*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.*

# SYNERGIC USE OF MODELING & SIMULATION, DIGITAL TWINS AND LARGE LANGUAGE MODELS TO MAKE COMPLEX SYSTEMS ADAPTIVE AND RESILIENT

Souvik Barat[1], Dushyanthi Mulpuru[1], Himabindu Thogaru[1], Abhishek Yadav[1], and Vinay Kulkarni[1]

[1]Tata Consultancy Services Research, Pune, INDIA

## ABSTRACT

Modeling and Simulation (M&S) has long been essential for decision-making in complex systems due to its ability to explore strategic and operational alternatives in a risk-free manner. The emergence of Digital Twins (DTs) has further enhanced this by enabling real-time bidirectional synchronization with physical systems. However, constructing and maintaining accurate and adaptive models and DTs remains time- and resource-intensive and requires deep domain expertise. In this paper, we introduce an adaptive Decision-Making Framework (DMF) that integrates Large Language Models (LLMs) and Model-Driven Engineering (MDE) into the M&S and DT pipeline. By leveraging LLMs as proxy experts and synthesis aids, and combining them with MDE to improve reliability, our framework reduces manual effort in model construction, validation and decision space exploration. Here, we present our approach of DT construction and discuss how it improves agility, reduces expert dependency and can act as a pragmatic aid for making enterprise robust, resilient and adaptive.

## 1 INTRODUCTION

Modeling and Simulation (M&S) has played a foundational role in decision-making for several decades by offering a systematic way to predict system behavior under diverse scenarios and providing a risk-free experimentation environment for evaluating the efficacy of strategic and operational changes prior to implementation. The potential of M&S has been extensively exploited in domains, such as aerospace (Balaban et al. 2009) and advanced manufacturing (Jain et al. 2001), particularly where: (a) physical trials are expensive or impractical and (b) systems can be sufficiently represented using analyzable models and mathematical equations. The evolution of Digital Twins (DTs) has further expanded the capabilities of traditional M&S (Boschert and Rosen 2016) over the past decade by establishing bidirectional links between digital models and their physical counterparts as discussed by (Tao et al. 2019). By providing faithful representations of real systems and seamless data synchronization, DTs enable continuous monitoring, predictive diagnostics, operational optimization and the exploration of design alternatives.

Despite their clear potential, both traditional M&S and DTs face inherent challenges in enterprises and social systems that exhibit techno-socioeconomic characteristics (Barat et al. 2022), including system-of-systems complexity, nonlinearity, uncertainty and emergence. Developing simulatable models for these systems is time-consuming, effort-intensive and highly reliant on domain expertise (Kritzinger et al. 2018). Moreover, enterprises operating in dynamic environments require not only optimization but also adaptability and resilience to internal and external changes - this implies that their digital representations, whether models or DTs, must also be adaptive in addition to faithful.

Through our earlier work applying M&S and DTs to a wide range of enterprise and societal problems, we have found that industry is increasingly seeking a surrogate environment that is capable of closely observing the system and providing context-sensitive recommendations while acting as decision aids for evaluating strategic and operational alternatives. Furthermore, these surrogate environments must adapt as enterprises and their environments evolve over time.

979-8-3315-8726-0/25/$31.00 ©2025 IEEE 

We propose a Decision-Making Framework (DMF) that helps construct enterprise models and DTs, utilize the constructed DTs and adapt to changing conditions with reduced effort and excessive dependency on Subject Matter Experts (SMEs) by synergistically integrating Large Language Models (LLMs) and Model-Driven Engineering (MDE) into the M&S and DT pipeline. LLMs, with their pretrained knowledge and cross-domain generalization abilities, can act as proxy domain experts during the modeling and validation phases, as discussed in (Barat et al. 2025). Moreover, LLMs can serve as synthesis aids for decision-makers to help navigating meaningful alternatives during the simulation phase. By combining MDE with LLMs (Kulkarni et al. 2023), we address key LLM limitations, such as hallucination, making them more dependable. This synergy makes the construction and validation steps more agile, allowing for rapid model and DT generation or adaptation. We also introduce a goal-driven LLM-based decision space navigation mechanism that produces recommendations without any training, thus addressing the limitations of both traditional M&S and traditional AI-based control approaches, such as reinforcement learning (RL).

In this paper, we argue that our approach not only enhances the utility of M&S and DTs but also contributes to the development of more agile and responsive decision-making aid for complex systems including enterprises. The rest of the paper is organized as follows: Section 2 motivates the need for agile and cost- effective modeling in M&S, reduced dependency on domain experts and stakeholders and better support for decision space exploration. We justify our argument by discussing the desired decision-making properties of contemporary enterprises, the state-of-the-art decision-making processes, highlighting their challenges and sharing our experience using M&S in enterprise decision-making. Section 3 discusses our hypothesis on the use of LLMs in the M&S workflow, and Section 4 presents a brief literature overview on LLM applications in M&S. Section 5 outlines our proposed approach, and Section 6 discusses its application to an industry-scale problem along with early observations before concluding with directions for future work.

## 2 ENTERPRISE DECISION-MAKING: ROBUSTNESS, RESILIENCE & ADAPTABILITY

Generalized enterprise decision-making processes that conceptually align with the foundational work of (Simon et al. 1972), can be represented using three core concepts: Goals or objectives ($G$), measures or key performance indicator ($M$) and a set of controllable change in Enterprise $E$ in terms of actions or interventions, denoted as levers ($L$). A decision-making problem is, therefore, can be visualized as an optimization function describing goals $G$ over a set of measures $M$, where $M$ are projections of observable state variables values $SV_E$ of the enterprise $E$, i.e., $M \subset Proj(SV_E)$. A decision-making process aim to find a set of levers ($l^* \in L$) such that, each $m_i \in M$ must be either maximized or minimized based on the desired goals $G$. Thus, a decision-making activity can be expressed as:

$$\text{Find } l^* \in L \text{ such that min/ max } G(M(l^*(E))), \text{ where } M(l^*(E)) = m_i(l^*(E)) : m_i \in M \tag{1}$$

However, optimizing $G(M)$ for all $m_i \in M$ over a time period t is a hard problem for enterprises because the internal system $E$ and its surrounding environment $S$ evolve over time based on multiple internal and external non-deterministic factors or uncertainties $\Omega$. Consequently, enterprises often pursue *satisficing criteria*, introduced by (Simon et al. 1972), rather than strict optimization. In satisficing, the enterprise aims first to ensure that all $m_i \in M$ achieve threshold levels $\theta_i$ under all circumstances $\omega \in \Omega$, before attempting further optimization, therefore decision-making in the face of uncertainty can be framed as,

$$\begin{gathered}\text{Find } l^* \in L, \text{ such that } m_i(l^*(E), \omega) \geq \theta_i, \forall i, \forall \omega \in \Omega \\ \text{then min/ max } G(M(l^*)) \text{ subject to the above constraints over } \Omega\end{gathered} \tag{2}$$

**Robustness in Decision-Making:** Robustness of an enterprise refers to its ability to continuously achieve goals $G$ over future states or circumstances, i.e., Selected levers $l^* \in L$ ensure that $G(M(l^*(E), \omega(T)))$ stays within acceptable bounds, i.e., satisficing criteria defined in Equation 2, for all plausible $\omega(T) \in \Omega_{(Current \cup Future)}$, where $T$ is time span from current to definite future.

**Resilience in Decision-Making:** Resilience emphasizes the ability of an enterprise to recover after a disruption, i.e., a deviation from $m_i(l^*, \omega\ ) \geq \theta_i$ criteria. It can be characterized through a time-indexed recovery of $M$:

$$\exists t_1 \leq t_{disruption} + \triangle t, \text{ such that } m_i(l^*(E), \omega) \geq \theta_i, \forall i, \forall \omega \in \Omega, \quad (3)$$

where $\triangle t$ denotes an acceptable recovery period after disruption $t_{disruption}$.

**Adaptability in Decision-Making:** Adaptability extends robustness and resilience by emphasizing the proactive adjustment of enterprise interventions $l_t \in l^*$ at time $t$ based on the plausible evolution of the environment $\omega(T) \in \Omega_{(Current \cup Future)}$.

$$\text{Find } l^*(t) \in L \text{ such that } G(M(l_t(E), \omega(T))) \text{ satisfies Equation 2 over time span } T. \quad (4)$$

This requires continuous monitoring of $M$ and assessment of plausibility $\Omega$ over time to adjust $l^*(t)$ of Equation 4.

### 2.1 M&S and Digital Twin in Decision Making

Ensuring robustness and resilience while continuously adapting enterprises for emerging situation is challenging for contemporary enterprises. Computationally, it involves multi-objective optimization under uncertainty over a dynamic space ($E$, $\Omega$), and continuous trade-offs to ensure satisficing criteria.

Enterprises chiefly depend on M&S for exploring $L$ and selecting $l^*$ from $L$ that ensure satisficing criteria and optimize $E$ over $\Omega$ by approximately representing enterprise $E$ and its environment $\Omega$ as $R$, i.e., $R \approx$ ($E$, $\Omega$). Simulation experiments are then performed to project the impact of lever choices $l$ onto projected measures $M$ to estimate the behavior of $G(M(l(E), \omega))$ over uncertainty.

Emerging Digital Twin is an extended version of $R$, we term it as $RDT$ - it constitutes a dynamic, connected and continuously updated model that not only replicates the behaviour of enterprise $E$ and environment, but also closely synchronizes the states space of real enterprise and monitors environment to periodically update $\Omega$ with minimal latency, i.e., $RDT \cong$ ($E$, $\Omega$). Therefore, the simulation of digital twin can also help to effectively evaluate the behavior of $G(M(l(E), \omega(T)))$ with all plausible evolution conditions.

### 2.2 State-of-the-art Simulation Aided Decision Making Process

Most of the simulation based decision-making approaches consider a structured iterative process with 3 types of activities recommended by (Sargent 2013). The step include:

- **Model Construction:** Construction of $R$ or $RDT$ sufficiently captures $E$ and $\Omega$, establishes mapping from $R/RDT$ to $M$ and defines $L$ as transformation function or configuration of $R/RDT$. The construction activity includes two steps: capture domain knowledge and problem space (i.e., $\Omega$) as conceptual model, and convert conceptual model into simulation model or implementation using underlying simulation platform.
- **Verification and Validation (V&V):** Ensures that the constructed $R$ or $RDT$ faithfully represents the characteristics of ($E$, $\Omega$).
- **Simulation:** It first changes the structure, behaviour or parameter of enterprise model $R$ or $RDT$ through a lever $l$ and observes its impact on projected measures $M$ along time $t \in T$.

The use of simulation in decision-making is primarily observing $M$ under plausible variations in $\Omega$, and estimating $G(M(l(E), \omega))$ for all $l$ in $L$ to find optimal or satisficing levers $l^*$.

### 2.3 Challenges

The traditional process of constructing, validating and utilizing traditional models ($R$) or digital twins ($RDT$) for enterprise decision-making is inherently effort-, time-, and knowledge-intensive endeavor. First, the model construction step requires the careful abstraction and formalization of the enterprise state space $E$, identification of relevant measures $M$, selection of actionable levers $L$, and characterization of the uncertainty space $\Omega$ . This phase demands deep domain expertise, extensive data acquisition and iterative refinement to ensure fidelity to real-world behaviors. Moreover, it requires simulation experts to convert conceptual model into simulatable specification.

Second, the verification and validation (V&V) is a step that ensures different forms of validations, as suggested in (Sargent 2013), to ensure the faithfulness of the models and environment, i.e., constructed $R$ or $RDT$ accurately represents ($E$, $\Omega$) (i.e., conceptual validity), the conceptual model is correctly translated into implementation or simulation model (i.e., implementation validity), and finally compares simulation outputs and empirical observations to ensure operational validity. This step is also knowledge-intensive, and one needs to rely heavily on expertise in system dynamics, statistical testing and empirical methods.

Third, the simulation execution and analysis of simulation results step involves running multiple simulations with different $l \in L$ and $\omega \in \Omega$ to observe the effects of $l$ or combination of $l^* \in L$ on $M$, effect of plausible scenarios $\omega \in \Omega$ on $M$ or their combination. Interpreting simulation results with respect to *satisficing* criteria or evaluating the best set of levers, especially under conditions of nonlinearity, emergent behavior or multi-objective trade-offs, is cognitively demanding and resource intensive.

### 2.4 Our Experience with Enterprise-Scale M&S and Digital Twins in Decision Making

Based on our experience of using M&S in enterprises (Barat et al. 2019; Barat et al. 2022; Barat et al. 2024), we observed that the model or digital twin (DT) construction phase is an extremely time-, effort- and knowledge-intensive process. This phase demands close collaboration between domain experts and modelers and typically spans 12–16 weeks involving approximately 30–35 person-weeks (PW) of dedicated modeler effort and a similar amount of engagement from Subject Matter Experts (SMEs). An additional 10–15 PW of development time is needed to translate the conceptual model into a simulation model. Moreover, this collaboration is inherently iterative and requires continuous feedback loops across domain experts, modelers and developers. Similarly, the verification and validation (V&V) phase often requires several cycles over an additional 6–8 weeks. While the actual execution of simulations is comparatively less resource-intensive, analyzing simulation results to confirm satisficing properties and identifying optimal or near-optimal actions ($l^* \in L$) requires deep domain understanding and decision-making expertise.

We also encountered an additional challenge that the faithful representation $R$ or digital twin $RDT$ of enterprise $E$ is often not a long-lived artefact. Any substantial change in ($E$ or $\Omega$) or a redefinition of goals $G$ triggers a full or partial re-modelling cycle with similar effort and stakeholder involvement as the initial construction. This underscores the need for more agile construction process with less dependency on SMEs throughout the M&S centric decision-making journey.

## 3 HYPOTHESIS

LLMs possess several powerful capabilities that make them valuable aid to accelerate conceptual modeling, assist validation process and serve as reasoning aid towards decision space exploration. As vast repositories of cross-domain knowledge, LLMs can correlate information across contexts to enable the development of conceptual models as a proxy SME of specific domain. LLMs can also act as modelers by responding in structured formats from which models can be automatically generated. These models can then be transformed from conceptual to simulation specifications using a guided set of prompts, transformation rules or illustrative examples by leveraging capabilities, such as few-shot learning or code generation. They can infer complex relationships and navigate models to establish certain properties to ease validation step. Considering these capabilities, we aim to adopt LLMs into the M&S and DT lifecycle, and we believe such an approach substantially reduces manual knowledge engineering burdens, accelerate model construction and V&V phases and enhance simulation-driven decision-making. Specifically:

- **Model Construction Assistance:** LLMs can extract knowledge from unstructured enterprise data, formalize entity-relationship models, suggest Measures ($M$), identify possible interventions ($L$), and generate initial conceptual models. Given prompts about enterprise goals and structure, they can propose candidate state-space representations ($E$) and characterize uncertainty ($\Omega$).
- **V&V Support:** LLMs can validate conceptual models using known knowledge structures and suggest validation scenarios. They can evaluate cross-reference simulation outputs against benchmarks and flag anomalies for expert review.
- **Simulation Experimentation and Analysis:** LLMs can design experiments over $L$ and $\Omega$, and help interpret results against decision criteria like robustness, resilience, and adaptability. They can also generate readable summaries of system behavior under varying conditions.

Despite their strengths, LLMs have limitations that must be addressed. They may experience attention fading, overlooking critical details or losing coherence in extended contexts. Their responses can be non-deterministic, occasionally yielding inaccurate or misleading information. Hallucinations, plausible but fictitious outputs, also pose risks.

## 4 LLMS IN MODELLING AND SIMULATION

In recent years, the application of LLMs, such as ChatGPT, in M&S pipeline has emerged as a promising direction. To understand the current state of adoption, we conducted an Systematic Mapping Study (Petersen et al. 2008) on peer-reviewed papers indexed in Google Scholar covering the years 2022 to 2025 with keywords: ("Modelling and Simulation" OR "Modelling & Simulation" OR "Modeling and Simulation" OR "Modeling & Simulation") AND ("Large Language Model" OR "ChatGPT" OR "LLM"). Our SMS revealed a growing trend with significantly high numbers in 2024. Upon examining the nature of these papers, we classified them into three categories: core contribution papers proposing methods or frameworks (Frydenlund et al. 2024; Wang et al. 2025; Hu 2025; Xia et al. 2024; Junprung 2023), survey papers offering state-of-the-art reviews (Gao et al. 2024; Kambeitz and Meyer-Lindenberg 2025) and case studies applying LLMs in specific simulation contexts (Giabbanelli 2023; Liu and Yang 2024). The case studies notably span domains, such as education, healthcare and simulation process facilitation, highlighting the cross-domain applicability of LLMs in simulation.

Table 1 summarizes the major capabilities introduced in the literature. As shown in the table, their contributions cover different phases of M&S based exploration. (Frydenlund et al. 2024) investigated natural language to simulation model generation, allowing prose descriptions to be translated into discrete-event, system dynamics or agent-based simulation code. (Hu 2025) developed ChatPySD, a system enabling interactive system dynamics simulations through natural language queries. (Xia et al. 2024) introduced an LLM-based multi-agent architecture to dynamically optimize simulation parameters, while (Junprung 2023) explored human-like behavior generation through prompt engineering techniques in agent-based models.

As shown in the table, most of existing approaches focus on isolated phases, such as model generation or result interpretation, without addressing the full spectrum from conceptualization to decision-making. Moreover, their key focus remained on demonstrating the art-of-the possibilities in M&S using LLMs without thoroughly examining their limitations, such as issues related to accuracy, consistency and the need for human oversight. This gap underscores the necessity for holistic frameworks that not only leverage the strengths of LLMs across all stages of M&S but also mitigate their shortcomings to ensure practical applicability in industrial contexts.

## 5 APPROACH

We present an adaptive Decision-Making Framework (DMF) that includes a surrogate system for an enterprise to enable risk-free experimentation, identify suitable levers for achieving goals and generating insights in response to future uncertainties or internal changes to help the organization remain robust, resilient and adaptive amid evolving conditions. The role of the DMF within the enterprise is illustrated in

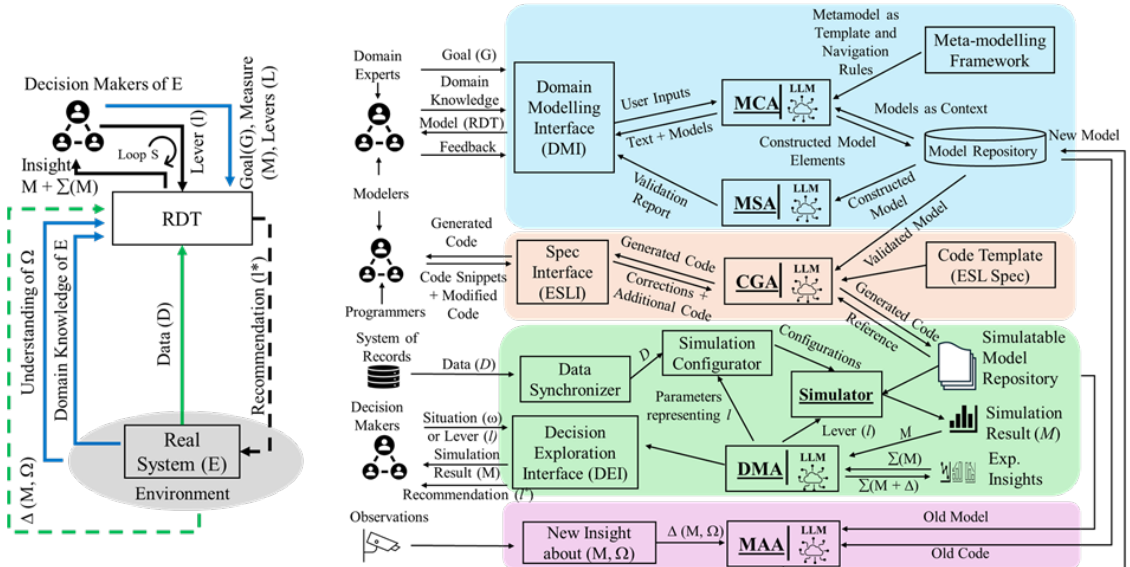
Table 1: Capabilities Proposed by Core Contribution Papers Integrating LLMs into M&S.

| Capability | Description | Reference |
|---|---|---|
| As Proxy SME to construct Model | Utilizing LLMs to extract domain knowledge, formalize conceptual models, and assist in initial model design through natural language processing. | (Frydenlund et al. 2024; Junprung 2023) |
| As Code Generator | Employing LLMs to translate conceptual models into executable simulation code for rapid prototyping and model implementation. | (Frydenlund et al. 2024) |
| As Guided Simulation Assistants | Leveraging LLMs to automate the design of experiments (DoE), parameter tuning and scenario generation for what-if scenario. | (Gao et al. 2024; Xia et al. 2024; Hu 2025). |
| As Simulation Result Synthesizers | Using LLMs to interpret and summarize simulation outputs to produce insights aiding in decision-making processes. | (Giabbanelli 2023). |
| Educational and Training Simulations | Use of LLMs into educational settings to enhance learning experiences through interactive simulations and feedback mechanisms. | (Liu and Yang 2024). |

Figure 1(a). As shown, the proposed approach centers on a faithful purposive digital representation (*RDT*) of the enterprise ($E$). Since the *RDT* is a purposive representation of a decision-making problem, one must start with constructing *RDT* by considering the Goals ($G$) as the purpose and leveraging domain knowledge about the enterprise under consideration and its environment including inherent uncertainties ($\Omega$). Once constructed, the *RDT* needs to be synchronized with system data by incorporating the system of record data collected from the enterprise and validated using validation techniques as discussed in Section 2.2. The validated and synchronized *RDT*, effectively a digital twin of $E$, can then be leveraged for decision space exploration by iteratively applying viable levers ($l$) and observing simulation outcomes $M$ to explore satisficing criteria and maximize the goals $G$ as outlined in Equation 2. The core components of DMF and supported workflows are shown in Figure 1(b).

While the supported workflows align with the traditional three-step process introduced by (Sargent 2013), we aim to reduce the excessive dependency on domain experts, decision-makers, expert modelers and programmers by leveraging LLM capabilities. The model construction phase (highlighted in blue) is supported by two LLM-based components: the Model Constructor Agent (MCA) and the Model Synthesizer Agent (MSA). The translation of the conceptual model into a simulation specification (shown in orange) is supported by an LLM-aided Code Generator (CGA), which enables a hybrid specification development interface combining traditional model-to-text transformations with developer in-the-loop LLM-based code generator.

The next phase, shown in green, corresponds to the traditional simulation phase supporting what-if scenario exploration in the M&S workflow. We leverage an agent-based simulator (Clark et al. 2017) to perform what-if scenarios, i.e., simulating $L$ and scenarios $\Omega$ to observe potential performance metrics $M$ under diverse scenarios. We further extend this phase to include (a) synchronization of the simulation model with system data (to adopt digital twin paradigm) and (b) recommendations for the next best lever ($l$) that could meet satisficing criteria and maximize goals. For this, we propose a Decision-Making Agent (DMA) that synthesizes simulation results relative to the goals and recommends optimal next steps. In addition to the traditional 3-step process, DMF also considers an additional step to monitor enterprise and its environment, and update *RDT* as they evolve. This step is highlighted using purple box in Figure 1 (b) and a building block, termed as Model Adaptation Agent (MAA), is introduced. Below, we describe the core concepts and rationale behind the key components presented in our approach.




(a) Role of DMF.

(b) An overview of DMF.

Figure 1: An Overview of Proposed Approach.

### 5.1 Model Constructor Agent (MCA)

We combine Model-Driven Engineering (MDE) and LLMs where LLM serves as knowledge repository and helps in information structuring and transformation, while MDE helps to break down a major step (e.g., domain model construction) into a sequence of manageable and unambiguous tasks. The use of MDE also ensures systematic navigation of supported metamodels for precise questions/prompts and metamodel-compliant model instantiation.

We adopted two metamodels from (Barat et al. 2025): GML metamodel and Agent metamodel, to represent enterprise and its environment as $RDT$. The GML metamodel capture goal ($G$), measures ($M$) and levers ($L$) along with their relationships as shown in Figure 2(a), whereas Agent metamodel captures enterprise and its environment using a set of interacting Agents ($A$) having their own internal state, structure and behaviours defined using deterministic and non-deterministic Event-Condition-Action rules. In this approach, these metamodels not only define what knowledge to extract but also provide the structure for navigating the conceptual space systematically. They ensure comprehensive coverage of enterprise state space $E$, relevant measures $M$, actionable levers $L$ and uncertainty space $\Omega$. As illustrated in Figure 2(b), the model construction workflow uses Model Navigator to systematically guide the exploration of metamodel elements (depicted in Figure 2 (a)) and Context component that maintains previously extracted model instances. This provides structure for the Prompt Generator, which creates context-aware prompts for the LLM. The Model Navigator uses both GML metamodel and Agent metamodel. The Prompt Generator also considers the feedback received from SMEs through Domain Model Interface (refer Figure 1 (b)). We found reaching LLM with precise context and fine-grained ask is essential as it overcomes their key limitations, such as attention fading and hallucination.

The Model Creator receives responses from LLMs, and leveraging MDE techniques, transforms them into formal model instances conforming to metamodel constraints and stores them in the repository for later validation and code generation. This MDE-driven transformation ensures structural correctness and consistency of resulting models. These models can then undergo further refinement through the Model Synthesizer Agent (MSA), creating a continuous improvement cycle that enhances model quality over time. The human-in-the loop approach combining LLMs with MDE significantly reduces the time SMEs must

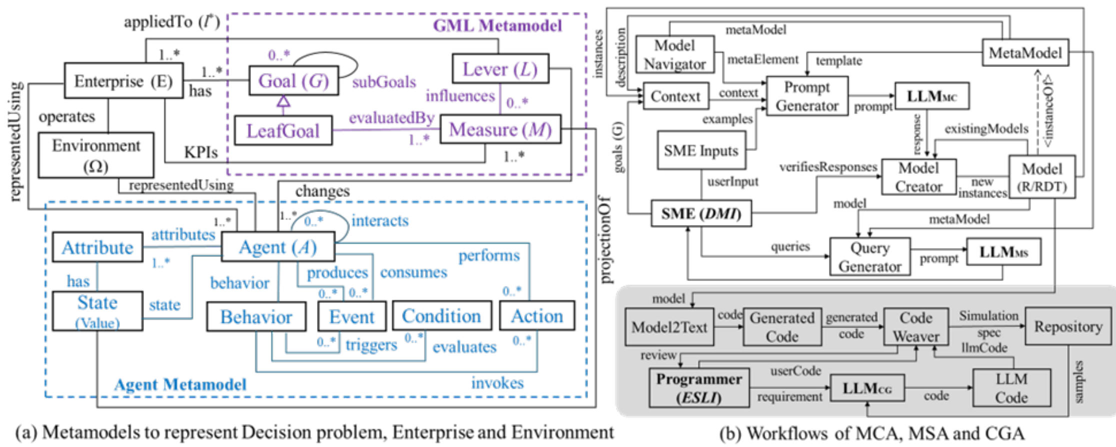



Figure 2: Model Construction, Validation and Code Generation.

spend on model construction. They contribute to the process in multiple ways: providing additional domain knowledge as inputs to the Prompt Generator and verifying/refining model elements as in Figure 1 (b).

### 5.2 Model Synthesizer Agent (MSA)

Models often suffer from inconsistencies, gaps that may not be apparent during initial construction. The model synthesizer agent employs an LLM agent with human oversight to validate and iteratively refine models. Unlike traditional validation tools that merely check for structural correctness, MSA acts as an intelligent agent that performs reviews across multiple dimensions including:

**Goal Structure:** Checks root goal alignment, sub-goal completeness, and hierarchical correctness.
**Measures and Levers:** Assesses KPI quantifiability, identifies missing or irrelevant KPIs and levers.
**Agent Models:** Checks missing actors, attribute coverage and behavior specifications.

Additionally, as shown in Figure 2(b), the Query generator component enables SMEs (through DMI) to interact with the model using natural language queries, which MSA processes to provide insights, explanations and recommendations about the model correction. This capability extends beyond validation to include model exploration and understanding. MSA incorporates human feedback in model refinement, creating a human-in-the-loop approach that leverages LLMs for model improvement while ensuring changes align with stakeholder intent. This significantly reduces validation effort while maintaining model quality.

### 5.3 Code Generator Agent (CGA)

The code generator agent translates conceptual models into executable specifications, bridging the gap between the abstract models and the executable code. We adopt a hybrid approach to translate conceptual model into simulatable specification, in our case ESL (Clark et al. 2017). We first use traditional model-to-text (M2T) transformation approach to convert captured conceptual model into ESL. However, it is observed that specifying behavioural specification using instance of a metamodel is not a pragmatic proposition. Therefore, we allow developers to add and weave missing code snippets by manual editing of the code snippets or producing code snippets using LLM where requirements and samples specification from code repository are provided to LLM (i.e., few-shot technique) as shown in Figure 2(b) within gray-colored box.

Essentially, as shown in Figure 2(b), the CGA systematically traverses the validated model structure and through the model2text component, maps model elements to code constructs according to the defined patterns. This process produces code that conforms to our simulation language specifications. The code weaver, a part of CGA workflow, integrates these specifications with necessary modifications to create complete and executable simulation code. It further incorporates feedback and code modifications from programmers. This human-in-the loop approach ensures correctness while leveraging LLM knowledge to improve the code-quality and significantly reducing development time and supporting faster iteration cycles as systems evolve.

### 5.4 Decision-Making Agent (DMA)

The DMA leverages LLM reasoning to synthesize simulation results, efficiently navigate the decision space and recommend optimal actions ($l*$) to achieve enterprise goals $(G)$ under varying conditions. Central to this navigational framework is a metamodel of decision space exploration, shown in Figure 3(a). It serves as a structured representation of the evolving relationships among Levers, Measures and Goals during decision space exploration and simulations.

As shown in the decision space exploration metamodel, conforming to the GML metamodel (Figure 2(a)), levers are represented as sets of changes in the *RDT* that mimic potential changes in the enterprise. These include either Model Changes - particularly in behavioral or event specifications (refer to Figure 2(a)), or changes in Attribute Values. Such elements are considered control parameters that decision-makers can manipulate within the enterprise or in simulation framework to understand the effects of those changes on Measures.

We characterize the relationships between these atomic changes and measures along two dimensions: direction of impact (i.e., polarity - Positive, Negative or Random) and magnitude (i.e., sensitivity - High, Medium or Low). These relationships are represented through a tertiary association in the decision space exploration metamodel (Figure 3(a)) and can be captured in a table, termed the Influence Table, as illustrated in Figure 3(b). This Influence Table serves as an interpretive tool clarifying how specific changes, and therefore levers, affect performance outcomes.

In our approach, we adopt a workflow, shown in Figure 3(c), to drive simulations, capture simulation results in a repository, and interpret them along the <polarity, sensitivity> tuple for the intersection of <change, measures> matrix using the Influence Table. Decision-makers can initiate simulations by selecting a specific lever $(l)$ of a simulation model from the repository. The Configurator block converts levers into appropriate parameter values and synchronizes the *RDT* with data $(D)$ collected from the system of record. The Simulator consumes the prepared run specification, along with all populated parameter values, to produce Measures $(M)$ trends over the simulation time span as simulation results.

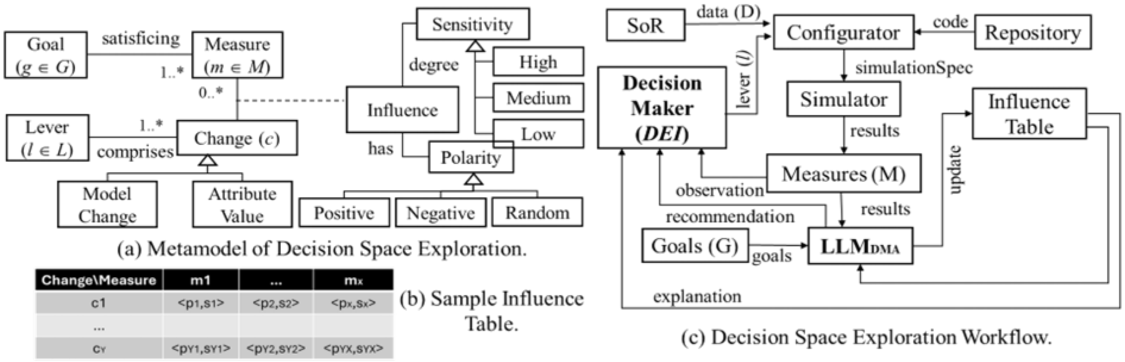
The DMA uses the LLM to update the Influence Table after each simulation run - similar to a policy update in Reinforcement Learning (RL). This simulation-led LLM-based continuous learning helps the DMA narrow down the decision space navigation options and assist in selecting the best possible levers.

### 5.5 Model Adaptation Agent (MAA)

This agent ideally should sense changes in (M, Ω), summarize those changes and leverage the LLM to automatically update the *RDT* and corresponding simulation code as shown in Figure 1(b). At present, it is a manual step where the involved stakeholders provide change insights; subsequently, those changes are analyzed and the *RDT* along with the associated simulation model are updated manually to keep the model in sync with reality. We are exploring the role and efficacy of LLMs in producing new *RDT*s and new simulation code from the combination of <old model, old simulation code> and change specifications, provided with appropriate context.

## 6 APPLICATION AND OUR EXPERIENCE

We have implemented DMF by integrating LLM APIs provided by ChatGPT (specifically, gpt-4o mini), Eclipse Modeling Framework (EMF) SDK v2.27 and EMF's Ecore v2.25 meta-modeling capability. In our implementation, EMF serves modeling needs through its Ecore modeling framework. Ecore is utilized to define and refine essential metamodels. It is also found to be useful for systematically traversing metamodels for producing textual description of metamodel elements and their associations using MOFM2T technique.



(a) Metamodel of Decision Space Exploration.
(b) Sample Influence Table.
(c) Decision Space Exploration Workflow.

Figure 3: Decision Space Exploration.

We applied the implemented approach in a real-world digital twin development project for a major parcel delivery organization, referred to here as Organization X for confidentiality. Organization X handles 20–30 million parcels daily across a vast network of collection centers, distribution hubs, packaging facilities and multimodal fleets (trucks, rail, air, ships). Facing rising labor and fleet costs, along with emerging demands such as one-day delivery, Organization X seeks to identify operational bottlenecks and explore opportunities to expand volume and services without compromising timely delivery. We consider this industry scale problem to evaluate the efficacy of our proposed approach.

## 6.1 Validation Methodology

We organized two teams for comparison. Team A consisted of 10 members: 2 SMEs familiar with Organization X, 2 logistics domain SMEs, 4 expert modelers and 2 developers experienced in ESL technologies. Team B, a lean team, included 1 SME (moderate domain knowledge, limited information about X), 1 modeler and 1 developer using the proposed LLM-based approach. While A team was working for a real-life case study presented in (Barat et al. 2024), team B were tasked with constructing a digital twin aligned with the same high-level goals.

## 6.2 Experience Summary

For Team A, the entire process spanned 10 weeks, comprising 4 PW of discussion sessions with SMEs from Organization X, and internally, 4 SMEs and 4 modelers dedicated 8 weeks of full-time work totaling approximately 68 person-weeks (PW) to construct and validate the digital model for Organization X. The conversion of the conceptual model to the ESL required an additional 12 PW from developers.

Team B started model construction with the same top-level goal and achieved $\sim$95% model similarity in identifying agents, their attributes, behaviours and interactions with 16 PW of effort compared to the 68 PW spent by Team A. We observed slight improvement using the proposed approach in the validation phase compared to the significant advantage in model construction phase. The development effort was reduced to 6 PW from 12 PW. We believe there is further scope for improvement in this step with a locally trained LLM tailored to the class of business problems and the ESL language. It is also noted that MCA agent produces better insights about potential uncertainty and variations $\Omega$. The principal differences between the outcome of team A and team B – possible uncertainties within $E$ were more species and contextual for organization X but outcome of B were overarching and holistic in nature describing potential future situation better.

In the decision space exploration phase, we found significantly less dependency on and burden for decision-makers to a) define the potential solution space (i.e., defining $L$) for a given set of goals $G$ (ref., Equation 2) and b) navigating solution space $L$ in a systematic manner to find $l^*$. However, we observed this improvement relative to purely human-centric decision space exploration, where decision-makers evaluate prior simulation results to derive the next best lever to explore. In contemporary enterprises, this navigation step is often aided by learning techniques, such as Reinforcement Learning (RL), as we have used previously (Barat et al. 2019). There, the decision space exploration was RL-aided but required around 16 PW of additional effort involving expert data scientists for developing and training the RL agent. We find the proposed approach requires less effort without substantial loss of precision. Moreover, RL approaches involve retraining costs whenever goals or enterprise/environmental conditions change.

With our experience we assert that the fundamental elements of an enterprise, $E$, to remain robust, resilient and adaptive (as outlined in Equations 2, 3, and 4) primarily hinge on accurately capturing $E$ (a task traditionally entrusted to modelers in M&S), anticipating and addressing uncertainties and evolution $\Omega$ over time (largely reliant on SMEs in conventional approaches), defining options $L$ to achieve goals $G$ (another SME-driven activity), and systematically selecting $l^*$ from a broad range of possibilities (often performed ad hoc under existing methods). The proposed approach enhances the precision in capturing $E$ and $\Omega$ comprehensively, which aids in the construction of digital twins (DTs), while also streamlining the navigation of $L$ to select $l^*$ effectively, showcasing the practical application and benefits of DTs. We observe that traditional methods heavily depend on SMEs, modelers, simulation experts and stakeholders, making them effort-intensive and time-consuming. The proposed approach substantially reduces these dependencies without compromising to make enterprise robust, resilient and adaptive for the future.

## 7 CONCLUSION

This paper presented an adaptive Decision-Making Framework (DMF) that synergistically integrates LLMs with Model-Driven Engineering (MDE), core M&S concepts and the concept of emerging digital twin to reduce the effort and excessive burden on domain experts, modelers and decision-makers while enhancing enterprise-scale decision-making capabilities. Building on a clear hypothesis, we demonstrated that when LLMs are systematically combined with supporting techniques, such as MDE and more robust code generation techniques, they can significantly reduce manual knowledge engineering, accelerate model construction, support validation and verification and synthesize insights from simulation results.

While we could not present the validation case study in detail due to space constraints, we discussed early experiences that show a substantial reduction in effort across all phases and highlight how expert dependency is minimized, particularly in model construction and decision space exploration, while maintaining meaningful precision. We believe the potential of this approach is considerable: LLMs, when locally fine-tuned for specific business classes, domain knowledge, and underlying tools and specifications such as ESL, could further enhance automation, adaptability, and the quality of decision recommendations.

## REFERENCES

Balaban, E., A. Saxena, P. Bansal, K. F. Goebel, and S. Curran. 2009. "Modeling, Detection, and Disambiguation of Sensor Faults for Aerospace Applications". *IEEE Sensors Journal* 9(12):1907–1917.

Barat, S., H. Khadilkar, H. Meisheri, V. Kulkarni, V. Baniwal, P. Kumar *et al*. 2019. "Actor Based Simulation for Closed Loop Control of Supply Chain Using Reinforcement Learning". In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*. May 13$^{th}$-17$^{th}$, Montreal, QC, Canada, 1802–1804.

Barat, S., V. Kulkarni, T. Clark, and B. Barn. 2022. "Digital Twin as Risk-Free Experimentation Aid for Techno-Socio-Economic Systems". In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems*. Oct 23$^{rd}$-28$^{th}$, Montreal, QC, Canada, 66–75.

Barat, S., D. Mulpuru, A. Yadav, R. Korabu, H. Thogaru, and V. Kulkarni. 2025. "Constructing Enterprise Digital Twins by Augmenting LLMs with MDE". In *Proceedings of the 18th Innovations in Software Engineering Conference*. Feb 20$^{th}$-22$^{nd}$, Kurukshetra, Haryana, India, 1–11.

Barat, S., A. Yadav, H. Thogaru, V. Kulkarni, and K. Bhattacharya. 2024. "Imparting Adaptiveness and Resilience to Parcel Delivery Networks: A Digital Twin Centric Simulation Based Approach". In *2024 Winter Simulation Conference (WSC)*, 2999–3010 https://doi.org/10.1109/WSC63780.2024.10838620.

Boschert, S., and R. Rosen. 2016. "Digital Twin—The Simulation Aspect". *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers*:59–74.

Clark, T., V. Kulkarni, S. Barat, and B. Barn. 2017. "ESL: An Actor-Based Platform for Developing Emergent Behaviour Organisation Simulations". In *Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection*. Jun 21st-23rd, Porto, Portugal, 311–315.

Frydenlund, E., J. Martínez, J. J. Padilla, K. Palacio, and D. Shuttleworth. 2024. "Modeler in A Box: How Can Large Language Models Aid in the Simulation Modeling Process?". *Simulation* 100(7):727–749.

Gao, C., X. Lan, N. Li, Y. Yuan, J. Ding, Z. Zhou, *et al*. 2024. "Large Language Models Empowered Agent-Based Modeling and Simulation: A Survey and Perspectives". *Humanities and Social Sciences Communications* 11(1):1–24.

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.

Hu, B. 2025. "ChatPySD: Embedding and Simulating System Dynamics Models in ChatGPT-4". *System Dynamics Review* 41(1):e1797.

Jain, S., N. Fong Choong, K. Maung Aye, and M. Luo. 2001. "Virtual Factory: An Integrated Approach to Manufacturing Systems Modeling". *International Journal of Operations & Production Management* 21(5/6):594–608.

Junprung, E. 2023. "Exploring the Intersection of Large Language Models and Agent-Based Modeling via Prompt Engineering". *arXiv preprint arXiv:2308.07411*.

Kambeitz, J., and A. Meyer-Lindenberg. 2025. "Modelling the Impact of Environmental and Social Determinants on Mental Health using Generative Agents". *npj Digital Medicine* 8(1):36.

Kritzinger, W., M. Karner, G. Traar, J. Henjes, and W. Sihn. 2018. "Digital Twin in Manufacturing: A Categorical Literature Review and Classification". *Ifac-PapersOnline* 51(11):1016–1022.

Kulkarni, V., S. Reddy, S. Barat, and J. Dutta. 2023. "Toward a Symbiotic Approach Leveraging Generative AI for Model Driven Engineering". In *2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS)*. Oct 1st-6th, Vasteras, Sweden, 184–193.

Liu, C., and S. Yang. 2024. "Application of Large Language Models in Engineering Education: A Case Study of System Modeling and Simulation Courses". *International Journal of Mechanical Engineering Education*:03064190241272728.

Petersen, K., R. Feldt, S. Mujtaba, and M. Mattsson. 2008. "Systematic Mapping Studies in Software Engineering". In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. June 26th-27th, Italy.

Sargent, R. G. 2013. "An Introduction to Verification and Validation of Simulation Models". In *2013 Winter Simulations Conference (WSC)*, 321–327 https://doi.org/10.1109/WSC.2013.6721430.

Simon, H. A. *et al*. 1972. "Theories of Bounded Rationality". *Decision and Organization* 1(1):161–176.

Tao, F., Q. Qi, L. Wang, and A. Nee. 2019. "Digital Twins and Cyber–Physical Systems Toward Smart Manufacturing and Industry 4.0: Correlation and Comparison". *Engineering* 5(4):653–661.

Wang, L., J. Zhang, H. Yang, Z.-Y. Chen, J. Tang, Z. Zhang, *et al*. 2025. "User Behavior Simulation with Large Language Model-Based Agents". *ACM Transactions on Information Systems* 43(2):1–37.

Xia, Y., D. Dittler, N. Jazdi, H. Chen, and M. Weyrich. 2024. "LLM Experiments with Simulation: Large Language Model Multi-Agent System for Simulation Model Parametrization in Digital Twins". In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*. Sep 10th-13th, Padova, Italy, 1–4.

## AUTHOR BIOGRAPHIES

**SOUVIK BARAT** is a Principal Scientist at Tata Consultancy Services (TCS) Research, India and a visiting researcher at Middlesex University London. His research interests include Digital Twin technology, Modelling and simulation of complex systems, and Model Driven Engineering. His email address is souvik.barat@tcs.com.

**DUSHYANTHI MULPURU** is a Scientist at TCS Research, India. Her current research interests include Digital Twins, Modelling and simulation of complex systems and Generative AI. Her email address is dushyanthi.mulpuru@tcs.com.

**HIMABINDU THOGARU** is a Researcher at TCS Research, Pune, India. Her current research interests are Software Engineering, Digital Twin Modelling and Simulation and Generative AI. Her email address is himabindu.thogaru@tcs.com.

**ABHISHEK YADAV** is a Researcher at TCS Research, Pune, India. His current research interests are Software Engineering, Digital Twin modeling and Simulation, and Generative AI. His email address is y.abhishek1@tcs.com.

**VINAY KULKARNI** is a TCS Fellow at TCS Research, Pune, India. His current research interests are Digital Twins, Adaptive Enterprises, and Software Engineering. Alumnus of Indian Institute of Technology Madras, Vinay is a Fellow of Indian Academy of Engineering. His email address is vinay.vkulkarni@tcs.com.