



HAL
open science

Agile Design Methods for Mechatronics System Integration

Matthieu Bricogne, Nadège Troussier, Louis Rivest, Benoît Eynard

► **To cite this version:**

Matthieu Bricogne, Nadège Troussier, Louis Rivest, Benoît Eynard. Agile Design Methods for Mechatronics System Integration. 10th Product Lifecycle Management for Society (PLM), Jul 2013, Nantes, France. pp.458-470, 10.1007/978-3-642-41501-2_46 . hal-01461888

HAL Id: hal-01461888

<https://inria.hal.science/hal-01461888v1>

Submitted on 8 Feb 2017

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.



Distributed under a Creative Commons Attribution 4.0 International License

Agile design methods for mechatronics system integration

Matthieu Bricogne¹, Nadège Troussier², Louis Rivest³ and Benoît Eynard¹

¹ Université de Technologie de Compiègne,
Department of Mechanical Systems Engineering,
CNRS UMR7337 Roberval
Rue du Docteur Schweitzer, BP 60319
60203 Compiègne Cedex, France
{matthieu.bricogne; benoit.eynard}@utc.fr

² Université de Technologie de Troyes,
Institut Charles Delaunay / CREIDD
CNRS UMR6972 STMR,
12 rue Marie Curie CS 42060
10004 Troyes Cedex, France
nadege.troussier@utt.fr

³ École de technologie supérieure,
Department of Automated Manufacturing Engineering,
1100 Notre-Dame Street West
Montréal, Québec
Canada H3C 1K3
louis.rivest@etsmtl.ca

Abstract. The multi-disciplinary nature of mechatronics product significantly increases complexity of the development process. In this paper, the benefits of agile design methods are presented for multi-disciplinary system integration.

After describing a model illustrating the relation between design state, decision making and multi-domain system integration, the conventional mechatronics system design process is illustrated. The weakness of “project planned” [1] management methods and their consequences on system integration are then pointed out. Finally, a framework to provide the necessary design and decision information is proposed to make agile design methods usage possible in this context.

Keywords: Mechatronics system design, multi-domain system integration, agile design methods, design-decision-integration model, Engineering action’s management system, collaboration workspaces

1 Introduction

Products are becoming increasingly complex, integrating technologies from several fields, such as mechanical engineering, electronic or electrical engineering and software engineering. Mechanical systems developed since the 80's have thus evolved from electro-mechanical systems with discrete electrical and mechanical parts to inte-

grated electronic-mechanical systems with sensors, actuators, and digital micro-electronics. These integrated systems are called mechatronics systems [2].

The multi-disciplinary nature of mechatronics “not only increases the complexity of products but also makes the product development process significantly more difficult” [3]. Engineers and designers use to be educated and to work discipline-wise. Some of them could have multidisciplinary skills but “cross-functional teams consisting of experts in several domains are mandatory to develop multi-disciplinary products” [3]. Several collaboration issues appear in such teams. One of them is the fact that multi-disciplinary product development leads to inter-disciplinary problems and integration issues. They are generally difficult to anticipate, hard to detect and even more hard to solve [3]. This paper aims to present some agile design methods and to propose a framework providing the design and decision information necessary for the implementation of these methods.

This paper is divided in two main parts. The first one proposes a model to illustrate the relation between design state, decision making and multi-domain system integration. This illustrating model is used to point out the weakness of “project planned” [1] project management, generally used to support mechatronics system design process. Its consequences in term of multi-disciplines system integration are then pointed out. The second part proposes a framework to provide the necessary design and decision information to make “agile design methods” [4] usage possible. These agile design methods are presented as a multi-disciplines integration enabler.

2 Mechatronics system multi-disciplines integration

Multi-disciplinary mechatronics system design could lead to integration issues. In order to explain what is the impact of multi-disciplines integration on the product, the different levels of mechatronics system’ integration are presented below.

Penas et al [4] describe four different mechatronics system’ integration levels (**Fig. 1**). The first kind of integration is called "separated components". In this case, components are designed separately and are just incorporated in the same system thanks to cables. The second level of integration corresponds to the concept of "joined components". The mechanical component will be designed in order to place the electrical and/or the electronic components in juxtaposition with each other. Distances between components have been reduced. The third kind of integration is called "inserted": electronic components are spread out into the whole system. There is no real integration. Finally, the ultimate integration level is the "merged" components: electronic is integrated as close as possible to the mechanical and electrical components. Parts are gathered in a coherent and functional manner and mechanical parts can also be used as signal transmitter. The contributions of this integration are various:

- Physical integration: spatial and weight optimizations are provided,
- Functional integration: detection, communication, control/information processing allow the system to provide new functionalities and to be reliable.



Fig. 1. The different integration levels in mechatronics system design [5]

Multi-disciplines integration is a key objective in a mechatronics system design perspective; to point out such objective, an illustrating model is proposed in the next section. This model is used to highlight the weakness of process-planned project management, generally used to support mechatronics system design process.

3 Decision – Design – Integration illustrating model

3.1 Decision – Design for “single discipline” product design

During product design projects, decisions are made and design information is created. Noël et al [6] describe the product–process–organization model (PPO) model where both type of information (Decision and Design) are linked. For example, the project objectives, part of the “decision framework” [6] have a great influence on the “design framework” [6]. On the other hand, technical constraints have to be taken into consideration by project leaders, which are parts of decision framework.

Fig. 2 illustrates the kind of interdependence that could traditionally exist between design framework and decision framework. It is composed of two axes. On the “Decision” axis, several design reviews are considered as the main steps during the design project decision process. On the “Design” axis, the main steps are based on various product deliverables resulting from the different stages of design process proposed by Beitz et al [7]. The polyline illustrates the “macro” evolution of a design project and the relation that can exist between the decision and the design information. During the design reviews, the overall progress is measured. Information about the project overall progress is usually seen as limited between project reviews, due to the lack of consolidated information. This idea is represented on the graph by the dashed line between points.

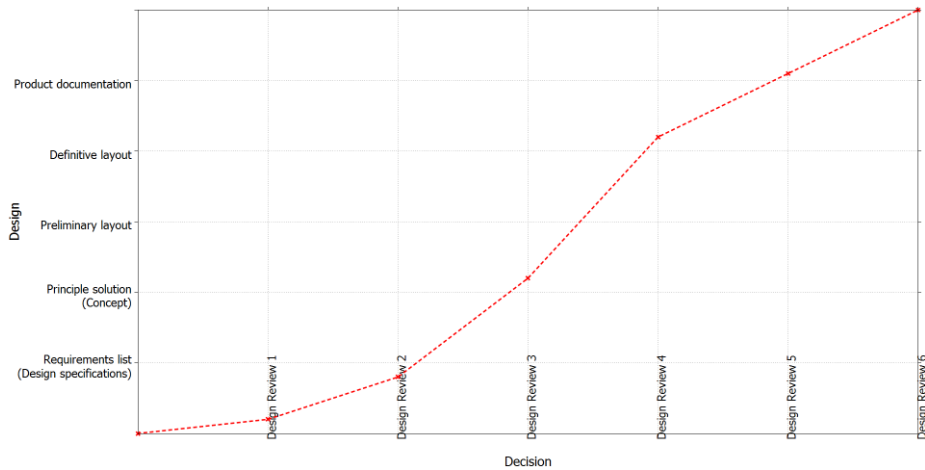


Fig. 2. Decision - Design model for “single discipline” product design

3.2 Decision – Design – Integration for multidisciplinary product design: application to mechatronics system design

In the previous section, a model illustrating the relation between decision and design information during product design involving only one discipline has been presented. In this, section, this model will be enriched to take into consideration the multi-disciplines integration objective in the mechatronics system design context.

Project planned mechatronics system design process.

Mechatronics system design is usually presented as a linear process [8]. Integration is then proposed as a specific step, in particular when the hardware part and the software part have to be integrated. Aca et al. present Product Lifecycle Management (PLM) system key benefit as its ability to facilitate the coordination of the activities among geographically distributed team members [9]. **Fig. 3** illustrates how the client request is divided among the different teams (software team, electrical/electronic team and mechanical team) and how the project manager organizes the different activities. This division is often presented as a necessary method to cope with design complexity [9]. Due to this division, only few people involved on the project have a general overview on the design problems faced by the different teams. The different interfaces between the different domains are defined and ratified in the early phases. To ensure a proper integration of the different modules, these interfaces have to remain stable. That is why if a design problem occurs in a specific discipline, this problem is generally treated directly by the team concerned, even if a global solution, i.e. implicating several disciplines, could be more efficient or provides a more integrated product. This way of organising the mechatronics system design process is called in this paper “project planned” process in contrast with agile design methods which will be presented later in this paper. To illustrate the lack of this process in term of product integration, the Decision – Design – Integration model will be used in the next section.

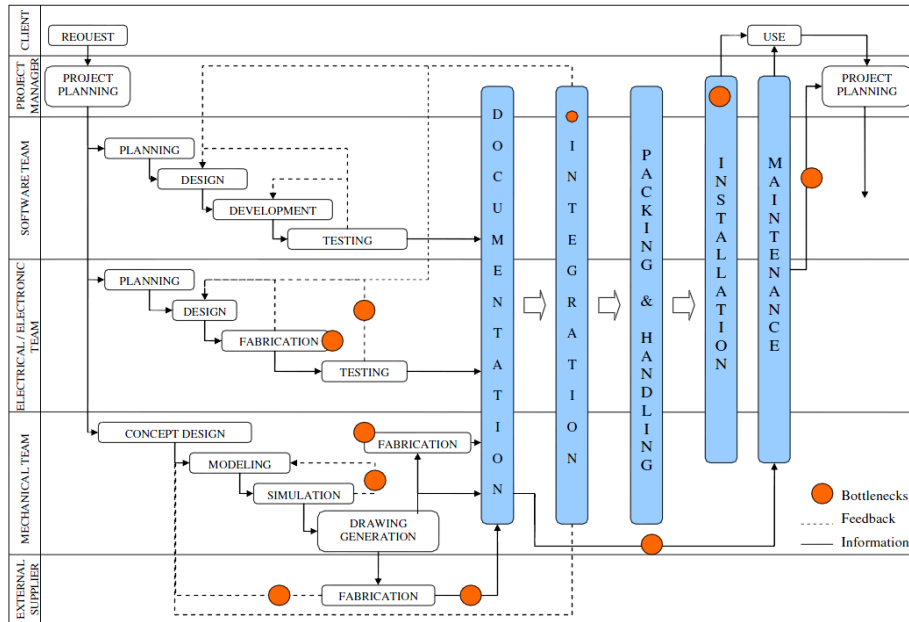


Fig. 3. Project planned mechatronics system design process [9]

Decision – Design – Integration illustrating project planned mechatronics system design process.

As described in the previous section, project planned mechatronics system design process leads to product integration weakness. Fig. 4 illustrates the lack of this process thanks to the Decision – Design – Integration model. It points out the fact that the different teams are performing their activities independently, sharing information mainly during the different design reviews. Some of these design reviews are dedicated to system integration, generating the steps on the graph. Between these design reviews, neither precise nor consolidated information about the system to design is available. This idea is represented on the plot by the dashed straight lines.

The red (mechanical domain), the green (electrical/electronic domain) and the blue (software domain) dashed lines are independent. Every discipline is managing its own modules’ design according to the requirements defined in the early phases of the project and is trying to respect the interfaces specified. As a result, the elevation on the graph is low, meaning that resulting integration is relatively poor.

In order to face this multi-discipline integration issue, agile design methods are proposed: in the next section, the general spirit of these methods is shortly described and some of their fundamental principles are presented as a key for multi-discipline integration. The Decision – Design – Integration model is finally used to illustrate the benefits of these methods in the mechatronics system design context.

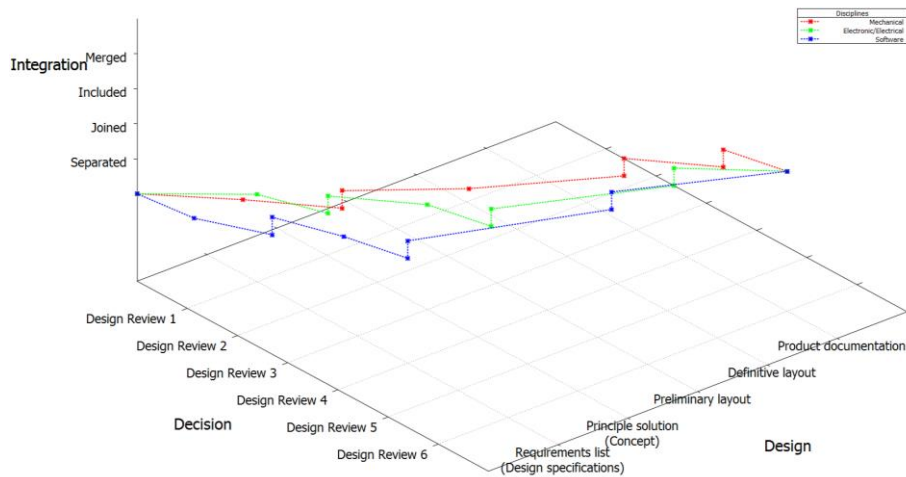


Fig. 4. Decision – Design - Integration model illustrating project planned mechatronics system design

Agile methods for mechatronics system design.

Agile methods are well documented for software design [1] and product manufacturing purposes [4]. However the relevance of these methods for concurrent design methods has not been so much studied. Matthews et al summarize agility benefits as “the ability to react rapidly to changes in the environment, whether expected or not” [4]. They also insist on the fact that concurrent design methods lack the ability to respond to unpredicted changes like a late customer request, a designer failure, or some other external environmental impact.

Agile design methods can be valuable for such problems, but also to support multi-disciplinary system integration. With the multidisciplinary design problems that can occur, the different cross disciplines activities etc., more exchange between designers, but also a unified, up-to-date and dynamic referential to collaborate [10] are required. In this section, three agile methods concepts are considered for mechatronics system design among the twelve principles of the “Agile Manifesto”¹ which is at the base of all the agile software methods:

- The first one is related to the early, frequent and continuous deliveries. At any time, all the data concerning the mechatronics system to design have to be available for design reviews.
- The second one concerns the fact that all the requirements do not have to be frozen at the beginning of the project. Requirements can be modified or added even late in development.
- The third one relates to the fact that managers, designers, developers have to work together daily on the project. A good overview has to be shared among project team members.

¹ <http://agilemanifesto.org/>

The choice made by the authors about these three concepts is mainly guided by literature reviews and by the mechatronics systems design characteristics. For example, the fact that mechatronics system design generally involves a great number of designers makes that it is necessary to cope with the need to adapt the agile methods to large-scale organization [11] [12]. Geographically distributed teams, or extended enterprise characteristics also implies that some of the principles are no more applicable [13] [14].

After the brief description of the three agile methods concepts considered in this paper, the expected benefits linked to their usage will be presented in the next section. To do this, the Decision – Design – Integration illustrating model is used.

Decision – Design – Integration illustrating agile design methods for mechatronics system design.

In order to be able to contribute to effective integration during the mechatronics system design, the experts coming from the different disciplines have to frequently share information: information about the design, information about the activities, information about the design problems, etc.

Fig. 5 assumes the role of continuous information sharing for multi-disciplinary system integration. On the plot, each point corresponds to a quick review that can be performed daily by the project leaders based on the consolidated information provided to them. If inter-disciplinary problems or integration issues are detected, global solutions are provided. This ongoing project management can assist multi-discipline integration, leading to a better resulting integration. This result is illustrated on the graph by the height of the final curves' points.

In summary, this graph illustrates the need for a unique repository for all the mechatronics system data. This does not mean that no specific teams have to be created, but they have to continuously share design data into the same repository. The graph also illustrates the influence of regular project leaders' decisions, based on consolidated and dynamic indicators.

In the next section, a framework to provide such repository, in order to facilitate the multi-discipline integration, and such indicators, in order to dynamically manage the project, is presented. This framework is presented as the anchor to support agile methods for mechatronics system design.

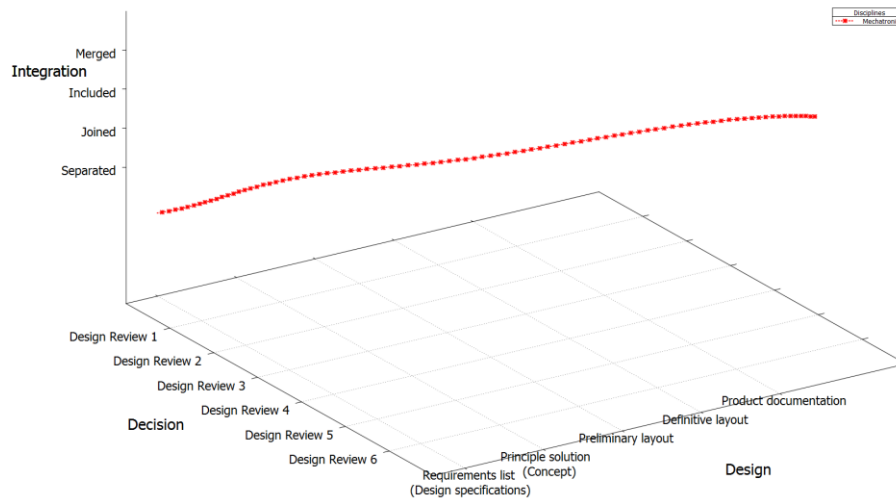


Fig. 5. Decision – Design - Integration model illustrating agile methods for mechatronics system design

4 A framework to support agile methods for mechatronics system design

As presented in the previous sections, agile methods are based on constant project reconfigurations. These reconfigurations are human decisions, but they are based on up-to-date indicators provided by the different systems supporting mechatronics system design. In this section, two main systems are presented as a framework to support agile design methods. The first one is dedicated to engineering actions management and the second one is focused on collaboration based on data exchange. The link between these two systems is finally presented.

4.1 Engineering actions management system

Engineering action's description

During a design project, the client requests are translated into technical requirements, driving the different design activities [9]. But other reasons influence and drive the design of a system. For example, design problems that occur are also generating specific demands and activities that are, depending on their severity, not formalized. These requests are generally informal and are considered as “daily work”. The last type of activities influencing the design a system is the engineering change request/order or bug correction for software engineering. Although they have own singularities, these prescriptive and reactive activities can be managed the same way, and in the same decision support management system. Indeed, they are all called in this paper “Engineering Actions” (EA).

Engineering actions contain different information. First of all, the EA's creator has to describe the context of the demand, the system concerned, the expected result and

eventually a description of the problem encountered. He also has to state on the severity of the demand. The severity can in fact be evaluated on a scale, for example from 1 to 4, thanks to the question “how will operate the system if this engineering action is not performed?”. To customize this engineering actions management system to specific companies’ needs, the EA can also be flagged by the creator, according to the flags available to him. When this definition stage is performed, the creator assigns the created EA to a specific person or to a domain leader if he does not know who will be in charge of it. The domain leader reads the description and, again, assigns the EA to a member of his team. During all this time, the EA’s status is “Opened – Under analysis”. When an engineer or a designer considers that he can play a role in the EA resolution, he changes the EA’s status to “Opened – Under treatment”. With the same principle, the designer will change the EA’s status according to the job performed: “Opened – Solution found”, “Opened – Solution in progress”, “Opened – Test in progress”. If he is the only actor concerned by the work to be performed, he will then deliver his modifications into the data management system and change the EA’s status to “Closed – corrected”. Otherwise, he could also assign the EA to another designer to perform other tasks on the same EA.

This kind of scenario could be really helpful for multi-discipline integration purpose. For example, if a mechanical part has to be modified to thermally insulate a new electronic device, a unique EA can be created. This EA is first assigned to the mechanical designer who defines a bounding box for the electronic card before transferring the EA to the electronic designer. This one design the card and transfer back the EA to the mechanical designer, specifying during the transfer the maximum temperature to respect. Finally, the mechanical designer modifies the part accordingly and closes the EA.

As showed above, this engineering actions management system differs from Product Data Management (PDM) project management modules in particular because activities are much more dynamic. When, in PDM, an activity is assigned to a person manually or because a workflow has been triggered, an EA is created by a specific person, assigned to a designer that can decide to work on the EA or to transfer it.

In this section, the EA has been described and briefly compared to traditional PDM activities management system. In the next section, the EA management system benefits for agile design methods will be presented.

Engineering action’s management system to support agile design methods

Beyond the fact that EA allow to manage trans-disciplines activities, it also provides useful indicators on the project progress, the project maturity, the burden of the teams, etc. Project progress can be measured thanks to the number of EA representing the required functionalities still opened. For the project maturity, the EA representing the unsolved problems can be summed up in order to determine whether the system is converging, e.g. the number of EA flagged “regression” decreases, or not. The burden of a team can be calculated thanks to the backlog of each team members, i.e. the opened EA affected to each team member.

All these calculated indicators, based on the raw indicators coming from the EA management system, are considered as a great support to decision makings. Quick review,

daily meeting, executive review, standup meetings are the different terms used in the agile methods community to speak about these meetings where new priorities are defined, problems are shared, and information about one project or about the different running projects is exchanged.

In this section, the benefits of EA management system for agile design methods usage have been pointed out. This system mainly provides indicators to support decisions makings in a multi-disciplines integration purpose. In the next section, this multi-disciplines integration will still be focused thanks to simplified data exchange.

4.2 Data exchange and collaboration through workspaces

The interest of a multi-disciplinary data management system has often been underlined [16]. But most of the authors continue to imagine a platform with a common interface addressing specific domain data management system, especially for hardware and software parts. Some other initiatives present the perspectives a full integrated platform [10] [17]. Whatever the solution, the workspace (WS) concept is a way of collaborating in a global environment. A WS is a set of Configuration Items (CI) with a specific version associated to each CI. A CI can be a document, an article depending on the domain involved and on the solution adopted. The only prerequisite on the nature of the CI is just that the CI has to be manageable in a version control system. For example, in the software engineering domain, a CI can be a source code file, a portion of this source code, an object class of this source code, etc. In the mechanical domain, a CI can be a parameter, a feature, a part, a product, an assembly or a set of these data.

The need for WS is linked to the fact that the design of a system generates a huge amount of data. In fact, engineers/designers are focused and interested by a subset of these data. WS allow them to be temporally insulated to be able to work on one or several engineering actions in a specific context. Insulation does not mean that they are not working under the control of the data management system. They are just not affected by the changes that can occur due to other engineering actions.

One of the major characteristic of the WS is that they have a tree structure (Fig. 6) and a “son WS” can only be composed of CI that are parts of the “parent WS”.

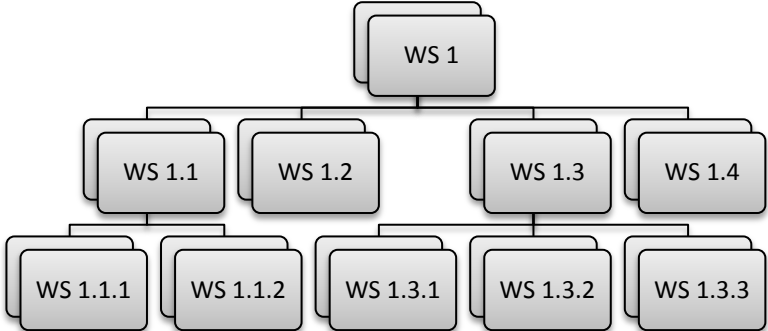


Fig. 6. Workspaces organized with a tree structure

Only workspaces located at the ends of branches are owned by the designers. The other workspaces are integration workspaces. In software engineering, in which automating testing is possible, automated or manual tests are performed in such integration workspaces.

The modifications performed by designers into their workspaces to realize an engineering action remain private until he decides to promote his modification to the parent workspace. In contrary, the fact of taking into consideration the modification performed by others from the parent workspace is called synchronization. So, the normal data exchange between two child workspaces is series of promotions followed by series synchronizations. For example, on **Fig. 6** a specific version delivered in WS 1.1.1 will reach WS 1.3.2 after passing through WS 1.1, WS 1 and WS 1.3. Some other direct exchange mechanisms could also be used.

There are multiple advantages to the usage of this system in a mechatronics system design context. First, the same workspace can be composed of CI coming from different domains. For example, if a change is required on an electronic card due to a supplier change, an engineering action is created by the project leader. The electronic designer changes the layout of the device and writes a note directly in the source code to modify before transferring the engineering action to the developer. Impacts are then easier to manage. Another advantage is the fact that the same CI in different versions can exist at the same time, allowing the designers to be focused on their engineering action and to be able to delay the merging action [18].

4.3 Engineering actions and workspaces: a way to link design and decision information

In the two previous sections, engineering actions management system and collaboration workspaces have been presented. They have been respectively pointed out as decision support system and data integration enabler. In order to provide a fully integrated framework, a link has to be provided between both functionalities. This connection is described in this section.

In fact, when a designer promotes his modification to the parent workspace, it is for a specific reason, described by one or several engineering actions. This means that during the promotion, the engineering action identifier has to be provided. With this information, it is possible to track from the engineering actions management system the impacted CI and their versions. In contrary, from a CI, it is possible to track which engineering actions contribute to the evolution of the CI.

Coming back to the Decision – Design – Integration illustrating model, engineering actions can also be seen as links between decision makings and design data. By giving higher priority to specific engineering actions during short design reviews, the project managers can influence the design of the mechatronics system. By analysing engineering actions exchange and tickets closure information, they also can control the way integration problem are continuously solved by designers and developers.

Fig. 7 illustrates how engineering actions can influence the curvature of the graph. On this figure, the points correspond to short design reviews, e.g. daily reviews, and engineering actions are represented thanks to labels.

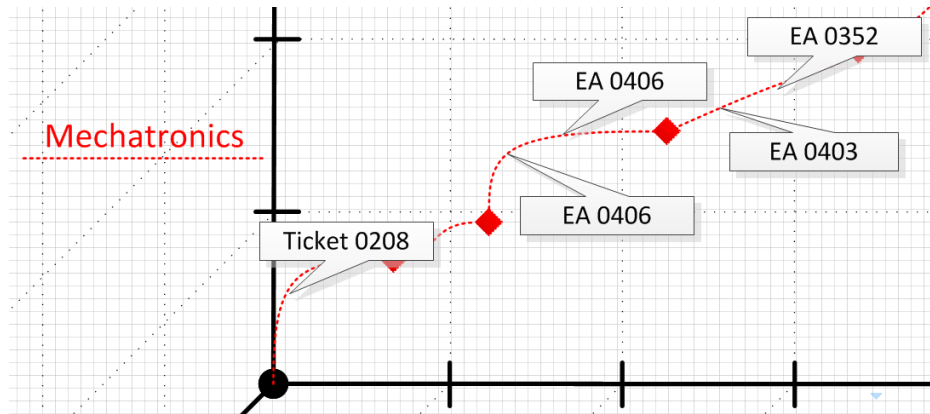


Fig. 7. Decision – Design - Integration model illustrating the tickets' role on mechatronics system integration

In this section, a framework based on engineering actions management system and collaboration workspaces has been proposed as a support to apply agile design methods in the mechatronics system design context. This framework shall provide a great number of indicators to project leaders, but also to share more information about systems data, especially in order to improve multi-disciplinary for mechatronics system design.

5 Conclusion

To design mechatronics systems, a unified, up-to-date and dynamic referential has to be provided to the project actors. The main role of this system, which could be a PLM system, is to federate design data and to assist multiple-disciplinary integration. But it could also be a support for new managing methods like agility design methods.

In this paper, a model to illustrate the benefits of agile design methods in term of multi-disciplinary integration has been suggested. A framework based on engineering actions management and collaborative workspaces to support decision processes has also been envisioned. This framework has been shortly described and future work will be focused on further framework description, its implementation and illustration on a case study in order to evaluate this proposition.

References

1. Sommerville, I., 2010. Software Engineering. Addison Wesley; 9 edition (March 13, 2010), New York, New York, USA.
2. Isermann, R., 2007. Mechatronics design approach, in: The Mechatronics Handbook. CRC Press, pp. 2–3.
3. Tomiyama, T., D'Amelio, V., Urbanic, J., ElMaraghy, W., 2007. Complexity of Multi-Disciplinary Design. CIRP Annals - Manufacturing Technology 56, 185–188.

4. Matthews, P., Lomas, C., Armoutis, N.D., Maropoulos, P.G., 2006. Foundations of an agile design methodology. *International journal of Agile Manufacturing* 9, 29–38.
5. Penas, O., Plateaux, R., Choley, J.-Y., Rivière, A., 2009. The different complexity levels in mechatronics design process, in: 3rd International Conference on Software, Knowledge, Information Management and Applications - SKIMA. Fes.
6. Noël, F., Roucoules, L., 2008. The PPO design model with respect to digital enterprise technologies among product life cycle. *International Journal of Computer Integrated Manufacturing* 21, 139–145.
7. Beitz, W., Pahl, G., Wallace, K., 2003. *Engineering design: a systematic approach*. Springer.
8. Shetty, D., Kolk, R.A., 2010. Mechatronics System Design, in: *Mechatronics System Design: SI*. Christopher M. Shortt, pp. 1–40.
9. Aca, J., Ramos, M., Serrano, J.L., Ahuett, H., Molina, A., 2006. Concurrent Engineering of Mechatronics Products in Virtual Enterprises: Selection and Deployment of a PLM System for the Machine Tool Industry, in: Luo, Y. (Ed.), *Co-operative Design, Visualization, and Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 318–326.
10. Bricogne, M., Troussier, N., Rivest, L., Eynard, B., 2010. PLM perspectives in mechatronics systems design, in: *Advances in Production Management Systems, APMS2010*. Cernobbio, Como, Italy, pp. 1–8.
11. Kettunen, P., Laanti, M., 2008. Combining agile software projects and large-scale organizational agility. *Software Process: Improvement and Practice* 13, 183–193.
12. Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., Kahkonen, T., 2004. Agile software development in large organizations. *Computer* 37, 26–34.
13. Sutherland, J., Viktorov, A., Blount, J., Puntikov, N., 2007. Distributed Scrum: Agile Project Management with Outsourced Development Teams, in: 40th Annual Hawaii International Conference on System Sciences (HICSS'07). IEEE, p. 274a-284a
14. Smits, H., Pshigoda, G., 2007. Implementing Scrum in a Distributed Software Development Organization, in: *AGILE 2007*. IEEE, pp. 371–375.
15. Abrahamsson, P., Warsta, J., Siponen, M.T., Ronkainen, J., 2003. New directions on agile methods: a comparative analysis, in: 25th International Conference on Software Engineering, 2003. Proceedings. IEEE, pp. 244–254.
16. El-Khoury, J., Redell, O., Torngren, M., 2005. A tool integration platform for multi-disciplinary development, in: *Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference On*. IEEE, pp. 442–449.
17. Do, N., Chae, G., 2011. A Product Data Management architecture for integrating hardware and software development. *Computers in Industry* 62, 854–863.
18. Bricogne, M., Rivest, L., Troussier, N., Eynard, B., 2012. Towards PLM for mechatronics system design using concurrent software versioning principles, in: Rivest, L., Bouras, A., Louhichi, B. (Eds.), *Product Lifecycle Management. Towards Knowledge-Rich Enterprises*. Springer Berlin Heidelberg, pp. 339–348.