

Article

A Framework for Conversational Digital Twins: Integrating Generative AI for Operational Event Simulation

Pablo Vicente-Martínez ¹, Adrián Chust-Ros ¹, Emilio Soria-Olivas ², María Ángeles García-Escrivà ^{3,*},
Edu William-Secin ⁴ and Manuel Sánchez-Montañés ⁵

¹ SPV Scala, Gran Canaria, 35100 San Bartolomé de Tirajana, Spain; c.datos12@salascale.com (P.V.-M.); c.datos34@salascale.com (A.C.-R.)

² Intelligent Data Analysis Laboratory (IDAL), Department of Electronic Engineering, Universitat de València, 46022 Valencia, Spain; emilio.soria@uv.es

³ Fundación Canaria Living Lab, 35017 Las Palmas de Gran Canaria, Spain

⁴ Department of Economics and Business Management, Institute of Tourism and Sustainable Development (TIDES), Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain; edu.william@ulpgc.es

⁵ GNB, Department of Computer Science, Universidad Autónoma de Madrid, 28049 Madrid, Spain; manuel.smontanes@uam.es

* Correspondence: coordinacionit@canariaslivinglab.org

Abstract

Operational planning for large-scale events involves high uncertainty, heterogeneous data sources, and complex resource allocation decisions. Although digital twins and simulation models can support this process, their adoption is often limited by the technical expertise required to configure scenarios and interpret outputs. This paper introduces a Conversational Digital Twin Framework (CDTF) that integrates generative AI with simulation-based decision support, enabling users to configure operational scenarios through natural language. The framework comprises four layers: data integration, a simulation engine combining machine learning and discrete-event simulation components, a conversational AI interface, and a visualization layer for reports and dashboards. It is validated through a TRL4 prototype for football stadium operational planning at the Gran Canaria Stadium, using historical match records, external contextual sources, and synthetically generated operational variables where detailed stadium data were unavailable. The validation focuses on architectural feasibility, workflow integration, conversational configuration, and operational viability. Results show that the agent recognized user intent in 93.3% of evaluated prompts, extracted all explicitly stated parameters in the tested cases, detected all evaluated invalid inputs, and completed the full simulation workflow in all tested scenarios. Average conversational responses took less than six seconds, while complete simulation workflows required approximately 1 min. These findings suggest that conversational interaction can facilitate the access to simulation-based planning and support interactive what-if analysis for event operations, while future work must validate predictive performance with real operational datasets.

Keywords: digital twin; conversational AI; generative AI; event simulation; software framework; machine learning; simulation architecture



Academic Editor: Paolo Bellavista

Received: 13 May 2026

Revised: 14 July 2026

Accepted: 19 July 2026

Published: 30 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Planning the operations of large scale events, including sporting competitions and cultural festivals, poses significant challenges for organizers, venue managers, and local

authorities. These events involve complex interdependencies between resource allocation, attendance forecasting, logistics management, and real-time decision-making under uncertainty [1,2].

Digital Twin (DT) technology has emerged as a promising paradigm for addressing these challenges by representing and analyzing complex physical systems through virtual models that can support simulation and decision-making under different operational conditions [3,4]. Digital twins enable the exploration of alternative scenarios in system operation, allowing the analysis of potential system behaviors, performance variations, and the impact of external conditions through simulation-based approaches [5,6]. However, existing digital twin implementations often require significant technical expertise for configuration, integration, and operation, which can limit their accessibility and adoption by non-technical domain experts [7,8].

1.1. Digital Twins for Simulation and Decision Support

Digital twins have been successfully applied across manufacturing [9], smart cities [10], healthcare [11], and infrastructure management [12]. These implementations leverage continuous data streams from IoT sensors, historical databases, and external APIs to maintain synchronized virtual representations of physical assets [13]. Machine learning models embedded within digital twins enable predictive analytics, anomaly detection, and optimization of operational parameters [14,15].

In the domain of event management, digital twins offer unique value by simulating crowd dynamics, resource consumption patterns, and service delivery performance [16]. Prior research has explored digital twin frameworks for modeling complex systems and optimizing operational processes across industrial and service contexts [5]. Similarly, simulation-based approaches have been proposed to support decision-making and performance evaluation in large-scale and dynamic environments [3]. However, these systems typically require manual parameter configuration through graphical interfaces or direct manipulation of configuration files, creating a steep learning curve for non-technical users.

1.2. Generative AI and Scenario Generation

The advent of Large Language Models (LLMs) and generative AI has opened new possibilities for human–computer interaction in complex systems [17,18]. Generative models can understand natural language queries, extract structured information, and generate contextually appropriate responses [19]. In the context of digital twins, generative AI can serve as an intelligent interface layer that translates user intentions expressed in natural language into formal simulation configurations [20]. This approach has been explored in industrial settings, where conversational agents assist operators in configuring production schedules [21] and in urban planning, where LLMs facilitate stakeholder engagement in scenario exploration [22]. However, the integration of generative AI with digital twin architectures remains fragmented, lacking a unified framework that addresses the full lifecycle from natural language interaction to simulation execution and result visualization.

1.3. Conversational Interfaces for Complex Systems

Conversational AI has demonstrated effectiveness in democratizing access to technical systems across multiple domains. Virtual assistants and chatbots now support tasks ranging from data analysis [23] to code generation [24]. The key advantage of natural language-driven systems lies in their ability to enable users to express analytical or programming intentions in natural language, reducing the need to learn domain-specific syntax or low-level implementation details [23,24].

For simulation and modeling systems, conversational interfaces can support scenario specification and enable more intuitive interaction with simulation environments [25,26].

Natural Language Processing (NLP) techniques enable intent recognition, entity extraction, and dialogue management, transforming user utterances into structured commands [27,28]. However, existing implementations often focus on narrow use cases and lack generalizability across different simulation domains.

1.4. Problem Explication

Following the Design Science Research methodology, the explication of the problem reveals that despite advances in digital twins, generative AI, and conversational interfaces, a critical gap remains: there is no established framework that systematically integrates these three paradigms to create accessible, intelligent simulation systems for operational planning. Existing digital twin implementations are typically focused on specific industrial applications and decision-support integration, often lacking advanced interaction capabilities and flexible, intelligent user interfaces [7,8]. Conversely, generative AI applications in simulation remain largely experimental and are not grounded in robust architectural frameworks suitable for real-world deployment [29].

This paper addresses this gap by proposing a comprehensive framework for Conversational Digital Twins, which are digital twin systems enhanced with generative artificial intelligence to support natural language configuration, scenario exploration, and result interpretation. Unlike existing digital twin architectures that rely on complex graphical interfaces, the originality of our solution lies in the seamless, lightweight architectural integration of a generative AI layer with a robust simulation engine. Consequently, the primary contribution of this study is this specific architectural integration and the resulting conversational configuration mechanism, which directly democratizes access to advanced simulation capabilities for non-technical decision-makers.

1.5. Define Requirements and Research Questions

Following the identification of the problem, this research defines the requirements for a solution by formulating the following Research Questions (RQs), which guide the design and development of our framework:

1. RQ1 (Technical & Architectural Requirements): How can generative AI be architecturally integrated with simulation-based decision support systems to enable non-technical users to configure operational scenarios through natural language?
2. RQ2 (Validation Requirements): How does the proposed conversational integration perform in terms of intent recognition, configuration accuracy, and system latency within a real-world operational planning context?
3. RQ3 (Generalizability Requirements): What are the architectural principles that allow this framework to be generalized to other domains requiring simulation-based decision support under uncertainty?

Answering the RQs within current literature: Regarding RQ1, existing literature typically approaches digital twins through complex graphical interfaces or direct file manipulation, while generative AI applications in simulation remain largely experimental and lack a unified architectural framework for real-world deployment. Addressing RQ2, current evaluations of conversational agents in complex systems often focus on narrow use cases (e.g., data analysis or code generation), leaving a gap in holistic validations that measure end-to-end latency and configuration accuracy in simulation workflows. Finally, concerning RQ3, while digital twins have been successfully applied in manufacturing, smart cities, and healthcare, frameworks that decouple the complexity of the simulation model from the user interaction paradigm to allow cross-domain transferability are still underdeveloped in the state of the art (SOTA).

1.6. Paper Organization

The remainder of this paper is organized as follows. Section 2 presents the CDTF, describing its conceptual architecture and the role of each component layer, including infrastructure, technology stack, and integration patterns. Section 3 presents the validation case study for simulation of sporting events, demonstrating conversational interaction examples and prediction outputs. Section 4 discusses the implications of our findings, framework generalizability, and identified limitations. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Design and Development of the Conversational Digital Twin Framework

In alignment with the Design and Develop phase of the Design Science methodology, this section presents the artifact: the CDTF. It is a layered architecture designed to integrate generative AI capabilities with simulation systems for operational planning. The framework addresses the key challenge of making complex digital twin systems accessible to non-technical users while maintaining computational rigor and predictive accuracy. We first describe the conceptual architecture and the role of each layer, then detail the technical implementation of the prototype developed to validate the framework's feasibility.

2.1. Conceptual Architecture

The CDTF is structured around four interconnected layers, as illustrated in Figure 1. Each layer serves a distinct functional role while maintaining loose coupling to ensure modularity and extensibility:

1. **Data layer:** Responsible for data acquisition, integration, and preprocessing from heterogeneous sources including historical databases, real-time APIs, and external data providers.
2. **Simulation engine (digital twin core):** Encapsulates the computational models that represent the physical system's behavior, including discrete-event simulation components and machine learning models for prediction. In this proof of concept, the simulation pipeline was adapted to the specific operational planning requirements of football events at the Gran Canaria Stadium. However, the architecture and simulation workflow were designed with modularity and extensibility in mind, enabling their adaptation to different operational domains and event-planning use cases beyond the stadium context.
3. **Conversational AI layer:** Provides a natural language interface powered by generative AI models, enabling users to configure simulations, query system state, and interpret results through dialogue.
4. **Visualization layer:** Presents simulation outputs through interactive dashboards, reports, and graphical representations tailored to different stakeholder needs.

The framework follows a request–response cycle: users express intentions in natural language, the conversational AI layer translates these into structured configuration parameters, the simulation engine executes the corresponding computations, and results are presented through the visualization layer. This architecture decouples the complexity of simulation modeling from the user interaction paradigm, enabling domain experts to leverage sophisticated analytical capabilities without requiring programming [7,8].

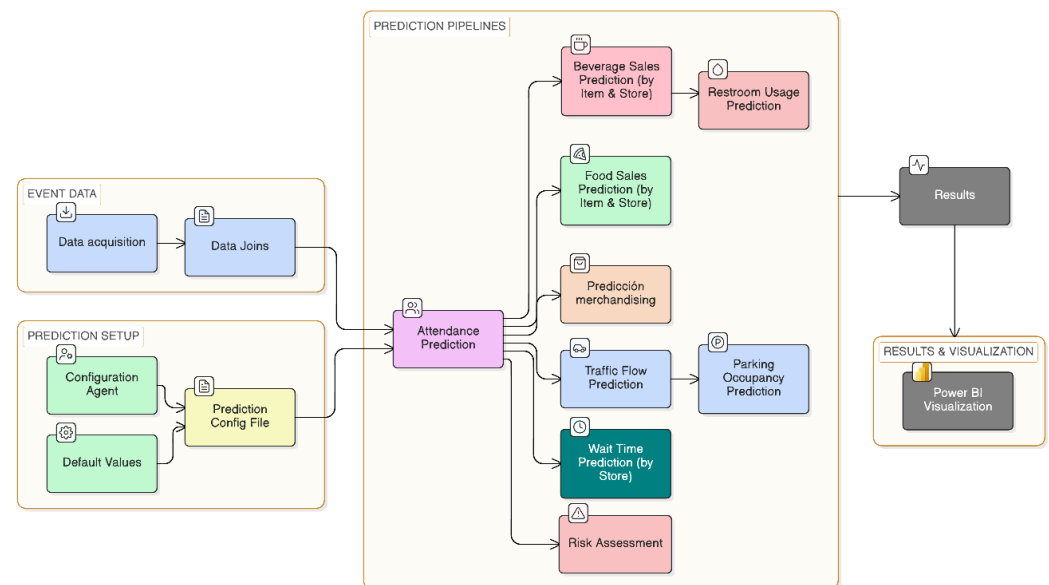


Figure 1. Conceptual architecture of the CDTF showing the four main layers and their interactions.

2.2. Data Layer

The data layer serves as the foundation for all simulation activities by ensuring reliable access to the diverse data sources required for digital twin operations.

2.2.1. Data Sources and Integration

Operational digital twins typically require three categories of data [12,13]:

- **Historical data:** Time-series records of past system behavior stored in relational databases, including operational metrics, resource consumption patterns, and contextual variables (e.g., weather conditions, attendance records).
- **Real-time data:** Current system state obtained through IoT sensors, monitoring systems, or operational databases that reflect live conditions.
- **External contextual data:** Information from third-party APIs that provide environmental context, such as weather forecasts, social media trends, public transportation schedules, or ticketing systems.

The framework employs an Extract-Transform-Load (ETL) pipeline that periodically queries these sources, applies data cleaning and normalization procedures, and stores the processed information in a unified operational database [15]. Data validation mechanisms ensure integrity through schema checking, range validation, and consistency verification across related datasets.

2.2.2. Data Versioning and Simulation Identification

A critical design decision in the framework is the use of a `simulation_id` field that tags all data records. The baseline configuration (`simulation_id = 0`) represents the reference state derived from historical data, while incremental identifiers mark scenario variants generated through conversational interaction. This versioning approach enables:

- **Traceability:** Each simulation run can be reconstructed by retrieving all data associated with its identifier.
- **Comparison:** Different scenarios can be evaluated against the baseline or against each other.
- **Reproducibility:** Simulation results can be validated by re-running configurations with identical parameters.

This design pattern aligns with best practices in computational reproducibility and digital twin lifecycle management [6,12].

2.3. Simulation Engine (Digital Twin Core)

The Simulation Engine constitutes the computational heart of the framework, implementing the mathematical and statistical models that predict system behavior under various operational conditions [9,30].

2.3.1. Model-Agnostic Architecture

The framework adopts a model-agnostic design philosophy, allowing different types of predictive and simulation models to coexist within the same infrastructure. This flexibility is achieved through a standardized interface that defines:

- Input schema: Expected format for configuration parameters and input features.
- Execution protocol: Method signatures for training, prediction, and state updates.
- Output schema: Structured format for returning predictions and diagnostic information.

This abstraction enables the integration of various modeling paradigms including machine learning algorithms and discrete-event simulation models.

2.3.2. Machine Learning Models for Prediction

In the prototype implementation, we employ supervised learning models to predict key operational variables such as attendance, resource consumption, queue lengths, and service utilization. To ensure adaptability to varying levels of data availability, the simulation engine is fully configuration-driven through YAML files. These files explicitly define the input features, target variables (e.g., `total_attendance`, `total_sells`, `available_parking_spaces`, or `avg_queue_seconds`), data joins, and temporal granularities for each predictive task.

Rather than relying on a single static algorithm, the framework implements an automated pipeline for model selection and training. For a given operational scenario, the system applies an 80/20 train-test data split (adjustable via configuration) and executes an internal feature selection process. It then performs hyperparameter tuning across multiple algorithm families, primarily Random Forest [31], XGBoost, CatBoost, and LightGBM. The pipeline evaluates the optimized models from each family using standard regression metrics—including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared values—and automatically selects the best-performing model for the final prediction step.

Once the optimal model is selected and trained, the prediction workflow follows these steps:

1. Feature engineering: Transform raw input variables (date, weather, event characteristics) into model-ready features as specified in the configuration schema.
2. Model loading: Retrieve pre-trained models from persistent storage, avoiding retraining overhead during simulation.
3. Prediction generation: Apply models to the configured scenario parameters to generate forecasts.

2.4. Conversational AI Layer

The Conversational AI Layer represents the most distinctive component of the framework, enabling natural language interaction with the digital twin system. This layer interprets user intentions, extracts relevant parameters, and generates human-readable responses.

2.4.1. Large Language Model Integration

The framework leverages state-of-the-art Large Language Models (LLMs) to provide conversational capabilities. In the prototype implementation, we integrate Google's Gemini 2.5 Flash Lite model via API, chosen for its:

- Instruction following: Ability to understand and execute complex instructions that require multiple actions.
- Contextual understanding: Capacity to maintain multi-turn dialogues.
- Structured output generation: Support for generating formatted data (e.g., YAML) alongside natural language.

The conversational agent, powered by a Large Language Model (LLM), operates through a carefully designed prompt engineering strategy that includes system instructions defining the agent's role, capabilities, and behavioral guidelines; a schema specification providing formal descriptions of configuration parameters and valid value ranges; and tool definitions specifying the functions the model can invoke (e.g., modifying the configuration or executing the simulation). The system prompt (detailed in Appendix A.1) was developed iteratively by the research and development team, without the involvement of independent end-users. The optimization process was qualitative, focusing on ensuring the correct invocation of tools under varying conditions and refining user intent detection. Particular attention was given to handling multi-intent queries, enabling the agent to execute several concurrent instructions or chained actions within a single conversational turn.

2.4.2. Natural Language Understanding Pipeline

Internally, the LLM of the agent processes user messages by recognizing their intent (such as configuring a simulation, querying results, or modifying parameters), extracting relevant entities like dates, numerical values, categorical options, and constraints, validating this information against allowed ranges and business rules, and finally generating actions by translating the validated parameters into structured tool calls that interact with the simulation engine.

Error handling mechanisms ensure graceful degradation when the LLM fails to extract parameters correctly, prompting the user for clarification rather than proceeding with invalid configurations.

2.5. System Integration Architecture

The conversational AI agent and digital twin communicate through two primary integration mechanisms:

2.5.1. Configuration File Management

The conversational interface manipulates a structured configuration file that parameterizes simulation execution. This file includes:

- Event metadata: Date, time, location, event type.
- Attendance projections: Expected visitor counts, demographic distributions.
- Environmental conditions: Weather forecasts, external events.
- Operational parameters: Resource availability, staff schedules, pricing policies.

The conversational agent updates this file according to user requests, while the digital twin uses it to configure the prediction process. This approach based on file exchange keeps the components loosely connected, so each one can evolve separately as long as the structure of the configuration remains consistent.

2.5.2. RESTful API Communication

The system exposes two main endpoints that support the simulation workflow: one for initialization and one for results retrieval. The initialization endpoint receives a YAML configuration, validates it, assigns a unique simulation identifier, and starts the process, returning the identifier along with a status indicating that execution has begun. The results endpoint allows checking the state of a simulation using its identifier and, once completed, provides access to a generated report with the results, while ongoing or failed executions return only status information. These endpoints were created using FastAPI [32] and are accessed by the conversational layer, which links user requests to system execution in a seamless way.

2.6. Visualization Layer

The visualization layer presents simulation results in formats suitable for decision-making. This layer must balance comprehensiveness with clarity, ensuring that stakeholders with varying levels of technical expertise can interpret outputs effectively.

Multi-Modal Result Presentation and Dashboard Design

The framework supports a multi-modal visualization strategy that enables users to access simulation outputs at different levels of detail and interactivity. Results can be presented directly within the conversational interface as structured tabular data, including raw prediction values, comparison metrics, and statistical summaries, or as programmatically generated visualizations illustrating trends, distributions, and comparative analyses. In addition, the framework supports interactive dashboard environments, such as Microsoft Power BI [33], which provide advanced analytical capabilities including filtering, drill-down exploration, and cross-dimensional analysis through persistent links generated by the conversational agent.

The dashboard layer is designed following established decision support system principles to facilitate rapid interpretation and operational analysis. Key performance indicators (KPIs) are prominently displayed using suitable visual encodings, such as gauges, sparklines, and heat maps, enabling at-a-glance comprehension of relevant operational conditions. Contextual navigation mechanisms allow users to transition from high-level overviews to detailed analyses through intuitive interactions, while temporal visualizations support the identification of trends, anomalies, and scenario-based variations over time.

This multi-modal approach allows users to quickly assess high-level results within the conversational environment while retaining the flexibility to perform deeper exploratory analyses in dedicated visualization platforms. Dashboard updates are triggered multiple times per day, ensuring that displayed information reflects recently generated predictions without requiring manual refresh cycles.

2.7. Technology Readiness Level 4 (TRL4): Implementation Details

This subsection describes the technical realization of the CDTF through a TRL-4 prototype developed for operational planning of sporting events in tourist destinations. The implementation demonstrates the framework's feasibility while validating core functional requirements in a controlled environment.

2.7.1. Validation Mechanisms

The prototype implements multiple security layers appropriate for TRL-4 validation:

- Input validation: Pydantic schemas enforce type checking, range constraints, and structural requirements on all API inputs, preventing injection attacks and malformed requests [34].

- Database access control: PostgreSQL roles and permissions restrict write access to authorized services, with read-only credentials used for visualization layers.

While production deployment would require additional security measures (e.g., OAuth 2.0 authentication, end-to-end encryption, comprehensive audit logging), the current implementation provides sufficient protection for controlled testing environments.

2.7.2. Framework Extensibility and Generalization

A key design consideration is the framework's adaptability to domains beyond event planning. The modular architecture facilitates extension in several dimensions:

- Data source integration: New ETL connectors can be added to the data layer without modifying downstream components, enabling connection to domain-specific databases or APIs.
- Model diversity: The Simulation Engine's standardized interface allows substitution of Random Forest models with alternative algorithms (e.g., neural networks, agent-based models) appropriate for different prediction tasks.
- Conversational capabilities: The prompt engineering approach can be customized for specific domains by updating system instructions and few-shot examples, without changing the underlying LLM integration.
- Visualization templates: Power BI dashboards can be adapted with domain-specific metrics and visual designs.

This extensibility positions the framework as a general-purpose blueprint for developing conversational digital twins across manufacturing [9], healthcare [11], urban planning [10], and other domains requiring simulation-based decision support.

3. Artifact Evaluation: Sporting Event Simulation Case Study

Following the Design Science Research methodology, this section focuses on the Artifact Evaluation step. It presents the evaluation of the developed CDTF through a case study focused on operational planning for large-scale sporting events at a football stadium in a tourist destination. The objective of this validation is not to demonstrate production-grade forecasting accuracy, but to assess the feasibility and usefulness of the proposed architecture at TRL4 level. This study is explicitly framed as a feasibility and workflow-integration investigation rather than a validation of predictive accuracy for operational deployment. In particular, the evaluation focuses on three complementary dimensions: the ability of the conversational agent to transform natural language requests into valid simulation configurations, the computational feasibility of executing complete simulation workflows within operationally acceptable response times, and the capacity of the system to generate decision-support outputs for resource allocation and logistics management.

3.1. Artifact Evaluation Against State-of-the-Art Approaches

To properly evaluate the proposed architecture, we establish a comparative framework against competing approaches identified in the literature, explicitly including recent advancements and fundamental analytical frameworks suggested in the state of the art [35–39]. As outlined in our requirements (RQ1 and RQ3), a set of desired features is necessary for an effective conversational digital twin.

Table 1 presents a design-science evaluation comparing our artifact (CDTF) against baseline methodologies: traditional manual configuration, off-the-shelf LLMs without rule-based validation (as tested in Section 3.2), and existing SOTA predictive digital twins and analytical approaches as proposed in recent literature [35,36,38,39].

Table 1. Comparison of desired architectural features against competing approaches.

Desired Feature (Requirements)	Traditional GUI/ Manual Config.	Off-the-Shelf LLM (Baseline)	Existing SOTA Approaches [35,36,38,39]	Proposed CDTF (Our Artifact)
Natural Language Scenario Configuration	No	Yes	No	Yes
Strict Parameter Validation & Schema Enforcement	Yes	No *	Yes	Yes
Automated End-to-End Simulation Workflow	No	No	Yes	Yes
Multi-modal Result Visualization	No	No	No **	Yes
Decoupled Architecture for Cross-Domain Transfer	No	Yes	No	Yes

* Prone to hallucination and extraction errors without architectural constraints. ** Typically restricted to static dashboards without conversational multi-modal feedback.

3.2. Case Study Context

The prototype was developed for the Gran Canaria Stadium, a 32,400-seat football arena hosting regular matches in Las Palmas, Canary Islands, Spain. The operational planning challenge involves coordinating multiple interdependent resources, including food and beverage inventory, merchandising stock, parking capacity, cleaning staff and inventories, and security personnel. Traditional planning relies on fixed capacity allocations based on average historical attendance, resulting in either resource wastage during low-attendance events or service degradation during high-demand matches [16].

The digital twin was calibrated using a combination of historical match records from three complete football seasons (2021–2024) encompassing 114 matches, publicly available contextual data, and synthetically generated operational variables, since real concession, staffing, or security datasets were not available. External data sources include AEMET (Spanish Meteorological Agency) for weather data, Google Trends for search interests, and occupancy rates for tourist accommodations in the proximities obtained from the Institute of Statistics of the Canary Islands (ISTAC) and the National Institute of Statistics of Spain (INE).

Although the framework includes mechanisms for monitoring predictive performance through standard regression metrics, these metrics are not reported as evidence of real-world accuracy in this validation because several operational target variables were generated synthetically.

Generation of Synthetic Operational Data

To ensure complete transparency and reproducibility regarding the TRL4 validation, it is necessary to detail how the synthetic operational variables were generated. The synthetic data set was created using a combination of the Python 3.11 Faker library [40] (to generate realistic data schemas resembling point-of-sale and operational logs) and heuristic rule-based estimations defined by the research team.

It is important to explicitly state that these synthetic generation rules were not validated by stadium domain experts, nor were they derived from historical proxy distributions. Their sole purpose was to generate structurally sound, logically plausible data pipelines to test the end-to-end architectural workflow (i.e., verifying that the simulation engine could ingest data, execute the models, and output results to the visualization layer without architectural failures). Table 2 summarizes the main synthetic variables, their generation rules, and the underlying assumptions used to populate the simulation databases.

Table 2. Summary of synthetic variables, generation rules, and underlying assumptions.

Variable Category	Generation Method	Rules & Ranges	Underlying Assumption (Unvalidated)
Attendance Breakdown (Season tickets, Locals, Tourists)	Heuristic estimation	Season tickets: ~65% of total. Locals: ~25%. Tourists: ~10%.	Proportions reflect a plausible demographic split for a major sporting event in a highly touristic destination.
Concession Sales (Food, Drinks, Merch)	Python Faker & Rule-based	Transaction logs generated with Faker. Volume scaled linearly with attendance (0.5 to 1.5 items per attendee).	Higher attendance directly correlates with proportional increases in food, beverage, and merchandising transactions.
Queue Lengths	Rule-based	Base time (0–2 min) + penalty multiplier based on stadium occupancy (up to 15 min).	Queues grow exponentially as the stadium approaches its maximum capacity.
Parking Availability	Rule-based	100% availability minus a factor of local and tourist attendance arriving by car.	Season ticket holders and locals represent the majority of parking pressure.

3.3. Conversational Agent Validation

The conversational interface enables planners to configure simulation scenarios through natural language dialogue. Table 3 presents representative interaction examples demonstrating intent recognition, parameter extraction, and system responses.

Table 3. Representative conversational interactions with the digital twin agent.

User Input	System Response
“Create configuration for Las Palmas vs Sevilla match with 32,400 attendance”	Configuration created with <code>simulation_id = 12</code> . Attendance set to 32,400. Other parameters initialized to historical averages. Simulation ready to execute.
“Simulate 10% more tourists than usual with rainy weather”	Configuration updated: <code>tourist_attendance</code> increased by 10%, <code>weather_condition</code> set to “rain” (probability 0.8). <code>simulation_id = 13</code> assigned.
“What happens if it’s a high-risk match on a sunny day with 12,300 spectators?”	Configuration created: <code>attendance = 12,300</code> , <code>risk_level = high</code> , <code>weather = sunny</code> . Note: Other parameters will be imputed from historical data. Proceed with simulation?
“Show me the predictions for simulation 13”	Simulation 13 status: finished. Downloading Excel report with predictions. Dashboard link: [PowerBI URL].

These interactions highlight the system’s capability to: (1) extract numerical values and categorical parameters from natural language, (2) handle incomplete specifications by applying intelligent defaults, (3) maintain conversation context across multiple turns, and (4) provide both immediate feedback and detailed analytical outputs.

The examples also illustrate the practical role of the conversational layer as an abstraction mechanism over the simulation configuration process. Instead of requiring users to manually edit structured configuration files or interact directly with API endpoints, the agent converts operational questions into executable simulation scenarios. This is particularly relevant in stadium planning contexts, where decision-makers may need to rapidly explore alternative assumptions regarding attendance, weather, match risk, tourist inflow, or resource availability without technical knowledge of the underlying simulation engine.

The results presented in Table 4 indicate that the conversational layer successfully identified user intent in the majority of the evaluated prompts and accurately extracted all explicitly specified parameters in the tested cases. Parameter extraction achieved an

accuracy of 100% across 50 evaluated parameters, encompassing numerical inputs (e.g., expected attendance), categorical variables (e.g., weather conditions or match risk levels), and relative adjustments (e.g., percentage increases in tourist attendance). This level of accuracy is particularly significant, as errors in parameter extraction could prevent the digital twin simulation engine from initiating and hinder system usability for non-technical users, who might otherwise encounter unclear or difficult-to-interpret error messages.

Table 4. Conversational agent validation results.

Metric	Value	Evaluation Setup
Intent recognition accuracy	93.3%	30 user prompts
Parameter extraction accuracy	100%	50 extracted parameters
Invalid input detection rate	100%	20 cases asking for invalid values
Full simulation workflow completion rate	100%	10 scenarios

The system also detected all invalid values in the evaluated test cases. These cases included out-of-range attendance values, unsupported weather categories, inconsistent scenario definitions, and requests involving unavailable parameters. Instead of executing invalid simulations, the agent either rejected the input or requested clarification. This behavior supports the use of the conversational layer as a controlled interface rather than as an unrestricted natural language wrapper over the simulation engine. Finally, the full simulation workflow completion rate reached 100% across the 10 evaluated scenarios. In this context, a completed workflow means that the user request was transformed into a valid configuration, the simulation process was executed, and the results were made available through the corresponding reports and dashboard output. This confirms that the prototype implemented successfully integrates the conversational interface, configuration management, simulation execution, result persistence, and visualization components into a single end-to-end workflow.

Although the evaluation set of 30 prompts might appear limited in a general NLP context, the research team specifically designed it to cover the full operational scope of the framework at a TRL-4 level. The agent's action space is highly constrained by design: its core capabilities are strictly limited to viewing current settings, updating parameters within a defined schema, triggering the simulation API, and providing the resulting report links. Because the variance in user interaction is primarily syntactic rather than functional, expanding the evaluation set yields diminishing returns once the system demonstrates a reliable mapping of natural language to these specific, bounded actions.

Despite these high performance metrics, the evaluation revealed specific limitations in handling ambiguous or out-of-scope requests. Complex operational adjustments still require human oversight or a wider set of configuration parameters, as demonstrated by the following failed cases observed during testing:

- **Subjective constraints:** When presented with an underspecified request such as *"Change the hotel occupancy rate to reflect the off-season"*, the agent failed to request additional information and instead inferred a value autonomously, responding *"Great, I'll change the hotel occupancy rate to 35% since it's the off-season"*. In this scenario, the expected behavior would be to initiate a clarification loop and request an explicit occupancy value from the user, rather than making unsupported assumptions about simulation parameters.
- **Out-of-scope requests:** When asked to *"Change the weather to indicate that there is a dust storm and visibility is low"*, the agent responded *"I'm sorry, but the option to add a dust storm as a weather condition during the event is not included in the settings. Please suggest a valid value referring to rain probability."* Although the request referred to a parameter

outside the supported configuration schema, the response demonstrates a limitation in handling out-of-scope instructions, as the agent failed to clearly explain that the requested modification could not be represented within the available simulation parameters.

To establish a baseline comparison, the same evaluation set of 30 prompts was processed by an off-the-shelf, unconstrained LLM agent lacking the framework's strict rule-based architecture and schema validation. This methodological choice aimed to isolate the contribution of our specific architectural constraints. While this baseline system produced valid configurations in only 56.67% of cases, our framework achieved much higher completion rates. In contrast, our framework enforces strict, lightweight schema validation (using Pydantic) at the extraction layer. If the LLM generates an invalid configuration, the schema validation reliably forces the agent to retry and correct the generation until it successfully aligns with the system's requirements, guaranteeing that the simulation engine never receives malformed data.

However, we fully acknowledge that the reported performance figures (e.g., 93.3% intent recognition accuracy and 100% extraction accuracy) must be interpreted with caution. The evaluation process was conducted on a limited test set of 30 prompts that were tightly constrained to the operational scope of the TRL-4 prototype. While the research team designed this set to cover the functional action space of the framework, expanding the evaluation set yields diminishing returns only when the variance in user interaction is primarily syntactic. Ultimately, these results demonstrate architectural feasibility and correct workflow integration within a highly controlled environment, rather than providing a definitive quantitative benchmark for open-ended conversational interactions. Additionally, the latency performance of the prototype during this validation phase is summarized in Table 5.

Table 5. End-to-end response times observed during prototype validation.

Metric	Value	Evaluation Setup
Average conversational response time	<6 s	50 cases
Average parameter modification time	5 s	30 cases
Average complete simulation workflow time	1:07 min	10 scenarios
Excel report generation time	2 s	10 scenarios

The observed response times, summarized in Table 3, indicate that the framework can support interactive scenario exploration. General conversational responses and parameter modification requests were completed in a few seconds, which is compatible with a dialogue-based planning workflow. Complete simulation workflows required approximately one minute on average, including configuration handling, simulation execution, result persistence, and report availability. This execution time is acceptable for pre-event operational planning, where users typically compare a limited number of alternative scenarios rather than requiring real-time control-loop execution.

An important observation is that simulation time did not increase substantially when attendance values were raised up to the maximum capacity of the selected stadium. This suggests that, for the evaluated range of scenarios, the computational cost of the prototype is dominated by fixed workflow components, such as model loading, simulation of attendee behavior, and report generation, rather than by the number of simulated spectators alone. As a result, full-capacity scenarios can be explored without a noticeable degradation in user experience.

Excel reports were generated almost immediately once the simulation output was available, within the normal response time perceived by the user when interacting with the

agent. This result is relevant from an operational perspective, since it allows planners to obtain portable reports for offline analysis, communication with stakeholders, or archival purposes without introducing an additional bottleneck in the workflow.

3.4. Scope of the Validation

The validation was designed as a system-level assessment of the proposed architecture rather than as an isolated benchmark of individual predictive models. This decision is motivated by the nature of the available data. While the prototype incorporates real historical and contextual information, including match records, weather data, tourism indicators, and search interest signals, several fine-grained operational variables had to be synthetically generated due to the lack of access to complete stadium management datasets. These include detailed concession sales, movement patterns, queue observations, cleaning operations, parking occupation dynamics, and security incidents.

Consequently, the numerical results presented in this section should be interpreted as evidence of feasibility, integration, and operational plausibility. They demonstrate that the framework can receive natural language planning requests, generate valid simulation configurations, execute the corresponding digital twin workflow, persist the outputs, and expose the results through reports and dashboards. They should not be interpreted as evidence of final predictive accuracy in a production environment.

This validation scope is consistent with the TRL4 nature of the prototype, where the main objective is to demonstrate that the core technological components operate together in a controlled experimental environment. A higher-readiness validation would require access to real transaction-level sales data, access-control logs, parking occupancy measurements, staffing records, and incident management data collected during live events.

3.5. Simulation Outputs and Predictions

Figure 2 illustrates sample outputs generated for a simulated scenario. In line with the validation scope described above, these outputs are interpreted as operational decision-support indicators rather than as validated production forecasts. The digital twin generates estimates for multiple operational variables, including:

- Attendance distribution: Total spectators, demographic breakdown (local vs. tourist), arrival time patterns.
- Resource consumption: Food items (sandwiches, snacks), beverages (soft drinks, beer), merchandising (jerseys, scarves).
- Service Utilization: Queue lengths at concession stands, restroom occupancy, parking lot capacity.
- Operational costs: Staffing requirements for cleaning, security, and customer service.

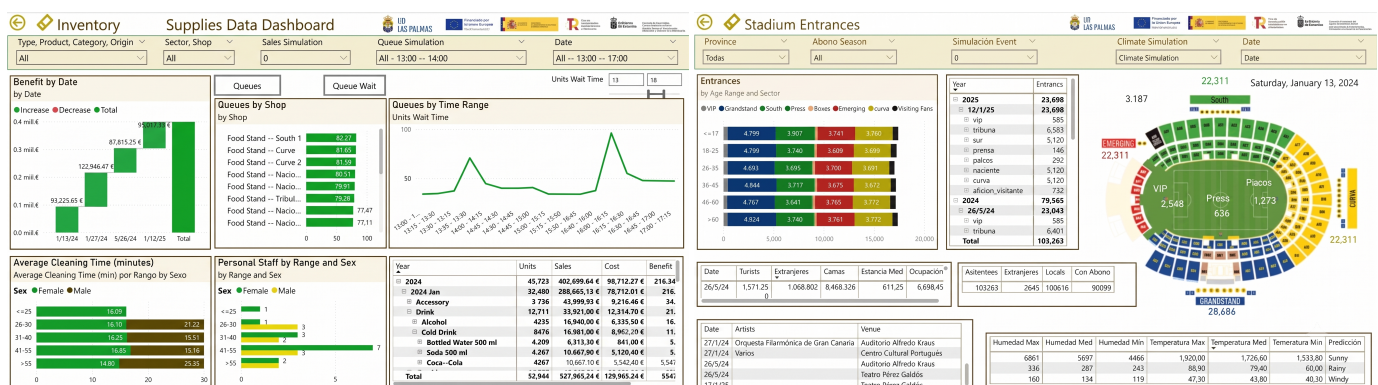


Figure 2. Sample prediction outputs showing forecasted product consumption and simulated attendance patterns.

The Power BI dashboard provides interactive exploration capabilities, allowing planners to filter by time period, compare multiple scenarios, and drill down into specific resource categories. In the implemented prototype, simulation outputs are persisted in the operational database and subsequently reflected in the dashboard through scheduled refresh mechanisms. This allows the dashboard to act as a decision-support layer where users can compare baseline and alternative scenarios without directly accessing the underlying database or simulation engine.

Performance Analysis

The performance results indicate that the implemented prototype is suitable for iterative pre-event planning. Conversational interactions were completed within a few seconds in most cases, while complete simulation workflows required approximately one minute on average. From an operational perspective, this response time enables planners to test several alternative configurations during the same planning session, for example comparing baseline, high-attendance, rainy-weather, or high-risk match scenarios.

The limited variation in execution time across attendance levels suggests that the current implementation is not strongly constrained by the number of spectators up to the stadium capacity considered in this case study. Instead, the overall latency appears to be primarily associated with orchestration overhead, model execution, data persistence, and report generation. This behavior is appropriate for the current TRL4 prototype, although future versions could further reduce latency by parallelizing independent prediction tasks, optimizing database access patterns, and caching frequent conversational or configuration operations.

The generation of Excel reports reinforces the suitability of the framework for decision-support workflows in which results need to be shared, archived, or further analyzed outside the conversational interface.

3.6. Scenario Exploration and Operational Plausibility

Beyond execution time and workflow completion, the prototype was evaluated according to its ability to support scenario exploration. The purpose of this assessment was to verify whether changes in the input configuration produced coherent variations in the generated operational indicators. This is particularly important in the absence of complete ground-truth operational datasets, since the value of the framework at TRL4 level lies in enabling planners to compare plausible alternatives under controlled assumptions.

The tested scenarios included variations in expected attendance, weather conditions, tourist attendance, match risk level, and concurrent events in the surrounding area. In qualitative terms, the generated outputs followed the expected operational direction. Higher attendance scenarios increased estimated food and beverage demand, parking pressure, restroom utilization, and cleaning requirements. High-risk match configurations increased the recommended security allocation and modified the expected pressure on access-control operations. Rainy-weather scenarios produced changes in mobility and service utilization patterns, while increases in tourist attendance affected consumption-related indicators and merchandising demand.

These results suggest that the digital twin can be used as a what-if analysis tool for pre-event planning. Rather than producing a single static forecast, the framework enables users to compare multiple operational hypotheses and observe their expected impact on resource allocation. This capability is especially relevant for stadium operations, where planning decisions are influenced by multiple interacting factors and where domain experts often need to reason under uncertainty.

4. Discussion

This section interprets the validation results of the CDTE. The discussion focuses on the implications of using this system, the architectural lessons derived from the case study, the limits imposed by the partial use of synthetic operational data, and the future developments required before deployment in real stadium operations.

4.1. Interpretation of Validation Findings

The validation results indicate that the proposed framework is technically feasible as a TRL4 prototype for conversational simulation-based planning. The agent achieved 93.3% intent recognition accuracy and always completed the full simulation workflow, confirming that the prototype integrates natural language configuration, scenario generation, simulation execution, result persistence, report generation, and dashboard visualization into a single operational pipeline.

These results should be interpreted primarily as evidence of system-level feasibility rather than as a final predictive validation. The prototype demonstrates that non-technical users can configure and execute operational scenarios without directly editing configuration files or interacting with API endpoints. This is a relevant contribution for digital twin adoption, since one of the practical barriers to simulation-based planning is the gap between domain expertise and the technical knowledge required to operate simulation tools [8].

The observed response times also support the suitability of the framework for pre-event planning workflows. General conversational responses were completed in less than six seconds on average, while complete simulation cycles took approximately one minute. These results are compatible with iterative planning sessions in which users compare a small number of alternative scenarios prior to an event.

The framework's modular architecture proved effective in isolating component responsibilities. This separation of concerns is a key design principle that facilitates iterative improvement [14].

A relevant observation from the validation is that execution time did not vary substantially when the simulated attendance increased up to the maximum capacity of the selected stadium. This suggests that, within the evaluated range, the total workflow time is attributable to fixed processes rather than to the attendance value alone. This behavior is acceptable for the current prototype, although future deployments could still benefit from optimization through caching, parallel execution of independent tasks, and infrastructure scaling [41].

4.2. The Democratization of Simulation Through Conversation

A central contribution of this work is the use of generative AI as an accessibility layer for simulation-based decision support. Traditional digital twin systems often require users to understand data schemas, parameter definitions, configuration files, or programming interfaces [7]. In contrast, the proposed framework allows domain experts to express planning intentions using operational language, such as asking the system to simulate a high-attendance rainy match or to modify the expected number of tourists.

This interaction model does not eliminate the need for rigorous simulation models, data validation, or expert interpretation. Instead, it changes how users access these capabilities. The conversational layer supports a human-in-the-loop approach in which the system assists planners in scenario exploration [9,11,12].

4.3. Exploratory Anecdotal User Observations

During the development of the prototype, we gathered early exploratory data from three non-technical users who interacted with the system. It is important to emphasize that

this does not constitute a systematic evaluation of human–computer interaction (HCI) or a formal usability study. Rather, these observations provide preliminary qualitative context regarding how non-technical stakeholders might experience the conversational interface.

These early observations highlighted potential perceived strengths:

- Accessibility: Users noted that they could configure simulations without directly editing configuration files or calling API endpoints.
- Transparency: The explicit feedback from the agent regarding the interpreted parameters and calculation of the default values was perceived as helpful to understand how requests were translated into simulation inputs.
- Actionability: The generated outputs were considered sufficiently granular to support preliminary discussions about resource planning.
- Iterative exploration: Users appreciated the ability to modify assumptions between turns to compare alternative scenarios.

Although these limited data points align with the central design assumption—that conversational interaction can lower the technical barrier to simulation-based planning—they must be treated strictly as exploratory insights. In future work, a rigorous and systematic HCI evaluation involving a representative sample of operational planners and stakeholders will be required to formally assess usability, user trust and the impact of the system on decision-making processes.

4.4. Framework Generalizability and Transfer Potential

Although the prototype was validated in the context of sporting event operations, the proposed architecture is not specific to football stadiums. Its four-layer structure can be transferred to other operational planning domains where historical or contextual data is available, alternative scenarios need to be explored, and stakeholders require decision-support outputs without directly interacting with simulation code.

Examples of potential transfer include:

- Venue and event management: concerts, festivals, trade fairs, and conference centres, where planners must estimate attendance, queues, staffing, security, cleaning, and resource consumption.
- Tourism and destination management: planning services around seasonal demand, weather conditions, accommodation occupancy, visitor flows, and concurrent events.
- Transport and mobility hubs: airports, ports, railway stations, or bus terminals, where passenger flow, parking, staffing, and congestion management are central planning problems [10].
- Healthcare capacity planning: hospital departments or emergency units, where managers need to explore patient flow, staff allocation, bed availability, and waiting times [11].
- Logistics and warehousing: facilities where demand variability, stock levels, workforce planning, and service times can be analysed through scenario-based simulation [15].

In all these cases, the reusable contribution is not a specific predictive model, but the architectural pattern: a data layer connected to heterogeneous sources, a simulation engine adapted to the domain, a conversational layer for scenario configuration, and a visualization layer for decision support. The LLM’s generalization capability means that prompt engineering suffices to adapt the conversational layer to new domains [17,18]. The main effort required for transfer would be the replacement or adaptation of domain-specific models, data connectors, validation rules, and dashboard indicators.

4.5. Limitations and Boundary Conditions

Despite promising validation results, the framework operates within important constraints that define its current applicability and highlight areas for future development.

4.5.1. Technology Readiness Level

The prototype corresponds to a TRL4 validation, where the objective is to demonstrate that the main technological components operate together in a controlled environment. The system has not yet been validated under live operational conditions or used to make decisions with real consequences during an event. Moving toward higher readiness levels would require:

- Access to real operational data: integration with point-of-sale systems, access-control logs, parking sensors, staffing systems, cleaning records, and incident management platforms.
- Field validation: comparison of simulated outputs with observations collected during real matches or events.
- Production infrastructure: deployment in a scalable and monitored environment with logging, backup mechanisms, and failure recovery.
- Real-time integration: Connectivity to live data streams rather than periodic batch updates.
- Multi-user operation: support for different stakeholder roles, permissions, and simultaneous planning sessions.
- Data governance and compliance: alignment with GDPR requirements, especially if future versions process personal, behavioural, or location-related data.

4.5.2. Use of Synthetic Operational Data

At the current technology readiness level, the use of synthetic operational variables enabled controlled testing of the complete conversational digital twin pipeline. This approach supported the evaluation of core framework functionalities, including configuration handling, simulation execution, report generation, dashboard integration, and scenario comparison workflows. The availability of synthetically generated data allowed the prototype to demonstrate end-to-end architectural feasibility and workflow integration in the absence of fully accessible operational datasets.

4.5.3. Economic Indicators and Cost Modelling

The current prototype focuses primarily on operational quantities, such as expected stock volumes, staffing requirements, parking pressure, and resource utilization. However, decision-making in stadium operations also depends on economic indicators.

For example, the system could compare the expected revenue from food and beverage sales against procurement and staffing costs, estimate the cost of overstocking or understocking, or quantify the economic impact of alternative staffing strategies. This would extend the framework from operational planning toward cost-aware decision support, allowing users to evaluate scenarios according to both logistical feasibility and expected economic performance.

4.5.4. Conversational Understanding Limits and Tool Use

The conversational layer achieved 93.3% intent recognition accuracy in the validation prompts, but this does not imply perfect understanding of arbitrary natural language. The main limitations identified during the performed qualitative assessment were related to complex or ambiguous user requests. In some cases, users had to rephrase queries that combined multiple constraints or used vague operational terminology. Complex requests

involving multiple constraints, ambiguous terminology, or implicit assumptions may still require clarification or rephrasing. This limitation is especially relevant in operational contexts, where different stakeholders may use informal or organization-specific vocabulary to refer to the same planning variables.

In addition, the conversational agent could be extended with tool-use capabilities beyond configuration editing and simulation execution. For instance, Model Context Protocol (MCP) servers or equivalent tool interfaces could allow the agent to query the operational database using SQL, retrieve previous simulation results, compare historical scenarios, inspect dashboard metadata, or trigger additional data validation routines.

These extensions would increase the functionality of the conversational layer, but they would also require stronger governance mechanisms. Because the framework relies on an interface mediated by a large language model capable of modifying simulation parameters, it is exposed to specific AI-related vulnerabilities. A concrete governance strategy must explicitly address risks such as prompt injection (where malicious inputs attempt to hijack the agent's instructions), unauthorized tool use (where the agent attempts actions beyond its permitted scope), incorrect parameter mapping (where the LLM misinterprets user intent and maps it to the wrong variables), and hallucinated assumptions (where the agent invents non-existent constraints or data points). Mitigating these risks requires strict, schema-driven input validation (such as our Pydantic layer that forces correction of malformed data), read-only database roles for the agent, explicit clarification loops, and auditable execution logs.

4.5.5. Scope of Automation

The framework automates scenario configuration, simulation execution, report generation, and dashboard integration, but it does not automate final operational decision-making. The generated outputs should be interpreted strictly as decision-support inputs that help planners compare alternatives, identify potential bottlenecks, and reason about resource allocation. Consequently, a strict human-in-the-loop protocol is mandatory. The necessity of human approval before any operational decision is a foundational safety constraint of this architecture. The system must not automatically trigger real-world actions (e.g., ordering stock, deploying personnel, or altering security protocols) based on its own simulation outputs. Final decisions regarding procurement, staffing, security, and mobility management remain the sole responsibility of human experts.

This distinction is particularly important because some inputs are uncertain and some operational variables are synthetic in the current prototype. The system can help structure and accelerate planning discussions, but it should not be interpreted as an autonomous planning authority. This design philosophy aligns with best practices in human–AI collaboration, where systems augment rather than supplant expert judgment [42].

4.5.6. Computational and Deployment Requirements

The current implementation relies on external LLM API access and a prototype-oriented execution environment. Although the observed response times were acceptable for pre-event planning, production deployment would require a more detailed assessment of cost, reliability, latency, and data governance. Depending on the operational context, organizations could consider different deployment strategies, including API-based access to more capable LLMs, smaller self-hosted models for constrained tasks such as parameter extraction, or hybrid approaches that combine both.

The choice of deployment architecture would also depend on privacy and integration requirements. Stadium operators or public institutions may prefer on-premises or private-

cloud deployments when sensitive operational data are involved, whereas less sensitive scenario exploration could rely on managed cloud services.

4.6. Implications for Digital Twin Evolution

The proposed framework reflects a broader evolution in digital twin systems: from technical simulation environments used mainly by specialists toward interactive decision-support tools accessible to operational stakeholders. In this view, the digital twin is not only a computational representation of a physical system, but also an interface for exploring operational assumptions, comparing scenarios, and communicating results through the use of LLMs [43].

The case study illustrates this shift in the context of stadium planning. A planner can describe an event scenario in natural language, the system converts the request into structured parameters, the simulation engine generates operational indicators, and the outputs are exposed through reports and dashboards. This interaction pattern can make simulation-based planning more usable, especially in organizations where domain expertise is distributed across users who may not have technical modelling skills.

At the same time, this evolution introduces new research and engineering challenges. Conversational digital twins must provide transparent parameter interpretation, prevent invalid configurations, expose uncertainty appropriately, and maintain a clear boundary between decision support and automated decision-making. Addressing these challenges will be necessary for moving from controlled TRL4 prototypes toward trusted operational systems.

4.7. Limitations and Future Work

The main limitation of the current work is that the validation was conducted at TRL4 level and relied partly on synthetically generated operational variables. Although the prototype integrates real contextual and historical sources, including match records, weather data, tourism-related statistics, and search trend indicators, detailed operational datasets such as transaction-level concession sales, queue measurements, parking occupancy, staffing logs, cleaning records, and security incidents were not available. As a result, several variables that are critical for event planning had to be simulated to enable end-to-end testing of the framework.

Therefore, the results should be interpreted as evidence of architectural feasibility, workflow integration, and operational plausibility, rather than as proof of production-level predictive accuracy. However, the use of synthetic data was appropriate at this technology readiness level, as it enabled controlled evaluation of the complete conversational digital twin pipeline, including configuration handling, simulation execution, report generation, dashboard integration, and scenario comparison workflows.

Future work will proceed along three main directions.

First, the framework should be validated with real operational datasets collected during live events. This would enable the transition from system-level feasibility assessment to predictive and operational validation. In particular, future studies should include quantitative comparisons between generated outputs and observed data related to resource consumption, attendance flows, parking usage, staffing needs, and incident patterns. Longitudinal evaluations assessing decision quality improvements and operational cost savings would further strengthen evidence of practical value.

Second, the simulation engine should be extended to support a broader range of modeling paradigms. This includes agent-based simulation for crowd dynamics, discrete-event simulation for service processes, and optimization techniques for resource allocation. In addition, the integration of economic indicators—such as product costs, selling prices,

staff costs, waste costs, and service-level penalties—would allow the system to evaluate scenarios not only from an operational perspective, but also in terms of financial impact.

Third, the conversational layer should be enhanced to improve robustness and analytical capability. This includes the use of more powerful LLMs, domain-specific prompt engineering, structured clarification dialogues, and curated examples derived from real planner interactions to better handle complex or ambiguous operational requests. Furthermore, tool-use capabilities such as MCP-based integrations could enable the agent to query operational databases, retrieve historical simulations, generate SQL-based summaries, compare scenarios, and provide richer interpretations of dashboard outputs. These extensions should be implemented with appropriate validation mechanisms, access control policies, and auditability to ensure safe operation.

5. Conclusions

This paper introduced the CDTF, an architectural approach that integrates generative AI with simulation systems to make operational planning tools more accessible to non-technical users. By enabling natural language configuration of simulation scenarios, the framework addresses one of the practical barriers that limits the adoption of digital twins in operational environments: the need for users to interact with complex configuration files, APIs, or modelling tools.

5.1. Summary of Contributions

The research contributions span conceptual, technical, and validation dimensions:

Conceptual framework: We presented a four-layer architecture composed of a data layer, a simulation engine, a conversational AI layer, and a visualization layer. This structure provides a blueprint for building conversational digital twins in which simulation complexity is decoupled from user interaction. As a result, domain experts can configure and explore scenarios through natural dialogue rather than through programming interfaces or manual configuration files.

Technical implementation: We implemented a TRL4 prototype for operational planning in football stadium events. The prototype integrates a Large Language Model, Gemini 2.5 Flash Lite, for natural language interaction; Random Forest models and discrete-event simulation components for scenario computation; a structured configuration and simulation identification mechanism for traceability; and Power BI and Excel reports for result visualization and sharing. The implementation demonstrates that these components can be orchestrated into an end-to-end workflow from conversational input to simulation execution and decision-support output.

Validation case study: We validated the prototype in a case study focused on the Gran Canaria Stadium. The evaluation showed that the conversational agent correctly recognized user intent in 93.3% of the evaluated prompts, extracted all explicitly stated parameters in the tested cases, detected invalid inputs in all evaluated invalid scenarios, and completed the full simulation workflow in all 10 tested scenarios. General conversational responses were produced in less than six seconds on average, while complete simulation workflows required approximately 1:07 min. These results support the feasibility of the framework for interactive pre-event planning at TRL4 level.

5.2. Impact and Implications

The main implication of this work is that conversational interfaces can facilitate the access to simulation-based decision support. In the evaluated prototype, users were able to configure scenarios, modify assumptions, execute simulations, and access results without directly interacting with the underlying configuration files, APIs, database, or simulation

engine. This interaction model is especially relevant for operational planning contexts in which domain experts need to compare alternatives but may not have modelling or programming expertise.

For stadium and event operations, the framework can support proactive scenario exploration around attendance, weather, tourist presence, risk of incidents associated with the match, stock requirements, parking pressure, cleaning needs, and security planning. More generally, the proposed architecture can be transferred to other domains where planners need to explore what-if scenarios using heterogeneous data sources and simulation models, such as venue management, tourism planning, transport hubs, healthcare capacity planning, or logistics operations.

Beyond its immediate application, the framework contributes to research on human–AI collaboration by illustrating how large language models can mediate between domain experts and specialized computational systems [42,43]. The conversational layer does not replace simulation models or expert judgement; rather, it provides a more accessible interface for configuring scenarios and interpreting decision-support outputs.

5.3. Concluding Remarks

The integration of conversational AI with digital twin technology represents a promising direction to make simulation-based planning more accessible to operational stakeholders. The CDTF provides a conceptual architecture and a TRL4 prototype showing that natural language can be used to configure simulation scenarios, execute a digital twin workflow, and deliver outputs through reports and dashboards.

The validation results indicate that the approach is feasible for interactive pre-event planning, particularly in contexts where users need to compare alternative assumptions before making operational decisions. At the same time, the current prototype remains an intermediate step toward deployment. Its predictive validity must be assessed with real operational data, and future versions should incorporate stronger data governance, richer economic indicators, and the use of more advanced conversational tools.

As large language models continue to evolve and digital twin adoption expands across sectors, conversational interfaces may become an important component of future decision-support systems. The framework presented in this paper contributes to this direction by identifying a practical architecture, demonstrating its feasibility in a controlled case study, and outlining the main challenges that must be addressed before conversational digital twins can be reliably deployed in operational environments. By lowering barriers to simulation access while maintaining analytical rigor, conversational digital twins have the potential to transform the way organizations plan, optimize, and adapt to operational uncertainty.

Author Contributions: Conceptualization, E.S.-O., M.S.-M. and E.W.-S.; methodology, A.C.-R., M.Á.G.-E. and P.V.-M.; software, A.C.-R.; validation, M.Á.G.-E., M.S.-M. and P.V.-M.; writing—original draft preparation, E.S.-O., P.V.-M. and A.C.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been carried out within the framework of the Spain Living Lab project (Grant Reference 1/1/2024-0412093852—SLLC16-01), funded by the Canarian Agency for Research, Innovation and the Information Society (ACIISI), Department of Universities, Science, Innovation and Culture of the Government of the Canary Islands, under the RETECH Programme, contributing to milestones 251, 252 and 253 of Component 16 of the Recovery, Transformation and Resilience Plan (PRTR), and co-funded by the European Union—Next Generation EU.

Data Availability Statement: The data and code supporting the findings of this study are available from the corresponding author (coordinacionit@canariaslivinglab.org) upon reasonable request.

Acknowledgments: The authors would like to thank Antonio Fernández Baldera for the valuable feedback and revisions provided during the preparation of this manuscript.

Conflicts of Interest: Authors Pablo Vicente-Martínez and Adrián Chust-Ros were employed by the company SPV Scala. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. System Prompt and Configuration

Appendix A.1. Agent System Prompt

The appendix details the system prompt designed to direct the behavior of the conversational agent. It specifies the agent's role, its permitted actions, and the constraints governing its interactions, thereby promoting consistent and well-structured functioning within the proposed architecture.

Listing A1: System Prompt of the Conversational Agent.

You are a specialist assistant in managing configuration files for simulation models.

You can help users:

- Check or display the initial or current configuration (with or without changes) (tool: view_current_config)
- Update a configuration parameter (tool: change_config_value)
- Update multiple configuration parameters (tool: change_config_value, useful for some validation rules)
- View the configuration file structure (below)
- Save changes (tool: save_config_changes)
- Create a simulation using the event simulation API (tool: simulate_event)
- Download the report for an event simulation (Excel) (tool: deliver_event_report)

Instructions:

- Use the tools when necessary.
- Display the current configuration settings to the user when requested.
- Notify the user of any changes made and ask if they want to save the changes before exiting.
- Save the changes only if the user requests it.
- Report any errors received by the tools.
- To create a simulation, use the 'simulate_event' tool and provide the user with the simulation ID and status. If the status is still in progress, you can tell the user that they can check back later to verify if the process has completed. Do not tell them that they will be notified when the simulation is complete.
- To retrieve the report for a completed simulation, use 'deliver_event_report' with the ID provided by the user. Inform the user of the simulation's status or whether the file has downloaded successfully. Do not disclose the file path to the user.

Important rules regarding tools:

- You can chain multiple tools together without asking the user for confirmation at each intermediate step. Execute all necessary steps at once and display the final result.

Configuration structure:

```
{config_structure}
```

Use the available tools as needed and always explain the changes you make.

Appendix A.2. Simulation Engine Configuration File

The following section details the file employed by the system to configure the simulation of attendee behavior and resource usage. It establishes the event details, variables to simulate, crowd size, and other settings. All null values are replaced during the execution with the default values bases on previous configurations used.

Listing A2: Structure of the Configuration File in YAML.

```
# USER INPUTS FOR ATTENDANCE CONFIGURATION
event_id: 3
is_derby: false
imp_high: null
imp_low: null
imp_normal: true
weather_sunny: true
probability: 0.6
season_ticket_attendance: null
dayofweek: 5
day: 26
month: 5
year: 2024

# USER INPUTS FOR DRINKS CONFIGURATION
available_hotel_beds: null
average_stay_nights: null
benefits: null
hotel_occupancy_rate: 85
international_tourists: null
local_attendance: null
product_category_cold_drink: null
product_type_drink: null
province_las_palmas: true
total_attendance: null
total_cost: null
total_tourists: 19000
tourist_attendance: null
average_income_per_person: null
year_of_rent: 2024

# USER INPUTS FOR FOOD CONFIGURATION
net_gross_net: true
product_category_light_snack: true
product_type_food: true

# USER INPUTS FOR MERCH CONFIGURATION
product_category_shirt: null
product_type_clothes: null

# USER INPUTS FOR QUEUES CONFIGURATION
hour: 15
minute: 15
second: 0
```

```

seconds_from_start:  -900
store_1:              true

# USER INPUTS FOR RESTROOM CONFIGURATION
drink_total_sells:    null
pcat_restroom:        true
ptype_supply:         true

# USER INPUTS FOR RISK CONFIGURATION
matchday:             null
away_position:        5
home_form_points:     9
away_form_points:     7
team_ud_las_palmas:   true
is_holiday:           false
is_weekend:           true
crime_threats:         null
crime_damages:         null
crime_thefts:          null
crime_injuries:        null
crime_burglary:        null
crime_robbery_with_violence: null
crime_thefts:          null
crime_total_crimes:    null
crime_degrading_treatment_violence: null
value:                null
start_hour:           null
start_minute:         null
start_second:         null

# USER INPUTS FOR TRAFFIC AND PARKING CONFIGURATION
timestamp:            null
latitude:             null
longitude:            null
capacity:             null
id:                   null
hits:                 null
avgsp:                null
time:                 "15:15:00"

```

References

1. Pott, C.; Spiekermann, C.; Breuer, C.; ten Hompel, M. Managing logistics in sport: A comprehensive systematic literature review. *Manag. Rev. Q.* **2024**, *74*, 2341–2400. [\[CrossRef\]](#)
2. Rabadi, G.; Khallouli, W.; Salem, M.; Ghoniem, A. Planning and management of major sporting events: A survey. *Int. J. Plan. Sched.* **2015**, *2*, 154–178. [\[CrossRef\]](#)
3. Fuller, A.; Fan, Z.; Day, C.; Barlow, C. Digital twin: Enabling technologies, challenges and open research. *IEEE Access* **2020**, *8*, 108952–108971. [\[CrossRef\]](#)
4. Singh, M.; Fuenmayor, E.; Hinchey, E.P.; Qiao, Y.; Murray, N.; Devine, D. Digital twin: Origin to future. *Appl. Syst. Innov.* **2021**, *4*, 36. [\[CrossRef\]](#)
5. Liu, M.; Fang, S.; Dong, H.; Xu, C. Review of digital twin about concepts, technologies, and industrial applications. *J. Manuf. Syst.* **2021**, *58*, 346–361. [\[CrossRef\]](#)
6. Madni, A.M.; Madni, C.C.; Lucero, S.D. Leveraging digital twin technology in model-based systems engineering. *Systems* **2019**, *7*, 7. [\[CrossRef\]](#)

7. Kunath, M.; Winkler, H. Integrating the digital twin of the manufacturing system into a decision support system for improving the order management process. *Procedia CIRP* **2018**, *72*, 225–231. [CrossRef]
8. Barricelli, B.R.; Casiraghi, E.; Fogli, D. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access* **2019**, *7*, 167653–167671. [CrossRef]
9. Tao, F.; Cheng, J.; Qi, Q.; Zhang, M.; Zhang, H.; Sui, F. Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3563–3576. [CrossRef]
10. Deng, T.; Zhang, K.; Shen, Z.-J. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. *J. Manag. Sci. Eng.* **2021**, *6*, 125–134. [CrossRef]
11. Croatti, A.; Gabellini, M.; Montagna, S.; Ricci, A. On the integration of agents and digital twins in healthcare. *J. Med. Syst.* **2020**, *44*, 161. [CrossRef] [PubMed]
12. Rasheed, A.; San, O.; Kvamsdal, T. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access* **2020**, *8*, 21980–22012. [CrossRef]
13. Jones, D.; Snider, C.; Nassehi, A.; Yon, J.; Hicks, B. Characterising the Digital Twin: A systematic literature review. *CIRP J. Manuf. Sci. Technol.* **2020**, *29*, 36–52. [CrossRef]
14. Kritzing, W.; Karner, M.; Traar, G.; Henjes, J.; Sihn, W. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* **2018**, *51*, 1016–1022. [CrossRef]
15. Lu, Y.; Liu, C.; Wang, K.-I.K.; Huang, H.; Xu, X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101837. [CrossRef]
16. Sharma, A.; Kosasih, E.; Zhang, J.; Brintrup, A.; Calinescu, A. Digital twins: State of the art theory and practice, challenges, and open research questions. *J. Ind. Inf. Integr.* **2022**, *30*, 100383. [CrossRef]
17. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901. Available online: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf (accessed on 4 January 2026).
18. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunschill, E.; et al. On the Opportunities and Risks of Foundation Models. *arXiv* **2021**. [CrossRef]
19. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models Are Unsupervised Multitask Learners. OpenAI Technical Report. 2019. Available online: <https://api.semanticscholar.org/CorpusID:160025533> (accessed on 29 December 2025).
20. Zhou, Z.; Lin, Y.; Jin, D.; Li, Y. Large Language Model for Participatory Urban Planning. *arXiv* **2024**. [CrossRef]
21. Elbasheer, M.; Laili, Y.; Longo, F.; Solina, V.; Tao, Y.; Veltri, P.; Zhang, Y.; Zhang, L. Natural language-driven production planning: Integrating large language models with automatic simulation model generation in manufacturing systems. *J. Intell. Manuf.* **2025**. [CrossRef]
22. Jauhiainen, J.S.; Hakanpää, S.; Honkasaari, H.-P.; Kivilompolo, N.; Kurri, M.; Lehtiranta, L.; Nurminen, M. Generative AI in participatory urban planning: Synthetic inhabitants and experts. *Land* **2026**, *15*, 407. [CrossRef]
23. Narechania, A.; Srinivasan, A.; Stasko, J.T. NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries. *arXiv* **2020**, arXiv:2008.10723. [CrossRef]
24. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.d.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating Large Language Models Trained on Code. *arXiv* **2021**, arXiv:2107.03374. [CrossRef]
25. Menzel, T.; Bagschik, G.; Isensee, L.; Schomburg, A.; Maurer, M. From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment. In *2019 IEEE Intelligent Vehicles Symposium (IV)*; IEEE: Piscataway, NJ, USA, 2019; pp. 2383–2390. [CrossRef]
26. Dengler, G.; Bazan, P.; German, R.; Lalbakhsh, P.; Liebmann, A. A conversational human-computer interface for smart energy system simulation environments. In *2023 Winter Simulation Conference (WSC)*; IEEE: Piscataway, NJ, USA, 2023; pp. 2978–2989. [CrossRef]
27. Tur, G.; De Mori, R. (Eds.) *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*; John Wiley & Sons: Hoboken, NJ, USA, 2011. [CrossRef]
28. Young, S.; Gašić, M.; Thomson, B.; Williams, J.D. POMDP-based statistical spoken dialog systems: A review. *Proc. IEEE* **2013**, *101*, 1160–1179. [CrossRef]
29. Liu, X.; David, I. AI simulation by digital twins: Systematic survey, reference framework, and mapping to a standardized architecture. *Softw. Syst. Model.* **2025**, *25*, 731–756. [CrossRef]
30. Grieves, M.; Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*; Kahlen, F.J., Flumerfelt, S., Alves, A., Eds.; Springer: Cham, Switzerland, 2017; pp. 85–113. [CrossRef]
31. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]

32. FastAPI. FastAPI Framework, High Performance, Easy to Learn, Fast to Code, Ready for Production. Available online: <https://fastapi.tiangolo.com/> (accessed on 20 December 2025).
33. Microsoft Power BI Documentation. Microsoft. Available online: <https://docs.microsoft.com/en-us/power-bi/> (accessed on 29 April 2026).
34. Pydantic: Data Validation Using Python Type Hints. Pydantic. Available online: <https://docs.pydantic.dev/> (accessed on 29 April 2026).
35. Li, J.; Chen, Z.; Huang, G.; Fu, G. A dynamic simulation framework for evaluating the impacts of urban flooding on transportation systems. *Int. J. Disaster Risk Sci.* **2026**, *17*, 182–196. [CrossRef]
36. Srinurak, N.; Sukwai, J.; Mishima, N. The simulation-based analysis focusing on street obstruction of evacuee mobility to mitigate disaster risk: Chiang Mai Historic City. *Heritage* **2025**, *8*, 546. [CrossRef]
37. Wikipedia Contributors. Polynomial-Time Reduction. Wikipedia, the Free Encyclopedia. 2026. Available online: https://en.wikipedia.org/wiki/Polynomial-time_reduction (accessed on 28 December 2025).
38. Lebioda, K.; Petrovic, N.; Pan, F.; Zolfaghari, V.; Schamschurko, A.; Knoll, A. Are requirements really all you need? Using LLMs to generate configuration code: A case study in automotive simulations. *IEEE Access* **2025**, *13*, 145115–145126. [CrossRef]
39. Noyman, A.; Hu, K.; Larson, K. TravelAgent: Generative agents in the built environment. *Environ. Plan. B Urban Anal. City Sci.* **2026**, *53*, 377–397. [CrossRef]
40. Faker. Faker: A Python Package That Generates Fake Data. Available online: <https://faker.readthedocs.io/> (accessed on 14 July 2026).
41. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.F.; Dennison, D. Hidden Technical Debt in Machine Learning Systems. In *Advances in Neural Information Processing Systems*; Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2015; Volume 28, pp. 2503–2511.
42. McMillan-Major, A.; Aschenbrenner, S.; Bhattacharjee, A.; Dash, S.; Ovalle, A.; Judd, N.; Mitchell, M. Whose Opinions Do Language Models Reflect? In *Proceedings of the 40th International Conference on Machine Learning*; PMLR: Honolulu, HI, USA, 2023; Volume 202, pp. 24094–24111.
43. Wang, X.; Li, J.; Liu, Y.; Chen, H. Conversational AI for Industrial Automation: A Survey. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 1821–1837.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.