Towards a User-Centric Systems Engineering Approach

Markus M. Peter
Elias Bader
Christian Neureiter
Josef Ressel Centre for Dependable System-of-Systems Engineering
Salzburg University of Applied Sciences
Salzburg, Austria
e-mail: {first name}.{last name}@fh-salzburg.ac.at

***Abstract*—In the face of increasing complexity in product development, the adoption of systems engineering, particularly model-based systems engineering, has become essential. However, incorporating systems engineering in companies encounters resistance due to concerns about additional expenses among others. While concepts like domain-specific systems engineering aims to enhance user acceptance by tailoring domain-specific languages, we argue for a more comprehensive approach. This research proposes a shift toward user-centric systems engineering, establishing the groundwork by defining the user's role and outlining means of support. The key aspects identified for a such an approach include facilitating tool interoperability to allow users to leverage familiar tools, reducing complexity in tools and modeling languages, and adopting a lean thought model to overcome barriers associated with formal systems engineering. User-centric systems engineering aims to involve the user throughout the entire systems development lifecycle, ensuring their needs and perspectives are integrated into every phase—from initial requirements to system deployment and maintenance. By focusing on the user's role and emphasizing interoperability and simplicity, this research aims to enhance the acceptance of systems engineering and provide a basis for future studies, ultimately improving system development outcomes across industries.**

Keywords-model-based systems engineering; domain-specific systems engineering; user-centric systems engineering

I. Introduction

In the contemporary landscape, characterized by digitalization, global competition, and sustainability, the complexity of products and product development is increasing. This transformation, coupled with a growing trend towards service-enabled systems, results in a new realm of complexities for companies. To effectively tackle this complexity, both at the product and organizational levels, a structured methodology is needed. In that regard, the adoption of Systems Engineering proves invaluable. It is also worth mentioning that model-based systems engineering (MBSE) has proven to be especially suitable for addressing complexity. MBSE is centered around evolving a system model, compromising system specification, design, validation, and configuration management leading to a "single point of truth" making it easier to maintain consistency and assure traceability [1]. However, the introduction of systems engineering and especially MBSE in companies seems challenging. While there are various reasons why the introduction can be challenging, research like in [2], [3] and [4] underscores the hurdles associated with a lack of acceptance and motivation, both at the managerial level and among specialized personnel. In [4], it is further stressed that the willingness of all stakeholders is required for a successful implementation. This willingness can be linked to certain hurdles making the introduction of systems engineering unattractive for current employees. These hurdles are for example the lack of an amortization concept for increased modeling work, and too complex tools with a lacking integration into existing IT infrastructure [2]. This is also reaffirmed by [3] stating that a systems engineering tools are not generally applied in a wider context to support a collaborative environment integrating different technical domains. However, there are already methodologys adressing user acceptance through exploiting particular application domains as common ground. So called domain-specific systems engineering (DSSE) is an approach within systems engineering that focuses on tailoring engineering methodologies, practices, and tools to specific domains or industries. Exemplary domains where research towards DSSE has already taken place are the Smart Grid domain [5] or industrial automation systems [6]. The research on this topic is ongoing for years and accordingly well-founded. In [7], a review on DSSE is given including a summary of existing work as well as reviewing the general approach of DSSE. And while it is stated to increase the practical applicability of MBSE they conclude that DSSE cannot be seen as a final solution. They stress that early implementations fell short due to a poor user acceptance, stressing that a user-centric perspective is needed. So, while this approach points in the right direction it does not go far enough. Instead, further evolution is necessary to turn the domain-specific into a user-centric systems engineering (UCSE) approach that better addresses the individual needs of the participating stakeholders. Our systems engineering maturity model depicted in Figure 1 Illustrates this intended evolution. A further description of the individual maturity levels is given in Table 1. Concluding in this short paper, we argue that the lack of acceptance both in the management as well as on the user side of systems engineering concepts lead to the need of a user-centric approach to systems engineering. We opt to address the challenges by specifically looking at the needs and hurdles of the individual user or user groups applying systems engineering concepts. The remainder of this paper is structured as follows: Section II explores the role of the user in systems engineering. Section III outlines key approaches to

TABLE I. MATURITY LEVELS AND SCOPE IN SYSTEMS ENGINEERING

Maturity Level	Scope and Focus
Level 0	Interdisciplinary engineering focuses on aligning individual disciplines. Engineers work with their corresponding design and development tools.
Level 1	In classic Systems Engineering / MBSE, the whole system comes into focus, and an alignment between technical and business aspects is sought. model-based systems engineering tries to align different models and integrate them using object-oriented modeling concepts, often based on general-purpose modeling Languages such as SysML. Modeling is done mainly by architects.
Level 2	Domain-specific systems engineering strives to establish a common understanding among all stakeholders. Technical modeling concepts are hidden behind domain-specific modeling concepts. Model manipulation is mainly done by architects. The intention of established models is (1) to be understood by all stakeholders and (2) to be compatible and interoperable with models from complementary domains.
Level 3	User-centric systems engineering strives for the active participation of different stakeholders in the modeling process. It acknowledges the individual needs of stakeholders and offers model access for various tools with a different focus. Further, it acknowledges the different needs according to the development logic realized.

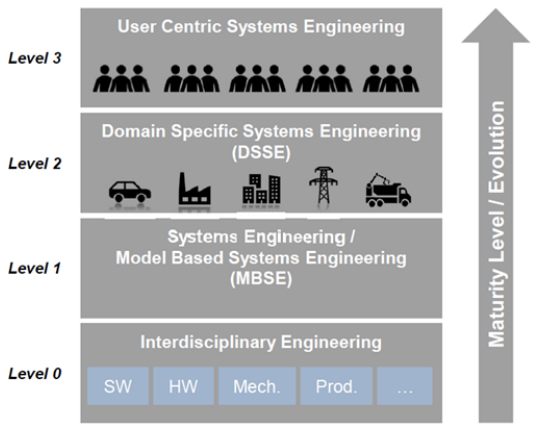

Figure 1. Systems Engineering maturity levels.

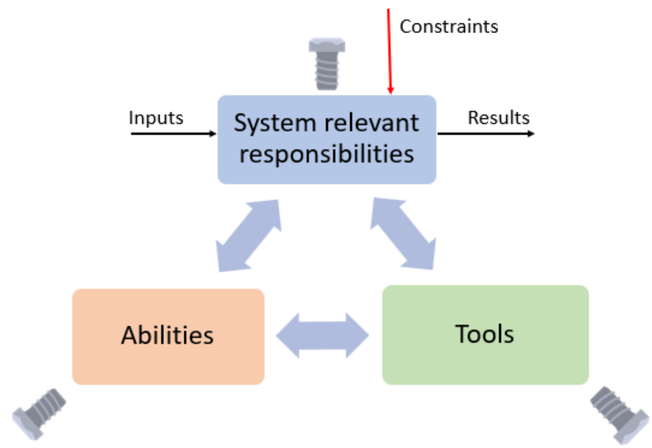

Figure 2. User definition.

support the user, focusing on reducing complexity, enhancing tool interoperability, and adopting a lean thought model. Finally, Section IV presents conclusions and outlines future research directions for advancing UCSE in both theory and practice.

II. UNDERSTANDING THE USER ROLE IN SYSTEMS ENGINEERING

In undertaking a research inquiry into a user-centric systems engineering approach, the foundational step involves a definition of the term user within this specific context. A most rudimentary definition can be formed asking what the tasks of a user are and how he fulfills them. Considering this, the characterization of a user unfolds through a combination of their responsibilities, their abilities and the tools needed to fulfill the responsibilities. Where the responsibilities can be defined as a set of tasks intertwined with specific constraints—whether these could be temporal considerations, the mandatory utilization of designated tools or similar. The ideal supported user has the ability to perfectly use a tool to complete a responsibility in its entirety. This definition leads to Figure 2 depicting a basic user with responsibilities, tasks, and abilities. These three characteristics can be thought of as adjusting screws working together collectively shaping the degree of support provided to a user. If any one of these screws is not properly adjusted, it can lead to problems or dissatisfaction for the user. For instance, the absence of a specific ability can result in responsibilities being inadequately fulfilled or not met at all. Addressing this scenario involves strategic options such as enhancing the user's ability through training, reducing their task load to allocate more resources, or deploying tools that alleviate task complexity. However, it is essential to consider both individual responsibilities and the interconnectedness of responsibilities among users. Poor information flow or unclear responsibilities can create unnecessary complications. Ensuring a well-organized user structure involves managing these interdependencies. For instance, using diverse tools may require frequent information conversion, whereas consistent tool usage reduces this burden. Therefore, supporting users in the systems engineering process requires focusing on their responsibilities, abilities, and tools, as well as the overall working structure.

III. How to support the user in Systems Engineering

This section lays the groundwork for advancing research in the direction of a user-centric systems engineering approach. As previously outlined, we have identified three focal points for supporting the user: their abilities, tools, and conceptual support through an exploration of their responsibilities and interconnections with other users. Subsequent sections will provide a forward-looking perspective on the anticipated research for each of these three aspects. This analysis is informed by both the drawbacks observed in DSSE and insights gathered from interviews with company partners implementing systems engineering in specific projects.

A. Tools: Facilitate tool interoperability

As outlined in the introduction a huge hurdle for the introduction of systems engineering is the significant number of different tools and the missing acceptance to learn new tools. As for modeling systems the de facto standard is OMG's System Modeling Language (SysML). And despite the broad acceptance, different limitations exist, such as poor interoperability between various tools or a lack of precision. To address these shortcomings, in 2017, the OMG issued a Request for Proposal (RFP) for the specification of SysML v2. As stated on the OMG homepage, "the emphasis of SysML v2 is to improve the precision, expressiveness, interoperability and the consistency and integration of the language concepts relative to SysML V1.x. [...] the language will be specified as both a SysML profile of UML and as a SysML metamodel" [8]. In parallel to the SysML v2 RFP, the SysML v2 API and Services RFP focused on services to operate on SysML v2 models and connect SysML v2 models with models in other disciplines. As stated by OMG, this API shall be "implemented by SysML v2 modeling environments and shall support a wide range of operations related to the model query, construction, view/viewpoint management, analysis, management and transformation for SysML v2 models" [9]. According to the OMG roadmap, the final specification of SysML v2 will be submitted in the first quarter of 2023, and an acceptance is expected no later than in the first quarter of 2025. Despite the long timeline until the final standard's release, it is already possible to explore the capabilities. This can be accounted for by the tight integration of the systems engineering community during the development of the specification. To be more precise, the current state of specifications has been made publicly accessible at GitHub with a close to monthly release frequency. Moreover, in parallel with the specifications, a pilot implementation has been developed and made accessible via the same repository. Though the standardization process of SysML v2 is in a very early stage, the specification is already quite mature. It indicates a clear direction, and the impact of SysML v2 on our research is significant as it delivers a plethora of new opportunities on the one hand and is expected to solve issues, such as tool interoperability on the other hand. This affects not only modeling environments but also the capabilities of model verification and validation or the integration with other tools. Moreover, the pilot implementation's existence will allow us to explore the envisioned capabilities of having a bridge between different tools and experiment with new ideas.

B. Abilities: Reducing complexity

The introduction of MBSE to existing development processes results in inherent overhead. Every user is required to possess proficiency in the additional systems engineering relevant tools and comprehend and use certain modeling languages. The inherent complexity of these tools is a huge barrier for users accepting the introduction of systems engineering. However, this complexity can be reduced through utilizing advanced Large Language Models (LLM), such as GPT-4, which enables the user to explain and express certain views of a system model in natural languages. Especially for user without Systems engineering background, AI and LLM could simplify the entry into systems engineering processes by reducing the inhibition threshold for systems engineering tools and eliminating misconceptions by translating and explaining constraints in natural languages. Repetitive and error-prone tasks could be supported by the AI and suggestions for improvement can be incorporated directly during development. The potential to support system engineering with AI is promising. While fully integrated solutions are yet to be realized, current capabilities allow for interaction and the creation of document-based system views. AI integration to model-based processes would fully utilize the potential and ensure interoperability of these models. Despite these limitations, beginners in the field can already benefit from the use of Large Language Models, by making the first steps more accessible and less intimidating. In this regard, in [10] an initial investigation was conducted to explore the feasibility of large language models for user-centric model-based systems engineering.

C. Responsibilites: A more flexible thought model

To support a user through his responsibilities, it is necessary to refine them, ensuring they are optimized, devoid of redundant tasks and unnecessary constraints. Aside from constraints tied to specific tools and abilities (e.g., mandatory tool usage), the optimization of responsibilities hinges on refining the underlying systems engineering processes. The central aspect, in this case, is the need for a developing and modeling concept that enables individual tailoring to reduce unnecessary tasks and complexity. This idea of proactively identifying and removing unnecessary overhead is not new. This approach is a well-established concept known as lean. The origins of lean can be related to lean manufacturing as part of the" Toyota Way" [11] and can be dated back to the post-war ages. It was intended to improve production cycle times by identifying and eliminating activities that do not add value ("waste") for the customer [12] [13]. The establishment of a thought model aiming at "reducing waste" since that has been adopted for different applications such as lean software development, lean project management, or lean product development. As such, lean appears to be a valid thought model for challenging and organizing MBSE approaches which often appear to be clumsy and overkill for engineers. Literature research, in that case, yields only a few outcomes, such as a 2010 contribution to the INCOSE International Symposium targeting "lean enablers for systems engineering" [14], or a relatively comprehensive but still general book published by Oppenheim in 2011 [15]. In the recent past, different publications came up specifically targeting the integration of lean and MBSE, such as Brusa's considerations on manufacturing [16], or the different approaches from Buczacki or Allen investigating lean product development [17] respectively model management [18]. We posit that the lean concept could serve as a valuable thought model for further extending the DSSE approach towards UCSE.

IV. Conclusion and Future Work

In conclusion, this study highlights the shift towards increasingly complex systems, underscoring the imperative role of systems engineering. However, the lack of acceptance for additional Systems engineering expense is challenging. While existing research, such as domain-specific systems engineering aims to improve user acceptance through tailored languages for domain experts, we believe this goes not far enough and propose the necessity for research towards a more user-centric systems engineering approach. With this research we want to lay the foundational groundwork by delineating the concept of a user and exploring avenues for their support. Several key elements are identified for a user-centric systems engineering paradigm. Firstly, there is a critical need to facilitate tool interoperability, allowing users the flexibility to utilize familiar tools. Secondly, the reduction of complexity in tools and modeling languages is essential, thereby minimizing barriers to acquiring new skills. Lastly, recognizing formal systems engineering approaches as potential obstacles, we advocate for the adoption of a lean thought model. This model aims to streamline processes and eliminate unnecessary complexities, addressing the challenges associated with traditional approaches. Looking ahead, critical areas for future research will encompass exploring the ramifications of SysML v2. The evolving landscape of SysML-v2 harbors the potential for heightened interoperability, facilitated by the establishment of an API that broadens the scope for interaction with various tools. Furthermore, the study advocates for an intensified focus on lean systems engineering and its role in eliminating unnecessary complexities. Future research endeavors should explore and expand upon the practical implementations and benefits of lean systems engineering in the context of modern systems development. Additionally, the integration of AI-supported modeling, presents a frontier for exploration. Investigating how artificial intelligence can enhance the modeling process, reduce complexities, and augment user capabilities will likely be a key area of interest. Future research should strive to unravel the potential of AI-supported modeling and its impact on user-centric systems engineering. These research trajectories are geared towards advancing the UCSE approach, envisioning a future characterized by improved interoperability, efficiency, and user-centricity. This ensures that systems engineering, with its holistic and structured approach, aligns seamlessly with diverse user needs.

Acknowledgment

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development and the Christian Doppler Research Association as well as the Federal State of Salzburg is gratefully acknowledged.

References

[1] A. M. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and research opportunities", *Systems Engineering*, vol. 21, no. 3, pp. 172–190, 2018.

[2] R. Dumitrescu, A. Albers, O. Riedel, R. Stark, and J. Gausemeier, "Engineering in germany-the status quo in business and science, a contribution to advanced systems engineering", *Paderborn*, 2021.

[3] T. Huldt and I. Stenius, "State-of-practice survey of model-based systems engineering", *Systems engineering*, vol. 22, no. 2, pp. 134–145, 2019.

[4] D. Wilke, R. Grothe, L. Bretz, H. Anacker, and R. Dumitrescu, "Lessons learned from the introduction of systems engineering", *Systems*, vol. 11, no. 3, p. 119, 2023.

[5] C. Neureiter, M. Uslar, D. Engel, and G. Lastro, "A standards-based approach for domain specific modelling of smart grid system architectures", in *2016 11th System of Systems Engineering Conference (SoSE)*, IEEE, 2016, pp. 1–6.

[6] I. Scheeren and C. E. Pereira, "Combining model-based systems engineering, simulation and domain engineering in the development of industrial automation systems: Industrial case study", in *2014 IEEE 17th International symposium on object/component/service-oriented real-time distributed computing*, IEEE, 2014, pp. 40–47.

[7] C. Neureiter, C. Binder, and G. Lastro, "Review on domain specific systems engineering", in *2020 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, 2020, pp. 1–8.

[8] Object Management Group, *Systems Modeling Language (SysML®) v2 Request For Proposal (RFP)*, Jun. 2018.

[9] Object Management Group, *Systems Modeling Language (SysML®) v2 API and Services Request For Proposal (RFP)*, Jun. 2018.

[10] E. Bader, D. Vereno, and C. Neureiter, "Facilitating user-centric model-based systems engineering using generative ai", in *Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering - MBSE-AI Integration*, 2024, pp. 371–377, ISBN: 978-989-758-682-8. DOI: 10.5220/0012623200003645.

[11] T. Ohno, *Toyota production system: beyond large-scale production*. Productivity press, 2019.

[12] K. Abu Bakar, M. F. Mohd. Sam, and M. I. Qureshi, "Lean manufacturing design of a two-sided assembly line balancing problem work cell", in *Symposium on Intelligent Manufacturing and Mechatronics*, Springer, 2021, pp. 249–259.

[13] J. Womack, D. Jones, and D. Roos, *The machine that changed the world, based on the Massachusetts Institute of Technology 5-million-dollar 5-year study on the future of the automobile*. New York, NY: Rawson Associates [u.a.], 1990, VIII, 323, ISBN: 0892563508.

[14] B. W. Oppenheim, "Lean enablers for systems engineering", in *INCOSE International Symposium*, Wiley Online Library, vol. 20, 2010, pp. 1996–2092.

[15] B. W. Oppenheim, *Lean for systems engineering with lean enablers for systems engineering*. John Wiley & Sons, 2011.

[16] E. Brusa, “Synopsis of the mbse, lean and smart manufacturing in the product and process design for an assessment of the strategy" industry 4.0".”, in *CIISE*, 2018, pp. 21–30.

[17] A. Buczacki, “An integrative model based systems engineering (mbse) and lean-based approach for development of new complex products”, *Multidisciplinary Aspects of Production Engineering*, vol. 2, no. 1, pp. 425–434, 2019.

[18] C. Allen, M. Di Maio, G.-D. Kapos, and N. Klusmann, “Mddp: A pragmatic approach to managing complex and complicated mbse models”, in *2016 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, 2016, pp. 1–8.

 ISBN: 978-1-68558-210-4