

# **SYSTEM OF SYSTEMS LIFECYCLE ENGINEERING APPROACH INTEGRATING SMART PRODUCT AND SERVICE ECOSYSTEMS**

**Forte, Sven (1);  
Göbel, Jens Christian (1);  
Dickopf, Thomas (2)**

1: Institute for Virtual Product Engineering (VPE), University of Kaiserslautern;  
2: CONTACT Software GmbH

## **ABSTRACT**

Due to the increasing smartization of products and the ecosystems in which they are typically embedded, holistic considerations of future value-added systems are becoming increasingly important. This also determines fundamentally new challenges for the engineering of the future's smart Products. In order to be able to address the increasingly fuzzy system boundaries associated with this, this contribution introduces a System of Systems Engineering Lifecycle Concept considering smart products and services as core components of connected System of Systems ecosystems. Main characteristics of smart products and system of systems are discussed and in a real existing scenario of a sustainable landfill the presented System of Systems Engineering Lifecycle Concept is used as a System of Systems Framework for the given Use-Case.

**Keywords:** Systems Engineering (SE), System of Systems, System of Systems Engineering Lifecycle, Early design phases, Integrated product development

## **Contact:**

Forte, Sven  
University of Kaiserslautern  
Institute for Virtual Product Engineering (VPE)  
Germany  
forte@mv.uni-kl.de

**Cite this article:** Forte, S., Göbel, J. C., Dickopf, T. (2021) 'System of Systems Lifecycle Engineering Approach Integrating Smart Product and Service Ecosystems', in *Proceedings of the International Conference on Engineering Design (ICED21)*, Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.552

# 1 INTRODUCTION

Due to the ever-progressing smartization of products, including the associated digital product twins (Göbel and Eickhoff, 2020; Stark and Damerou, 2019), new market potentials are emerging, especially in the context of disruptive new business models for cross-product and cross-platform offerings that require the comprehensive information technology interaction of several heterogeneous products and their infrastructure components (Geisberger and Broy, 2015; Abramovici and Herzog, 2016). However, this determines fundamentally new challenges for the engineering of these products and infrastructure components. Due to the high demand for deep integration and interaction of several smart products, the need for standardized communication paths (interfaces) between the products and the associated complexity in engineering increases significantly. Especially with this expected future highly complex and dynamic product ecosystems with too fuzzy system boundaries, which can only be considered to a limited extent by traditional systems engineering approaches, it can be assumed that a decisive role especially will also be played in the civil area system of systems (SoS) in the engineering of these smart (product) ecosystems like smart mobility services including shared-scooters, -bikes, trains, infrastructure and other services (e.g., payment). System of Systems approaches have already been applied successfully in military and spaceflight use cases, for example, in the description of highly complex missions and mission scenarios with a focus on communication (Smith et al., 2011). One central research question is how the future system of systems in the context of smart products and their belonging ecosystems can be engineered to be used reliably and in a systematic way for system validation and improvement purposes considering MBSE concepts. This contribution provides an overview of the SoS engineering research activities focusing on the development, validation, operation, and optimization of future system of systems and the integration of single capsuled systems. The implementation and validation of the presented concept took place on an SoS research testbed at the University Kaiserslautern using a platform technology provided by CONTACT Software.

## 2 STATE OF THE ART

### 2.1 Smart Products as core SoS-Components

Cyber-physical systems with a certain degree of autonomy and the capabilities to communicate most likely over specific networks (e.g., internet) and to interact with different stakeholders (Tomiya et al., 2019; Porter and Heppelmann, 2015; Mühlhäuser, 2008) are discussed as smart products as shown in Figure 1 below. These products often are integrated into heterogeneous smart environments (Mühlhäuser, 2008) to maximize customer value by connecting and integrating various smart products, services and connecting infrastructures with each other. Furthermore, Tomiya et al. (2019) describe characteristics where smart products distinguish themselves from conventional products through different characteristic capabilities. Intelligence is one of the key characteristic of smart products, which enables autonomy and is indispensable when interacting with humans. Connectivity allows smart products to interact through networks and link to other products. The connectivity and intelligence can serve as enablers for new product-related services. Thus service integration reflects another characteristic of smart products and the data-driven ability to make life-cycle-related decisions due to their capability to sense and analyze their environment's data and information. With the development of traditional mechatronic products, there are well-defined system boundaries within which the product is designed and later operates. In the multidisciplinary integration of several such smart products, however, these boundaries merge more and more and go from clearly defined to increased uncertainty, which leads to new challenges in product engineering. However, smart products' interfaces and the new disruptive business models they offer create a decisive added value for companies to bring future innovations to the customer, especially by the integration of different smart products into targeted SoS to enable extended functions and services. However, this would mean that there should be an open exchange of information and open communication between the individual systems to enable these data-driven business models and services. Here the classical cybertronic products will, in the future, more and more come up against their limits. Due to monolithic interfaces, these products usually have an insufficient level of standards for open data exchange. By integrating the smart products, including their System Lifecycle Management (SysLM) approaches (Eigner et al., 2017a), into a model-based SoS engineering lifecycle, the collaboration-enhancing exchange of standards and interfaces for the adequate system integration of a system into future smart SoS will be enabled. In particular the integration of the SysLM of those smart

products are enabling furthermore a seamless data- and information-driven integration as also demand-oriented reconfiguration and comprehensive new data-driven product-related services.

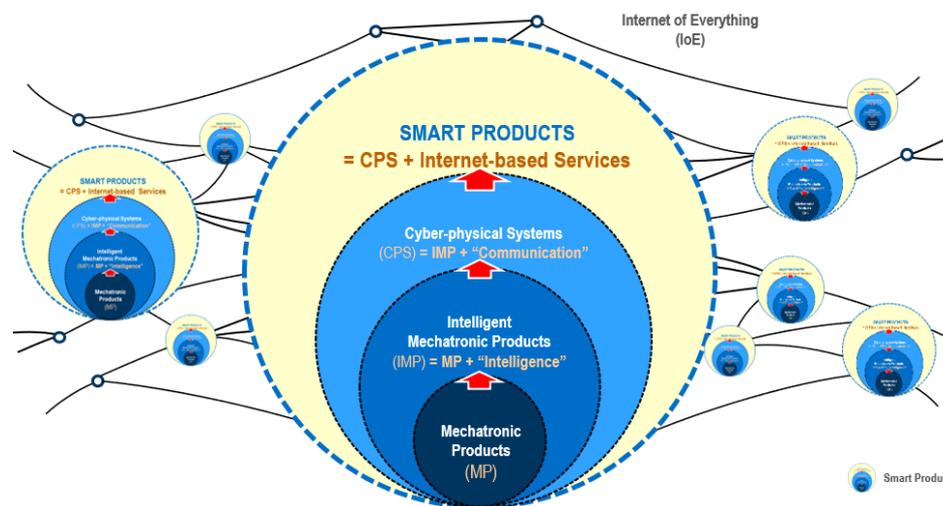


Fig. 1: Connected smart product engineering lifecycles based on Abramovici (2019)

## 2.2 System of Systems (SoS)

Nowadays, systems engineers are face with multi-integrated complex systems and the situation that system boundaries are getting crossed (Dahmann and Henshaw, 2016; Gorod et al., 2008). As a result systems engineers have to deal with SoS instead of single capsuled systems. Of an SoS, there are different existing definitions in the literature. The International Council on Systems Engineering (INCOSE) defined an SoS as “an System of Interest (SoI) whose elements are managerially and/or operationally independent systems. These interoperating and integrated collections of constituent systems usually produce results unachievable by the individual systems alone” (Walden et al. 2015). Maier (1998) postulated five key characteristics to decide if an SoI can be understood as an SoS. The characteristics take into account that the SoS constituent systems are handled like systems in their own right (Dahmann and Henshaw, 2016, Walden et al., 2015). These characteristics are operational independence of the components, managerial independence of component systems, geographical distribution, emergent behavior, and evolutionary development processes.

Based on Maier (1998) and Dahmann & Baldwin (2008), four SoS types could be identified. The different types can be categorized based on their authority relationships between the SoS and the constituent systems (Dahmann and Henshaw, 2016). Directed SoS are centrally managed SoS, where the constituent systems maintain their independence but are subordinated to the central purpose. Acknowledged SoS are SoS where constituent systems are independent, but changes and objectives are based on cooperative agreements. On the SoS level, there are designated managers and resources. Collaborative SoS show no central management. The constituent systems collaborate voluntarily to fulfill the recognized purpose of the SoS. Virtual SoS rely on invisible mechanisms. There is no central management and no agreed purpose that occur the SoS randomly/by accident. As stated in the standard ISO/IEC/IEEE 15288:2015 (ISO, 2015) the architecture for an SoS is a framework for the organization and integration of capabilities of existing systems and new systems into an SoS capability.

The definition of the SoS architecture begins with the de facto architecture. Design decisions can lead to more than one way to realize the different stakeholder concerns and top-level SoS requirements. New requirements for the constituent system and their effects and the architecture constraints are part of the consideration (ISO, 2015). The two most used military frameworks today are the US Department of Defense Architecture Framework (DoDAF) and the UK Ministry of Defense Architecture Framework (MODAF) (Hause et al., 2017). The Object Management Group (OMG) developed the Unified Architecture Framework (UAF) based on DoDAF and MODAF. UAF extends these military frameworks’ scope and delivers a generalized approach for commercial and military architectures. The framework is using a Model-Based Systems Engineering (MBSE) approach to deliver a standard representation for enterprise architecture (OMG, 2020). UAF is based on the standardized OMG Systems Modelling Language (SysML, 2019) and uses different viewpoints and views to describe systems with

different stakeholders (OMG, 2020). Additional SoS engineering approaches are described by Cook and Unewisse (2020) and are therefore not further described in this paper.

### 3 SYSTEM OF SYSTEMS LIFECYCLE ENGINEERING APPROACH

Due to the increasingly fuzzy interfaces in smart Product ecosystems to the external environment, new holistic approaches to future System of Systems and their Lifecycle are becoming more and more important. In particular, the value-added phases e.g. design phases and the Lifecycle Management as an integral approach are in focus. To enable an early system of systems design, verification, and integration of single systems, e.g., smart products and services in more extensive system of systems environments as well as the operation and management of those highly complex and independent systems of systems, new information-driven smart lifecycle management concepts and optimization approaches, as presented in Figure 2 below, are becoming more and more attention intending to cope with the ever-increasing complexity and dynamic in the systems development nowadays. This research, to provide a holistic system of systems concept based on smart products and services as its core elements, is based upon the previous work on the VPESystemDevelopmentMethodology (Dickopf, 2020) as well as the closed-loop systems engineering approaches (Dickopf, 2020; Dickopf et al. 2021). New methods such as agile engineering methods (Dove, R. and Schindel, B. (2019)) are also taken into consideration here for example, in order to support future agile design of interdisciplinary processes for the engineering of e.g. partial models along the SoS Engineering Lifecycle. In this case, the given article focuses on the SoS design, validation, and operation rather than the engineering of integrated smart products, as further described in Göbel and Eickhoff (2020).



Figure 2: System of systems (SoS) engineering lifecycle concept

#### 3.1 SoS Design

In the context of single individual systems in the field of Model-based Systems Engineering (MBSE), there are already general processes and methods that support holistic system analysis and description (ISO, 2015; Eigner et al., 2017b). For this purpose, beginning in the innovation-road mapping phases of smart systems, mostly stakeholder requirements are described, and based on these, functions are derived which the system should fulfill. However, explicit operational aspects in the context of the utilization-ecosystem are mostly not included in the necessary detail level (Bleisinger et al., 2020) to create a shared understanding of the stakeholders in the involved value networks. By strategic and operation nature, these aspects are represented as views in enterprise architecture frameworks and are thus incorporated into the enterprise architecture. As described in chapter 2.3, there are several frameworks, especially for describing operational architectures for military missions, including an essential function of capability management and its analysis in this context. They aim to create a uniform understanding of interfaces

and principles and ensure that all involved parties' subsystems are interoperable. Using the UAF Framework as an example, strategic- and operational-views are integrated into the overall architecture and represent a suitable possibility for the holistic description of a solution. This concept only focusses parts of the UAF framework's operational and structural views to describe the given SoS Scenario Use-Case (chapter 4.1). Based on the described capabilities of the SoS, the basic abstract SoS architecture can be developed to derive first interfaces and functions as well as requirements for solution systems. On the one hand, this enables subsequent solution systems to secure their functions against the SoS at an early stage to create the best possible integration in the intended context of use. On the other hand, the first description of relevant test cases in the SoS to secure the SoS and to maximize its functionality will be developed. However, this also requires holistic methodological approaches to developing the SoS itself and developing the solution spaces. For the description of the solution spaces i.e. smart product systems, the VPESystemDevelopmentMethodology was developed and subsequently validated in practice-use of an example of an excavator by Dickopf (2020). There are already requirements for a cross-disciplinary method for modeling smart products embedded in an SoS, which are currently being further developed and are not discussed in this paper. In this context, Cook and Pratt (2014) also describe several success factors for SoS engineering methodologies based on Department of Defence (2008), Cook et al. (2013) and Adams and Keating (2011) as shown in Figure 3 below.

Methodology	Principles/Success Factor	Methodology	Principles/Success Factor
Coverage (scope, scale depth)	Maintain understanding of the nature of constituent systems	Type	Focus on design strategy & trades
	Include SoS capability evaluation		SoSE ≠ TSE
	Holistic process – not just technical		Architecture-informed – not driven
	Support different SoS applications	Inputs, intent or concept	Balance of top-down & bottom-up analysis
	Continuity of design & trade analysis		SoS must be recognized
	Continuous characterization of SoS		Minimal set of principles
Characteristics (what it is "like")	Adaptable & flexible		Practice-based discipline
	Multi-methodological		Metrics to assess quality of inputs
	Tailored & tailorable		Long-term roadmap of SoS
	Outcome spaces – multi view & scale	First iteration is pragmatic / investigatory	
	Incremental & evolutionary	Gain constituent system involvement	
	Transportable (multi-use)	Guide action	
	Grounded in theory/philosophy	Needs management & leadership expertise	
	Provide consistency & replicability	Methods within	Use of "glueware"
Account for & minimize external influences	End-to-end analysis essential		
	Open systems & loose coupling		
		Discovery engineering	
		Development environment	

Figure 3: System of systems methodologies success factors (Cook and Pratt, 2014)

Adams and Keating (2011) identified nine attributes needed for an SoS engineering methodology to be sustainable. These factors are transportable, theoretical and philosophical grounding, a guide to action, significance, consistency, adaptability, neutrality, multiple utilities, and rigorous. Department of Defence (2008) identified five emerging principles for SoS systems engineering to implement the core elements mentioned in the guide with success. These five emerging principles are addressing organizational as well as technical issues in making SE trades and decisions, acknowledging the different roles of systems engineers at the system versus the SoS level and the relationship between the SE done at the two levels, conducting balanced technical management of the SoS, using an architecture based on open systems and loose coupling, focusing on the design strategy and trades both when the formal SoS is first established and throughout the SoS evolution.

### 3.2 SoS Validation and Integration

Validation in the context of the systems engineering discipline is commonly understood to verify the system's design and ensure the stakeholders' requirements are satisfied in the highest possible way (ISO, 2015), which is already a significant challenge at single systems itself. Furthermore, in highly complex and dynamic SoS, it is usually impossible to map this to specific individual system components. Depending on the characteristics of the SoS, uncertainty can always be observed in the validation of the entire system for many parts of the SoS. Proven concepts such as the closed-loop systems engineering "model-in-the-loop" described in Dickopf et al. (2019) should be extended for future SoS concepts to keep these uncertainties manageable. This extension will enable a comprehensive validation of the individual capsuled systems and a holistic, single-system-integrated validation of the SoS. Furthermore, it also includes the tracing of the requirements of a SoS with regard to the interfaces of the system components and the communication networks and the value chain involved in the execution of the SoS.

### 3.3 SoS Operation and Reconfiguration

As a core of operational data, IoT-Platforms are playing an increasingly important role in the aggregation and visualization of usage data. Due to the high connectivity of smart products and services, primarily data-driven services, more and more operational data becomes interesting for further use beyond the development and provision of new features in the context of engineering. By efficiently networking the digital twins of the individual systems represented in the SoS not only new services could be developed and monitored in the context of the SoS, but also a detailed twin of the SoS can be developed in any level of abstraction, based on the individual systems if required. In the context of SoS engineering and operation, the operator of an SoS usually focuses on monitoring the individual SoS Key-Performance-Indicators (KPI). Here, a efficient management of the SoS can be supported by a strong integration of the product usage data, e.g., in Internet of Things (IoT) platforms. If the engineering data from the development of the SoS and the individual systems are also available in the corresponding Systems Lifecycle Management (SysLM) approaches, even deeper integration of product usage data and engineering data can be used. E.g. an integrated SoS service development, which enables not only new innovative functions and features for the products but also completely new business models along the value chain within the SoS ecosystem. As well a high availability of the SoS should be guaranteed to minimize the failures of subsystems and comprehensive SoS features. Methods of predictive maintenance can achieve this. Analyzing different existing approaches in this research, we build on an approach of information-driven condition monitoring of the single systems, which generates a superior digital SoS twin based on the single systems' digital twins and contains all SoS operation relevant information and states of the single systems. The behavior of individual systems can be tracked through the continuous integration and provision of usage data with context-relevant methods and, in the event of irregularities, draw attention to the need for intervention by the SoS operator.

Optimization as a concept of data-driven system improvement using usage information back in the development phase for individual products or product populations is often optimized through planning strategies (Albers et al., 2015), bringing the system architecture close to the digital twins. Dickopf et al. (2019) present a Closed-loop Systems Engineering approach (CLE) to improve and optimize a system step by step, reconfigure a running system, or bring the system under development to the application faster. Paquin (2014) has identified three essential strategic features with which successful companies can successfully minimize complexity. These are simulation, virtual prototyping, and improvement loops. The improvement loops are crucial here, as continuous data and information feedback from the usage/simulation back into the development allows a recalibration and thus improvement or optimization of the system. This also applies to the system environment. Here, context-related usage data can be used to optimize the system as such and the SoS itself, in which the individual system is embedded. In the SoS context, based on the given CLE approach for smart products, a further improvement loop was introduced, SoS-in-the-loop.

SoS-in-the-loop addresses the improvement concept based on the context-related usage data of the participating individual systems and the resulting overall analysis of the SoS-based functions and features (capabilities, strategic requirements etc.), especially for developing SoS architecture and behavior. Thus, by purposeful integration of the context-referred use data of the existing single systems, the SoS being in the reconfiguration can be reconfigured adequately. Thus the operational sequences within the SoS can be arranged more robustly, which affects mainly the efficiency in the operation and the general availability of the SoS.

### 3.4 SoS End of Lifecycle

As SoS are mostly long-term projects and, due to their characteristics, often do not have a concretely defined end of their lifecycle, the end of the lifecycle is not only reached with the end of an SoS project in the context of a conventional cooperation project. Instead, aspects such as characteristic changes of the SoS are an indicator for a possible change into a new life cycle and the case when a large part of the systems is separated from the system network and thus forces a reconfiguration. Furthermore, an SoS can also fundamentally change its context of use and thus its target focus, both passively and actively, by reconfiguring a large part of the active individual systems. The individual systems primarily drive a passive change and a possible operator of the SoS can only have a limited influence on it. As smart products can in addition could have different discipline-specific lifecycles due to their interdisciplinarity. An active change is again primarily driven by the SoS itself and reconfigures the SoS according to a

change, e.g., the model-in-the-loop concept and the provision of temporary SoS-based services, which can be directly related to a single participating system but do not necessarily have to be. However, this is mostly always accompanied by a change of the characteristics as described above. A further focus, in the Phase SoS End of Lifecycle is a direct communication into the respective roadmapping and early development phases of future Systems and System of Systems in order to validate these as early as possible (e.g., by simulation of the single system against the SoS or also general collaboration within the value chain for the exchange of requirements).

## 4 EXEMPLARY SOS ENGINEERING USE-CASE

### 4.1 Use-Case Scenario

Based on a real existing scenario of a sustainable landfill on outskirts in Kaiserslautern, Germany, the application scenario describes a directed system of systems according to Maier (1989), which can be found on construction sites, in mines and quarries, and the mentioned landfills. The overall system itself consists of individual systems that can be subdivided into subsystems depending on their level of granularity. Essential systems in all of the applications, as mentioned above, are the motor vehicles that do the actual work. Depending on their level of innovation, they can either communicate autonomously in the sense of smart cyber-physical systems or are operated with renewable energies in the sense of an ecological footprint. In the sense of an ecological footprint, a wind farm has been erected on the site, consisting of several wind turbines. The energy generated is made available to the other SoS components and fed into the urban energy supply. In the SoS itself, the energy is provided to the vehicles via charging stations, whereby an additional system is added to the SoS. The application scenario described here can also be regarded as a subsystem of an even larger SoS by feeding energy into the urban energy network. The vehicles in the application scenario do not only have to belong to the operator. For this reason, the SoS also includes a weighbridge. It allows the service provided to be remunerated and the service required to be invoiced. Thus the application scenario contains a systemic structure based on the individual systems and several business models. Furthermore the dynamic demand-oriented reconfiguration of the smart products based on changing requirements in the SoS-Context as one specialty. Since the application scenarios mentioned earlier are open terrain scenarios, both the prevailing weather and the environment, i.e. the terrain itself, can bring along potential dangers. For this reason, a security system monitors the entire SoS, which uses sensors to detect external influences, but also has an entrance and exit control system to keep track of who is on the premises. Figure 4 shows both the real existing scenario of the landfill on the outskirts of Kaiserslautern and a pictorial representation of the described scenario, which forms a basis for this research and is implemented in a continuously expandable demonstrator.



Figure 4: Real (left) and pictorial (right) representation of the SoS use-case scenario

### 4.2 Use-Case specific SoS Lifecycle Engineering

The use-case scenario as described above, have been designed under consideration of the System of System Lifecycle Engineering concept shown in Figure 2 and is described in the following according to its Lifecycle Phases. During the design phase of the SoS (as described in chapter 3.1), the UAF

framework was used to design the given SoS scenario from chapter 4.1. In Figure 5, the A section shows examples of the UAF-Views of the Model: Capability (upper left), Operational Taxonomy (lower left),

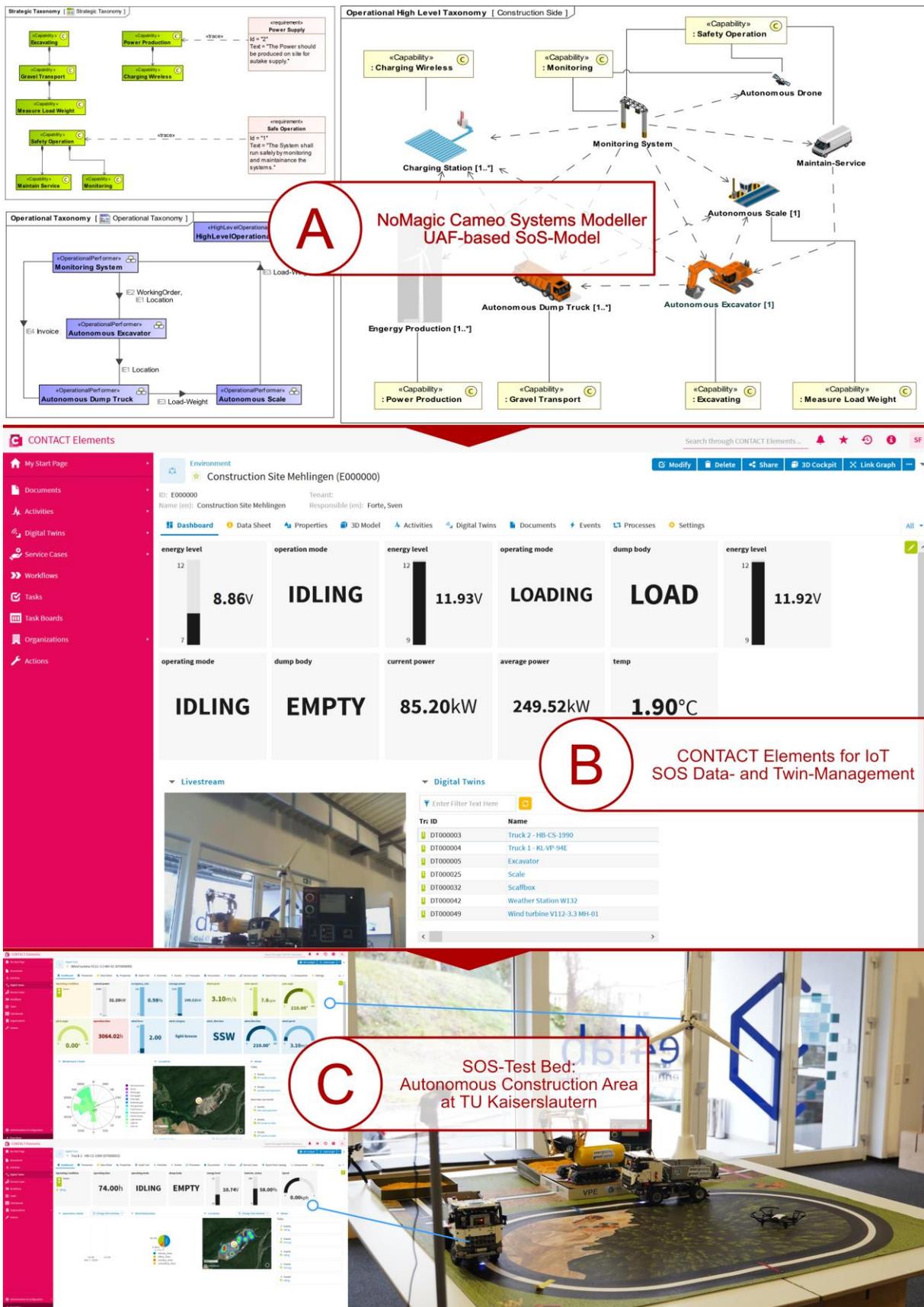


Figure 5: Exemplary SoS use-case implementation architecture

and High-Operational-Architecture (right). The model has been designed with No Magic Cameo Systems Modeller (NoMagic, 2020) and the UAF-Plugin. The partial systems have been modelled and integrated into the SoS according to the VPESystemDevelopmentMethodology and were validated against the SoS using the MQTT-based IoT connection for operational data-exchange of the given digital twins. A section of the SoS twin is exemplarily illustrated with the Elements for IoT platform by CONTACT Software in Figure 5 Section B. Finally, the SoS testbed of the autonomous construction site with the partial systems excavator, dump truck 1, dump truck 2, scale system, autonomous drone, security system, wireless charging system, and wind turbines to supply the charging infrastructure of the vehicles in Section C on the right. A dashboard of the wind turbine with its KPI and telemetry data is shown on the upper left. On the lower left the dashboard of the operational data of the scale system.

## 5 OUTLOOK

The contribution presents a model-based system of systems lifecycle engineering approach with smart products as its core components, enabling an early SoS design and an early verification and integration of the smart products into the SoS environment to enable cross functionalities and services. The various phases of the System of Systems Lifecycle Engineering Approach: (1) SoS Design, (2) SoS Validation and Integration, (3) SoS Operation and Reconfiguration, and (4) SoS End of Lifecycle were described therefore in chapter 3. The smart product engineering was explicitly not considered in detail in this paper, concerning the corresponding literature. The system of systems lifecycle engineering approach was put into practice after its discussion in chapter 3 by using an exemplary use case described in chapter 4 and demonstrated in detail by several selected examples from modelling and implementation in Figure 5. Further research on the system of systems lifecycle engineering approach is needed, particularly in the detailed consideration of the required information technology infrastructure components of the SoS and the participating systems and possibilities for the temporary and dynamic reconfiguration of the SoS, especially with a focus on smart product-based services. For this purpose, the presented testbed will be successively extended by the discussed aspects in order to be able to represent a holistic System of Systems Lifecycle Engineering and Management, using the example of the given-use-case.

## REFERENCES

- Abramovici, M. (2019), "Smart Products", in Chatti, S., Laperrière, L., Reinhart, G. and Tolio, T. (Eds.), CIRP Encyclopedia of Production Engineering, Springer Berlin Heidelberg, Berlin Heidelberg, pp. 1574–1578. [http://doi.org/10.1007/978-3-662-53120-4\\_16785](http://doi.org/10.1007/978-3-662-53120-4_16785)
- Abramovici, M. and Herzog, O. (Eds.) (2016), Engineering im Umfeld von Industrie 4.0: Einschätzungen und Handlungsbedarf, acatech Studie, Herbert Utz Verlag, München.
- Adams, K.M. and Keating, C.B. (2011), "Overview of the systems of systems engineering methodology", International Journal of System of Systems Engineering, Vol. 2 No. 2/3, p. 112-119. <http://doi.org/10.1504/IJSSE.2011.040549>
- Albers, A., Bursac, N. and Wintergerst, E. (2015), "Produktgenerationsentwicklung - Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive", *Stuttgarter Symposium für Produktentwicklung (SSP) 2015*, Stuttgart, June 19, 2015, Fraunhofer IAO, Stuttgart.
- Bleisinger, O., Forte, S., Apostolov, C. and Schmitt, M. (2020), "Applying Model-based Systems Engineering to the Development of autonomous Vessel Functions", Proceedings of the Design Society: DESIGN Conference, online, May, Cambridge University Press, online, pp. 2455–2464. <http://doi.org/10.1017/dsd.2020.170>
- Cook, S.C., Nowakowski and M. Unewisse (2013), "Principles for Designing System of Systems Engineering Approaches for the Australian Defence Force", Proceedings of SETE 2013, Canberra, Australia, April 30 - May 1.
- Cook, S.C. and Pratt, J.M. (2014), "Towards designing innovative SoSE approaches for the Australian defence force", International Conference on System of Systems Engineering (SOSE), 9-13 June, 2014, Adelaide, Australia, IEEE, Piscataway, pp. 295–300. <http://doi.org/10.1109/SYSOSE.2014.6892504>
- Cook, S.C. and Unewisse, M.H. (2020), "A Capability Engineering Lifecycle Framework Based on Insights from Australian Defence", INCOSE International Symposium, Vol. 30 No. 1, pp. 886–901. <http://doi.org/10.1002/j.2334-5837.2020.00761.x>
- Dahmann, J.S. and Baldwin, K.J. (2008), "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering", 2nd Annual IEEE Systems Conference, April 07-10, 2008, Montreal, IEEE, pp. 1–7. <http://doi.org/10.1109/SYSTEMS.2008.4518994>
- Dahmann, J.S. and Henshaw, M. (2016), "Introduction to Systems of Systems Engineering", INSIGHT, Vol. 19 No. 3, pp. 12–16. <http://doi.org/10.1002/inst.12100>

- Department of Defense (2008), *Systems Engineering Guide for System of Systems - Version 1.0*. [online] Office of the Deputy Under Secretary of Defense for Acquisition and Technology. Available at: <http://acqnotes.com/wp-content/uploads/2014/09/DoD-Systems-Engineering-Guide-for-Systems-of-Systems-Aug-2008.pdf> (accessed November 5 2020)
- Dickopf, T., Apostolov, H., Müller, P., Göbel, J.C. and Forte, S. (2019), “A Holistic System Lifecycle Engineering Approach – Closing the Loop between System Architecture and Digital Twins”, *Procedia CIRP*, Póvoa de Varzim, Portugal, May 8-10, 2019, Elsevier Procedia, pp. 538–544. <http://doi.org/10.1016/j.procir.2019.04.257>
- Dickopf, T. (2020), *A holistic Methodology for the Development of Cybertronic Systems in the Context of the Internet of Things*, Dissertation, Technische Universität Kaiserslautern.
- Dickopf T., Forte S., Apostolov C. and Göbel J.C. (2021), *Closed-Loop Systems Engineering – How MBSE and IoT together support system improvement!*, *CSD&M Asia 2020, 2021* (to be published)
- Dove, R. and Schindel, B. (2019), “Agile Systems Engineering Life Cycle Model for Mixed Discipline Engineering”, *INCOSE International Symposium*, Vol. 29 No. 1, pp. 86–104. <https://doi.org/10.1002/j.2334-5837.2019.00591.x>
- Eigner, M., Dickopf, T. and Apostolov, H. (2017a), “System Lifecycle Management - An Approach for Developing Cybertronic Systems in Consideration of Sustainability Aspects”, *Procedia CIRP - The 24th CIRP Conference on Life Cycle Engineering*, Kamakura, Japan, March 8-10, 2017, Elsevier Procedia, pp. 128–133. <http://doi.org/10.1016/j.procir.2016.11.210>
- Eigner, M., Koch, W. and Muggeo, C. (2017b), *Modellbasierter Entwicklungsprozess cybertronischer Systeme: Der PLM-unterstützte Referenzentwicklungsprozess für Produkte und Produktionssysteme*, Springer Vieweg, Berlin. <http://doi.org/10.1007/978-3-662-55124-0>
- Geisberger, E. and Broy, M. (2015), *Living in a networked world: Integrated research agenda Cyber-Physical Systems (agendaCPS)*, Herbert Utz Verlag, München.
- Göbel, J.C. and Eickhoff, T. (2020), “Konzeption von Digitalen Zwillingen smarterer Produkte”, *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Vol. 115 No. special, pp. 74–77. <http://doi.org/10.3139/104.112301>
- Gorod, A., Sauser, B. and Boardman, J. (2008), “System-of-Systems Engineering Management: A Review of Modern History and a Path Forward”, *IEEE Systems Journal*, Vol. 2 No. 4, pp. 484–499. <http://doi.org/10.1109/JSYST.2008.2007163>
- Hause, M., Bleakley, G. and Morkevicius, A. (2017), “Technology update on the Unified Architecture Framework (UAF)”, *Insight*, Vol. 20 No. 2, pp. 71–78. <http://doi.org/10.1002/inst.12153>
- ISO (2015), *ISO/IEC/IEEE International Standard, Systems and software engineering - System life cycle processes*, 15288:2015, pp.1-118, May 15, 2015. <https://dx.doi.org/10.1109/IEEESTD.2015.7106435>
- Maier, M.W. (1998), “Architecting principles for systems-of-systems”, *Systems Engineering*, Vol. 1 No. 4, pp. 267–284. [http://doi.org/10.1002/\(SICI\)1520-6858\(1998\)1:4<267::AID-SYS3>3.0.CO;2-D](http://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D)
- Mühlhäuser, M. (2008), “Smart Products: An Introduction”, in Mühlhäuser, M., Ferscha, A. and Aitenbichler, E. (Eds.), *Constructing Ambient Intelligence, Communications in Computer and Information Science*, Vol. 11, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 158–164. [http://doi.org/10.1007/978-3-540-85379-4\\_20](http://doi.org/10.1007/978-3-540-85379-4_20)
- NoMagic (2020), *Camo Systems Modeller*. [online] No Magic Inc.. Available at: <https://www.nomagic.com/products/cameo-systems-modeler> (accessed November 5 2020)
- OMG (2020), *Unified Architecture Framework (UAF) Domain Metamodel, Version 1.1*. [online] OMG. Available at: <https://www.omg.org/spec/UAF/1.1/DMM/PDF> (accessed November 7th 2020).
- Paquin R. (2014), *The Systems Engineering Closed Loop Process: The Key for Validation*, Aberdeen Group, Waltham.
- Porter, M.E. and Heppelmann, J.E. (2015), *How Smart, Connected Products Are Transforming Companies*. [online] Harvard Business Review. Available at: <https://hbr.org/2015/10/how-smart-connected-products-are-transforming-companies> (accessed November 9 2020).
- Smith, J., Harikumar, J. and Ruth, B. (2011), *An Army-Centric System of Systems Analysis (SoSA) Definition*, ARL-TR-5446, Army Research Laboratory.
- Stark R and Damerou T. (2019), “Digital Twin”, In: Chatti S, Laperrière L, Reinhart G, Tolio T (Eds.), *The International Academy for Production Engineering, CIRP Encyclopedia of Production Engineering*, Springer-Verlag, Berlin, Heidelberg, 2019, [https://doi.org/10.1007/978-3-642-35950-7\\_16870-1](https://doi.org/10.1007/978-3-642-35950-7_16870-1)
- SysML (2019), *OMG Systems Modeling Language Version 1.6* [online]. OMG. Available at: <http://www.omg.org/spec/SysML/About-SysML/> (accessed November 7 2020)
- Tomiya, T., Lutters, E., Stark, R. and Abramovici, M. (2019), “Development capabilities for smart products”, *CIRP Annals*, Vol. 68 No. 2, pp. 727–750. <http://doi.org/10.1016/j.cirp.2019.05.010>
- Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D. and Shortell, T.M. (2015), *Systems engineering handbook: A guide for system life cycle processes and activities*, Wiley, Hoboken NJ.