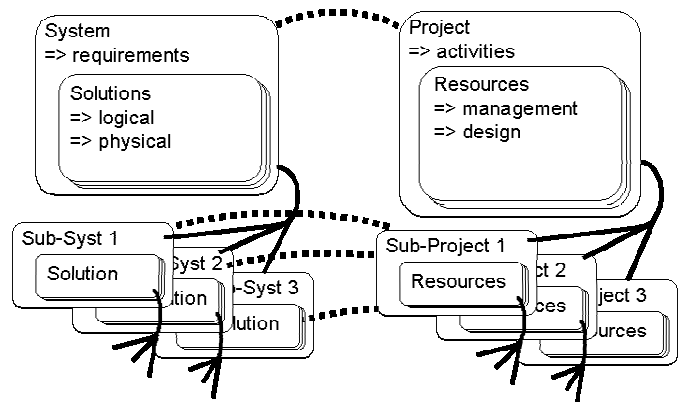


Figure 2 – Interaction relevant to entity existence

In order to clarify the meaning of the figure 2 we assume the following.

For the design process :

- requirements can correspond :
  - on a first side with informal characteristics that gather textual description, drawings or description of what is expected,
  - on a second more formal side variables or attributes that describe requirements or expected performances that can be quantified with num-

bers (as required mass, electric consumption, reliability...),

- solutions describes always :
  - on a first side a list of components and/or sub-assemblies, drawings, textual description,
  - on a second side the effective characteristics or performances of the proposed design (as effective mass, electric consumption...).

Logical solutions describe abstract design principles while physical ones gather :

- elementary physical components at the lowest decomposition level,
- physical sub-assemblies at upper levels.

When a solution needs sub systems, the description of this solution contains the requirements of the sub systems.

For the planning process :

- activities can roughly be associated with tasks dealing with : requirement definition, solutions search, association of requirement with solution and management. Each activity can be described :
  - on a first side with textual description,
  - on a more formal side with numerical attributes as at least duration, and quite often cost, risk or quality ratio for example,
- resources are required to perform the activity.

When solution search becomes too complicated, the search activity is decomposed in sub project at a lower level with respect to sub-systems identification.

These simple interactions bring in the following question. Is it possible to have a system design initialization without an already existing project ? It seems that the design resources (mainly humans) necessary to initialize the system design should be identified before by the project planning process. Therefore it seems that the decision propagation and relevant interaction is more oriented from planning towards design.

This is not so obvious, because many industrial situations show pre-design steps achieved without control of any project. By contrast, when dealing with decomposition, it seem that the origin of the event is most of the time on the design side and is a consequence of the system complexity. For this decomposition case, the interaction propagates decision from system design to project planning.

### 3.2 Interaction with knowledge embedded in cases

When design is less creative and becomes more routine, it is possible to store in a database passed design and planning as cases. Knowledge relevant to the system and project is thus embedded in these cases. This knowledge can be therefore used for some case base reasoning (CBR) assistance (Aamodt and Plaza 1994).

If CBR is rather frequent in design, it is not often used in planning. For our interaction goal, we consider that if CBR can be used in system design it should be extended to project planning. We consider therefore that CBR and reuse mechanisms are available for design and planning thanks to two databases. Consequently, the interaction between the two processes considers that a case retrieval in the design process implies a similar mechanism in project planning. The symmetrical interaction from project planning towards system design has not been considered, because the user most of the times looks for a system with some planning data and not for a project with some system information.

The interaction is thus oriented design towards planning. This allows interesting retrievals operation mixing technical and planning data as for example: (System = engine) AND (Power > 10kW) AND (cycle-time < 2 months) . Three issues can exist:

- no case can be considered and the mechanism is withdrawn,
- a close solution exists but must be adapted,
- a solution match exactly and is used without any modification.

In the second case, the case (requirements/solutions/activities/ resources) is imported in the current design and planning solution and some “cleaning” is achieved for adaptation. The third case is a simple substitution of the current system design by the retrieved system while the associated design project is just closed (no more design needed).

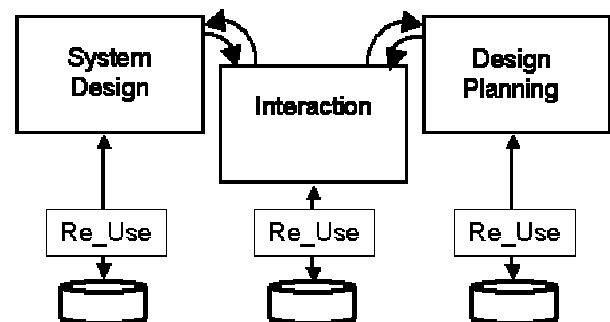


Figure 3 – Case based interactions

If this proposition sounds rather simple, it hides a problem of case descriptors. In the previous illustration we mention descriptors as “system” , “power” or “cycle time”. We assumed that all previous designed system and planned project could be characterized with these descriptors. It is clear that if previous designed systems gather different kinds of systems (as plane, drone, crane, boat..) case descriptors should be different according with the kind of system. It is therefore necessary for each kind of system or sub-systems on which some case base reasoning should be conducted to define a set of case descriptors. These case descriptors can be included into a set of key parameters corresponding with what we call system concept and project concept.

A *system concept* permits to characterize a *system*. A set of *system concepts* permits to build a domain ontology, i.e. a hierarchical classification of concepts. The most general concept is the "Universal" one. The ontology is defined using a tree of *system concepts*. The lowest concepts of the hierarchy are the most specialized ones and the highest the most general. A concept of the ontology is described by a set of variables used to characterize a system (for example : length, mass, electric consumption...). A concept is associated to its own variables and it also inherits those of its ascendants. The association of a concept to a system permits to associate automatically appropriated variables to this system in order to define the requirements and to characterize the solutions.

The *project concepts* are similar to the *system concepts*. A project concept permits to characterize a same kind of projects (plane design project for instance) by a same set of variables, such as the duration of the task, its cost, its associated risk and sometimes resource category or competency requirement. The project concepts are gathered into an ontology of hierarchical project concepts defining the project domain. Each concept in the hierarchy inherits the variables of its ascendants.

A question remains open and concerns the restriction to reuse a case for just one given abstraction level or the possibility to reuse a case with all its detailed lower levels. In other words, you retrieve at level 3 and you hesitate to reuse level 3, 2 and 1. It sounds rather logical that a design solution at a given level does not guarantee a solution at a lower level. For example a small plane with a single engine design in 2007 cannot guarantee that the engine used in 2007 still exists in 2009. It sounds therefore rather cautious to restrict the reuse mechanism to a single level. However, nothing forbids to launch reuse processes for each sub-system/sub-project existing at the lower level.

### 3.3 Interaction with explicit knowledge

When design is very routine, moreover reuse and solution storing, it is possible to try to identify some rules or

constraints that associate design and planning entities. These rules are often fuzzy and hidden in the memory of humans and “old” company experts.

The mechanism supporting the interaction assumes that, for each type of systems (plane, crane...), it exists a key set of design parameters (for example, if we consider a plane: power, flight range, mass...) that impact mainly project planning. Most of the time, this set of key parameters overlaps strongly with the set of variables that describe the ontology concept and the set of case descriptors that are used for the case retrieval process of the previous section. The key parameters in the project planning process are mainly relevant to activity durations, resource selections or even activity alternative identifications. Interactions between the two processes are supported by rules or constraints. As constraints do not have any predefined direction, interactions can work from design to planning and also in the reverse way from planning to design. For example in figure 4, a constraint relevant to a plane stating that ( flight-range > 500 AND power > 5kW ) is compatible with (cycle-time = 4to6 months AND project-leader = M.CX) can be propagated in both directions.

It has been shown in (Aldanondo *et al* 2008) that this kind of interaction can be considered as a constraint satisfaction problem (CSP) and thus supported by constraint filtering approaches. In order to do so, it is necessary, for each system concept and associated project concept, to establish a constraint model that link some system variables with some project variables called filtering variables. Once such a model is setup with relevant constraint filtering process that rely mainly on arc consistency for discrete variables (Bessière *et al* 1993) and bound consistency for continuous variables (Lhomme, 1993), each time a filtering variable has its definition domain reduce with a user input constraint filtering allows to propagate consequences on remaining variables.

The single drawback of this interaction mode is relevant to the availability of valid knowledge that links key parameters of design and planning. Some work is undergoing in order to try to mix this approach with the previous one in order to compensate the occurrence of “explicit knowledge hole” with knowledge embedded in cases.

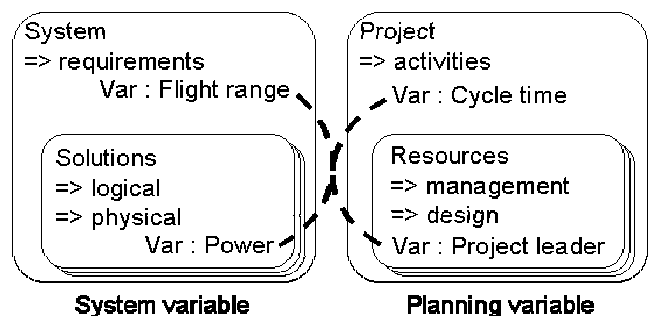


Figure 4 – Constraint based interactions

### 3.4 Conclusion on decision propagation

Three mechanisms have been shortly introduced. They rely on the four entities : requirement and solution on the system side and activity and resource on the project side. In order to describe or characterize these entities, it is possible to identify five kinds of information supports :

- informal text or drawings that are mainly used for aiding user interpretation without any capability to support formal automated interactions, this corresponds with the lowest level of knowledge,
- descriptive variables that formally characterize an entity but without any capability to support formal interaction,
- variables that belong to a system concept or project concept that must be considered for any system design and project planning,
- variables that are case descriptors used for simultaneous system and project retrieval,
- variables that are filtering variables used for constraint propagation.

Decision propagation between design and planning rely mainly on the last three subsets of variables. As explain at the end of the previous section, the identification, validation and maintenance of these variables lists is a key issue for interaction.

## 4 INTERACTIONS AND PROCESS CONTROL

The three previous interactions between design and planning have been set according to some knowledge availability (entity existence conditions / availability of passed cases / definition of rules and constraints). We will now consider two control or follow-up issues involving interactions. The first one considers some consistency test between the states of each entity while the second one deals with follow up activities and dashboard.

### 4.1 Interaction with consistency test

The goal of this interaction is to enforce the consistency of the entities of the design and planning processes:

- during the top down decomposition processes of design and planning,
- once the bottom level reached, during the bottom up processes that integrates and consolidates the solution.

In that purpose, we assume that :

- any system must include a set of goal/indicators/requirements relevant to performance (for example power, mass, volume...) and cost (component or material cost)
- any associated project should also be characterized with goal/indicators relevant to time (activity duration of the design process) and cost (cost of design activities and if necessary cost of manufacturing/transportation activities). Sometimes, risk is often added to both.

During the top-down processes of design and planning, and more accurately at the decomposition step (system/sub-system and project/sub-project), the previous goals or requirements at a given level are spread or distributed at a lower level according to some algebra (sum, average, min, max...). A "checking" Boolean attribute is associated to each system and each project (figure 5). It becomes :

- "true" when all the upper level goals have been distributed at the lower level while respecting the algebra,
- "false" if no solution respecting upper level goals has been found.

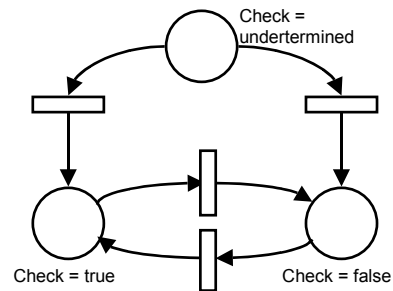


Figure 5 - Check attribute states

This can be seen in the automaton of figure 5 showing the three states and possible evolutions showing two others transitions :

- from true to false. This means that once a solution has been found at a given level, the initial goals or requirements are modified in the stronger sense and the propose solution is not valid anymore. Typical case, the customer changes his mind.
- From false to true. This means that if any solution could not be found at a given level, the initial goals can be modified in the lighter sense allowing a solution. Typical case, in order to reach a solution a customer requirement is modified.

This allows to check that the design or planning is globally finished at a given abstraction level with a consistent top down propagation of the design and planning goals. At the bottom level, when there is no more decomposition need, elementary component selections or "on the shelf solution" must be selected. The lowest level solu-

tions for both design and planning are also characterized by the previous indicators that must (in order to be included in a valid solution) respect the expected goals/indicators of the lower level.

A second “validation” Boolean attribute is thus associated to each system and each project. It becomes “true”:

- at the lower level, when the solution for design and planning respects expected goals (as shown in figure 6 for sub-system 3 and sub-project 3 which are not decomposed anymore)
- for upper levels, when all validation attributes of all children sub-system/sub-project are “true”.

This attributes allows to assemble in a bottom up approach components and sub-assemblies on the design side and activities and sub projects on the planning side.

An automaton similar to the one of figure 5 has been designed for the validation attribute. The single difference lies in the fact the transition true to false is not accepted anymore. Because, once in the bottom up consolidation step, accepting a stronger requirement from the customer means that a solution with current goals and requirements has been found until the lowest level, and is canceled due to stronger requirements. In that case, the idea is to consider a new design problem.

These two Boolean attributes permit to control the consistency of the two processes. The interaction corresponds with the checking of the behavior of these attributes that, for any couple system/project (check-system, check-project, valid-system, valid-project), should at least follow the obvious sequence (U, U, U, U) -> (T, T, U, U) -> (T, T, T, T). Other more detailed and complex sequences that gather entities from consecutive levels are currently investigated. Any gap occurrence from this sequence corresponds with an alert relevant to an inconsistency between the two processes.

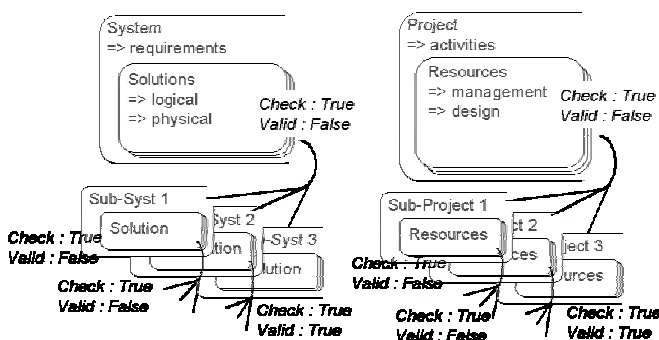


Figure 6 – Consistency based interactions

#### 4.2 Interaction with compromise on alternatives

This second mechanism operates thanks to the existence of design alternatives and project alternatives. The idea

is to assume that more than one solution (for a system and for a project) can be simultaneously investigated at any abstraction level. At a high level, this means that global architecture variants that can have a strong impact on system design and project planning (for example a small plane with one or two engines). At a lower level, this corresponds more with component alternatives or competition between suppliers or sub-contractors.

At any level, any investigated alternative is in fact a whole association of system and project and is therefore characterized with the indicators of the previous section (performance, duration, cost, risk). The interaction between system design and process planning can occur when it is necessary to select the appropriate alternative. Indeed, quite often, a solution preferred by the design team is not the one that would have been selected according to planning issues. For example a very efficient component can require a long integration design and relevant development cycle that can be incompatible with a customer due date. In the other way, a close and easy to manage sub-contractor selected by project planning may not be able to provide a sufficient level of quality for design.

Alternative selection becomes therefore some kinds of compromise search that is most of the time multi-criteria, gathering aspects relevant to performance, durations, cost and risk issues. The interaction mechanism consists in the possibility to be able at any time to setup a dashboard showing for each investigated solution (that can of course results of combination of alternatives) the distribution of the different indicators.

System Indicators	Alternative		Alternative		Planning Indicators	
	1	2	1	2	1	2
System Cost	240	300	Process cost	4	1.5	
Risk level (1 to 5)	3	2	Risk level	2	2	
Syst_perf_1 - power	52	55	Cycle time	18	24	
Syst_perf_2 - mass	26	20				
Syst_perf_3 - quality	16	21				
Checki attribute	True	True	Checki attribute	True	True	
Valid attribute	False	False	Valid attribute	False	False	

Figure 7 – Follow-up dashboard based interactions

## 5 CONCLUSION

The goal of this communication was to present and to discuss mechanisms that support interactions that can exist between system design and process planning. This, in order:

- to avoid iterations or loops in the design and planning processes
- to detect inconsistencies that can occur between these two processes

- to permit finally a better control of system design and process planning.

A related work survey has shown that if many works have been achieved, they mainly show causal links between design and planning. Very few of them have tried to describe or assist processes interactions. An industrial survey confirms that these interactions clearly exist and deserve some investigations. With respect to axiomatic design, DSM domains and system engineering recommendations, we have proposed five interaction mechanisms between system design and project planning. Three of them rely on the availability of domain knowledge and correspond with some consequence propagation between the two processes. The two others are more associated with the global behavior of the two processes and seem to attract the interest of companies as final users.

The study of these interactions is in a completion phase, each of them should be individually tested and evaluated with software mockups and later gather in a software test pilot. The whole three years research project, called ATLAS, that has provided these results reach his mid term now. We hope to be able soon to confirm the interest of the proposed interactions between design and planning.

#### ACKNOWLEDGMENT

The authors wish to thank their partners in the ATLAS project especially: C. Baron, D. Estève, M. Malbert, C. Merlo, T. Roux, M. Zolghadri, and the leaders of the World Competitiveness Cluster Aerospace Valley. The authors also thank the French National Research Agency for supporting this project.

#### REFERENCES

- Aamodt A., Plaza E., 1994, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, Vol. 7:1, pp. 39-59.
- Abeille J., Goris G., Vareilles E., Roux T., Aldanondo M., Coudert T., 2009, Couplage de la Conception de Produit et de la Planification de Projet: Première Analyse des Pratiques Industrielles", *Proc. of CGI 09*, Tarbes France 2009.
- Aldanondo M., Vareilles E., Djefel M., Baron C., Product and Process Configuration: A constraint based approach, *Proc. of IEEM 2008*, Singapore.
- Bessière, C. et Cordier, M. (1993). Arc-consistency and arc-consistency again. *Proc of AAAI 93*, pp 108\_113, Cambridge MA.
- Brown D.C., Chandrasekaran B., 1985, Expert systems for a class of mechanical design activity, *Knowledge Engineering in Computer-Aided Design*. North-Holland. pp. 259-282.
- Goncalves-Coelho A, 2004, Axiomatic Design and the Concurrent Engineering Paradigm, *Proc. of COSME*, Brasov Roumania, 2004.
- Lhomme, O. (1993). Consistency techniques for numeric CSP. *Proc of International Joint Conference on Artificial Intelligence*, pp 232\_238, Chambéry, France.
- Lindemann U., 2007, A vision to overcome "chaotic" design for X processes in early phases, in *Proc. of ICED*, Paris, France.
- Pahl G. and Beitz W., 1996, *Engineering Design - A Systematic Approach*, 2nd edition, London Springer.
- PMI Corporate Author, 2004, *A Guide to the Project Management Body of Knowledge: (Pmbok Guide)*, Project Management Institute.
- Stewart D. and Tate D., 2000, Integration of Axiomatic design and project planning, *Proc. of first Int Conference on Axiomatic Design*, Cambridge USA, p 285-289.
- Suh N., 1990, *The principles of design*, Oxford Series.