

Towards Smart Product Lifecycle Management with an Integrated Reconfiguration Management

Michael Abramovici, Jens Göbel, Philipp Savarino, Philip Gebus

► **To cite this version:**

Michael Abramovici, Jens Göbel, Philipp Savarino, Philip Gebus. Towards Smart Product Lifecycle Management with an Integrated Reconfiguration Management. 14th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2017, Seville, Spain. pp.489-498, 10.1007/978-3-319-72905-3_43. hal-01764182

HAL Id: hal-01764182

<https://hal.inria.fr/hal-01764182>

Submitted on 11 Apr 2018

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.



Towards Smart Product Lifecycle Management with an integrated Reconfiguration Management

Michael Abramovici, Jens Christian Göbel, Philipp Savarino*, Philip Gebus

Chair of Information Technology in Mechanical Engineering (ITM),
Ruhr-University Bochum, Germany
philipp.savarino@itm.rub.de

Abstract. Recent ICT innovations determine dramatic changes of traditional products towards intelligent, connected Smart Products. These product-related changes also imply the need for a fundamental adaptation and enhancement of traditional Product Lifecycle Management approaches. Analyzing the main characteristics of Smart Products reveals that lifecycle management approaches for Smart Products especially have to extend their focus on the product use phase. A core challenge in this context is to provide suitable methods and IT tools for the reconfiguration of Smart Products across different engineering domains. This contribution introduces a conceptual approach for the reconfiguration of Smart Products, which considers the dynamical changes of virtual and physical product instances based on the virtual product twin. This approach was implemented and validated prototypically in a model-environment considering smart vehicles, that were temporarily reconfigured during their use phase

Keywords: Reconfiguration, Smart Products, Product Lifecycle Management, Virtual Twin

1 Configuration of traditional and reconfiguration of Smart Products

Recent innovation in Information and Communication Technology (ICT) have begun to dramatically change traditional products towards intelligent, connected Smart Products (SP). Characteristic features of Smart Products are their capability to communicate and interact with their environment and other Smart Products by using Internet-based services [1]. They have a high degree of personalization and autonomy, a large number of multidisciplinary components, the ability to react in real-time and their dynamic reconfiguration potential during their whole lifetime [2]. Especially the high amount of software components enables a tremendous potential for IT-based reconfiguration, e.g. like a (temporary) parking assistant for a smart vehicle. However such a reconfiguration requires suitable methods and IT tools e.g. for the management of virtual and physical product instances that exceed common Product Lifecycle Management approaches [3].

A reconfiguration can be generally described as a modification of a product instance

to meet new requirements [4]. Considering Smart Products, reconfiguration processes during the use phase primarily aim at:

- The general technical improvement of product classes (e.g. the implementation of new software releases for a new generation of smart vehicles).
- The individual improvement of product instances by enhancing the functional amount (e.g. IT-based parking assistance systems for smart vehicles) or IT-services (e.g. a cloud-based, individually improved energy management for smart vehicles).

As a conceptual basis for the methodological reconfiguration of Smart Products, existing product configuration approaches providing initial customer specific virtual product models can be used. The normative standard ISO 10007 defines configuration units, which are entities within a configuration that provide a specific end use function. These configuration units are part of a configuration management model that also addresses the change management of these units during a reconfiguration [5]. However, this requires knowledge of all product instance configuration units along the entire product lifecycle. The management of configuration units can also be done by using product configurators aiming at the mass customization of products. They base on system of rule describing dependencies and requirements between configuration units. These rules have to be maintained and updated along a product's lifecycle, too [6]. The IEC 61499 standard defines function blocks to manage control unit configurations with a focus on (agile) manufacturing systems [7]. This methodological approach has been enhanced in several research works by addressing reconfiguration issues with a focus on electronic or software components [8,9].

Smart vehicles can be regarded as a representative example for a Smart Product due to its complexity and commercialization, which is also part of cross sectoral smart systems (e.g. smart energy systems). The analyses of reconfiguration processes of smart vehicles have shown that in contrast to the previously mentioned approaches for an initial, customer specific configuration of virtual product model variants a Smart Product configuration additionally requires:

- A continuous development and enhancement of virtual product models considering the compatibility to existing virtual product models that are part of product instances in the use phase.
- An individual management of all product instances across all engineering domains.
- A synchronization of the initial virtual product configuration models and the multiple reconfiguration variants of all product instances.
- A continuous development and enhancement of the configuration knowledge for all physically existing product instances.
- An integration of external service providers that can be integrated into the reconfiguration process of Smart Products during their use phase.

In our previous contribution at the PLM16 conference we presented a conceptual approach considering virtual product twins as integrated components of Smart Products [10]. As a result of our follow-up research activities this paper focusses on the ability of virtual twins to serve as an enabler for Smart Product reconfiguration.

2 Virtual twins as enabler for the reconfiguration of Smart Products

While the traditional product development concept can be sufficient for the management of the initial virtual product configuration models, the management of virtual product instance models for the reconfiguration of Smart Products along the entire lifecycle is characterized by an enormously high complexity. The consistent, situational consideration of virtual product models including the integration of product use data exceeds traditional Product Lifecycle Management methods. In this chapter, we will show how virtual product twins can serve as an approach for the integrated consideration of virtual product lifecycle models and product use data.

2.1 Virtual product twin concept

As mentioned in section 1, product configurations of Smart Products dynamically change along the entire product lifecycle. This leads to a new spectrum and history of virtual product models based on previous product instance versions. Specific characteristics of Smart Products like autonomy and IT-based reconfiguration services however require also the consideration of product use data as an extended lifecycle management approach. Therefore, the concept of the virtual product twin is introduced for the management of all the virtual product models as well as the product use data (Figure 1). In this context, different terms addressing similar research questions, like for example product avatars as a product-instance centric management concept, arose in the past decade [11]. In the scope of Smart Products, the virtual twin can be considered as the notion, where the data of each stage of the product lifecycle is transformed into information and is made seamlessly available to subsequent stages [12]. The transformation into information can be e.g. the intention to analyze the data within its semantic context or the use of the data to predict the product's behavior.

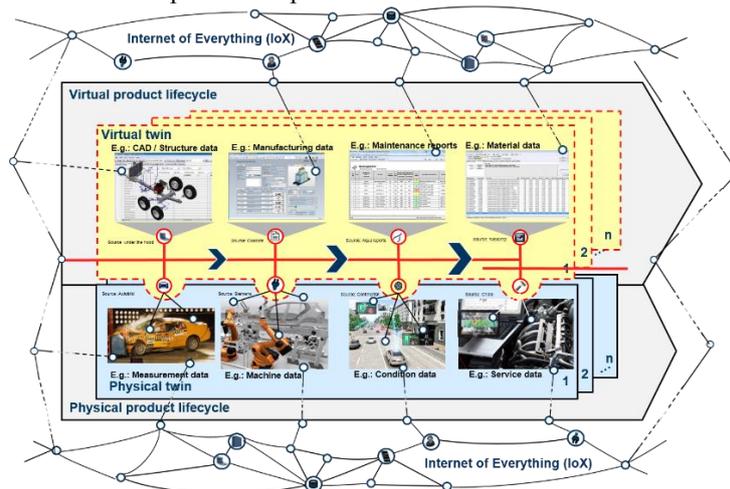


Fig. 1. Conceptual model of a virtual product twin for the lifecycle of a Smart Product.

As every virtual twin has a physical twin, the separation between a virtual and a physical lifecycle is necessary. As Smart Products combine highly interdisciplinary components from different engineering domains like mechanics, electronics and informatics, the virtual twin needs to synchronize context-specific multi-disciplinary models with the respective physical twin. This integration enables e.g. a reconfiguration like a temporary IT-service for a customer based on product structure data, CAD-models and software component versions [10]. Exemplary specific product instance models and data generated along both virtual and physical product lifecycles are shown in Figure 1. In the early phases of the virtual product lifecycle, e.g. the product structure is defined, while in the following phases the management of maintenance reports is focussed. Analogously, the pictured physical lifecycle describes data that is primarily generated in different lifecycle stages by the physical product twin. This can refer to service data in a product's late lifecycle stage for example, which can be used for conclusions about a product's component condition and the possibility of further usage. However, data from the physical product lifecycle cannot be ignored when aiming at the composition of a product's comprehensive digital copy. E.g., product condition data is of essential importance in order to describe the current behavior or status of product. Therefore, they must be part of the virtual twin.

Considering that the data from every lifecycle phase is transformed into information and is made seamlessly available to subsequent phases, the amount of information available through the virtual twin increases together with the product's lifecycle stages. Consequently, this refers to a higher spectrum of context-specific deployment possibilities of the virtual twin of a Smart Product in the late phases of the product's lifecycle.

2.2 Reconfiguration lifecycle of Smart Products during their use phase

The changes in engineering affected by the transformation from traditional to Smart Products demand a rethinking regarding the relations between the generic variant development in the traditional product development phase and the affiliated product configuration. The introduced product reconfiguration lifecycle management approach (figure 2) addresses these changes.

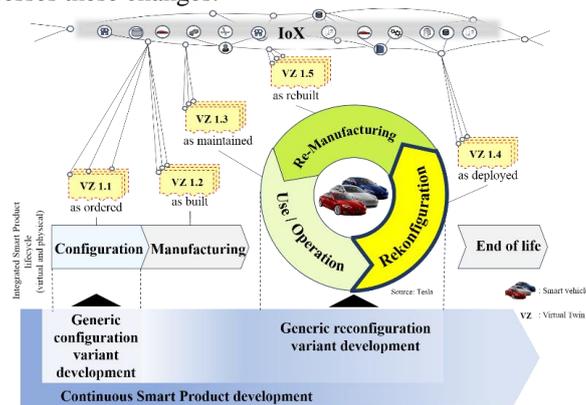


Fig. 2. Reconfiguration lifecycle in the product use phase.

The given approach includes a continuous Smart Product development phase that can be regarded as a lifecycle-attending instead of an early-lifecycle only element, which is addressed by traditional Product Lifecycle Management methods. Consequently, the provider of a Smart Product is highly involved in all lifecycle phases. The initial early phases of Smart Products lead to generic, modular, virtual product models and structures, which are the basis for the configuration of individual, customer-specific product variants. Common technical solutions for the customer-specific configuration are for example online product configurators that are highly popular in the automotive sector (e.g. BMW). The resulting configuration however describes the product only to a specific point in time or in a defined status for the distribution that can be subject to various reasons of change. Hence, the initial configuration by the customer is only one state (as ordered) that can be part of traditional PLM-approaches. However, this state varies dynamically along the entire lifecycle e.g. due to stochastic influences or necessary adjustments. Likewise, the product condition can be defined as an as built configuration after the manufacturing and distribution process, before entering a cyclic process in the use phase consisting from use/operation (as maintained), reconfiguration (as deployed) and re-manufacturing (as rebuilt). The different shapes of the physical product twin along the entire product lifecycle require simultaneously the adjustment of the virtual product twin (compare the different versions of the virtual product twins in figure 2). The generic reconfiguration options during the use phase are part of the continuous product development. Therefore, the customer can individualize his product further while choosing from variants that did not necessarily had to be existent during the very first configuration. Regarding a smart vehicle, this can refer to either mechanic, electronic or software components or more likely, to a combination of those that can be reconfigured depending on their compatibility. For example an IT-service such as a parking assistant can be offered as a (temporary) reconfiguration providing a given sensor technology, although this service has not been part of the company's product portfolio at the time of the initial configuration. A core challenge is the synchronization of the initial virtual product configuration models with the multiple, reconfigured variants of the product instances. The necessary models regarding the configuration knowledge for the introduced reconfiguration cycle in Figure 2 are product structure models (e.g. BOM or structure trees) including the attached product models (e.g. metadata of the product components or CAD-models). Source for this data can be Product Lifecycle Management (PLM) systems, however due to the complexity of Smart Products the necessary knowledge can be scattered over different PLM systems in an engineering collaboration partner network [13]. In order to offer reconfiguration options during the use phase as a result of the continuous product development, this knowledge needs to be centralized as it is part of the virtual product twin (compare chapter 2.1).

Hence, the virtual twin can be considered as a fundamental element when enabling a reconfiguration such as the previously introduced example of a parking assistant. As a first step, the product instance configuration has to be identified which can be done by analyzing the given product structure. After that, the necessary structural components for the parking assistant have to be compared to those identified regarding the compatibility of the reconfiguration option. This step does not necessarily include only

configuration knowledge that has been created in-house, but has to consider engineering-knowledge from the whole partner network as mentioned above. After the rule-based compatibility checks have been conducted successfully, the implementation of the component-related reconfiguration can be initialized. Considering the implementation of the reconfiguration as a transformation of the product instance, this must be consequently regarded as a transformation of the virtual product twin as well.

3 Model environment for the reconfiguration of Smart Products

This chapter introduces a realization of the previously presented approach of the reconfiguration of Smart Products based on virtual products twins by transferring it to a model environment. The setup of this model environment consists of a conceptual and experimental part that were deduced from the use case in chapter 4.1.

3.1 Considered Use case

The use case in this chapter shall emphasize the benefit of the reconfiguration cycle while underlining the need for virtual twins in order to profit from a continuous product development. It describes the reconfiguration process of a smart vehicle, which will be the basis for the conceptual design of a reconfiguration platform followed by the implementation by means of a developed software-prototype. The basic idea of this use case is that a customer is owner of a smart vehicle whose functions can be temporarily or constantly altered. At first, the physical product instance is identified and connected to the respective virtual twin. A compatibility check considering the product instance configuration and all possible reconfiguration options follows. This includes all options that are currently available according the continuous product development process. The compatible, offered reconfiguration option in this use case is a parking lot detection assistant. Upon activation of the IT-service by the customer, the virtual twin changes analogously as part of the reconfiguration process. After the implementation, the IT-service enables the retrieval of free parking lot positions detected by sensors from other smart vehicles as soon as they pass an empty parking box. Finally, the customer can navigate to the free parking lot.

3.2 Components of a cloud-based reconfiguration platform

In order to realize the reconfiguration approach for Smart Products in their use phase based on virtual product twins, the cloud-based reconfiguration platform architecture has been developed. It allows the integration of the necessary product instance data as well as the management of the reconfiguration options and rules (Figure 3).

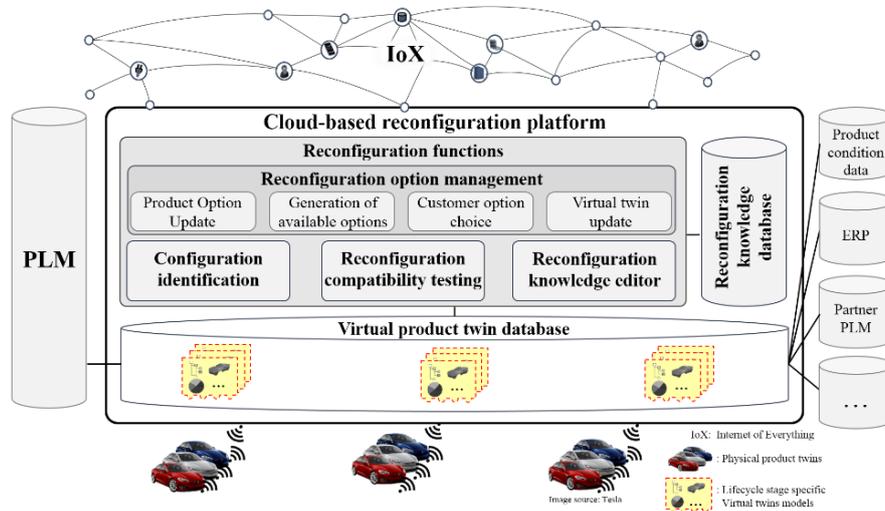


Fig. 3. Extension of the traditional PLM-concept by a cloud-based reconfiguration platform for Smart Products.

The reconfiguration platform consists of two main sections: Functional modules that combine all operational units for the reconfiguration process and data sources that provide the necessary data for the reconfiguration process.

The central unit of the data providing section is the **virtual product twin database**. It assures the assignment of virtual product models from the entire lifecycle to the physical product instances via wireless connection e.g. as part of a service of an Internet of Things (IoT) platform. The virtual product models for the virtual product twin database are supplied by Product Lifecycle Management (PLM), e.g. by considering product structure data, and enterprise resource planning (ERP) systems of the entire partner network, as the complexity of Smart Products requires a highly flexible engineering-collaboration. Additionally, product condition data from the physical lifecycle has to be considered as part of the virtual product twin database as well. They can be used to trigger reconfiguration events, e.g. when passing a threshold.

Core of the function modules is the **reconfiguration option management**. The product option update is responsible for the consideration of the continuous product development notion. The basic idea is to keep the possible (re)configuration variants up to date in the sense of including newly developed variants while excluding those who are not part of the company's portfolio anymore. The second module is needed in order to generate only available reconfiguration options for each product instance. This assures that only those reconfiguration options are available for a product instance whose compatibility has been positively checked before. The customer choice option module is giving the customer the possibility to choose from the available reconfigurations, which can be realized by a graphical user interface. The fourth module is responsible for the continuous updating process of the virtual product twin. After the customer has chosen a reconfiguration option, the product instance changes as well as the virtual product twin.

Regarding the **configuration identification**, two different kinds of identification layers need to be considered: A general configuration is linking the physical product instance to all the related virtual product models as well as to the related physical lifecycle data. This refers e.g. to the assignment of the related CAD-models, current product structure or the localization of the present product position. The second layer addresses the product-internal selection of configuration units that are affected by the reconfiguration process (specific configuration identification).

The **reconfiguration compatibility testing** module checks the current product instance against missing configuration units that exclude any possible reconfiguration options. One core challenge is assuring the compatibility of a product instance with software and/or hardware components that were developed after the product instance (forward compatibility), e.g. by using neutral data formats. On the other hand, backward compatible reconfiguration options are offering potential for various implementations.

The compatibility between a product instance configuration and all possible reconfiguration options can be tested by applying a rule-system, which define the requirements and dependencies between the configuration units. These rules can check for example the availability of necessary sensors in a product instance configuration before offering an IT-service like a parking assistant. These rules are managed in a **reconfiguration knowledge database** and are based on the rules at the time of the initial product configuration but have to be updated respectively enhanced due to the continuous product development. This database is source for the **reconfiguration knowledge editor**. The editor is required to adjust the rules from the reconfiguration knowledge database to the existing product instance configuration with all the requirements and dependencies of the implemented configuration units.

3.3 Technical implementation and validation of the platform

The cloud-based reconfiguration platform described in chapter 4.2 was implemented and validated in a model environment. To model the characteristic properties of a Smart Product, the technical features of two Lego Mindstorm robots were augmented by using a Raspberry Pi to enable Internet-connectivity of sensor data like speed, position or distance to objects. Core of the reconfiguration platform is the cloud-platform “ThingWorx” commercially available by the provider “PTC”. The service oriented architecture offers a broad range of interfaces, e.g. to the PLM-software “Windchill” (also PTC). Hence, ThingWorx offers the possibility to link physical lifecycle data (in our case retrieved by the sensors from the Lego Mindstorm robots like position data and ultrasonic data) with virtual lifecycle data (in our case product structure data from Windchill). Technically, the robots were described in ThingWorx as “things” and the product structure data as well as the sensor data were described as the things’ “properties”. According to previously defined query intervals, the BOM-data and data values from the ultrasonic-sensors as well as the robots’ positions are updated constantly in order to monitor the product instance configuration and condition continuously. After an algorithm had checked the compatibility between the reconfiguration option “parking lot detection assistant” and the given product structure the software update was

implemented. Following a necessary reboot, the robot was able to cyclical query the position of the other robot. Regarding the semantic context of the given use case the position of the Lego Mindstorm robot was always then transferred if the ultrasonic sensor at the side of the robot was exceeding a previously defined value. This position was indicating a free parking lot (see figure 5).

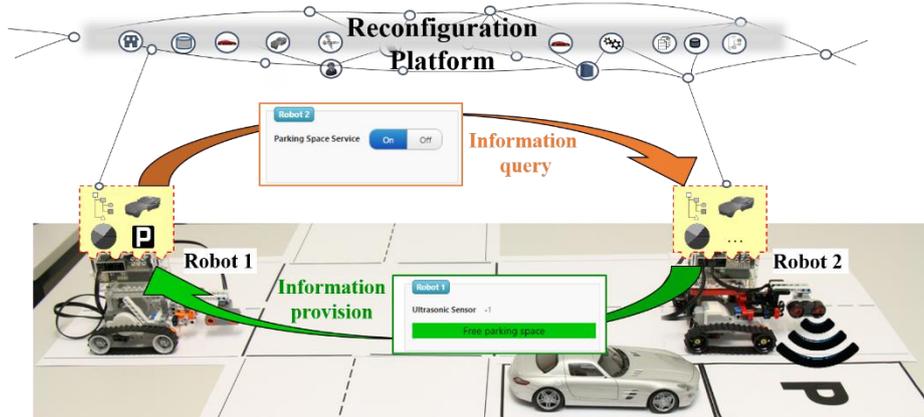


Fig. 4. Reconfiguration example based on virtual product twins.

4 Conclusion

Smart Products offer a tremendous potential for reconfiguration during their use phase due to their multidisciplinary properties. Especially the broad range of IT-services can enable an augmentation of functions by means of Internet technology in combination with hardware. This contribution showed the necessity of a continuous product development that is supported by a Product Lifecycle Management in order to access the full potential of reconfiguration offers for Smart Products. However, the dynamical reconfiguration of Smart Products is neither conceptual nor technological sufficiently addressed to the present day. Hence, a conceptual reconfiguration platform based on virtual product twins was introduced in order to take one step towards a higher reconfigurability of Smart Products during their use phase. Fundamental core of the virtual product twin concept was an integrated Product Lifecycle Management. The approach is especially characterized by its openness regarding the integration of more functional modules and data sources in order to transfer it to different Smart Product-Service Systems (Smart PSS). Challenges like latency and availability of needed Internet infrastructure, data security and access management questions as well as the scalability for a high number of product instances are still remaining. Current research activities focus semantic data management models and the integration of product behavior models into the reconfiguration cycle.

References

- [1] M. Abramovici, Smart Products, in Lapierrière L, Reinhart G, (Eds) CIRP Encyclopedia of Production Engineering, Springer, Berlin/Heidelberg 1-5., Springer, Berlin Heidelberg, 2014.
- [2] M. Abramovici, H.B. Dang, J.C. Göbel, P. Savarino, Systematization of IPS2 Diversification Potentials Using Product Lifecycle Data, *Procedia CIRP* 47 (2016) 288–293.
- [3] E. Lutters, F.J. van Houten, A. Bernard, E. Mermoz, C.S. Schutte, Tools and techniques for product design, *CIRP Annals - Manufacturing Technology* 63 (2) (2014) 607–630.
- [4] T. Männistö, T. Soininen, J. Tiihonen, R. Sulonen (Eds.), Framework and conceptual model for reconfiguration. Configuration Papers from the AAAI Workshop (AAAI Technical Report WS-99-05)(pp.59-64), AAAI Press, 1999.
- [5] ISO 10007:2003, Quality management systems- Guidelines for configuration management.
- [6] L.L. Zhang, P.T. Helo, A. Kumar, X. You, Implications of product configurator applications: An empirical study, in: *IEEM 2015: 2015 IEEE International Conference on Industrial Engineering and Engineering Management* 6-9 December 2015, Singapore, Singapore, Singapore, IEEE, Piscataway, NJ, 2015, pp. 57–61.
- [7] DIN EN 61499, International Electrotechnical Commission 61499-1: Function Block - Part 1 Architecture, 2013.
- [8] W. Lepuschitz, A. Zoitl, M. Vallée, M. Merdan, Toward Self-Reconfiguration of Manufacturing Systems Using Automation Agents, *IEEE Trans. Syst., Man, Cybern. C* 41 (1) (2011) 52–69.
- [9] S. Olsen, J. Wang, A. Ramirez-Serrano, R.W. Brennan, Contingencies-based reconfiguration of distributed factory automation, *Robotics and Computer-Integrated Manufacturing* 21 (4-5) (2005) 379–390.
- [10] M. Abramovici, J.C. Göbel, P. Savarino (Eds.), Virtual Twins as Integrative Components of Smart Products: Proceedings PLM16 International Conference on Product Lifecycle Management, 11-13 Juli Columbia, SC, USA, 2016.
- [11] K.A. Hribernik, L. Rabe, K.D. Thoben, J. Schumacher, The product avatar as a product-instance-centric information management concept, *IJPLM* 1 (4) (2006) 367.
- [12] R. Rosen, G. von Wichert, G. Lo, K.D. Bettenhausen, About The Importance of Autonomy and Digital Twins for the Future of Manufacturing, *IFAC-PapersOnLine* 48 (3) (2015) 567–572.
- [13] D. Penciu, A. Durupt, F. Belkadi, B. Eynard, H. Rowson, Towards a PLM Interoperability for a Collaborative Design Support System, *Procedia CIRP* 25 (2014) 369–376.