Automated Generation of Simulation Models for Production and Logistics Processes Using LLM-Based Multi-Agent Systems

Roman Krämer*, Jens Heger

Institute for Production Technology and Systems (IPTS), Leuphana University Lüneburg, Universitätsallee 1, 21335 Lüneburg, Germany; *roman.kraemer@leuphana.de

SNE 36(1), 2026, 9-14, DOI: 10.11128/sne.36.tn.10762
Sel. ASIM SST 2024 / ASIM SPL 2025 Postconf. Publ. 2025-09-30
Received Improved English: 2025-12-01; Accepted: 2025-12-20
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Discrete-event simulation (DES) is a well-established method for analyzing and optimizing complex production and logistics systems. However, its use is often limited by high modeling effort and the need for specialized expertise. This article presents a novel multi-agent system based on Large Language Models (LLMs) that automates the creation and validation of simulation models. Building on a previous approach of ours, the new system employs an agent-based architecture designed to address issues such as context loss and the need for manual validation. Specialized agents handle tasks from requirements elicitation to results evaluation. The implementation uses open source frameworks "LangGraph" to structure agent interactions and "SimPy" to model the simulation logic. A case study demonstrates that the system can automatically and reproducibly model a complete production scenario from text-based descriptions. The results show realistic modeling and a significantly reduced modeling effort compared to both a manual approach and our previous system. The proposed approach lowers reliance on expert knowledge and makes simulation-based methods more accessible to non-specialist users.

Introduction

Growing complexity in modern production and logistics systems combined with volatile markets is driving demand for powerful planning and analysis tools. In this context, discrete event simulation has long established itself as a key method for planning and optimizing such systems.

Particularly in complex, dynamic systems with stochastic influences, where static or deterministic methods reach their limits and prove inadequate, simulation studies provide possibilities for accurate process modeling, identification of weak points and creating a sound foundation for decision-making as well as optimization measures [1].

In practice, however, their application is hampered by high modeling costs: Beyond requiring deep understanding of the specific production processes, practitioners need expertise for modeling and specialized software, most often necessitating external consultation. The resulting coordination processes are time-consuming and thus costly, often making simulations economically viable only for large scale projects. While these challenges aren't new, recent advances in the field of artificial intelligence (AI) and especially large language models (LLMs) present new opportunities for simplifying the modeling process and execution of simulation studies.

In our previous work [2], we presented a monolithic approach in which an LLM handled the entire modeling process from user interaction through code generation to output of results. While this approach successfully generated functional Python code for simulation models using the framework "SimPy", a critical limitation became apparent during the modeling process: the LLM lost focus over longer contexts, requiring numerous manual correction loops. The effort was merely shifted from model creation to code validation, which meant successful use of such an approach remained limited to experts still.

This paper extends that approach and introduces an LLM-supported multi-agent system, in which specialized agents with narrowly defined task focuses handle the automated creation of simulation models based on natural language text descriptions. To evaluate the improvements compared to the previous approach, we revisit the production scenario from our previous paper.

1 State of the Art

The rapid advancement of LLMs is driving their increasing adoption in corporate contexts. Systematic studies focusing on production environments indicate that LLMs are especially used for communication and information provisioning, predominantly as chatbots in scenarios of direct interaction with employees [3].

LLMs with added retrieval augmented generation (RAG) approaches find utilization in structured delivery of knowledge from companies' internal databases and, in some cases, serve as worker support tools for processes in production related tasks [4]. Within the domain of simulation and modeling, LLMs have been proposed and used to assist with simulation software usage by helping users navigate interfaces and provide context-specific recommendations for problem-solving [5]. Approaches for translating natural language descriptions into simulation models and automated code generation for simulation models have been investigated for general suitability using simple examples, particularly in the field of production logistics [6]. These approaches often focus on specific aspects, such as modeling of queues [7], through use of prompt engineering without further task decomposition into subtasks.

Consequently, focus drift and loss of focus during extended interactions is observed, necessitating manual correction efforts. Therefore, such approaches are only suitable for experienced users. Further research has explored approaches for automated structuring of customer requirements that can be imported into simulation environments for tasks such as layout planning [8].

LLM-based multi-agent systems for complex process automation represent an active area in current research. Of particular note is an approach for automated scientific manuscript generation [9]. One such manuscript generated by the system outlined in that approach was submitted to a scientific conference workshop and received ratings exceeding the average peer review acceptance threshold [10].

While multi-agent systems are extensively used experimentally, they have not yet explicitly found their way into the field of production and logistics simulations [11]. Approaches employing multi-agent systems for parameterizing exemplary, simplified simulation models have been described, though not within production or logistics contexts [12]. To our knowledge, no existing approach addresses the complete pipeline for generating simulation models in production and logistics contexts with a focus on accessibility for users without simulation expertise.

Facilitating the use of simulation as a methodology therefore remains an unresolved challenge.

2 Approach & Implementation

Building on the findings from our previous approach, and in particular with regard to the loss of focus, the limitations of a monolithic approach are addressed by following the example of real interdisciplinary collaboration. Therefore, both a clear division of labor and a structured validation strategy are introduced, establishing a collaborative multi-agent system. Since the system corresponds to a directed graph, "LangGraph" is used for implementation. This layer of abstraction also enables a flexible system that functions independently of a specific LLM or LLM provider. Additionally, the Python framework "SimPy" is used for implementing the simulation logic. The presented system, which automatically generates, runs and evaluates simulation models based on a natural language input description, aims to enable users without in-depth simulation knowledge to independently conduct simulations for decision support.

The agent-based architecture of the system comprises a total of five roles:

- **Agent 1 - Requirement Elicitor Agent:** Conducts a structured conversation with the user to collect all necessary requirements (e.g., entities, processes, resources, simulation goals) and summarizes them to the user. After user confirmation, no further input by the user is necessary.
- **Agent 2 - Simulation Plan Agent:** Creates a detailed simulation plan from that summary with all model components, logic, schedules, resources, error rules, and metrics as a basis for code generation.
- **Agent 3 - Code Generator Agent:** Uses the simulation plan to generate modular Python code with functions, classes and data collection mechanisms.
- **Agent 4 - Code Validator Agent:** Checks the code for correctness, logical consistency and complete implementation of requirements. Provides feedback for revision, if necessary, to the code generator, which revises the code and resends it to the code validator.
- **Agent 5 - Scenario Tester Agent:** Tests the model against the collected goals and objectives. Checks the implementation and limitations in the evaluation. If necessary, provides feedback for improvement to the code generator, which revises the code and resends it to the code validator.

Each agent works sequentially with built-in validation and confirmation steps, creating a reliable and modular pipeline for converting production system descriptions into executable SimPy simulation models (see Figure 1).

The used LLMs are "gpt-4.1-2025-04-14" from OpenAI and "claude-sonnet-4-20250514" from Anthropic, both representing the current state-of-the-art at the time of creation of this work. These models were chosen due to their availability and performance, but particularly for their large context windows of 1,047,576 tokens (approx. 750,000 words of English text) for GPT-4.1 and 200,000 tokens (approx. 150,000 words of English text) for Claude Sonnet 4 [13, 14], as maintaining contextual coherence proved to be a critical factor in the previous monolithic approach.

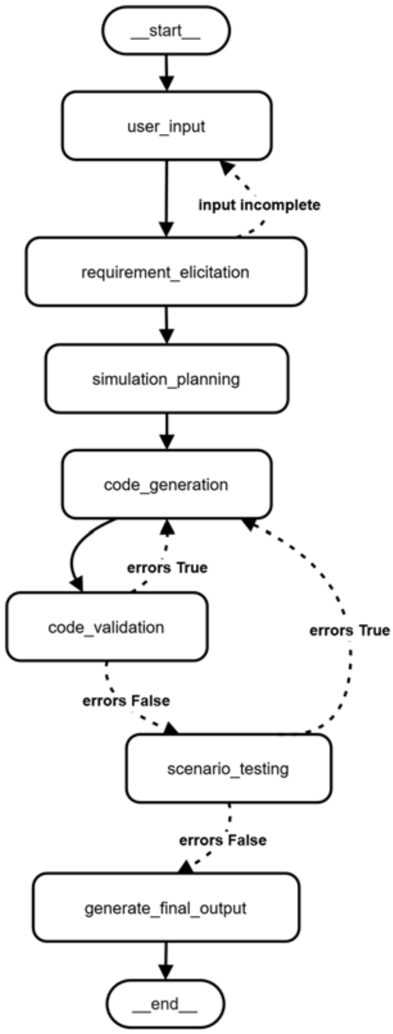

Figure 1: Structure and workflow of the multi-agent system using "LangGraph" for agent orchestration.

The experiments showed that reasoning models such as Anthropic's Claude Sonnet are well-suited for complex tasks such as code analysis and generation, but due to their reasoning capabilities, they also tend to digress when performing narrowly defined, structured tasks such as collecting explicitly defined requirements.

Based on these findings, Agent 1 uses OpenAI's non-reasoning model, while the other agents use Anthropic's reasoning model. The task and role descriptions (system prompts) for the individual agents were developed based on the procedures described in VDI 3633 Part 1 using the system's outputs for iterative refinements.

The complete implementation with instructions for setup and operation, as well as the scenario description for the initial prompt is available at the web link: https://github.com/romankraemer/SNE2025

3 Case Study

To validate and evaluate the improvements by the multi-agent system presented, we revisit the production scenario example from our previous work [2].

The scenario comprises a raw material storage, a buffer storage, a finished goods warehouse and two production machines. Two automated guided vehicles (AGVs) handle transport operations between storages and machines using a pull-based logic. At start, AGV-1 retrieves the required quantity of raw material for one product from the raw material storage and loads it into machine-1, which then begins processing.

Upon completion of this processing step, the intermediate product is transferred via a chute to the buffer storage located behind machine-1. AGV-2 collects the intermediate product, transports and loads it into machine-2. After the processing step in machine-2, the now finished product slides via a chute into the finished goods warehouse located directly behind machine-2.

At simulation start, the raw material storage contains 100 units of material. This initial quantity is chosen based on the processing times just to ensure that no material shortage occurs over the simulation period. One unit of raw material is required to manufacture one product.

The processing, handling, and transport times are modeled as deterministic, constant values in order to ensure reproducibility and traceability of the results. The time steps are shown in Table 1:

Time steps	Time in minutes
AGV-1 transport task: material storage – machine-1	2 (pickup, transport, unload, return)
AGV-2 transport task: buffer – machine-2	2 (pickup, transport, unload, return)
Processing time: machine-1	10
Processing time: machine-2	20
Simulation duration	480

Table 1: Production system properties and parameters.

As the only modification to the scenario, the AGV time steps were divided into four sub steps (pickup, transport, unload, return to start position), each with a duration of 0.5 minutes. So, the absolute cycle times remain unchanged. Notably, this was explicitly requested by Agent 1 during requirement elicitation, indicating an understanding of pull-based control in production simulation contexts and importance for clearly defined simulation states.

This scenario was deliberately designed with a bottleneck at machine-2 which has double the processing time compared to machine-1. The simulation objective is to model and track inventory levels in units across the three storages as well as the utilization of the production machines and AGVs. This data is then compared with the simulation results from our previous work.

4 Results

Following the methodology described in Chapter 2, the scenario including parameters, simulation objectives and desired output values is provided to the multi-agent system as a text description. Queries from the Requirement Elicitor Agent are addressed accordingly.

The multi-agent system captured the production system in our scenario correctly and reproducibly given the same or similar input (accounting for variations based on Agent 1's clarifying questions), generated a consistent simulation model, and executed it successfully. The results are visualized via a graphical dashboard and additionally stored in suitable data files for further use. The following diagrams are based on the output data files.

A comparison of the inventory level diagrams shows plausible results and overall consistency between the previous approach with manual correction loops during modeling (see Figure 2) and the now automated multi-agent system approach (see Figure 3).

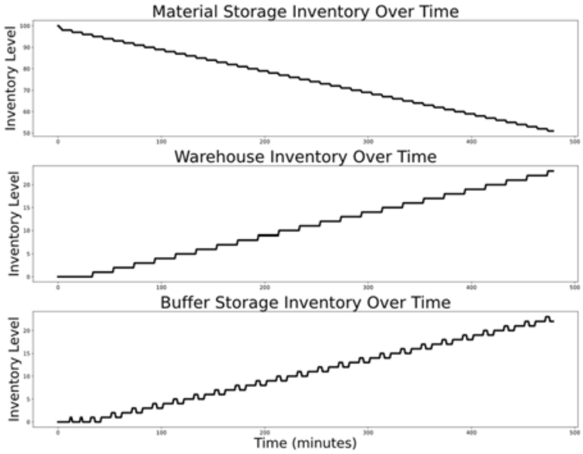

Figure 2: Inventory levels for material storage, warehouse, and buffer storage using the previous approach.

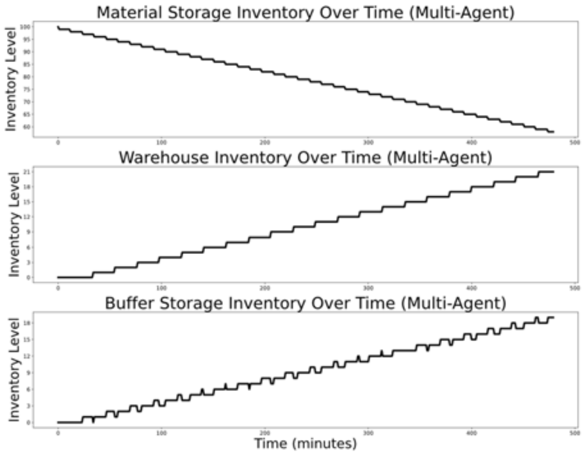

Figure 3: Inventory levels for material storage, warehouse, and buffer storage using the new approach.

However, direct comparison reveals some differences. Most notably, the production scenario achieves higher throughput with the previous approach.

While 23 products reach the finished goods warehouse in the simulation model created with the previous approach, only 21 do so with the new multi-agent system.

This difference stems from the previous system's implementation of AGVs and machines without sufficiently defined states and constraints: The AGVs effectively act as buffers, waiting with their payloads for the machines to become available. Since the transport and handling times are shorter than the processing times for both machines, the raw materials and intermediate products become available in the machines without any delay whenever the machines finish a task once the production system reaches steady state.

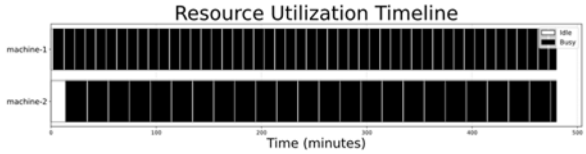

Figure 4: Machine utilization with previous approach.

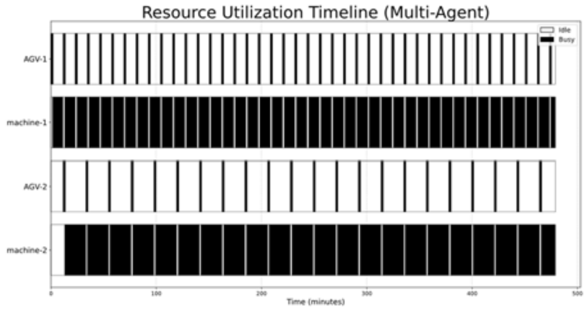

Figure 5: AGV and machine utilization with new approach.

This is evident in the machine utilization diagrams for the previous approach (cf. Figure 4).

Agent 1's system prompt specifies asking precise questions about the production scenario, including the definition of initial and final states as well as any control logic. After eliciting requirements from the user, the multi-agent system implemented a pull-based control for the AGVs, which is visible in the utilization diagram as machine idle time (cf. Figure 5).

In contrast to the previous simulation model, the pull-based control introduces realistic delays between the end of processing and the start of processing a new product due to transport and handling times. This yields lower but more realistic throughput values, which are also reflected in the utilization rates for both machines and AGVs during the simulation (cf. Table 2).

Resource	Prior approach	Multi-agent system
AGV-1	20,4% (98 min)	17,5% (84 min)
AGV-2	10,4% (50 min)	9,2% (44,2 min)
Machine-1	99,6% (478 min)	86,5% (415,2 min)
Machine-2	97,1% (466 min)	90,5% (434,4 min)
Finished products	23	21

Table 2: Utilization of individual resources in the models.

For utilization calculations of AGV-1 and AGV-2 in the previous approach, only raw travel times were used. If times during which the AGVs were waiting for the machines to become available were considered, the utilization would be closer to 100%.

In addition to creating diagrams and metrics, the new system includes a "Scenario Tester Agent". It compares the simulation model against the requirements collected by Agent 1, and after running the simulation, evaluates the results upon successful completion. If applicable, the agent issues recommendations for possible optimization actions. An exemplary recommendation for the production scenario with bottleneck added at machine-2 as used in this case study was:

"... If increased throughput is desired, improvements should focus on machine-2's processing time or capacity, as it is the limiting factor in the production system."

The system correctly identified the bottleneck at machine-2 and understood the simulation's objective, providing the user with specific recommendations for optimizing the production system.

5 Summary and Outlook

This paper demonstrates the potential of LLM-based multi-agent systems for production simulation and optimization.

Compared with our previous approach, the effort for requirements specification was significantly reduced. Now after an initial description, only a few targeted questions from the agent system were needed to generate a technical simulation plan which served as the basis to generate executable simulation models.

Novel to this approach is the complete automation of the process, from code generation, validation and execution to the analysis of simulation results and derivation of optimization recommendations. This addresses the main limitation of the prior approach, where majority of the effort was shifted from modeling to code validation as model complexity increased.

For our exemplary production scenario, an end-to-end run incurred an average cost of approx. US$0.20. This low cost, combined with the high degree of automation, shows potential for making simulation-based methods far more accessible even to users without deep domain expertise.

Nonetheless, the quality of the results remains highly dependent on the accuracy and completeness of the user-provided information. A basic understanding of key concepts and processes in modeling and simulation is still beneficial.

In the experiments conducted, it became evident that purely technical errors are easy to identify and correct. More challenging, however, is the handling and integration of domain knowledge, which often exhibits context-dependent variations or is based on experience-driven judgments. Nevertheless, the system demonstrates how simulation-based methods can be made more accessible with significantly less effort compared to manual model creation.

Acknowledgements

The project "TrICo - Transformation through Innovation and Cooperation in Communities" was funded by the Federal Ministry of Education and Research under grant number 03IHS284A. The author is responsible for the content of this publication.

Publication Remark

This contribution is the improved extended English version of the German conference versions for **ASIM SST 2024** - 27th Symposium Simulation Technique - Munich 2024 and **ASIM SPL 25** - 21. ASIM-Fachtagung Simulation in Produktion und Logistik - Dresden 2025:

Krämer R, Heger J. Beschleunigte Erstellung von Simulationsmodellen für Produktions- und Logistikprozesse mithilfe von GPT-basierten Large Language Models. ASIM 2024 Tagungsband Kurzbeiträge, ARGESIM Report 46, e-ISBN 978-3-903347-64-9, p 21-24. Vol. DOI 10.11128/arep.46

Krämer R, Heger J. Automatisierte Generierung von Simulationsmodellen für Produktions- und Logistikprozesse mithilfe LLM-basierter Multi- Agenten-Systeme. Simulation in Produktion und Logistik 2025. Techn. Univ. Dresden. DOI: 10.25368/2025.290

References

[1] Sutherland R, Özkul F, Wenzel S. Studie zum Einsatz der ereignisdiskreten Simulation in Produktion und Logistik. In Rose O, Uhlig T, editors. ASIM SST 2024 Tagungsband Langbeiträge. *27. ASIM Symposium Simulationstechnik*; 2024 Sep. Wien: ARGESIM Publisher Vienna. 185–192. DOI 10.11128/arep.47.a4716

[2] Krämer R, Heger J. Beschleunigte Erstellung von Simulationsmodellen für Produktions- und Logistikprozesse mithilfe von GPT-basierten Large Language Models. In Rose O, Uhlig T, editors. ASIM SST 2024 Tagungsband Kurzbeiträge. *27. ASIM Symposium Simulationstechnik*; 2024 Sep. Wien: ARGESIM Verlag. 21–24. DOI 10.11128/arep.46.

[3] Finkel P, Wurster P, Radler R. Large Language Models im Produktionsumfeld - Eine Potenzialanalyse zur Transformation von Produktionsabläufen in modernen Fabriken. *Industry 4.0 Science*. 2024; 40(6): 48–55. DOI 10.30844/I4SD.24.6.48.

[4] Heredia Álvaro JA, Barreda JG. An Advanced Retrieval-Augmented Generation System for Manufacturing Quality Control. *Advanced Engineering Informatics*. 2025; 64(1). DOI 10.1016/j.aei.2024.103007

[5] Anylogic. Exploring the Utility of ChatGPT as a Java Scripting Aid for AnyLogic Modeling. 2023. https://www.anylogic.de/blog/unlocking-the-power-of-simulation-modeling-in-chatgpt/. Accessed 18 Jul 2025.

[6] Jackson I, Jesus Saenz M, Ivanov D. From natural language to simulations: applying AI to automate simulation modelling of logistics systems. *International Journal of Production Research*. 2024; 62(4): 1434–1457. DOI 10.1080/00207543.2023.2276811.

[7] Obinwanne U, Feng W. A Case Study on AI to Automate Simulation Modelling. In Feng W, Rahimi N, Margapuri V, editors. *Software and Data Engineering.* Cham: Springer Nature Switzerland. 2025. 179–186. DOI 10.1007/978-3-031-75201-8_13

[8] Tinsel E-F, Lechler A, Riedel O, Verl A. Concept of an Initial Requirements-Driven Factory Layout Planning and Synthetic Expert Verification for Industrial Simulation Based on LLM. In IEEE, editor. 2024 IEEE 22nd International Conference on Industrial Informatics (INDIN). *2024 IEEE 22nd International Conference on Industrial Informatics*; 2024 Aug; Beijing, China: IEEE. 1–6. DOI 10.1109/INDIN58382.2024.10774366

[9] Lu C, Lu C, Lange RT, Foerster J, Clune J, Ha D. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. 2024. http://arxiv.org/pdf/2408.06292. Accessed 18 Jul 2025.

[10] Yamada Y, Lange RT, Lu C, Hu S, Lu C, Foerster J, et al. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. 2025. https://arxiv.org/pdf/2504.08066. Accessed 18 Jul 2025.

[11] Guo T, Chen X, Wang Y, Chang R, Pei S, Chawla NV, et al. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In Larson K, editor. Proc. of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24; 2024 Aug; Jeju, Korea: International Joint Conferences on Artificial Intelligence Organization. 8048–8057. DOI 10.24963/ijcai.2024/890

[12] Xia Y, Dittler D, Jazdi N, Chen H, Weyrich M. LLM experiments with simulation: Large Language Model Multi-Agent System for Simulation Model Parametrization in Digital Twins. In Facchinetti T, editor. *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2024. 1–4. DOI 10.1109/ETFA61755.2024.10710900

[13] Anthropic. Models overview. 2025. https://docs.anthropic.com/en/docs/about-claude/models/overview. Accessed 18 Jul 2025.

[14] OpenAI. Models: GPT-4.1. 2025. https://platform.openai.com/docs/models/gpt-4.1. Accessed 18 Jul 2025.