Advancing Model-Based Production System Engineering: A Unified Framework for Virtual Commissioning and Discrete-Event Simulation

Martin Birtic and Anna Syberfeldt
Department of Intelligent Automation
University of Skövde
Högskolevägen, Box 408, 54128, Sweden
email: Martin.Birtic@his.se

KEYWORDS
Virtual commissioning, discrete event simulation, digital twins, model-based systems engineering

ABSTRACT

Production simulation holds great promise for industrial applications. Virtual commissioning and discrete event simulation are production simulation techniques that are both economically and operationally justified in theory. However, their practical use is limited by high initial costs and expertise, as well as time and effort requirements. This study proposes a model-based engineering framework with a focus on developing and utilizing these two types of techniques in parallel, with the aim of reducing overall costs while simultaneously harnessing the benefits of both methods' complementary strengths. The proposed framework serves as a basis for future development of methodologies, processes, and tools aimed at streamlining the joint and parallel creation and utilization of said models through simulation-driven systems development. The study presents the framework and an illustrative example that demonstrates its feasibility and practical utility.

INTRODUCTION

Simulation is emerging as an essential tool for managing the increasingly digitalized production systems. By digitizing and representing production systems virtually, advanced algorithms can be applied to extract insights that remain inaccessible through conventional methods, thereby supporting the transition to truly *smart manufacturing* (Kusiak 2018). However, as digitalization advances, emerging technologies not only introduce computational capabilities into production but also increase system complexity. Notable examples include cyber-physical systems, the Internet of Things, and cloud manufacturing (Zhong et al. 2017). Increasing system complexity poses significant challenges to the design, development and operation of these systems. Digitalization tactics aim to address this problem area by using even more digitization tools such as cyber-virtual representations (sometimes referred to as digital twins) of systems to enable more accurate predictions, informed decisions and informed plans (Tao et al. 2018) despite growing complexity. One such tactic is modeling and simulation, which enables systems engineers to adopt a model-based approach and use dynamic, visualized models, instead of working with system development based on a flat and static document-driven methodology (Herter and Ovtcharova 2016). There is a large body of knowledge on model-based system development and from an economic and resource perspective, this literature shows that the approach often requires a large early investment in system development, while the actual benefits of the technology may only become apparent at a later stage (Madni and Purohit 2019). This can lead to uncertainty and reduce incentives for industrial companies to adopt the technology, which may be a factor in the technology not being widely adopted. Therefore, it is important to investigate how the initial investment can be reduced or how activities during this phase can be improved to accelerate and clarify both the early and later advantages and gains of using virtual models for system development.

This study is based on the assumption that the initial cost could be reduced or better utilized by the simultaneous and synergistic development of multiple types of simulation models. In parallel development, activities such as requirement elicitation, data collection, and pre-study insights can be performed collaboratively, while the dual simultaneous development and application of different models may generate value that surpasses what they would achieve individually. There are several simulation methods available which also address different aspects of production systems (de Paula Ferreira et al. 2020). Two of the most common in industrial applications have been selected, namely Virtual Commissioning (VC) and Discrete-Event Simulation (DES). Despite extensive research into the use of different simulation methods (hybridization) (Mustafee and Powell 2018), there is a clear lack of studies that investigate exactly how VC and DES can be combined and exploited. Consequently, the aim of this study is to develop a framework for the simultaneous development of VC and DES models, which in the next steps enables the development of a methodology and a software platform for investigat-

ing this particular combination in detail. To structure the composition of the framework, the following research question has been posed:

RQ What are the key factors that should be integrated into a framework leveraging both VC and DES simulations for model-based system development?

To guide the reader, the rest of this article is structured as follows: The following section provide background regarding model-based systems engineering, modelling and simulation, and how the study's framework can contribute to future development of methodology. This is followed by a presentation of the proposed framework and an illustrative example. Finally, conclusions and future work are provided.

BACKGROUND

Model-Based System Engineering

Designing comprehensive production systems is difficult. There are many moving parts that need to be developed, forged and integrated over a long period of time. Many decisions must be made. One of the difficulties is making decisions that require predicting how the system will behave when finished and implemented. For example, how many buffers are needed, how large an AGV fleet should be, should there be mounted or movable industrial robots, how robust are the control programs, or how safe a specific design of a cell is for its operators. Traditional systems engineering attempts to answer these questions through a document-driven, linear method that is based on more or less static documents such as drawings, descriptive text and non-behavioral 3D models which are developed through sequential processes. Such a methodology can work for projects where the requirements are stable and the conditions that will prevail for the system are well known and follow conventions. Then again, there is presently an increase in the complexity of systems, while production conditions are progressively volatile (Estefan et al. 2007, Roos 2014). This makes traditional system engineering methods seem too rigid and inflexible to arrive at safe design decisions. Making design decisions while lacking knowledge of the system's performance under changing dynamic conditions can mean that suboptimal design decisions propagate and have to be corrected at later stages of development, or during operation, leading to increased rework costs and delays.

Model-Based Systems Engineering (MBSE) is a methodology that shifts from traditional document-based development to a model-driven approach (Estefan et al. 2007, Henderson and Salado 2021). This transition aims to bring the system under development to life at an early stage by faithfully replicating its dynamic behavior as defined by its current hypothesized specifications. By examining the anticipated behavior of an early design, the system concept can be evaluated and insights can be used to refine the design specifications. As a result, the system can be transitioned to a more mature state before being implemented in reality. The models can be formalized in various ways, for example through standardized modeling languages, where there are also modeling languages that can be executed in the form of simulations to get a hint about the dynamics that the current system solution could exhibit (Estefan et al. 2007). In addition, models can be created that contain detailed 3D representations of equipment and products. By subjecting such models to emulated physical laws, even more reliable system dynamics can predict and forecast system performance in different situations. Madni and Sievers (2018) advocates a unified and central model that could serve as a "single source of truth", and that such a model could offer significant potential to overcome the limitations of traditional, document-centric systems development. Ideally, the model is useful for most disciplines involved in systems development where it can contribute to understanding the future performance of the current system configuration for each discipline, enabling all disciplines to make verifiable improvements to the system within their domain of expertise. If traditional systems development can tend to end up in a silo situation where each discipline works separately to develop and refine documents, it is believed that a model-centric approach can overcome such phenomena through increased team communication and synchronization of terminology (Estefan et al. 2007, Madni and Sievers 2018). Madni and Purohit (2019) points out an important finding that MBSE requires higher initial investments than traditional document-centric system design methods but provides significant long-term savings and efficiency gains through improved defect detection, traceability, risk reduction and communication, which is also supported by other researchers (Carroll and Malins 2016, Henderson and Salado 2021), see Figure 1. The fact that these identified benefits are realized in later life cycle stages suggests that further research is needed to reduce initial costs, uncover and accelerate benefits in early life cycle phases, and to better exploit the models in later stages. Another interesting finding was done by Huldt and Stenius (2019) showing that MBSE is mainly applied during early technical phases, with limited integration into broader business processes that extend beyond the system design phase and into the operational phases of the system. They further noted that current tool sets fail to support full life-cycle management, particularly in configuration management and cross-domain collaboration. This highlights the need for frameworks in the field that can support methodology and tool development to improve the situation.

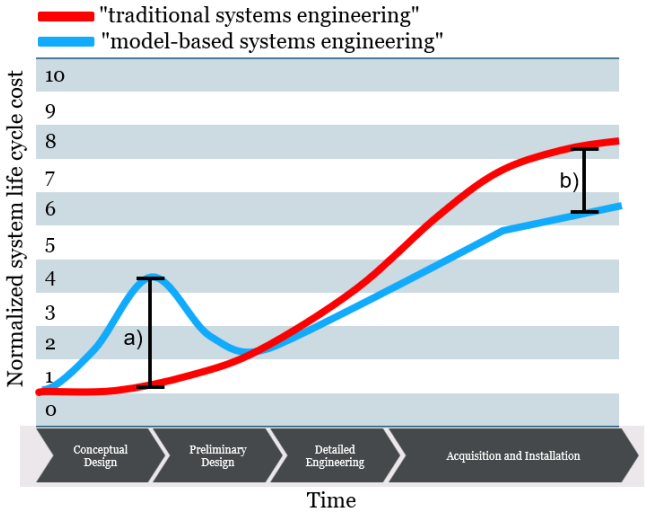

Figure 1: Attributed to Madni and Purohit (2019), the diagram presents normalized life cycle cost dynamics for traditional systems engineering versus model-based systems engineering across different life-cycle stages. While MBSE incurs higher initial costs (a), these investments convert into benefits such as enhanced system maturity, quality, clarity, and manageability in later stages of the life-cycle (b).

Modelling and Simulation

To realize model-based systems engineering, models are needed that can also be simulated and studied to reveal the insights they hold. Modeling and simulation is a field of study and knowledge that has developed processes and methods to enable the creation and simulation of digital representations of complex systems. Through simulation, hypothetical systems dynamics can be explored, future possible outcomes can be predicted under varying conditions, and processes and methods can be optimized without the high costs or risks associated with real-world testing (Banks 2005). Creating relevant models and simulation experiments that achieve useful simulation results often requires significant effort and resources, which has prompted the development of methods and protocols (Lorig 2019) to support model design and simulation studies. Such methods and tools support, among other things, modeling at an appropriate level of detail, effectively handling underlying assumptions, facilitating rigorous experimental development, and optimizing the utilization of simulation results. Simulation techniques encompass a wide range of methods, each targeting specific aspects of system behavior. In this study, we focus on two commonly adopted approaches that are integral to production system development.

Virtual Commissioning

VC in manufacturing uses advanced simulation and digital models to test and validate automation systems, control logic, and other production aspects before physical implementation (Striffler and Voigt 2023). By creating an accurate virtual representation of the system, it is ensured that equipment, software, and processes function as intended, reducing the risk of errors and integration issues. The model is built with a representative 3D model of the developed system, which includes components such as robots, conveyors, sensors, and mechanical moving parts. Products, which reflect the intended real-world objects, are included to varying degrees and details.By providing control system programmers with a simulation that accurately replicates equipment movements and product flow dynamics, control programs can be thoroughly refined and validated before real-world implementation. The simulation is performed in real time or near real time and often requires the integration of multiple software systems, such as PLC software, robot software, and database management. This allows for seamless and efficient validation of the entire production system before implementation begins. VC offers significant economic benefits by reducing ramp-up time, reducing costs, and improving overall system quality. For example, Shahim and Møller (2016) claims that VC enables a reduction in system ramp-up time and faster product introduction time, as well as improved product quality. In practical terms, case studies show that 50% of on-site person-hours could be saved which could translate into significant reduced labor costs or improved working conditions (Shahim and Møller 2016). Furthermore, Makris et al. (2012) has reported a reduction in ramp-up time, which impacts the overall installation time by 15–25% along with a significant reduction in investment costs by reducing the human resources required for troubleshooting by up to 15%. Reinhart and Wünsch (2007) further reinforce these findings by emphasizing that substantial reductions in production costs and lead times are achievable, along with significant enhancements in software quality at startup. Taken together, these studies support the economic case for adopting VC in modern automation industries.

Discrete Event Simulation

DES works at a higher level of abstraction than VC, focusing on event sequencing and flow dynamics rather than fine-grained details. In DES models, details such as PLC logic, physical laws and product geometry, as well as the mechanical movements and specifications of the machines, are excluded. Instead, robots, for example, are abstracted into ”processes” where details lie above control logic and signals at the level of abstraction where individual process steps and sequences are defined as a time interval. The layout may reflect the

intended design, but the focus is not on working with detailed layout features but rather on determining the most efficient sequence and placement of production and process steps. Consequently, as DES primarily focuses on event sequencing at process level and product flows it enables simulations that operate faster than real-time allowing extensive clock times to be represented within significantly shorter simulation runs (Banks 2005). Interesting simulation results from this type of simulations can reflect system dynamics at the process level, resource utilization under specified conditions, and identifying process bottlenecks or non-value-creating process steps (Mourtzis 2020). DES can be used to economically evaluate different manufacturing scenarios by calculating the net present value for each, and it can further identify non-value-added activities whose elimination could lead to a significant increase in monthly production (Montevechi et al. 2010), demonstrating that DES not only streamlines decision-making but also directly contributes to productivity improvements and cost reductions in complex manufacturing environments. Furthermore, DES can provide significant economic benefits to manufacturing systems by serving as a valuable tool for process optimization and decision support as shown by Velumani and Tang (2017) that performed simulations that enabled the identification of bottlenecks and buffer levels and provided practical insights that led to reductions in cycle times and improved throughput.

From Framework to Methodology: A Strategic Evolution

A framework in the context in which it is presented in this study should be seen as an overarching structure that defines and organizes the boundaries within which important components such as processes, methods, and tools operate, see Figure 2. Frameworks from this perspective outline the essential architecture of key components and their interrelationships, providing a foundation for the systematic use of these elements. A methodology on the other hand operates within the framework and is a "recipe" that integrates and prescribes the use of the operational elements: *processes* describe sequences of tasks within the key domains, *methods* describe the specific approaches to performing these tasks, and *tools* are support systems to improve efficiency and accuracy executing the methods (Martin 2020). Essentially, the framework establishes the context and structure, while the methodology leverages that structure to deliver a concrete, actionable plan for achieving the goals.

By further integrating taxonomies and ontologies within the framework, its elements can be organized hierarchically and semantically related, thereby providing a coherent framework that supports effective further development. Taxonomies and ontologies are closely related: *taxonomies* provide a hierarchical classification

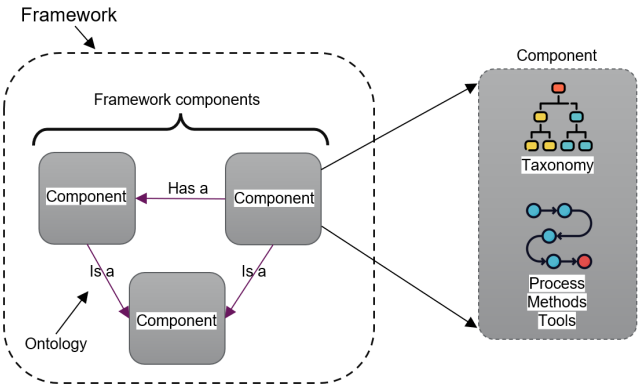

Figure 2: Overview of framework, components, and relationships.

of system elements, while *ontologies* define the semantic relationships and properties among those elements (Dori and Reinhartz-Berger 2003) . Incorporating both into a framework enables clear organization, enhances data integration, and supports effective life-cycle management by ensuring consistency and interoperability. In contrast, frameworks lacking these features could suffer from ambiguous structure and weak semantic grounding, leading to inefficiencies in communication and integration across framework and its processes, methods, tools, and methodologies.

Problem Description

The presented literature emphasizes the significant benefits of models and simulation methods in system development. However, despite both procedural and economic justifications for MBSE and simulation, widespread adoption is impeded by high upfront investments and delayed payoffs, highlighting the need for innovative approaches that lower initial costs, accelerate benefit realization, and extend the application of these models throughout the system life-cycle. This study aligns with this situation by examining the assumptions that by exploiting the system engineering process, multiple types of models can be built and simulated in a more cost-effective manner, while using both types of models can lead to the discovery of ways to use them in more ways during the operational phase of the system.

PROPOSED FRAMEWORK

Based on MBSE and modeling and simulation, a preliminary framework has been developed to facilitate the simultaneous and synergistic development of both VC and DES models, laying a solid foundation for future methodological innovations. Over time, it is intended to evolve into a robust suite of processes, methods, tools, and strategies for effective application.

The elementary components are presented below and are illustrated in Figure 3.

Business Case Modelling
Incorporating a business modeling domain within a modeling and simulation framework is crucial as it ensures alignment with real-world operational needs and decision-making structures, as highlighted by previous research, including the Rational Unified Process (Anwar 2014). By formalizing and modeling the underlying business problem, such as through business process representations, situational constraints and use cases, engineers can prevent inconsistencies and misalignments during system development. Although initial validation of business modelling is conceptually supported, further studies are needed to assess and explore how this type of business logic can be used more effectively to increase consistency across system development stages and mitigate design issues in later phases..

System Specification and Requirements Engineering
The specification of the system to be built is a critical aspect of model-based development, since it provides both a formalization of design intent and a foundation for iterative refinement. Previous research has emphasized the importance of model-based methods to improve the accuracy of system specifications, including the development of "semantically rich representations" of the system (Anwar 2014), the incremental development of executable digital versions, the use of directly simulable specification languages, and digital prototypes (Estefan et al. 2007). Once an initial specification has been translated into a testable VC or DES model, iterative simulations and experiments can be used to identify errors early and thereby improve and refine the specification so that it better corresponds to the requirements established in the business modeling.
Estefan et al. (2007) conducted a survey of MBSE methods and the study strongly emphasizes the use of requirements elicitation and management as a way to outline and gradually clarify the system's specifications. The work with requirements involves, in addition to a detailed review and translation of the requirements into functions, also an approach where each requirement is somehow linked to an executable test that can verify whether the system in its existing design can satisfy the requirement. By iteratively simulating and testing the models against the requirements, the models and the system specification are refined, which increases the probability that the final solution meets all specified criteria. The framework recognizes the integrated relationship between requirements, testing and experimentation and includes a dedicated domain for test- and experiment-driven development, which provides rigorous methods for verification and confirmation of each requirement. In addition, developing taxonomies and ontologies is essential for system specification, as they help define a central database and data model that standardize key concepts and relationships within the domain with the aim to provide consistency throughout the system.

Modelling and Simulation
Adopting an incremental model-based systems approach, where models are iteratively refined, simulated early where they are validated and verified, can lower overall project risk, reduce rework and thereby reduce long-term costs, something that is shown in the methods investigated by (Estefan et al. 2007). At the same time, Madni and Purohit (2019) shows that model-based approaches require greater investment in the early development phases, which can be discouraging, which is why it motivates the development of frameworks and tools to streamline the process. System models and simulations are ideally developed through an iterative, incremental process that successively improves the level of detail. Initially, simple, low-detail models could be used to attempt to render and clarify an early return on investment by enabling early simulation-driven insights. Then, the models would be gradually refined as the system specification is clarified through accumulated simulation data. Such strategic modelling and simulation could justify an approach that promotes parallel and collaborative development of VC and DES models to minimize initial costs and maximize long-term benefits through synergies in the modelling and simulation processes.

Tests and Experiments
Modeling and simulation can be viewed as a test- and experiment-driven methodology aimed at producing new knowledge through rigorous verification and validation, utilizing systematic and reproducible experiments (Lorig 2019). These principles are central to a test-driven development approach that uses tests to ensure that the system meets actual needs (validation), confirms correctness against precise criteria (verification) and maintains integrity even in the face of changes (reproducibility) (Gärtner 2012). In addition, in model-based systems engineering, there is an integrated interplay between requirements, testing and experimentation, where tests and experiments are based on explicitly defined requirements and specifications. Each requirement is systematically linked to the corresponding tests, and through targeted experiments, multiple tests can be carried out. To ensure rigor and traceability, requirements, tests, and experiments are formally managed through careful planning and documentation. Such a structured approach that connects these parts is essential for generating simulation data that is empirically relevant. It is hard to envision achieving high-quality, data-driven system development without such integrated components.

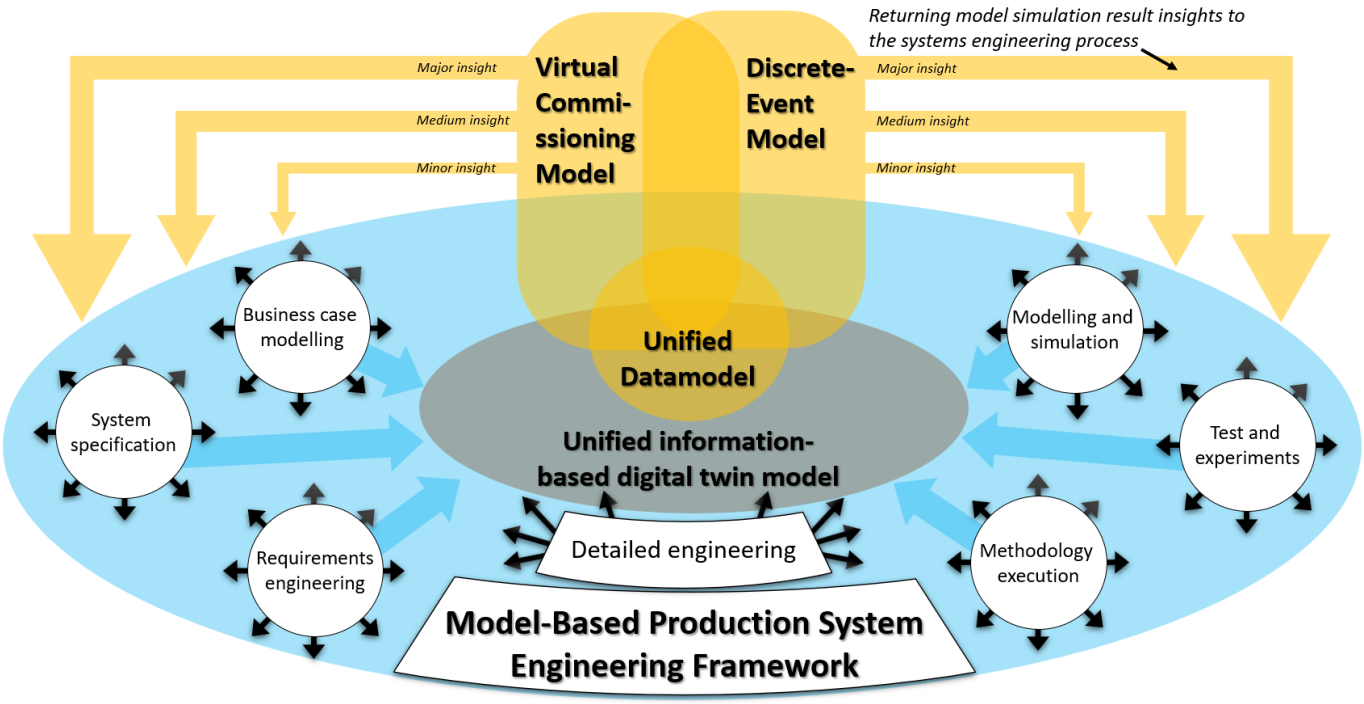

Figure 3: Six key areas of model-driven systems development are business case modelling, system specification, requirements engineering, modelling and simulation, testing and experiments, and methodology control. Collectively, they provide essential data for an informational digital twin that incorporates a consolidated data model, which in turn is used to instantiate both a virtual commissioning- and a discrete-event simulation model. These simulation models generate insights that can be applied throughout system development. The insights guide both strategic decision-making and detailed engineering efforts where intricate system components are designed.

Methodology Execution

Since modeling and simulating complex systems is a demanding task, research has been devoted to developing processes, methods, and tools that serve as guiding principles to ensure robust simulation methods. Lorig (2019) emphasizes that structured methods not only promote standardization and increase transparency, but also reduce methodological pitfalls by enabling continuous improvement. By introducing systematic workflows, these approaches help maintain rigor, clarity, and reproducibility in complex modeling studies, tests, and experiments. Modeling and simulation frameworks are inherently complex and require significant expertise to configure, use, and maintain. To meet this challenge, framework-related practical guides, protocols and tools need to be developed to promote consistent, efficient, and high-quality modeling and simulation practices within systems engineering. While several studies have emphasized the need for systematic guidelines, explicit methods are lacking to initiate the development of methodologies for simultaneous modeling of both VC and DES models as an integrated part of the system design, which is the long-term goal of this research. However, this current study only outlines the conceptual framework and the positioning of methodology within it. Further research is necessary to develop specific methodologies and to evaluate their advantages and limitations in managing cost and effort for efficient concurrent development of VC and DES models in MBSE approaches.

ILLUSTRATIVE EXAMPLE OF APPLICABILITY

This chapter illustrates an example where the proposed framework components have been used to initialize a systems engineering process of a demonstrator production cell for product quality control and defect removal. The example demonstrates that the proposed framework components can simultaneously contribute to the development of VC and DES models during the concept and system specification phase, a scenario driven by the challenges outlined in the study's introduction and background. A sample of the results are illustrated in Figure 4, accompanied by a summary description below.

Business Modelling Domain

Flowchart modeling was employed to represent the business logic, focusing on the issue of unintentional defective product deliveries to customers, which lead to significant costs. The objective is to develop a solution that mitigates these occurrences and minimizes negative impact.

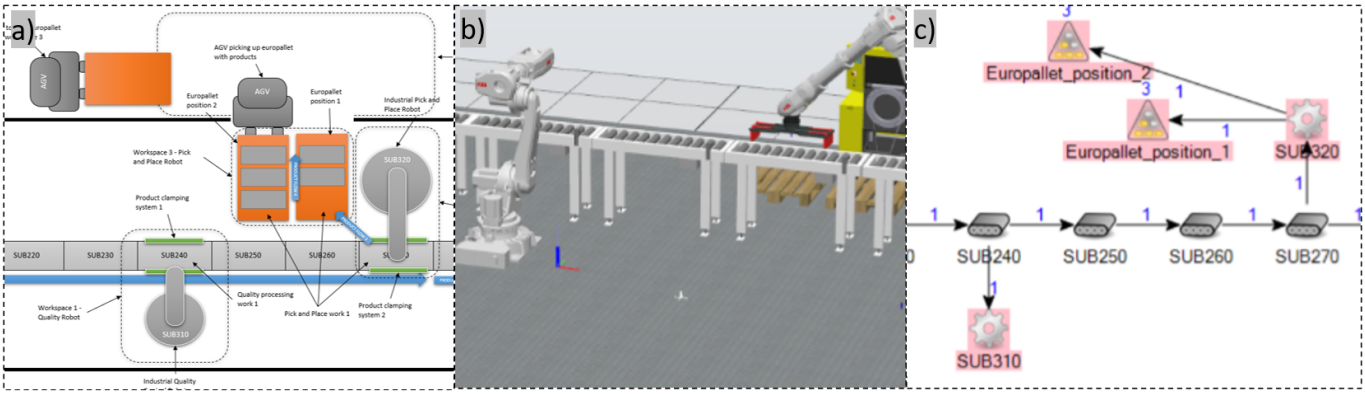

Figure 4: A detailed system layout (a) and corresponding specifications were developed through iterative virtual model building. The VC model solution (b) features a quality robot that inspects incoming products, followed by a robot that removes defective products in the subsequent sequencing stage. Additionally, a shared, information-driven digital twin was created using a queryable diagram database and the layout was also replicated in a DES model (c).

System Specification and Requirements Engineering

The system in the example was first specified at a high level in the form of a functional description that was iteratively improved during the development of a basic VC model. As details were resolved, a database and a query-enabled diagram database were developed with all current information about the intended system. This "knowledge graph" can be seen as the central "single source of truth" as mentioned in MBSE literature. The functional description of the system is as follows: Products arrive at the inbound conveyor, where they are stopped and inspected by an industrial robot equipped with a camera. If a defect is detected, the product's production data is updated accordingly. The products then move on to the next cell, where those with registered defect data are identified and removed from the flow by a pick-and-place robot that places them on pallets. Automated guided vehicles then transport the pallets with defective products to the outbound area for further processing or disposal.

Modelling and Simulation Domain

In this study, a system layout and three types of virtual models were developed through an iterative process, see Figure 4. A modeling of the problem led to an initial functional description of the system that was gradually decomposed and refined into clearer specifications. Parts of the evolving specification were continuously digitized and analyzed for insights to improve the specification. Rather than following a strict sequential approach, the modeling process was dynamic and adapted to discoveries and adjustments in real time. This suggests that a rigid sequence of activities may not be practical. Furthermore, the models were not simulated in this case, although simulation could have further improved the system specification - an aspect that will be explored in future research.

Test- and Experiment-Driven Development Domain

During the development of the example, a simplified requirements elicitation has been performed. The requirements have been documented in tabular form and each requirement has been decomposed and paired with corresponding acceptance criteria. These criteria have in turn been broken down into practical test steps. In the study, these tests have not been performed, which could lead to a more refined solution.

CONCLUSION

This study begins an exploration of the integration of VC and DES through parallel development and exploiting co-development synergies. As a first step, a basic model-based systems engineering framework has been established. A framework that leverages VC and DES should integrate several key areas. It should include business modeling, iterative system specification, and requirements elicitation to drive increasingly accurate and mature designs. It also makes sense to incorporate test-driven modeling and simulation processes that explicitly validate system specifications. In addition, the framework should provide mechanisms for capturing and accumulating data that can be used to instantiate both VC and DES models. Collectively, these components create a framework that can be used to further develop a methodology for the parallel and synergistic development and utilization of VC and DES models.

Limitations

A limitation of this research is that developing a complete methodology for the simultaneous creation of VC and DES models is highly complex, involving many components and requiring expert collaboration. A framework represents only one part of this multifaceted process, and fully evaluating it requires further develop-

ment. As the research is still in early stages, the proposed framework has not been thoroughly validated. However, it is grounded in literature and extensive simulation experience. To support feasibility, an illustrative example was developed, incorporating material from all proposed domains, though with varying levels of detail. Model-based systems engineering often aspires to a single, comprehensive model. While desirable, no standard model or software currently fulfills this role. Therefore, our research supports the use of multiple models. Nonetheless, we rely on one unified informational model in the form of a central knowledge graph consolidating all system information.

Future work

Future work will develop a methodology, processes, methods, and tools for parallel development of VC and DES models. The methodology will explore a model- and test-driven approach that starts with minimal detail and resource investment while generating fundamental simulation empirical data to drive system development.

REFERENCES

Anwar A., 2014. *A review of rup (rational unified process). International Journal of Software Engineering (IJSE)*, 5, no. 2, 12–19.

Banks J., 2005. *Discrete event system simulation*. Pearson Education India.

Carroll E.R. and Malins R.J., 2016. *Systematic literature review: How is model-based systems engineering justified?*

de Paula Ferreira W.; Armellini F.; and De Santa-Eulalia L.A., 2020. *Simulation in industry 4.0: A state-of-the-art review. Computers & Industrial Engineering*, 149, 106868.

Dori D. and Reinhartz-Berger I., 2003. *An OPM-based metamodel of system development process*. In *International Conference on Conceptual Modeling*. Springer, 105–117.

Estefan J.A. et al., 2007. *Survey of model-based systems engineering (MBSE) methodologies. Incose MBSE Focus Group*, 25, no. 8, 1–12.

Gärtner M., 2012. *ATDD by example: a practical guide to acceptance test-driven development*. Addison-Wesley.

Henderson K. and Salado A., 2021. *Value and benefits of model-based systems engineering (MBSE): Evidence from the literature. Systems Engineering*, 24, no. 1, 51–66.

Herter J. and Ovtcharova J., 2016. *A model based visualization framework for cross discipline collaboration in industry 4.0 scenarios. Procedia CIRP*, 57, 398–403.

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

Kusiak A., 2018. *Smart manufacturing. International journal of production Research*, 56, no. 1-2, 508–517.

Lorig F., 2019. *Hypothesis-Driven Simulation Studies*. Springer.

Madni A.M. and Purohit S., 2019. *Economic analysis of model-based systems engineering. Systems*, 7, no. 1, 12.

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

Makris S.; Michalos G.; and Chryssolouris G., 2012. *Virtual Commissioning of an Assembly Cell with Cooperating Robots. Advances in Decision Sciences*.

Martin J.N., 2020. *Systems engineering guidebook: A process for developing systems and products*. CRC press.

Montevechi J.A.B.; da Silva Costa R.F.; Leal F.; and de Pinho A.F., 2010. *Economic evaluation of scenarios for manufacturing systems using discrete event simulation based experiments. Brazilian Journal of Operations & Production Management*, 7, no. 1, 77–103.

Mourtzis D., 2020. *Simulation in the design and operation of manufacturing systems: state of the art and new trends. International Journal of Production Research*, 58, no. 7, 1927–1949.

Mustafee N. and Powell J.H., 2018. *From hybrid simulation to hybrid systems modelling*. In *2018 Winter Simulation Conference (WSC)*. IEEE, 1430–1439.

Reinhart G. and Wünsch G., 2007. *Economic application of virtual commissioning to mechatronic production systems. Production engineering*, 1, 371–379.

Roos G., 2014. *The constantly changing manufacturing context. Advanced Manufacturing-Beyond the Production Line Committee for Economic Development of Australia (CEDA)*.

Shahim N. and Møller C., 2016. *Economic justification of virtual commissioning in automation industry*. In *2016 Winter Simulation Conference (WSC)*. IEEE, 2430–2441.

Striffler N. and Voigt T., 2023. *Concepts and trends of virtual commissioning–A comprehensive review. Journal of Manufacturing Systems*, 71, 664–680.

Tao F.; Zhang H.; Liu A.; and Nee A.Y., 2018. *Digital twin in industry: State-of-the-art. IEEE Transactions on industrial informatics*, 15, no. 4, 2405–2415.

Velumani S. and Tang H., 2017. *Operations status and bottleneck analysis and improvement of a batch process manufacturing line using discrete event simulation. Procedia Manufacturing*, 10, 100–111.

Zhong R.Y.; Xu X.; Klotz E.; and Newman S.T., 2017. *Intelligent manufacturing in the context of industry 4.0: a review. Engineering*, 3, no. 5, 616–630.