A Federated Enterprise Architecture and MBSE Modeling Framework for Integrating Design Automation into a Global PLM Approach

Thomas Vosgien[1(✉)], Eugen Rigger[1,2], Martin Schwarz[3], and Kristina Shea[2]

[1] V-Research GmbH, Dornbirn, Austria
{thomas.vosgien,eugen.rigger}@v-research.at
[2] ETH Zürich, Zürich, Switzerland
kshea@ethz.ch
[3] Liebherr-Werk Nenzing GmbH, Nenzing, Austria
martin.schwarz@liebherr.com

Abstract. PLM and Design Automation (DA) are two interdependent and necessary approaches to increase the performance and efficiency of product development processes. Often, DA systems' usability suffers due to a lack of integration in industrial business environments stemming from the independent consideration of PLM and DA. This article proposes a methodological and modeling framework for developing and deploying DA solutions within a global PLM approach. This framework supports the identification of DA potentials and the definition of the DA task building blocks to support DA task formalization by practitioners. The aim is to make the specification and development of DA solutions more efficient and aligned with the business requirements and with the existing digital environments. This framework combines the usage of two standardized modeling languages to make the captured knowledge re-usable across heterogeneous PLM and DA applications. An industrial case study demonstrating the applicability of the framework is introduced and discussed.

Keywords: Design automation · Product Lifecycle Management
Enterprise architecture · Model-Based System Engineering · ArchiMate · SysML

1 Introduction

Product Lifecycle Management (PLM) has become a central management approach for managing product information, engineering processes and applications along the different phases of the product lifecycle [1]. Around 70% of costs for the market launch of new products are defined in the very early phases of the product lifecycle; i.e. product specification and development [2]. The engineering design departments of manufacturing companies are hence under increasing pressure to perform better in terms of low-time, high-quality and high value output that can provide competitive advantage for the organization [3]. Design Automation (DA) has already been identified as a key enabler for addressing these challenges [4] and is defined as the automatic running of a task or a sequence of tasks performed in an engineering design process [5] and can be divided

into two types: information handling (acquisition, retrieval, and analysis) and knowledge processing [6]. That is why, on one hand, DA should be considered as a key enabler of a PLM approach increasing design process efficiency and supporting different types of concurrent engineering and Design-for-X approaches (anticipating and integrating downstream activities' constraints as early as possible in the product development phase). On the other hand, the acquisition, formalization and re-use of the engineering knowledge consumed or generated by DA applications strongly rely on the capabilities and the usage of PLM enabling technologies (CAX and IT systems as well as their interfaces). Therefore, companies developing mechanical products, have to consider the advantages of Engineering DA (EDA), its realization, implementation as well as its applicability and integration in their specific business environment. However, there is a discrepancy between availability of DA methods stemming from academia and their industrial application [7]. Reasons for that are uncertainties with respect to awareness of available opportunities, recognition of potential of applying DA and ability to define the automation task [7]. In order to overcome above mentioned shortcomings and pave the way for more systematic implementation of DA, this paper introduces a methodological and modeling framework supporting the identification of DA potential within the product development lifecycle and the specification of the required DA task building blocks to clearly define the context of a design task and thereby support DA task formalization by practitioners. The framework combines the usage of two standardized and neutral modeling languages to make the captured knowledge computational and platform independent and to enable the re-use of this knowledge across heterogeneous PLM and DA systems. Section 2 evaluates the current state of the art with respect to enterprise architecture (EA) modeling methods and approaches supporting the specification and development of PLM approaches as well as with respect to system engineering (SE) approaches and computational design task definition. Section 3 introduces the proposals for an EA and Model-Based System Engineering (MBSE) methodological framework including a DA task formalization methodology. Section 4 illustrates the application of the methodology on an industrial case study: the formalization of an optimization task for dimensioning box-type booms of maritime cranes designed and manufactured by Liebherr Werk Nenzing GmbH (LWN). Finally, the results and limitations of the proposed framework and methodology are discussed in Sect. 5 before concluding the paper and presenting lines of future work.

2 Background

DA, as part of a PLM strategy, is not only a technical solution for automating design tasks but also a strategic answer that has to consider many aspects of the company such as: the strategic business drivers, the specific business processes and related requirements; the different authoring and IT applications or platforms used for implementing and/or integrating DA methods; the interfaces between interdependent business processes, authoring applications and IT systems enabling all these elements to interoperate together; the IT infrastructures hosting and enabling these applications to be efficiently integrated and used within and outside the company; the complexity of the

system to be designed, the formalization of the related engineering knowledge required for performing design tasks and finally the human factor, i.e. the user interaction, usability and user acceptance with respect to DA systems. One way to address and apprehend the complexity of such business digital environments is to use EA considering the different dimensions and elements listed above [8–10]. The second way is to evaluate methods and approaches for computational design task definition in order to enable practitioners to specify the required DA task building blocks.

2.1   EA Frameworks, SE Standards and PLM Applications

According to [9], "achieving alignment between business, application, information and technologies (IT) requires an integrated approach to all aspects of the enterprise" and EA is an important instrument to address this company-wide integration. Further, it provides "a coherent whole of principles, methods and models that are used in the design and realization of the enterprise's organizational structure, business processes, information systems, and infrastructure" [11]. However, as highlighted in [9], these domains are generally not approached in an integrated way; each domain speaks its own language, draws its own models, and uses its own techniques and tools. Therefore, it is important that the EA can be represented with relevant information and at the appropriate level of detail for all involved stakeholders [10]. For this purpose, several EA approaches, frameworks and methods have emerged since the 90's whether from the literature (e.g. Zachman, CIMOSA) or from standardization initiatives (e.g. IEEE-1471, ISO/IEC/ IEEE-42010, TOGAF). In literature, it is possible to distinguish between simple methods of representation (e.g. SADT, IDEFx) and reference architectures (e.g. CIMOSA, Zachmann, TOGAF). As highlighted in [12], most of these framework approaches aimed at representing business user's concerns with no direct link to IT implementation. Moreover, these frameworks and methods are generally complex to implement [10]. One reason of these difficulties is due to the existence and cohabitation, according to the viewpoints and domains, of different types of interrelated representations and modeling languages. The co-evolution and hence the consistency maintenance of these interrelated models across time as well as the interoperability between these models and the modeling tools implementing these languages represent major open-issues for efficiently implementing EA. The deployment of a PLM approach can only be achieved through the alignment between business processes, applications, information and technologies and should hence rely on EA modeling and monitoring. Nevertheless, few works can be found in the literature proposing and/or demonstrating the crucial role and contribution of EA for modeling, specifying and monitoring the architecture of the complex system of systems (considering simultaneously the system to be designed, its environment, its interfaces as well as the system for designing; i.e. resources such actors, CAx and IT systems) which is beyond a PLM approach. In [12], it is shown that most of the recent works in EA address the development of frameworks for interoperability, e.g. the IMAGINE and SIP projects. The latter focuses on interoperability through the implementation and evaluation of PLM standards, but also proposes to use EA and ArchiMate to model standards-based business collaboration scenarios and to model the test bed environment that will enable the execution/simulation of this scenario [13]. The SIP

project also considers standards and practices of both PLM and SE communities, since PLM and SE are closely related. As stated in [13], although the scope of application of PLM is larger than the one covered by SE and a PLM strategy can be efficiently deployed being SE processes independent. Whereas ISO-15288, EIA-632 and IEEE-1220 are standards for SE process formalization, SysML has been established as a product data exchange standard for requirements and system architecture models. One goal of this work is to study and adapt EA frameworks to specify and model DA business scenarios.

2.2 Computational Design Task Definition

Generally, implementing DA requires a deep insight in the design process to be able to capture and formalize the principles in the design domain. This typically requires a set of building blocks (i.e. components/modules), which can be combined in certain ways to result in the product fulfilling the customer's requirements. Depending on the purpose of the automation task, the assembling procedure can be fixed yielding exactly one solution, or capable of exploring various assembling strategies resulting in a solution space. In [14, 15], building blocks for definition of conceptual design task are presented. However, the context of a task with regards to design process is not considered. With the intention of providing an easy-to-use categorization of DA tasks, in [4], authors introduce a categorization that puts design tasks that are suitable for automation into context with a generic design process, so to close the gap between product states and formalization. With a focus on reusability of task related knowledge, in [16, 17], authors propose a hierarchical decomposition of a design task to the level of granularity that enables re-use of templates that can be adapted and integrated for the given design task. In [18], authors address the formulation of process templates introducing an ontology-based approach including verification of inputs by means of rules. However, neither the usage of a standardized language that enables reuse of knowledge in a broader context, nor the context with EA is considered within these studies. MBSE "is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities […]" [19]. The SysML language supports such a MBSE approach by providing graphical representations and the semantic foundation for modeling system requirements, behavior, structure, and parametric system representations. With a focus on formalization of simulation-based design tasks, [20, 21] show the applicability of SysML for integrating design and analysis models. However, further analysis is needed to streamline and standardize modeling in SysML for the various design tasks and guide the designers for specifying a design task. In this paper it is proposed to use design task specific modeling templates and SysML stereotypes for modeling task specific knowledge according to given DA task categories.

3   The EA-MBSE Methodological Framework for Design Automation Task Formalization

In this section, a modeling framework and methodology based on the open, independent and standardized ArchiMate architectural framework and language [22], is proposed.

ArchiMate divides the EA into a business, applicative and technological layer and is partly based on the IEEE-1471. It permits to describe, analyze and visualize architectures within and across business domains with a restrictive number of artifacts and relationships. The easy-to-use implementation of ArchiMate in the free open source tool Archi®, as well as the possibility to define and re-use pre-defined models templates, were also determining. Further, it is proposed to combine the usage of ArchiMate EA models with SysML models for the definition and implementation of DA task specific building blocks.

3.1   The EA Modeling Framework

The methodology on which the proposed framework has been built is illustrated on the sketch of ArchiMate views of Fig. 1.
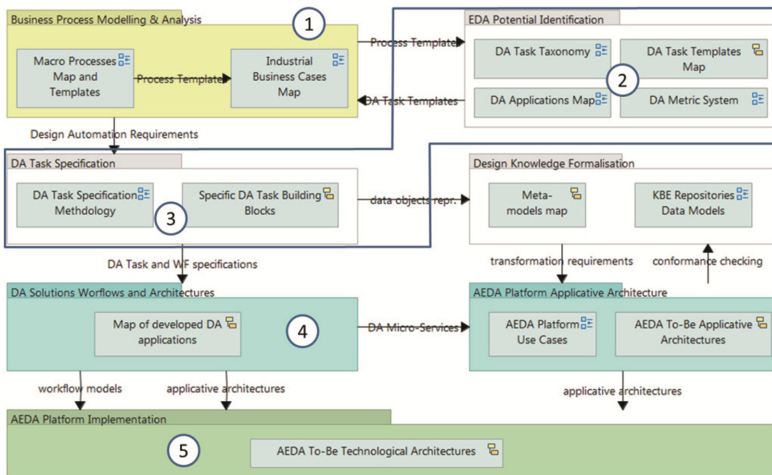

Fig. 1.   Overview of the methodological modeling EA Framework for integrating DA into a global PLM approach – the blue framed area (steps 2 and 3) matches the focus of the paper. (Color figure online)

The first step of the methodology is dedicated to the business process models as well as models supporting the identification of DA potential within these processes. The "business process modeling and analysis" package encompasses a set of business process templates that can be re-used and adapted for modeling industrial business processes and DA business scenarios. The second step of the methodology is the "EDA potential identification" for which the framework provides a taxonomy and a map of DA tasks that have been derived from [4] and positioned according to their domain(s) of application. For each DA task category, the framework also provides a set of DA task templates to be re-used for specifying a DA task within an industrial DA business scenario instantiating and/or combining these templates (third step). Whereas Fig. 2 illustrates the generic design task templates, Fig. 3 illustrates the template of a specific

design task category. Another package "Design Knowledge Formalization" comprises a set of meta-models for formalizing the engineering design knowledge required for performing the automated design task, as well as conceptual data models that intend to be implemented into knowledge-based repositories of DA systems. Finally, the instantiated and orchestrated DA task templates should provide all the information for fully specifying the DA solution workflows and architectures in the applicative layer (step 4) as well as the concrete implementation specifications (step 5). The focus of this paper is on steps 2 and 3, i.e. the specification of DA task building blocks through the re-use of identified DA task patterns and related ArchiMate templates.
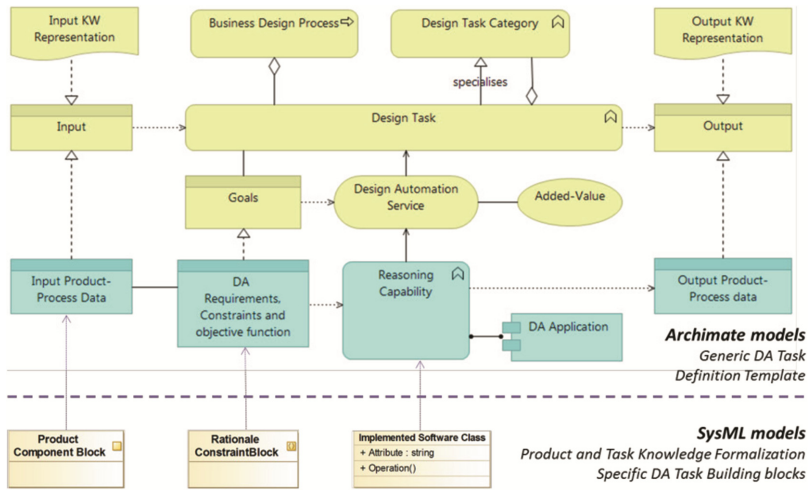

Fig. 2. Generic DA task definition template - Linking concept for integrating EA models in ArchiMate with product related system models in SysML.
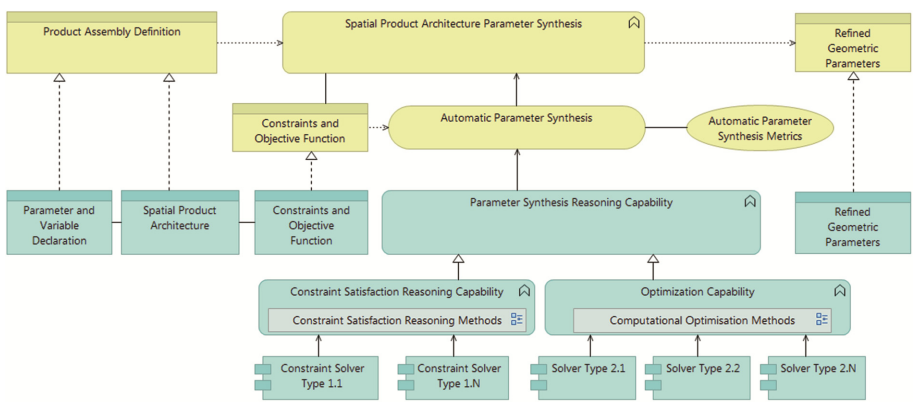

Fig. 3. Template for spatial product architecture parameter synthesis task

Input and output states as well as corresponding product knowledge are determining criteria for defining a design task. Further, the representations as well as problem solving strategy/reasoning technique, i.e. the reasoning capability, (Fig. 2) are key criteria for specifying the DA solution for a given task. Lastly, in analogy to [23], the goal of a task is investigated in order to account for the requirements, constraints and objectives. For reasons of "genericity" of the approach and for enabling re-use of formalized knowledge, the business and applicative elements of this DA task formalization should remain generic for each task category. In contrary, the reasoning capabilities vary according to the specific DA methods. This is illustrated in Fig. 3 that shows the DA task template defined for the design task category "Spatial Product Architecture Parameter Synthesis". The possible variations while instantiating such a template are related to the type of solver chosen for automating the task, for instance optimization methods or constraint solvers as well as the related knowledge representations for input and goals.

Further, Fig. 2 introduces the concept for linking the ArchiMate language with the SysML or UML language permitting to establish dependency and traceability relationships between the two. This work focuses on the specification of DA tasks and on modeling the related input and goals with SysML. Block Definition Diagram (BDD), constraint blocks as well as corresponding Parametric Diagrams (PD) are used for modeling the task knowledge, i.e. necessary equations and the corresponding relations to the product and task design parameters, variables and constraints. The following section introduces the generic methodology for formalizing a DA task in SysML and establishing the dependency and traceability relationships between ArchiMate and SysML models.

3.2 MBSE Methodology for Specifying Design Automation Tasks

Figure 4 shows the generic activity diagram relating the actions and inputs that are required for formally defining a DA task as well links to the specific activities that are implemented due to the distinct characteristics of each category. After DA potential has been successfully identified and the corresponding EA models have been defined, the formal definition of a design task is initiated. First, detailed product knowledge is a prerequisite for design task definition in order to be able to comprehensively describe in-/output states, e.g. product architectures, parameters, variables and relations. Next, task specific SysML profiles serve as a means to further guide and support the task modeling. Finally, instantiated EA task templates support the identification of boundary conditions and corresponding formalizations. The action "Define DA Task", shown in Fig. 4, is modeled using the SysML stereotype "structured action" to indicate the parallel occurrence of the actions "Define Input (Product) Knowledge" and "Define Control Knowledge". Whereas the first refers to definition of product architectures, parameters and relations, the latter defines the goals and requirements to guide the execution of the design automation task. For representation of both product knowledge and goals, BDDs are used for defining structures whereas PARs are used for definition of corresponding relations.
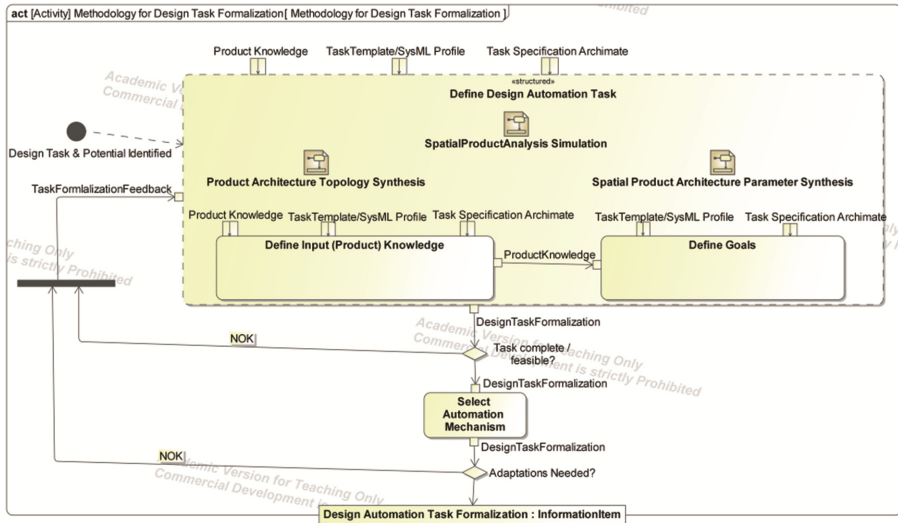
Fig. 4. SysML DA task formalization activity diagram.

After a task has been completely formalized, an appropriate automation mechanism has to be selected and translated from SysML to the corresponding formalization. Consequently, this translation has to be conducted for each DA method specifically.

4   Case Study: Design of Maritime Cranes' Box-Type Booms

The case study addresses the specification of an optimization task and related solution for the design of a box-type boom crane at Liebherr Werk Nenzing (LWN). Figure 5 provides a 3D illustration of the box-type boom, its components and design parameters. The objective is to minimize the costs of the middle section of the boom with respect to material of stiffeners and sheet metals as well as welding of stiffeners and sheet metals. As shown on Fig. 5, the boom is divided into multiple segments, each of which lies between two bulkheads or a bulkhead and the pivot- or end-section and is split into bottom plate, two (symmetric) side plates as well as a top plate. While the length of the boom, and the number and lengths of segments are given as input parameters, the thickness of each sheet metal as well as the number and type of stiffeners remain variables to be determined for each segment during the optimization procedure. Figure 5 shows the corresponding objective function and the complete formalization of the optimization problem, i.e. all constraints and variables. In order to satisfy the requirements stemming from the load case scenarios, the utilization within each segment has to be smaller than one. The utilization calculations are performed within all the plates of each segment with respect to stress, fatigue and buckling. Towards this end, LWN provides an external structural analysis tool.
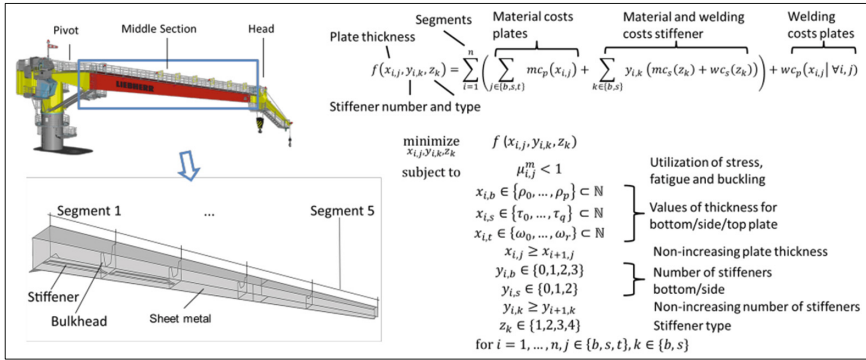
Fig. 5. Illustration of a maritime crane box type boom, cost optimization problem and related formalized objective function

The idea of this case study is to couple the task of spatial product architecture parameter synthesis (i.e. determining above mentioned variables) with the corresponding analysis. This is illustrated in Fig. 6, showing how two pre-defined DA task templates ("Spatial Product Architecture Parameter Synthesis" and "Spatial Product Analysis") are re-used, instantiated and combined to specify the automation of the business task "generate optimized design of the box-type boom".
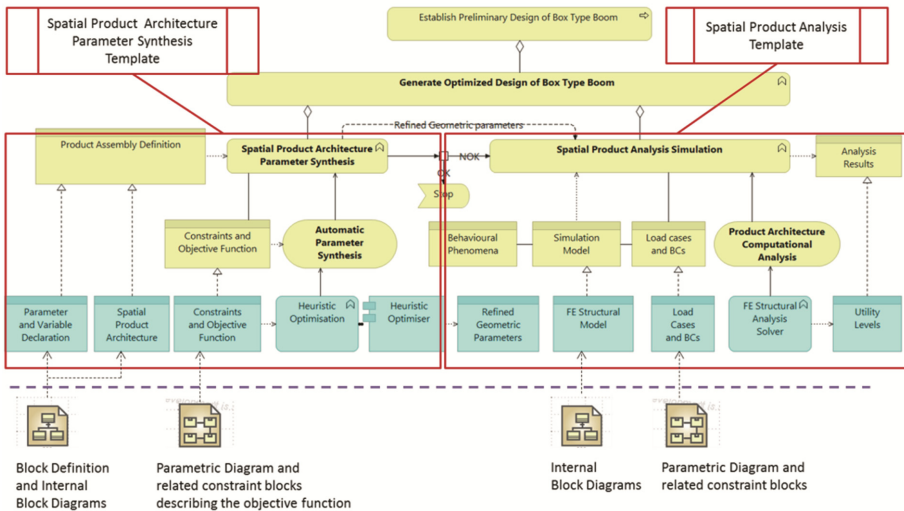

Fig. 6. Re-use and combination of two DA task templates for specifying the automated generation of optimized design of a box-type boom.

Figure 7 shows the BDD of the box-type boom for describing its architecture in terms of sub-components as well as related design parameters and variables. Figure 8 shows the PAR of the cost optimization function as introduced above. The constraint blocks

that are illustrated within the PAR of Fig. 8 symbolize the rules for linking the specific parts and parameters of the text and include both the equations as well as parameters needed for relating the elements. Thus, design parameters, variables and constraints required for performing the cost optimization task and for implementing this objective function are interrelated.
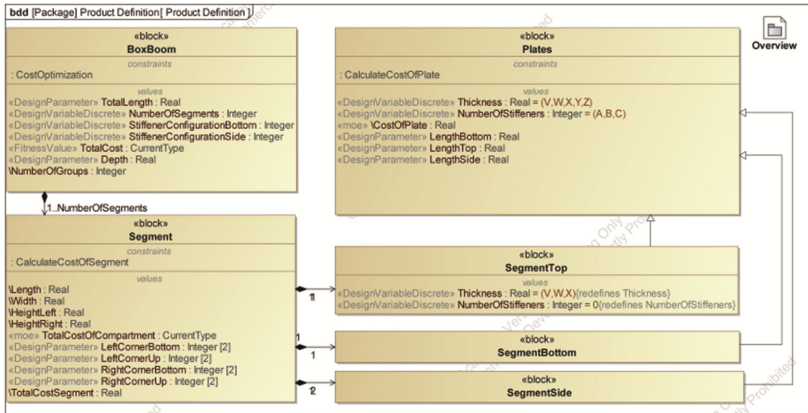

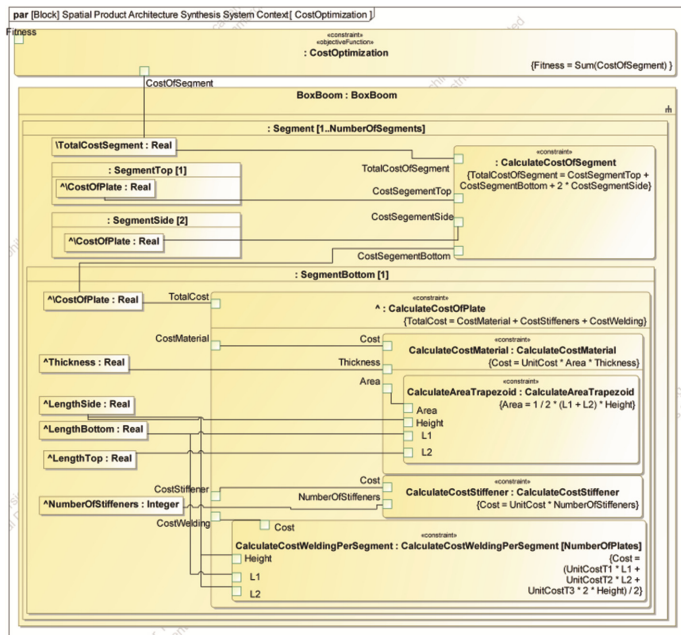**Fig. 7.** SysML block definition diagram of the box-type boom.

Fig. 8. SysML PAR of the cost optimization function and related constraint blocks.

This case study has permitted to demonstrate the applicability of the methodology on which the proposed EA and MBSE methodological framework for DA has been developed. The definition of the DA task building blocks has been performed based on predefined templates and is currently implemented for both heuristic and meta-heuristic solvers. Easy comparison of different solving strategies is hence enabled. For the shown case study, the design task has been formalized with respect to product knowledge describing the input (structure, parameters, variables, relations etc.) as well as the desired output state (constraints and objectives). Despite the usage of model libraries as well as corresponding SysML profiles, the expressivity SysML provides remains a challenge to the modeler and the corresponding interpretation for translation to a computable language.

5   Conclusion and Way Forward

In this paper, a federated EA and MBSE methodological framework for integrating DA into a global PLM approach has been introduced. This framework is built upon a systematic methodology for:

- ensuring the transition of academic methods to industrial practice through a comprehensible and comprehensive DA task categorization that allows practitioners to grasp the opportunities state-of-the-art DA offers;
- supporting the specification of industrial business cases and scenarios through business process modeling and re-use of business process templates;
- supporting the specification of the DA solutions to be developed: for each derived design task category, a DA task template is proposed to be re-used and instantiated in order to derive the building blocks required for the implementation of the appropriate DA method.

A case study addressing the specification of an optimization task and related solution for the design of box-type boom cranes at LWN has been used to demonstrate the applicability of the framework's methodology. Lines of future work comprise the completion of the EA framework with all the DA task templates required for each DA task category, the completion of the MBSE framework with SysML DA task profiles and stereotypes for each of these categories and the development of specific user interfaces to guide the designers for defining the DA task building blocks themselves. Future work should also include the development of interfaces that are restricted to the modeling capabilities needed for a specific DA task, rather than providing the entire expressivity of the SysML language. Further, mechanisms to assess the quality of the task definition need to be developed. Finally, in order to provide maintenance consistency and change propagation mechanisms while linking EA models, SysML models and the various platform-specific DA implementations, standardized linking semantics concepts should be investigated.

References

1. Abramovici, M.: Future trends in product lifecycle management (PLM). In: Krause, F.L. (ed.) The Future of Product Development, pp. 665–674. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-69820-3_64

2. Ehrlenspiel, K., Kiewert, A., Lindemann, U.: Cost-Efficient Design. Springer, Heidelberg, ASME Press, New York (2007)

3. Baxter, D., Gao, J., Case, K., et al.: A framework to integrate design knowledge reuse and requirements management in engineering design. Rob. Comput.-Integr. Manuf. **24**, 585–593 (2008)

4. Rigger, E., Münzer, C., Shea, K.: Estimating the potential of state of the art design automation - tasks, methods, and benefits. In: Proceedings of the DESIGN 2016 (2016)

5. Lund, J.G., Fife, N.L., Jensen, C.G.: PLM-based parametrics for design automation and optimization. Comput.-Aided Des. Appl. **2**, 37–45 (2005)

6. Cederfeldt, M., Elgh, F.: Design automation in SMEs - current state, potential, need and requirements. In: Proceedings ICED 2005, the 15th International Conference on Engineering Design: and Global Economics, p. 1507 (2005)

7. Cagan, J., Campbell, M.I., Finger, S., Tomiyama, T.: A framework for computational design synthesis: model and applications. J. Comput. Inf. Sci. Eng. **5**, 171 (2005). https://doi.org/10.1115/1.2013289

8. Whitman, L., Ramachandran, K., Ketkar, V.: A taxonomy of a living model of the enterprise. In: Proceedings of the 33rd Conference on Winter Simulation, pp. 848–855. IEEE Computer Society (2001)

9. Lankhorst, M.M.: Enterprise architecture modelling—the issue of integration. Adv. Eng. Inform. **18**, 205–216 (2004)

10. Moones, E., Vosgien, T., Kermad, L., Dafaoui, E.M., El Mhamedi, A., Figay, N.: PLM standards modelling for enterprise interoperability: a manufacturing case study for ERP and MES systems integration based on ISA-95. In: van Sinderen, M., Chapurlat, V. (eds.) IWEI 2015. LNBIP, vol. 213, pp. 157–170. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47157-9_14

11. Bernus, P., Nemes, L., Schmidt, G.J.: Handbook on Enterprise Architecture. Springer Science & Business Media, Heidelberg (2012). https://doi.org/10.1007/978-3-540-24744-9

12. Chen, D., Doumeingts, G., Vernadat, F.: Architectures for enterprise integration and interoperability: past, present and future. Comput. Ind. **59**, 647–659 (2008)

13. Moones, E., Figay, N., Vosgien, T., et al.: Towards an extended interoperability systemic approach for dynamic manufacturing networks: role and assessment of PLMStandards. In: Boulanger, F., Krob, D., Morel, G., Roussel, J.C. (eds.) Complex Systems Design and Management, pp. 59–72. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-11617-4_5

14. Dinar, M., Danielescu, A., MacLellan, C., et al.: Problem map: an ontological framework for a computational study of problem formulation in engineering design. J. Comput. Inf. Sci. Eng. **15**, 031007 (2015)

15. Dinar, M., Summers, J.D., Shah, J., Park, Y.-S.: Evaluation of empirical design studies and metrics. In: Cash, P., Stanković, T., Štorga, M. (eds.) Experimental Design Research, pp. 13–39. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33781-4_2
16. Yu, J., Cha, J., Lu, Y.: Design synthesis approach based on process decomposition to design reuse. J. Eng. Des. **23**, 526–543 (2012)
17. Panchal, J.H., Fernández, M.G., Paredis, C.J.J., et al.: A modular decision-centric approach for reusable design processes. Concurrent Eng. Res. Appl. **17**, 5–19 (2009)
18. Ming, Z., Yan, Y., Wang, G., et al.: Ontology-based executable design decision template representation and reuse. In: International Design Engineering Technical Conference and the Computer and Information in Engineering Conference. American Society of Mechanical Engineers (2015)
19. Friedenthal, S., Griego, R., Sampson, M.: INCOSE model based systems engineering (MBSE) initiative. In: INCOSE 2007 Symposium (2007)
20. Peak, R.S., Burkhart, R.M., Friedenthal, S.A., et al.: Simulation-based design using SysML part 1: a parametrics primer. In: INCOSE International Symposium, pp. 1516–1535. Wiley Online Library (2007)
21. Peak, R.S., Burkhart, R.M., Friedenthal, S.A., et al.: Simulation-based design using SysML part 2: celebrating diversity by example. In: INCOSE International Symposium, pp. 1536–1557. Wiley Online Library (2007)
22. The Open Group: ArchiMate 3.0 Specification. The Open Group, Reading, UK (2016)
23. Sim, S.K., Duffy, A.H.B.: Towards an ontology of generic engineering design activities. Res. Eng. Des. **14**, 200–223 (2003)