Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

DESIGN SPACE SPECIFICATION, EXPLORATION, AND SIMULATION FOR PRODUCTION SYSTEMS

Nick Paape

Control Systems Technology
Eindhoven University of Technology
Eindhoven, THE NETHERLANDS

ABSTRACT

Designing a production system using simulation can be a challenging cycle of specifying, modelling, simulating and evaluating the performance of each (re)design. This extended abstract presents a framework for automated design space exploration of production systems. The framework includes a formal design space specification language for production systems which can be used to generate potential designs, and a method for automated exploration and analysis of the specified design space using simulation.

1 INTRODUCTION

The design process of a production system is an intricate process, with many iterations of (re)design. Use of simulation in the design process allows for predictions on the product flow through the system, making it possible to compare design alternatives. However, this generally requires many iterations of specifying the design, constructing a simulation model, performing simulation experiments, and interpreting the simulation results. This can be a complex and time-consuming task, which is why the goal of this research project has been to develop a framework for simulation-based design space exploration of production systems.

Similar design space exploration frameworks have been developed in the fields of vehicle powertrains (Kort et al. 2020) and embedded systems (Kang et al. 2011). As noted in the latter, an effective design space exploration framework must describe methods for representation, analysis, and exploration of the design space. Our framework includes a formal design space specification language for representation of a production system's design space which can be used to generate potential designs, and a method for exploration and analysis of the specified design space using simulation.

2 DESIGN SPACE SPECIFICATION

This section describes the formal design space specification language, from which potential designs can be generated. From a product flow perspective, a production plant consists of resources (e.g. machines) and products which flow between these resources. Thus, the functional design of a production plant can be considered to consist of the resources in the system, and the connections between these resources (e.g. a conveyor bringing products from machine A to machine B). The proposed formal design space language has been described using Extended Backus-Naur Form (EBNF), and supports specification of (1) which types of resources can be used in the design, (2) which instances of these types there must be, and (3) constraints on the design, such as, for example, how many of a resource type are used or on how resource must be connected. This translates to the following EBNF notation: `specification := defComponentTypes, defComponentInstances, constraints`. How each of these definitions are expressed is omitted for brevity. Figure 1 shows an example of a possible specification, which describes the component types, instances and constraints. Note that not all instances need to be defined explicitely. Figure 2 shows an instantiation of a possible system design.

Component types

- There is component type **Infeed** with output ports (out)
- There is component type **MachineA** with input ports (in) and output ports (out1, out2)
- There is component type **MachineB** with input ports (in) and output ports (out)
- There is component type **Outfeed** with input ports (in)

Component instances

- infeed1 is an instance of Infeed
- outfeed1 is an instance of Outfeed

Constraints

- #Components(MachineA) ≤ 1 AND #Components(MachineB) ≤ 2
- For all instances iN of Infeed: There exists an instance iA of MachineA: iN.out connects to iA.in
- For all instances iB of MachineB: There exists an instance iO of Outfeed: iB.out connects to iO.in

Figure 1: Formal specification of the design space.

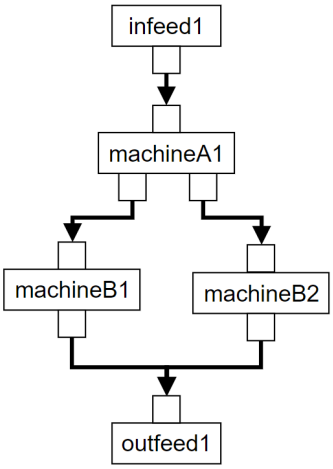

Figure 2: Instantiation of a possible design.

3 DESIGN SPACE EXPLORATION AND ANALYSIS

Exploration and analysis of the specified design space is achieved through the method shown in Figure 3. In this method, the design space is explored iteratively using an exhaustive search. In each iteration a potential design is generated. To analyze the performance of the design a model of it is constructed using a model library. To automatically construct the model requires that each of the resource types specified in the design space have a corresponding model component. A production system generally operates in a dynamic environment, which is why the constructed model is simulated for a range of production scenarios as defined by the system designer. After simulation the design is evaluated according to the chosen performance indicators. When all designs are evaluated, the method recommends the set of pareto-optimal designs.

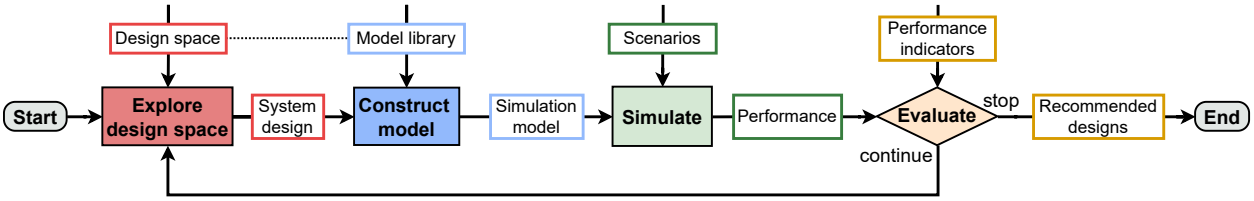

Figure 3: The proposed simulation-based method for design space exploration of production systems.

4 FUTURE WORK

So far the focus has been on the representation and analysis of the design space. In the future we would like to investigate more effective options for exploration, as exploring the design space using exhaustive search is only viable for a relatively small design space. We plan to investigate which (meta)-heuristics can be used to explore the design space, and to identify if we can use feedback from the 'evaluate' step to direct design space exploration.

REFERENCES

Kang, E., E. Jackson, and W. Schulte. 2011. "An Approach for Effective Design Space Exploration". In *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems. Monterey Workshop 2010. Lecture Notes in Computer Science, vol 6662*, edited by R. Calinescu and E. Jackson, 33–54. Springer Berlin Heidelberg.

Kort, A.-J., J. Wijkniet, A. Serebrenik, and T. Hofman. 2020, 11. "Automated Multi-Level Dynamic System Topology Design Synthesis". *Vehicles* 2(4):603–624.