

Meta-model of PLM for Design of Systems of Systems

Peter Hehenberger^{1,2}, Matthieu Bricogne², Julien Le Duigou², Benoit Eynard²

¹Institute of Mechatronic Design and Production, Johannes Kepler University, Linz, Austria
peter.hehenberger@jku.at

²Sorbonne Universités, Université de Technologie de Compiègne,
Department of Mechanical Systems Engineering, UMR CNRS 7337 Roberval,
CS 60319, 60203 Compiègne Cedex, France,
matthieu.bricogne@utc.fr, julien.le-duigou@utc.fr, benoit.eynard@utc.fr

Abstract. Mechatronic System Design involves close examination and further development of design methods, design processes, models and tools. The current trend in mechatronics involves networked mechatronic systems, or cyber physical systems (CPS), which can also be considered as a sub-part of Systems of Systems (SoS). Therefore data models for the description of the product lifecycle of SoS are necessary on the base of existing meta-models for single (mechatronic) systems. The paper shows a meta-model of PLM for the design of SoS and discusses the influence of the IT-architecture in supporting the PLM interoperability.

Keywords: Systems of Systems, Mechatronics, CPS, Meta-model, PLM.

1 Introduction

Within the framework of this work the product development process is discussed and analyzed from the basis of a mechatronic point of view or Cyber-Physical System (CPS) to extend to the overall consideration of the Systems of Systems (SoS) using meta-models of PLM for the design of the SoS. An important task in Modelling and Simulation of SoS is the consistency of the product model horizontally during all phases of the design process and vertically through all involved sub-systems.

Mechatronics may be defined as an interdisciplinary field of engineering science which aims to mutually integrate and interconnect mechanical engineering, electrical engineering / electronics and computer science (also often called information technology) in such a manner, that the interactions constitute the basis for the designing of successful products, e.g. (Bishop [1], DeSilva [2], VDI-2206 [3]). Mechatronics can be considered to be an integrative discipline which utilizes the various technologies in order to provide enhanced products, processes and systems.

Mechatronic systems usually consist of the following elements: it can be considered as two elements, one part focuses mainly on the energy flow, the other focuses in the information flow. The basic system is often a mechanical, electro-mechanical, electrical, hydraulic, pneumatic, thermo-dynamical, chemical or biological system and contains the realized process. The actuators use the information

that is provided to directly act on the basic system or the process which is taking place in an appropriate manner. The sensors observe the current properties/state of the basic system and the system environment (e.g. temperature). This may be done directly through measurement or indirectly via so-called observers. The provided information will be used by the control unit. The information is mostly available digitally and processes the sensor signals in order to create control variables for the existing actuators refers to the given driving parameters.

Mechatronic design emphasizes the competence integration between engineers skilled in specific domains such as mechanics, electronics and software. The interactions between product developers from these different disciplines are hindered by insufficient understanding between the disciplines and by missing common platforms for modeling of complex systems. As many sub-systems are delivered by external suppliers, there is a need for both horizontal integration within organizations and for vertical integration between the sub-system suppliers and the suppliers of the full systems. Therefore Lee [4] defines Cyber-Physical Systems (CPS) as the “integrations of computation and physical processes”. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa.

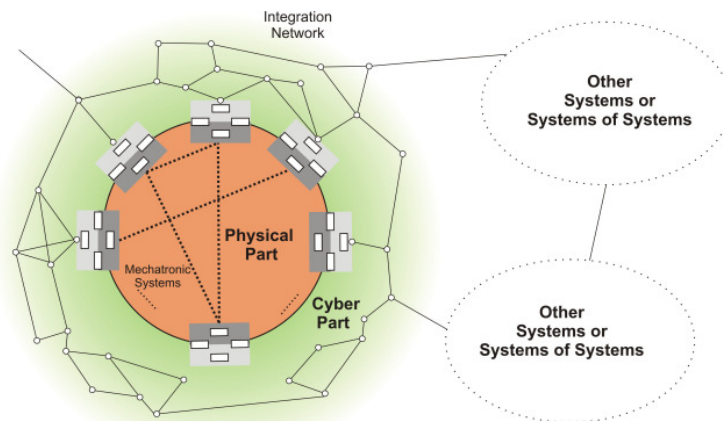


Fig. 1. Mechatronic Systems and Cyber Physical Systems into a Systems of Systems

System modeling and evaluation are also important topics of CPS description, for which improved tools and knowledge are ever claimed by the engineering profession. In many cases, very accurate system modeling is not reasonable to describe a complex system, as uncertainties and costs of quite detailed modeling may be so high that the drawbacks compared to simpler modeling become significantly overwhelming, so system-level models allow a multi-disciplinary engineering approach to be supported.

CPS System-level models need specific methods, languages and tools to support multi-view modeling to facilitate an interdisciplinary approach. More generally, this objective can be realized through multi-agent modeling, based on an engineering cloud structure. This also results in the usage of tools supported by Model-Based Systems Engineering (MBSE) [5]. CPSs are often dominated by one engineering

discipline. System-level models have to promote equal treatment of all engineering disciplines involved during product development and project execution.

Fig. 1 presents a structuring model to describe the relations between mechatronic systems and CPS (shades of grey). These core elements are all constituted of a HW-part and a SW-part. The interactions between them can occur in the physical domain (e.g. clash between two robots detected thanks to their sensors), represented in orange, or in the cyber part (e. g. dialog between these robots thanks to network protocols), represented in green. This cyber part is considered as the integration network. All these core elements are made up of several modules represented by the small white boxes. The mechatronic systems and the CPS located on the border of the circle are the parts of the considered Systems of Systems.

2 Modeling and Simulation of Systems of Systems

Systems of Systems (also called Systems-of-Systems and SoS) are large-scale concurrent and distributed systems that are comprised of complex systems [6]. A system is defined by its separation from the environment, which is everything that is not involved in the system. The system boundary defines the limit of the system with respect to its environment, with which it has interfaces for the exchange of mass, energy or information. The system boundary is typically not identical with the physical limits of a system or its components. A sub-system is an element of a system. The decomposition of a system in its system elements and relations between them and with the system environment results in a (dynamical) structure of the system. A number of important aspects of, and requirements on, system-level models for Systems of Systems Engineering (SoSE) are presented in [6], [7], [8], [9] and are shortly summarized below.

- **Rationale for system-level models:** There is a lack of models that describe structure and behavior of complex technical products in a clear and consistent way. The goal of system-level modeling is to represent the overall system in a more comprehensible way in order to remedy this unsatisfactory situation. System-level models are used to extract the main characteristics and relationships of a system with the aim of showing its requirements, structure, and behavior and to allow a holistic view of the (overall) system under consideration. This is one kind of holistic view enabling a higher level understanding of a complex system and the integration of its constituents. Another kind of holistic component models are capable of capturing the variety of domain specific properties required when integrating such models into a multi-domain model for the purpose of multi-objective optimization of functional and non-functional system properties.
- **Requirements on system-level models:** Due to the increasing complexity of modern technical systems, it becomes more and more difficult even for trained engineers to master the inherent relationships and dependencies between complex sub-systems. System-level models allow the description of the most important sub-systems and their relationships and dependencies of importance for the overall system. Systems often consist of a base-system

which can be extended by optional sub-systems. System-level models should support engineers in considering both (a) base system variants and (b) sub-system options. The basic principle of system-level modeling is to model only those aspects which are essential for the overall system or which are important across engineering disciplines. However, it is necessary to specify principles to decide which relationships and dependencies are to be modeled on the system level rather than on the level of the disciplines involved. Hence, it is important to ensure consistency in both directions - from system-level models to discipline specific models and vice versa.

- **Mechatronic systems and CPS sometimes constitute systems-of-systems:** To understand complex systems, it is convenient or even necessary to decompose them into separate sub-systems. Hence, mechatronic systems may be understood as part of Systems-of-Systems. The boundaries of the involved sub-systems have to be chosen such that the interfaces become clear to all engineering disciplines involved. System-level models therefore illustrate the dependencies between the sub-systems which themselves may consist of solutions from different engineering disciplines, and they provide a multi-level view of the overall system under consideration. To model these dependencies, the definition of well-defined interfaces between the individual systems is very important. Interfaces must support the tracking and communication not only of single values but also of more complex data types such as characteristic curves and key figures.

3 State-of-the-art analysis

As presented in the previous sections, mechatronic systems and Cyber Physical Systems (CPS) can be considered as components of Systems-of-Systems. To design such components, several information systems are currently used by companies to manage their related technical data. For instance, Product Data Management systems are generally used to manage hardware (HW) data whereas Software Configuration Management (SCM) systems are used to manage software (SW) data. PDM is generally considered as one of the most important components in Product Lifecycle Management (PLM) implementation. Indeed, PLM is defined as a systematic concept for the integrated management of all product- and process-related information through the entire lifecycle, from the initial idea to end of life [10]. Focusing on the basic functions (version control, concurrent development, configuration selection and workspace management), SCM is very similar to PDM, as the Application Lifecycle Management (ALM) can be compared to PLM. ALM “has emerged to indicate the coordination of activities and the management of artefacts (e.g., requirements, source code, test cases) during the software product’s lifecycle” [11]. ALM comes from the Configuration Management (CM) domain, which traditionally provides storage, versioning and traceability between all artefacts. It provides extra functionalities to support communication, reporting, traceability and tool integration, such as requirements and defects management, build and test facilities, etc., as illustrated in Fig. 2.

Even if ALM and PLM are converging on a certain number of aspects, challenges remain for mechatronic system and CPS design data management. For instance, having a common bill of materials combining all HW and SW components, defining a common data model managing the entire system at each design phase etc. are still issues. Back to SoS level, the different components, i.e. mechatronic systems or CPS, have to be designed in order to be able to interoperate. This implies some constraints at the data management level, e. g. in term of interoperability. The next paragraph describes the different architectures and techniques that could be used in order to make this data management system interoperable.

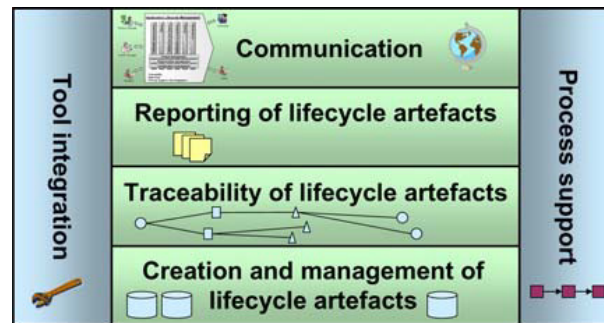


Fig. 2. Main elements of Application Lifecycle Management [11]

To be able to design SoS, the different Information Systems (PDMs and SCMs) supporting the design of the mechatronics and CPS must be interoperable to exchange and share information concerning HW and SW about the different components of the SoS. PDMs and SCMs must be able to communicate, cooperate and exchange services and data, thus despite the differences in languages, implementations, executive environments and abstraction models as defined by Wegner [12]. In several EU projects as ATHENA and INTEROP NoE and according to EIF [13], interoperability requires three different levels. The first one is the technical level to ensure the continuity of information flow through tools. The second one is the semantic level to ensure the understanding of information by the different systems. The third one is the organizational level to ensure the compatibility of the processes used by those systems.

Interoperability between PLM systems is a well-known issue and is addressed by several works at the three levels:

- **Technical level:** The two main strategies are mediator [14] and SAO [15]. In mediator strategy all the information go through a unique gate before being distributed. It is an efficient architecture in terms of agility of IS and total cost of ownership of interfaces [14]. SOA lies on web services that enable the exchange of information regardless of the environment in which the system is executed. SOA is also widely used to solve technical interoperability [15].
- **Semantic level:** Lots of works are based on an integration approach using one common format for all models, especially using STEP ISO 10303 [16].

Unified approaches are based on one common meta-model for all models. They are also well exploited, using for example the NIST Core Product Model [17] or a Product Process Organization meta-model [18]. But nowadays, the more used approach is the federation with the use of ontologies like OntoPDM [19], PROMISE [20] or OntoSTEP-NC [21].

- **Organizational level:** organizational level consists of defining objective, coherence and coordination of processes. It is less treated by PLM scientific literature because it is associated to human problems [22].

SCM interoperability is very few (or not) treated in scientific literature, but the same concepts remain applicable. At present, there is no real solution to solve SCM-PDM interoperability issues [23].

4 Proposed Meta-Model of SoS and Interoperability in SoS Engineering

This section shows a meta-model for describing the structure of a SoS and the relation to the involved information object, which are the base for covering the PLM-process.

Designing the principal structure of a SoS is vital and has to be done early in the engineering process. In this context it is necessary to have a clear understanding of elements of the design and of their relations, which are described in the meta-model (Figure 3).

The focus of this description is to capture the overall structure of the SoS, without considering different views necessary for the different application. The elements of the meta-model are

- **Systems of Systems:** is a set of systems (e.g. autonomous systems and their inter-connections) with independence integrating together to lead to a greater capability that fulfills the demand of the task.
- **System:** This is a kind of a basic element that reflects the SoS to be designed. While objects have to be engineered, systems allow logically grouping objects. Systems are elements of reuse on a more coarse level. Interfaces of systems have to be carefully designed in order to facilitate reuse.
- **System Information Object:** System information objects can explicitly be modeled and are used to design which information is stored about systems or subsystems.
- **Mechatronic/CPS Object:** Element of the System that is to be engineered. Typically, these objects can be decomposed into mechatronic modules. Mechatronic/CPS Objects are candidate elements for reuse and therefore should always be designed with reuse in mind.
- **Mechatronic/CPS Information Object:** Mechatronic information objects can explicitly be modeled and are used to design how information about physical mechatronic objects is stored. When dealing with the structure of a mechatronic system it has to be defined here for which physical mechatronic objects (and possibly their sub-elements) distinct mechatronic information objects should be provided. Ideally it can also be captured which essential

information has to be provided (e.g. CAD, FE-calculations) without going into details of data modeling or data storage structures.

- **Mechatronic Module:** Element of a mechatronic system that is supposed to be provided by suppliers and therefore is not developed in-house. In many cases it will depend on the specific context, whether an element of a mechatronic system is considered to be a mechatronic object or a mechatronic module. A mechatronic module can be of different granularity – it might be a simple nozzle but can also be the entire power train. While mechatronic/CPS objects have to be engineered, mechatronic modules are to be provided by suppliers. The identification of mechatronic modules is crucial for the overall design. As already mentioned above, mechatronic modules are not to be engineered by the organization but are elements that have to be provided by suppliers. It is therefore a strategic decision which parts of a mechatronic objects are engineered by the organization and which parts are provided by suppliers. The decoupling of mechatronic information objects from physical information objects allows structuring the information about physical mechatronic objects in different ways. Depending on the information needs and possibly on the importance of the physical mechatronic objects, different strategies can be used. By default, each physical mechatronic object has exactly one mechatronic information object; for sub-hierarchies of physical mechatronic objects one mechatronic information object might be sufficient for the entire sub-tree of physical mechatronic objects.

Usually, the whole system is in the form of a cross-discipline model. The problem is that the different disciplines, different modeling approaches and models or model descriptions use their integration still leaves something to be desired. The challenge is that the knowledge of the entire system does not equal the sum of knowledge from the corresponding disciplines. System design and evaluation are important topics for which improved tools and knowledge are ever claimed by the engineering profession. A system may be defined as an assembly of sub-systems, hardware and software components, and people designed to perform a set of tasks to satisfy specified functional requirements and constraints. On the lowest level, Modules contains discipline-specific Components, who are either user-defined or standardized parts of mechatronic modules: User-defined (parametric) Components comprise at least one feature with continuous variables (e.g., position and size). For instance, they can be used to create 3D models and manufacturing drawings. The Design Parameters thus specify the user-defined components and the range of allowed values. Standardized Components are described by Configuration Parameters regarding the allowed configuration options, which can for instance be defined in a variability model [24].

Amongst the variety of information available to the designer during the engineering process it is necessary to handle this information content in the PLM model. One important point in this context is that the information content expands or more generally changes depending on the engineering phase. In the planning phase we basically have the customer need and a list of requirements, quite often with uncertain information. In the detailed design the product properties will define all detail on a level of high granularity. The process of integration, enrichment, changes or deletions of related information can have multiple levels of detailing and several

iteration loops. The refinement of a domain model is the concretization and enrichment of information within the domain model. This engineering process is influenced by engineering activities of the design engineers using different knowledge bases.

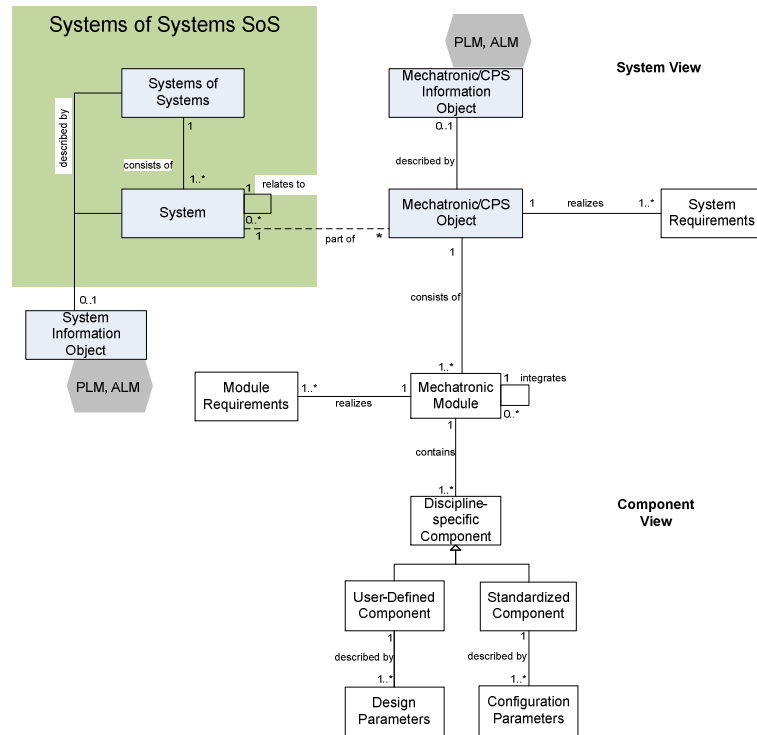


Fig. 3. Meta-model

There are different ways for the consideration of information enrichment during the engineering process: 1.) The existing domain model includes the specific characteristics, which are described on a low level of detailing (low number of artefacts, parameters). In the next design phases the domain model blows up (high number of artefacts, parameters). 2.) Additional aspects will be added during the design phase. 3.) Interaction between the model elements will be added.

Figure 4 shows a first solution concept for the integration of the several PLM/ALM-instances. To ensure the interoperability between the different PLM-ALM software, a Unified approach is considered in this work. Indeed, based on the proposed meta-model, all PDM and SCM instances can be linked to the same meta-model. Then a PDM item and a SCM item representing one mechatronic system could be link to the same mechatronics component object. The HW information contained in the PDM and the SW information from the SCM could be both linked to the mechatronics component information object offering complete information on HW/SW of each component of the SoS. The interoperability issues are mainly

addresses at the semantic level. The validation of the proposed model with an industrial case study will be realized in a next step after the description of all interfaces.

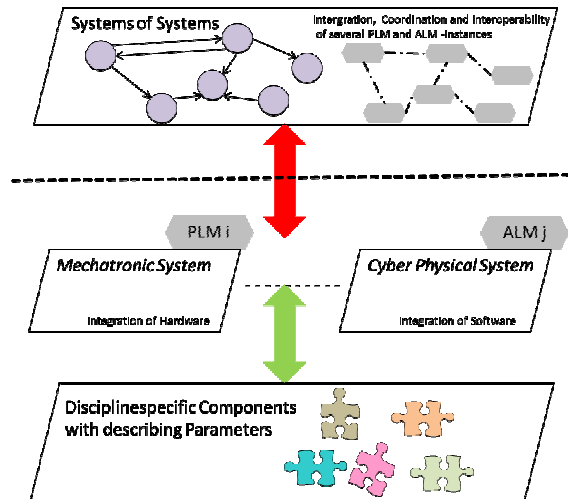


Fig. 4. PLM/ALM instances coordination and interoperability to support SoS design

5 Conclusions and Future Work

This paper discusses the support of Modelling and Simulation Tasks for SoS with the consistency of product model from Components, Mechatronics System, CPS and SoS and the Interoperability PLM/ALM for SoS. The main problems are described and first solution concepts are presented. Nevertheless there is a lot of work to prepare a PLM-Model for SoS and also to analyze the influence for the IT-architecture.

Acknowledgments. This work has been partially supported by the Linz Center of Mechatronics (LCM) in the framework of the Austrian COMET-K2 program and the Labex MS2T at the Université de Technologie de Compiègne.

References

1. Bishop, R.H.: Mechatronic Fundamentals and Modeling (The Mechatronics Handbook). CRC Press Inc, New York (2007)
2. De Silva, C.W.: Mechatronics – an integrated Approach”, CRC Press Inc, Boca Raton, London, New York, Washington DC (2005)
3. VDI 2206. Design Handbook 2206, Design Methodology for Mechatronic systems”, VDI Publishing Group, Düsseldorf, German. (2003)
4. Lee, E.A. Cyber Physical Systems: Design Challenges, In: Proc. 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC), (2008)

-
5. Haveman S.P., Bonnema G.M.: Requirements for high level models supporting design space exploration in model-based systems engineering, Conference on Systems Engineering Research (CSER 13) Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013. *Procedia Computer Science* 16 (2013)
 6. Jamshidi, M. (Ed.): *Systems of Systems Engineering: Principles and Applications*. CRC Press, ISBN 9781420065886 (2008)
 7. Bilal, M., Daclin, N., Chapurlat, V.: Collaborative Networked Organizations as System of Systems: A Model-Based Engineering Approach. In: *IFIP Advances in Information and Communication Technology*, vol. 434, 2014, pp 227-234. Springer, Heidelberg (2014)
 8. Stevens Institute of Technology, Castle Point on Hudson, Hoboken, N. 07030: Report on System of Systems Engineering (2006)
 9. Kopetz, H.: A Conceptual Model for the Information Transfer in Systems-of-Systems. In: *Proc. of ISORC 2014*, Reno, Nevada, June 2014. IEEE Press (2014).
 10. Saaksvuori, A., Immonen, A.: *Product Lifecycle Management*. Springer, Berlin (2008)
 11. Kääriäinen, J., Välimäki, A.: Applying Application Lifecycle Management for the Development of Complex Systems: Experiences from the Automation Industry. In: *Software Process Improvement*, vol. 42, Eds. Berlin, Heidelberg: Springer, 2009, pp. 149-160 (2009)
 12. Wegner, P.: Interoperability. *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 285–287 (1996)
 13. EIF: European Interoperability Framework, White Pages. pp. 1–40 (2004)
 14. Paviot, T., Cheutet, V., Lamouri, S.: Design and logistics IT federation through Product Lifecycle Support standard, in *PLM09, IFIP WG 5.1* (2009)
 15. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orhcard, D.: *Web Services Architecture*. W3C Working Group, W3C Working Group Note (2004)
 16. Lee, S.-H., Jeong, Y.-S.: A system integration framework through development of ISO 10303-based product model for steel bridges. *Automation in Construction*, vol. 15, no. 2, pp. 212–228 (2006)
 17. Sudarsan, R., Fenves, S.J., Sriram, R.D., Wang, F.: A product information modeling framework for product lifecycle management. *Computer Aided Design*, 37(13):1399–1411 (2006)
 18. Noël, F., Roucoules, L., Teissandier, D.: Specification of product modelling concepts dedicated to information sharing in a collaborative design context. *Advances in integrated design and manufacturing in mechanical engineering*, p. 135–146 (2005)
 19. Panetto, H., Dassisti, M., Tursi, A.: ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. *Adv. Eng. Informatics*, vol. 26, no. 2, pp. 334–348 (2012)
 20. Matsokis, A., Kiritsis, D.: An ontology-based approach for Product Lifecycle Management, *Comput. Ind.*, vol. 61, no. 8, pp. 787–797 (2010)
 21. Danjou, C., Le Duigou, J., Eynard, B.: Closed-Loop Manufacturing process based on STEP-NC, *International Journal on Interactive Design and Manufacturing*, <http://dx.doi.org/10.1007/s12008-015-0268-1> (2015)
 22. Vernadat, F.: Technical, semantic and organizational issues of enterprise interoperability and networking, *Annual Reviews in Control*, vol. 34, no. 1, pp. 139–144 (2010)
 23. Bricogne, M., Rivest, L., Troussier, N., Eynard, B. Concurrent versioning principles for collaboration: towards PLM for hardware and software data management, *International Journal Product Lifecycle Management*, vol. 7, no. 1, pp. 17–37, <http://dx.doi.org/10.1504/IJPLM.2014.065457> (2014)
 24. Lettner, D., Hehenberger, P., Nöhrer, A., Anzengruber, K., Grünbacher, P., Mayrhofer, M., Egyed, A. Variability and Consistency in Mechatronic Design, *Concurrent Engineering: Research and Applications*, published online DOI: 10.1177/1063293X15585008 (2015)