Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

BRIDGING EXPERTISE AND AUTOMATION: A HYBRID APPROACH TO AUTOMATED MODEL GENERATION FOR DIGITAL TWINS OF MANUFACTURING SYSTEMS

Lekshmi P.[1], and Neha Karanjkar[1]

[1]School of Mathematics and Computer Science, Indian Institute of Technology Goa, GOA, INDIA

ABSTRACT

We consider the problem of Automated Model Generation (AMG) for Digital Twins of manufacturing systems, particularly those represented as stochastic Discrete-Event Simulation (DES) models. Unlike fully data-driven approaches, where both model parameters and structure are inferred from event logs, we propose an expert-in-the-loop approach targeting systems whose structure changes only occasionally, with such changes being automatically detected. While machine states and parameters (such as task delays) are continuously inferred from data, the process remains expert-tunable via a Graphical User Interface-based, guided flow. A natural language description of system structure is translated into readable DES models using LLMs. The model is built as an interconnection of configurable components from a lightweight, open-source library (*FactorySimPy*), with parameters inferred seamlessly from data. We outline the proposed flow, its components, and results from a proof-of-concept implementation, and provide a detailed review of existing AMG approaches, highlighting key differentiating aspects of our framework.

1 INTRODUCTION

The concept of Digital Twins (DTs) has significantly accelerated the transformation of manufacturing systems by providing virtual replicas that dynamically reflect the state and behavior of real-world production environments (Matta and Lugaresi 2024). The underlying model of a manufacturing DT is typically a stochastic Discrete-Event Simulation (DES) model, where processes and interconnected physical components such as machines and conveyor belts are represented using process-based simulation models, and their parameters (such as task delays) are represented by random variates whose distribution is selected to match observed data from the real system. Traditionally, such DES models have been manually constructed by modeling experts, a method that is labor-intensive, error-prone, and unsuitable for real-time DT applications (Reinhardt et al. 2019a). For building responsive DTs, the underlying model needs to be adaptable, both in terms of its parameters (ensuring continuous synchronization with real-time sensor data) and its structure (accommodating occasional structural changes due to operational needs, such as equipment upgrades or layout modifications). To address these challenges, **Automated Model Generation** (AMG) approaches have emerged, aiming to streamline and accelerate the creation of simulation models directly from available digital data sources. AMG is essential for enabling rapid deployment and real-time responsiveness in Digital Twins, significantly reducing the time required to update simulation models when production environments evolve. The term AMG in manufacturing simulation began appearing prominently in the 2010s as Industry 4.0 and Digital Twin concepts gained traction. While the term encompasses methods to ease or assist model-building (for example, pulling system structure and parameters from digital data such as process plans, or control logic) (Popovics et al. 2012), it also encompasses approaches using algorithmic or AI-based techniques to infer model structure and behavior directly from data (for example, discovering process flow rules from event logs).

Recent developments in AMG predominantly emphasize fully data-driven methodologies, leveraging process mining techniques to infer both the structural and parametric elements directly from operational data, such as event logs (May et al. 2024; Tan et al. 2023; Lugaresi and Matta 2023). Despite their

success in domains such as material flow modeling, purely data-driven approaches face some critical challenges: limited model readability and tuning capabilities, and difficulty in systematically integrating expert knowledge into the automated model generation flow (Lugaresi 2024). Specifically, for manufacturing systems where the constituent components and processes are well-understood, and structural changes are infrequent and largely planned (such as assembly lines or consumer goods production), automating the model generation process is essential, and yet a fully automated inference from event logs might introduce unnecessary complexity and inaccuracies (Behrendt et al. 2025). To address these challenges, we discuss and present a prototype AMG flow that emphasizes model readability, tunability, ease of validation and integration of expert knowledge as central elements of the inference process, leveraging recent advances in LLM-based agentic workflows.

1.1 Guiding Principles and Motivation for an Expert-in-the-Loop Approach

- Expert domain knowledge regarding manufacturing system structure is typically well-established and reliable. For example, factory floor operators are intimately familiar with the factory's layout, machinery configurations, material flow paths, operational constraints, and standard operating procedures, even though they may not have specialized expertise in simulation modeling languages or DES modeling frameworks. Systematically integrating this domain knowledge into the model generation workflow can substantially reduce the complexity and ambiguities associated with purely data-driven approaches.
- Importantly, this integration of expert input should not necessitate specialized expertise in simulation modeling or programming from the user. Recent advancements in Large Language Models (LLMs) have opened new opportunities to translate natural language descriptions directly into formal simulation models (Jackson et al. 2024).
- Despite their potential, LLM-based translations that directly generate DES models may lack robust validation routines and automated sanity checks to ensure that the generated models are both accurate and usable. However, if model generation from coarse descriptions is framed as the **composition of pre-built, configurable components** from an open-source library using well-defined interconnection rules, the resulting models can be more easily validated and are often correct-by-construction.
- Unlike the Petri-net based models generated by process mining based AMG flows, configurable component-based models offer superior readability, extensibility and maintainability, crucial for practical industrial use (Heavey et al. 2014).
- While structural changes in such systems are typically infrequent and intentional, automated detection mechanisms are essential to identify these changes promptly and solicit verified expert input for updating the simulation model accordingly.
- Likewise, the fitting of model parameter distributions to observed data must be continuous and online, while also remaining expert-tunable, and allowing domain experts to guide the choice of distribution families, specify dependencies or correlation structures, and adjust fitting parameters as needed.

1.2 Proposed Expert-in-the-Loop AMG Framework

Motivated by these considerations, we propose an integrated AMG framework with an expert-in-the-loop approach targeted for generating stochastic DES models whose parameters are synched continuously to match real system data. The AMG flow decouples structural modeling, based on expert input about components and their connections, from real-time parameter fitting. The framework comprises three core components: **DataFITR**, **FactoryFlow** and **FactorySimPy** illustrated in Figure 1.

- **DataFITR** is an open-source tool for guided statistical fitting of real-time sensor data to probability distributions. It provides a graphical interface and can generate Python code for random variate

generation within stochastic DES models (Lekshmi et al. 2023). DataFITR is fully implemented and available as a standalone, cloud-hosted tool at DataFITR page (DataFITR 2023).
When used as a component within the AMG framework, it integrates with the other modules to continuously update model parameters based on sensor data.

- **FactorySimPy** is a core component of the AMG flow. It is a lightweight, open-source Python library that provides configurable simulation models for components in manufacturing systems. Systems are described as interconnected components with associated parameters, and the interconnection follows well-defined rules to enable automated validation. Unlike closed-API commercial tools, FactorySimPy features a modular and minimal design, is well documented, and is released under a permissive BSD-style license. This makes it reusable, extensible, and suitable for integration with AMG flows, custom visualizations, optimizers, and other workflows. An initial version is available at GitHub repository (FactorySimPy Repository 2025), with documentation and examples at documentation page (FactorySimPy Documentation 2025).
- **FactoryFlow** is the expert in the loop AMG flow that takes natural language descriptions of system structure and employs LLM-based agentic workflows to generate process-based, readable DES models. The resulting model is a network of parameterized component instances from FactorySimPy. FactoryFlow includes a graphical interface for editing intermediate model representations, refining structure and parameters, and visual validation through block diagram views. It integrates with DataFITR to associate components with inferred parameters by matching component instance names in data headers. A proof-of-concept implementation is available at GitHub repository (FactoryFlow PoC 2025) and is discussed in this work. As part of future work, the framework will be cloud-hosted, with an expanded graphical interface for performing simulations, parameter optimization, and systematic design space exploration.

1.3 Main Contributions and Paper Organization

The primary contributions of this paper are as follows. First, we provide a comprehensive review of the state of the art in AMG, with a focus on generating DES-based Digital Twins for manufacturing systems (Section 2). Second, we present a case for the expert-in-the-loop approach to Automated Model Generation (AMG), emphasizing its distinctions from and advantages over fully data-driven methods, particularly in systems with infrequent but deliberate structural changes (sections 1.1 and 2.5). Third, we describe the proposed AMG framework in detail (Section 3), outlining its guiding principles, modular architecture, and integration of human input with data-driven methods. We also present the implementation and validation of a proof of concept. Additionally, we report initial results from a scalability study (Section 4) and conclude with a summary of the key advantages, distinguishing features and limitations of our approach in general, and the current implementation in particular (Section 5), with a brief discussion of future work.

2 A REVIEW OF AUTOMATED MODEL GENERATION

Automatic Model Generation (AMG) encompasses methods to automatically derive simulation or digital-twin models from operational data, metadata, and design specifications, reducing manual modeling effort. Model generation has historically been a critical and time-consuming process, encompassing multiple iterations of data collection, input modeling, model specification and its translation to executable models and validation, requiring specific expertise. Recent trends such as Industry 4.0 and the opportunity to utilize real-time sensor data to drive Digital Twins demands rapid, adaptive, and resilient model creation for decision support and optimization. AMG flows utilize simulator interfaces and algorithmic methods to streamline and expedite model creation (Bergmann and Strassburger 2010; Fowler and Rose 2004). AMG has been successfully implemented across a diverse range of target model types (including Markov chains, Petri nets, and machine learning models), drawing inputs from a range of data sources (such as ERP systems, routing information, event logs, sensor and PLC data for machine states), and employing

techniques such as process mining and machine learning to infer system behavior and structure. This review focuses on discrete-flow manufacturing systems, such as assembly lines or consumer goods production facilities. Such systems exhibit clearly defined processes with stable structural configurations and infrequent operational changes, making them ideal for human-in-the-loop AMG approaches. The review focuses on discrete-event simulation (DES)-based Digital Twins (DTs) of manufacturing systems, emphasizing key dimensions such as adaptability, validation, modularity, and the required degree of human interaction. Literature from reputable conferences (e.g., Winter Simulation Conference, SIGSIM-PADS) and journals (e.g., Computers in Industry, IJPR) published predominantly post-2015 is reviewed.

AMG frameworks have found applications across manufacturing (Reinhardt et al. 2019b; Psarommatis and May 2023; Liu et al. 2019), logistics and supply chains (Ivanov 2024; Jackson et al. 2024), automotive and transportation (Huang et al. 2011), and dynamic industrial processes (Tan et al. 2023). Schwede and Fischer (2024) presents a review of the advances in simulation-based DTs in a discrete material flow system. AMG workflows reported in the literature can be broadly categorized into two classes: (1) purely data-driven approaches, which include process mining methods that rely on event logs, and machine learning-based techniques that infer machine states and behavior from sensor data; (2) hybrid or expert-in-the-loop approaches, which combine automated inference with domain expert input for improved interpretability and control. These include emerging methods based on Natural Language Processing (NLP) and Large Language Models (LLMs), which aim to generate model structure and parameters from coarse descriptions.

2.1 ML and Process Mining-based Methods

Process mining emerges prominently in AMG methods, facilitating automatic model generation by leveraging operational data. Lugaresi and Matta (2021a) propose a framework for automatic DES model construction from event logs, inferring both structural and parametric model elements. Lugaresi (2024) discuss methodological challenges and research directions within process mining. Similarly, May et al. (2024) combine process mining with machine learning and deep learning to infer resource allocation and sequencing logic. Friederich et al. (2022) introduce a conceptual framework integrating process mining with machine learning techniques to create Petri-net simulation models semi-automatically, requiring manual event labeling. Adaptive methods addressing evolving systems are relatively uncommon but critical. Tan et al. (2023) utilize timed-event logs to continuously adapt Petri-net models, employing data assimilation methods to discard outdated structures.

2.2 NLP and LLM-based Approaches

LLMs have recently emerged as promising tools for enhancing various stages of the AMG workflow. Jackson et al. (2024) propose an NLP-driven approach that translates textual system descriptions into simulation models, with final validation performed against real system behavior. Giabbanelli (2023) survey the potential roles of LLMs in simulation workflows, including the generation of conceptual models, summarization of simulation results, visualization narration, and debugging. Hattori et al. (2024) explore LLM integration within multi-agent simulation contexts, showcasing their utility in configuring agent behaviors through natural language. Carreira-Munich et al. (2024) introduce a two-stage LLM-assisted framework that transforms high-level natural language descriptions into executable DEVS models in PythonPDEVS.

2.3 Hybrid and Expert-in-the-loop Methods

Overbeck et al. (2024) propose adaptive simulation frameworks that automatically detect structural and parametric changes, necessitating human intervention only during discrepancies. Although the significance of validation in AMG has been highlighted in several flows (Overbeck et al. 2024; Lugaresi et al. 2023; Onggo and Currie 2024; Hua et al. 2022; Lugaresi and Matta 2021a), integrating human validation and input systematically at each step remains challenging in such flows. Expert input may be integrated only

during initial stages, such as event labeling (Friederich et al. 2022) or parameter specification (Krenczyk et al. 2016; Vaidya et al. 2024). GUI-based, and cloud-hosted AMG solutions are less common. One such example is proposed by Heavey et al. (2014) to utilize ERP and MES data to generate DES models using the SimPy-based ManPy library. However, this work does not address dynamic parametric or structural adaptability required for DTs.

Systematically integrating expert input, generating readable, maintainable and extensible model descriptions, and real-time adaptivity of the model are oftentimes conflicting requirements. Considering a subset of applications in order to make simplifying assumptions can be the key to addressing this challenge. For example, (Novák and Vyskočil 2022) suggests a decoupling of parameter extraction and model description; using a formal representation and static model of production operations while parameters such as operation duration times are acquired through process mining techniques. A similar strategy is employed the AMG flow we propose in this paper that decouples the structural modeling (which is eased using LLMs), and parameter fitting (which is real-time, and yet expert tunable through a systematic GUI), whist generating highly readable and self-validating models through an open component library (FactorySimPy).

2.4 Other Relevant Aspects of AMG

1. **Open-Source and Commercial Simulation Platforms:** Several frameworks depend on commercial simulation software like FlexSim (Burnett et al. 2008) or Arena (May et al. 2024). Platforms such as Tecnomatix and MindSphere also provide visual feedback and system monitoring (Overbeck et al. 2024; Lugaresi et al. 2022). AMG workflows reported in recent literature leverage either commercial (e.g., FlexSim, Arena, AnyLogic) or open-source (e.g., ManPy, Salabim, JaamSim) simulation tools. Commercial solutions offer user-friendly graphical interfaces but are often limited by closed or custom APIs (in contrast to popular programming languages for scripting or model generation) and can be expensive, offering limited opportunity for customization, and integration with automated workflows. Open-source tools provide flexibility but often lack application-specific component libraries (Huang et al. 2011).
2. **GUI-based and Cloud-hosted Flows:** Few AMG frameworks offer fully cloud-hosted workflows. Heavey et al. (2014) and Vaidya et al. (2024) (AWS-based) represent rare examples. Many GUI tools are primarily offline utilities aiding visualization rather than comprehensive modeling platforms (Burnett et al. 2008).

2.5 Comparison Between Data-driven and Expert-In-The-Loop Flows

The primary distinction between fully **data-driven** approaches (particularly process-mining based methods that generate DEVS or Petri-Net models) and **Expert-In-The-Loop (EITL)** methods such as ours lies in their underlying assumptions, application suitability, and resulting model characteristics.

- **Suitability:** Data-driven approaches are well-suited when the manufacturing environment is highly dynamic. Examples include job shops, reconfigurable cells, or make-to-order lines where product mix, routing, and resource states change unpredictably. In such contexts, event logs allow process-mining algorithms to discover emergent flow paths, update resource allocation rules on the fly, and keep the model synchronized with reality. When the plant layout is stable, such as assembly lines, packaging lines, or semiconductor fabs, repeatedly rediscovering an essentially fixed topology becomes wasteful. EITL flows use expert knowledge (for example, the station list or conveyor graph) and automation focuses on parametric calibration, detection of change and incremental updates. This approach lowers data-collection overhead, shortens model-generation time, and avoids fluctuation-induced noise. In systems where equipment layout resembles a mesh or line, process mining spends compute to rediscover obvious routes. Expert specification removes this redundancy, focusing automation on parameter fitting and stochastic calibration rather than topology discovery. Case studies show that structure inference time drops significantly when expert

templates are used, with no loss in KPI fidelity (Cimino et al. 2025; Heavey et al. 2014). However EITL flows may be a poor choice for systems where routing changes frequently and dynamically as expert input will be sought each time a structural change is detected.

- **Model Readability, Maintainability, and Tunability:** Data-driven methods typically generate Petri-net, Markov Chain, DEVS or similar formal models, resulting in opaque, difficult-to-interpret structures. Complex data-driven models frequently require additional manual tuning to mitigate excessive expressiveness, often labelled the "spaghetti effect" (Castiglione 2024; Lugaresi and Matta 2020). EITL can produce modular, structured simulation code that domain engineers can readily inspect and maintain. These models are inherently more tunable and less prone to redundant complexity.
- **Ease of Model Validation:** In data-driven methods that generate formal models such as Petri-Nets, Markov Chains or DEVS, outlier events, insufficient event logs or dynamic changes in the system structure can lead to models containing loops, deadlocks or livelocks that are extremely difficult to detect and require post-simulation debugging. FactoryFlow generates models as a composition of pre-validated configurable components. Deadlocks in the generated model are still possible with certain system descriptions, but easier to spot, visualize and rectify. Human checkpoints built into EITL flows detect and correct structural or parametric anomalies early.
- **Data Requirements:** Data-driven methods often require large, high-resolution logs for discovering the entire structure and state-space of the system. An example of an event log required for model generation is present in (Bayomie et al. 2022). Model accuracy can degrade when logs are sparse or noisy. EITL such as FactoryFlow decouple structural description and fitting of individual parameters. Parameters associated with longer data streams will be fitted with greater accuracy. The data streams can be much smaller, targeted for unknown or dynamically changing aspects of the system.
- **Recognition of Production Policies and Patterns:** Data-driven methods struggle to capture scheduling rules such as *first-available* or *round-robin* dispatching which statistically look similar to a uniform discrete distribution, and yet lead to significantly different model behavior (Lugaresi 2024). Consequently, operational nuances critical to realistic behaviour may be missed. Manual and hybrid flows such as FactorySimPy formalises resources and operations explicitly.
- **User Interface:** Many data-driven pipelines run offline in batch mode and offer limited interactive refinement. EITL flows depend heavily on an intuitive, interactive GUI where engineers iteratively adjust structure and parameters and receive immediate feedback. The quality of the interface can often be a bottleneck in the effectiveness of such a flow.

Fully data-driven AMG is most effective for data-rich systems that undergo frequent, unpredictable routing changes, while EITL flows are better suited to stable plants where structure changes rarely and transparency, tunability, and low modelling overhead are the primary concerns. Hybrid flows meet the central challenge of creating adaptable, stochastic DES models for manufacturing digital twins by combining automated parameter inference with domain expertise. A current gap exists for GUI-based tools that can integrate expert-input with real-time parameter-inference, using open-source component libraries.

3 FACTORYFLOW

In this section, we describe how the key design principles underlying our expert-in-the-loop approach outlined in Section 1.1 are operationalized in the design and implementation of the *FactoryFlow* framework. The FactoryFlow framework consists of the following main components, introduced in Section 1.2 and illustrated in Figure 1:

- **DataFITR** is responsible for guided parameter fitting and input modeling, from real-time streaming data. It can automatically infer input distributions (univariate as well as joint), estimate their parameters and correlation structures, and generate Python routines for random variate generation

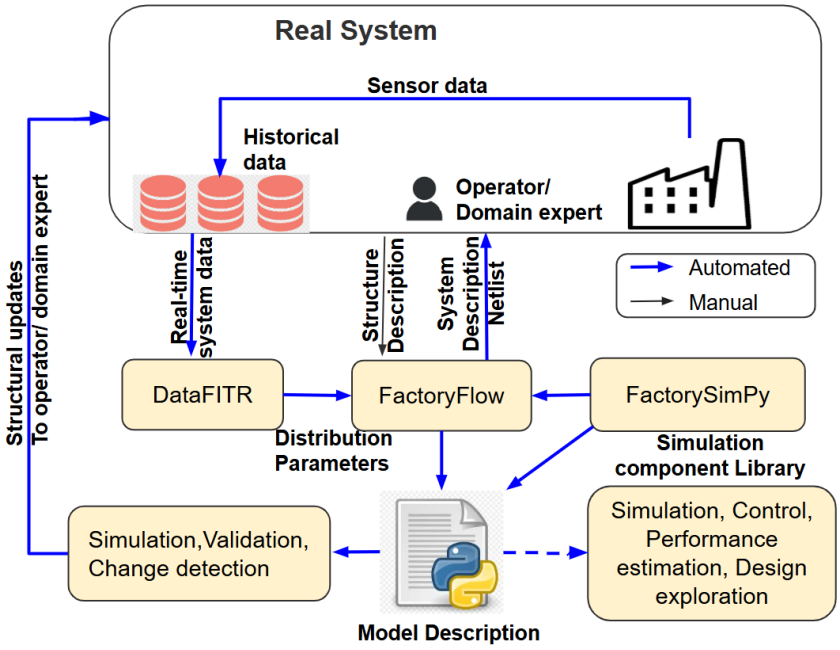

Figure 1: Core components and their integration in FactoryFlow.

to be used within a stochastic DES model. A standalone cloud-hosted version, with a Streamlit-based GUI interface is freely available. The current implementation accepts historical data in CSV format. Integration with real-time databases for supporting streaming data is ongoing.

- **FactorySimPy** is a lightweight, open-source component-based simulation library for manufacturing systems built using SimPy. It is minimal, well-documented and provides pre-validated and configurable models of components (such as machines, splits, conveyor belts and other item transportation mechanisms) that can be connected together using well-defined rules to generate correct-by-construction models.
- **FactoryFlow** is the overall AMG flow proposed in this work. A proof-of-concept implementation is available as open-source and is described here. It uses LLMs to translate coarse structural descriptions (in plain english) to executable models described as an interconnection of parameterized components from FactorySimPy. The parameter values are automatically inferred by DataFITR and associated with component instances using their IDs (instance names) from the data stream. The model descriptions are highly readable. A screenshot of the GUI provided in the PoC is shown in Figure 2. A block diagram is also generated to enable user validation and correction. The GUI supports direct editing of the generated model at each intermediate step. The current implementation uses Gemini 2.5 Pro model accessed via API calls, with custom prompt strategies to extract component and connection information from natural language.
- **FactoryFlowSim** is a guided, GUI-based flow for simulation, optimization and design-space exploration which is planned to be implemented and integrated with FactoryFlow. Users can either download the simulation model for local execution or run it in the cloud by specifying parameters like run length or termination conditions. It also supports visual validation and operator feedback loops. The module compares key performance indicators (KPIs) such as cycle time, resource utilization, and throughput between those generated by the simulation and the ones obtained from real system data. Deviations beyond thresholds trigger an alert for expert input. This module, and a guided GUI interface is currently under development.

Figure 2: A Screenshot of the PoC Implementation of FactoryFlow.

4 VALIDATION AND SCALABILITY

We present results of an initial validation and scalability study for the PoC implementation. The study aims to understand current issues, accuracy and extent of KPI match for the generated models and to project how the effort (both, human and automation) would scale with model size, complexity and number of parameters. To do this, we consider **three reference systems** of increasing size and complexity:

- **Model A** represents a small model, and is a linear sequence of 10 machines connected over buffers. The individual machines have diverse distribution types and parameter values for their task delays, work capacities, output port selection policies and also the number of buffers connected in parallel between them.

- **Model B** represents a model that is larger (100s of machines) but homogenous (low complexity). It simply consists of 10 parallel instances of the linear sequence of 10 machines (similar to model A) connected through a common source and sink. Although the parallel sequence instances are structurally similar, their parameter values are different and unique.
- **Model C** represents a model that is both large and complex. It has the same number of machines (and parameters) as Model B but a non-regular, heterogenous interconnection structure that merits a longer system description.

For each reference model, we build a ground-truth simulator that generates synthetic sensor data. DataFITR uses these data streams and coarse textual descriptions from FactoryFlow to infer parameters and create executable models in FactorySimPy, denoted **A_gen**, **B_gen** and **C_gen**. KPI accuracy is assessed by comparing matched distribution parameters and KPI values for each pair (A versus A_gen, B versus B_gen, C versus C_gen) and the results are summarised in Table 1. The KPIs show good agreement indicating an acceptable level of accuracy in the generated models. The same reference models were used to understand and quantify scaling of human effort, processing time, and description length (see Table 1). Decoupling structural description from parameter fitting lets the flow scale to large systems with regular topologies, while fitting cost remains proportional to the number of parameters. The trends in the time required by an expert for model description and that taken by the flow highlight the distinct advantage of using an AMG flow in contrast to manual model creation using simulation tools or programming that can take hours to days.

5 CONCLUSIONS, LIMITATIONS AND FUTURE WORK

Automated Model Generation (AMG) is set to play a critical role as an enabler for Digital Twins of manufacturing systems, particularly those using stochastic Discrete-Event Simulation (DES) models. Traditional manual modeling approaches are too slow and rigid to support the adaptability and responsiveness that Digital Twins require. At the same time, fully data-driven AMG approaches face challenges related to the systematic integration of expert knowledge, data availability and the readability and extensibility of the generated models. This is particularly relevant for systems where structural changes are infrequent and deliberate, such as in production assembly lines. This paper made the case for a hybrid, expert-in-the-loop approach to Automated Model Generation (AMG), and presented an outline and proof-of-concept

Table 1: A summary of the validation and scalability results. Details, model descriptions, and full comparison results are available at GitHub repository (FactoryFlow PoC 2025).

		Model A	**Model B**	**Model C**
Model size and complexity	Num of machines	10	100	100
	Num of connections	14	140	145
	Model parameters	20	200	200
	System description size (characters)	603	1087	1616
Modeling Effort (human) approximate time in minutes	System description	5-6	10-12	14-15
	DataFITR guided parameter fitting	2-3	2-3	2-3
	Model visual validation	2-3	4-5	8-9
Automated Model Generation Effort (time in minutes)	Parameter fitting delay (DataFITR)	< 1	4.45	4.98
	LLM based translation time in FactoryFlow (including API calls delay)	< 1	4.98	5.6
	Input token size (including prompt)	900	2594	2789
	Output token size	8393	20509	14425
Simulation Execution Time in seconds (for simulation time of 10^4)		48.97	429.15	348.45
Model Generation Accuracy (relative error between the KPIs generated by simulating the reference model and the generated model, for simulation time=10^4)	Avg system throughput (items processed per second)	0.25 %	0.12%	2.40%
	Avg cycle time (averaged across all items)	0.60%	0.49%	3.60%
	Total num of items processed	0.25%	0.12%	2.40%
	Avg time spent by M5 in processing state	0.41%	1.02%	2.60%
	Time avg occupancy of Buffer B_4_5	4.20%	1.20%	0%

implementation of FactoryFlow, a framework we have developed that embodies this idea. The framework consists of three key modules: **DataFITR**, for guided, online statistical modeling of system parameters; **FactoryFlow**, an LLM-based module that translates natural language descriptions into formal model structures; and **FactorySimPy**, a lightweight, open-source simulation library used by FactoryFlow to assemble models through rule-based interconnections of configurable components. The resulting models are modular, human-readable, and easily extensible, while being adaptive to real-time sensor data via on-line parameter fitting.

While the proposed framework demonstrates the effectiveness and utility of an expert-in-the-loop approach, there are certain limitations, both in the current implementation and in the approach itself. From an implementation standpoint, the framework currently supports only historical data via CSV uploads for parameter inference; integration with streaming data sources is ongoing. FactoryFlow depends on commercial LLM APIs (specifically Gemini 2.5 Pro), which may restrict access and reproducibility. Updates or changes to the LLM by the provider can result in glitches and a need for re-tuning the prompt. This can be overcome by transitioning to open-source, locally hosted LLM models such as LLaMA to offer competitive performance while eliminating licensing fees, and is planned as future work. FactorySimPy presently supports only discrete item flows (in contrast to material or fluid flows). Support for material flows (using hybrid simulation) and more complex behaviors is planned. Furthermore, while a web interface exists for parameter fitting and model generation, the simulation and design exploration interface (FactoryFlowSim) is still in progress and not fully integrated into the pipeline. From a broader methodological perspective, the framework requires expert input for describing structural changes in the system. While this supports flexibility and interpretability, it is suited for systems with largely static structures. Additionally, extending the component library in FactorySimPy or introducing new system behaviors requires expertise in simulation modeling and Python development, which may not be available to all users. These challenges are being addressed to improve accessibility, automation, and robustness of the framework in future iterations.

In summary, this work demonstrated a viable alternative to fully data-driven AMG approaches by combining human expertise, structured modeling abstractions, and advances in LLM-based AI tools to produce interpretable, adaptive, and validated DES models for manufacturing Digital Twins.

REFERENCES

Bayomie, D., K. Revoredo, S. Bachhofner, K. Kurniawan, E. Kiesling, and J. Mendling. 2022. "Analyzing Manufacturing Process by Enabling Process Mining on Sensor Data". In *Workshop of the Practice of Enterprise Modelling, PoEM 2022, November 23-25*. London, UK.

Behrendt, S., T. Altenmüller, M. C. May, A. Kuhnle, and G. Lanza. 2025. "Real-To-Sim: Automatic Simulation Model Generation for a Digital Twin in Semiconductor Manufacturing". *Journal of Intelligent Manufacturing*:1–20.

Bergmann, S., and S. Strassburger. 2010. "Challenges for the Automatic Generation of Simulation Models for Production Systems". In *Proceedings of the 2010 Summer Computer Simulation Conference*, SCSC10, 545–549. San Diego, CA, USA: Society for Computer Simulation International.

Burnett, G. A., D. J. Medeiros, D. A. Finke, and M. T. Traband. 2008. "Automating the Development of Shipyard Manufacturing Models". In *2008 Winter Simulation Conference (WSC)*, 1761–1767 https://doi.org/10.1109/WSC.2008.4736264.

Carreira-Munich, T., V. Paz-Marcolla, and R. Castro. 2024. "DEVS Copilot: Towards Generative AI-Assisted Formal Simulation Modelling based on Large Language Models". In *2024 Winter Simulation Conference (WSC)*, 2785–2796 https://doi.org/10.1109/WSC63780.2024.10838994.

Castiglione, C. 2024. "Automated Generation of Digital Models for Manufacturing Systems: The Event-centric Process Mining Approach". *Computers and Industrial Engineering* 197:110596.

Cimino, A., M. Elbasheer, F. Longo, G. Mirabelli, V. Solina, and P. Veltri. 2025. "Automatic Simulation Models Generation in Industrial Systems: A Systematic Literature Review and Outlook towards Simulation Technology in the Industry 5.0". *Journal of Manufacturing Systems* 80:859–882.

DataFITR 2023. "A Tool for Input Modeling". Accessed 09th September.

FactorySimPy Documentation 2025. "Documentation of the Package with Examples". Accessed 09th September.

Fowler, J., and O. Rose. 2004. "Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems". *Simulation* 80:469–476.

Friederich, J., D. P. Francis, S. Lazarova-Molnar, and N. Mohamed. 2022. "A Framework for Data-Driven Digital Twins of Smart Manufacturing Systems". *Computers in Industry* 136:103586.

Giabbanelli, P. J. 2023. "GPT-Based Models Meet Simulation: How to Efficiently use Large-Scale Pre-Trained Language Models Across Simulation Tasks". In *2023 Winter Simulation Conference (WSC)*, 2920–2931 https://doi.org/10.1109/WSC60868.2023.10408017.

Hattori, H., A. Kato, and M. Yoshizoe. 2024. "Integrating Large Language Models into Agent Models for Multi-Agent Simulations: Preliminary Report". In *2024 Winter Simulation Conference (WSC)*, 230–241 https://doi.org/10.1109/WSC63780.2024.10838923.

Heavey, C., G. Dagkakis, P. Barlas, I. Papagiannopoulos, S. Robin, M. Mariani *et al*. 2014. "Development of an Open-source Discrete Event Simulation Cloud Enabled Platform". In *2014 Winter Simulation Conference (WSC)*, 2824–2835 https://doi.org/10.1109/WSC.2014.7020124.

Hua, E. Y., S. Lazarova-Molnar, and D. P. Francis. 2022. "Validation of Digital Twins: Challenges and Opportunities". In *2022 Winter Simulation Conference (WSC)*, 2900–2911 https://doi.org/10.1109/WSC57314.2022.10015420.

Huang, Y., M. D. Seck, and A. Verbraeck. 2011. "From Data to Simulation Models: Component-based Model Generation with a Data-driven Approach". In *2011 Winter Simulation Conference (WSC)*, 3719–3729 https://doi.org/10.1109/WSC.2011.6148065.

Ivanov, D. 2024. "Conceptualisation of a 7-element Digital Twin Framework in Supply Chain and Operations Management". *International Journal of Production Research* 62(6):2220–2232.

Jackson, I., M. J. Saenz, and D. Ivanov. 2024. "From Natural Language to Simulations: Applying AI to Automate Simulation Modelling of Logistics Systems". *International Journal of Production Research* 62(4):1434–1457.

Krenczyk, D., B. Skolud, and M. Olender-Skóra. 2016. "Semi-automatic Simulation Model Generation of Virtual Dynamic Networks for Production Flow Planning". *IOP Conference Series: Materials Science and Engineering* 145.

Lekshmi, P., N. Karanjkar, and T. Lone. 2023. "DataFITR: An Open, Guided Input Modeling Tool for Creating Simulation-Based Digital Twins". In *Proceedings of the 13th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2023, Rome, Italy, July 12-14, 2023*, edited by G. Wagner, F. Werner, and F. D. Rango, 279–286: SCITEPRESS.

Liu, Q., H. Zhang, J. Leng, and X. Chen. 2019. "Digital Twin-driven Rapid Individualised Designing of Automated Flow-shop Manufacturing System". *International Journal of Production Research* 57(12):3903–3919.

Lugaresi, G. 2024. "Process Mining as Catalyst of Digital Twins for Production Systems: Challenges and Research Opportunities". In *2024 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC63780.2024.10838896.

Lugaresi, G., S. Gangemi, G. Gazzoni, and A. Matta. 2022. "Online Validation of Simulation-Based Digital Twins Exploiting Time Series Analysis". In *2022 Winter Simulation Conference (WSC)*, 2912–2923 https://doi.org/10.1109/WSC57314.2022.10015346.

Lugaresi, G., S. Gangemi, G. Gazzoni, and A. Matta. 2023. "Online Validation of Digital Twins for Manufacturing Systems". *Computers in Industry* 150:103942.

Lugaresi, G., and A. Matta. 2020. "Generation and Tuning of Discrete Event Simulation Models for Manufacturing Applications". In *2020 Winter Simulation Conference (WSC)*, 2707–2718 https://doi.org/10.1109/WSC48552.2020.9383870.

Lugaresi, G., and A. Matta. 2021a. "Automated Manufacturing System Discovery and Digital Twin Generation". *Journal of Manufacturing Systems* 59:51–66.

Lugaresi, G., and A. Matta. 2023. "Automated Digital Twin Generation of Manufacturing Systems with Complex Material Flows: Graph Model Completion". *Computers in Industry* 151:103977.

Matta, A., and G. Lugaresi. 2024. "An Introduction to Digital Twins". In *2024 Winter Simulation Conference (WSC)*, 1281–1295 https://doi.org/10.1109/WSC63780.2024.10838793.

May, M. C., C. Nestroy, L. Overbeck, and G. Lanza. 2024. "Automated Model Generation Framework for Material Flow Simulations of Production Systems". *International Journal of Production Research* 62(1-2):141–156.

Novák, P., and J. Vyskočil. 2022. "Digitalized Automation Engineering of Industry 4.0 Production Systems and their Tight Cooperation with Digital Twins". *Processes* 10(2):404.

Onggo, B. S., and C. S. M. Currie. 2024. "Extending Simulation Modeling Methodology for Digital Twin Applications". In *2024 Winter Simulation Conference (WSC)*, 3058–3069 https://doi.org/10.1109/WSC63780.2024.10838633.

Overbeck, L., S. C. Graves, and G. Lanza. 2024. "Development and Analysis of Digital Twins of Production Systems". *International Journal of Production Research* 62(10):3544–3558.

FactoryFlow PoC 2025. "GitHub Repository". Accessed 09th September.

Popovics, G., A. Pfeiffer, B. Kádár, Z. Vén, L. Kemény, and L. Monostori. 2012. "Automatic Simulation Model Generation Based on PLC Codes and MES Stored Data". *Procedia CIRP* 3:67–72.

Psarommatis, F., and G. May. 2023. "A Literature Review and Design Methodology for Digital Twins in the Era of Zero Defect Manufacturing". *International Journal of Production Research* 61(16):5723–5743.

Reinhardt, H., M. Weber, and M. Putz. 2019a. "A Survey on Automatic Model Generation for Material Flow Simulation in Discrete Manufacturing". *Procedia CIRP* 81:121–126.

Reinhardt, H., M. Weber, and M. Putz. 2019b. "A Survey on Automatic Model Generation for Material Flow Simulation in Discrete Manufacturing". *Procedia CIRP* 81:121–126. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.

FactorySimPy Repository 2025. "GitHub Repository". Accessed 09th September.

Schwede, C., and D. Fischer. 2024. "Learning Simulation-Based Digital Twins for Discrete Material Flow Systems: A Review". In *2024 Winter Simulation Conference (WSC)*, 3070–3081 https://doi.org/10.1109/WSC63780.2024.10838729.

Tan, W. J., M. G. Seok, and W. Cai. 2023. "Automatic Model Generation and Data Assimilation Framework for Cyber-Physical Production Systems". In *Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM-PADS '23, 73–83. New York, NY, USA: Association for Computing Machinery.

Vaidya, R., A. Mittal, and G. Nanaware. 2024. "Cloud based Simulation Platform (CSP): A Novel Way to Democratize Simulation Based Experimentation". In *2024 Winter Simulation Conference (WSC)*, 1446–1456 https://doi.org/10.1109/WSC63780.2024.10838992.

AUTHOR BIOGRAPHIES

LEKSHMI P is a Ph.D. candidate in the School of Mathematics and Computer Science at the Indian Institute of Technology Goa (IIT Goa). Her areas of interest include discrete-event simulation and Digital Twins. Her email address is lekshmi20231101@iitgoa.ac.in.

NEHA KARANJKAR is an Assistant Professor in the School of Mathematics and Computer Science at the Indian Institute of Technology Goa (IIT Goa). Her research interests include discrete-event simulation, parallel simulation and hybrid (mixed discrete-continuous) simulation. She is an IEEE senior member and has served as a member of the ACM India Education Committee. Her email address is nehak@iitgoa.ac.in and her website is https://nehakaranjkar.github.io.