electronics

MDPI

Article

Integrating Conversational AI Agents with Digital Twins: A Systems Engineering Approach to Complex Infrastructure Management and Predictive Decision-Making

Pablo Vicente-Martínez [1,*], Emilio Soria-Olivas [2], Sergio Sebastiá-García [3], Claudia Vizcaíno-Ramírez [3], Adrián Chust-Ros [1], María Ángeles García-Escrivà [3,*] and Edu William-Secin [4]

1 SPV Scala, Gran Canaria, 35100 San Bartolomé de Tirajana, Spain; c.datos34@salascala.com
2 Inteligent 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; data1@canariaslivinglab.org (S.S.-G.); marketing@canariaslivinglab.org (C.V.-R.)
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
* Correspondence: c.datos12@salascala.com (P.V.-M.); coordinacionit@canariaslivinglab.org (M.Á.G.-E.)

Abstract

Background: Managing complex infrastructure increasingly requires predictive, adaptive, and human-centered systems. Traditional approaches often struggle with operational complexity, fragmented data, and high technical barriers. **Methods:** This study presents a TRL4 proof of concept integrating a conversational AI agent with a user-adaptive digital twin for occupancy forecasting. Users can upload their own datasets, and dynamically configure prediction models (ARIMA, SARIMA, Random Forest, XGBoost) based on input variables such as occupancy or demand drivers. The AI agent, powered by Gemini 2.5 Flash Lite, functions as an orchestration layer, translating natural language instructions into data ingestion, model execution, and query actions. While the digital twin supports additional variables (energy, water, waste), these are envisioned for future work and were not part of the current validation. **Results:** Functional validation confirmed the system's capability to interpret user intentions accurately, adapt model training to the characteristics of user-provided data, and present results through convenient and comprehensible visualization methods. The integrated architecture demonstrated stable performance across multiple validation scenarios, achieving satisfactory prediction accuracy (within expected ranges for TRL 4). **Conclusions:** This work validates the technical and functional viability of integrating conversational AI agents with digital twins as an emergent system of systems, extending beyond conventional predictive pipelines by enabling context-specific modeling. The systems engineering approach reveals how such integration transforms reactive infrastructure management into proactive, data-driven, and human-centered decision-making processes, establishing a foundation for future developments toward higher technology readiness levels.

Keywords: digital twin; conversational artificial intelligence; systems engineering; infrastructure management; predictive maintenance; natural language processing; machine learning

Academic Editor: Pietro Manzoni

Received: 10 March 2026
Revised: 16 April 2026
Accepted: 21 April 2026
Published: 28 April 2026

1. Introduction

1.1. Context and Motivation

Modern infrastructure management increasingly faces challenges associated with growing operational complexity, resource optimization under uncertainty, and the need for predictive decision-making in dynamic environments. Large-scale facilities such as hotels, airports, hospitals, commercial centers, and industrial complexes operate as intricate socio-technical systems in which human activities, physical assets, resource flows, and environmental conditions interact in nonlinear and often unpredictable ways [1,2]. Traditional management approaches, typically reactive and based on historical averages or rule-based heuristics, struggle to capture the emergent behaviors and interdependencies characteristic of such complex systems [3,4].

The convergence of digital transformation, Internet of Things (IoT) sensing capabilities, and artificial intelligence (AI) has created unprecedented opportunities to reimagine infrastructure management through a systems engineering lens [5,6]. Two technologies have emerged as particularly promising: Digital Twins (DTs), defined as virtual representation of physical assets or processes that enable real-time monitoring, simulation, and optimization [7,8], and Conversational AI Agents, which are intelligent systems capable of natural language interaction that bridge the gap between human operators and complex computational models [9,10].

However, despite significant advances in both domains independently, the systematic integration of conversational AI agents with digital twins has received limited attention from a holistic systems engineering perspective. Most existing implementations treat these technologies as separate modules rather than as synergistic components of an integrated system of systems (SoS) [11,12]. This fragmentation may limit the potential for emergent capabilities—properties that arise from the interaction of subsystems but are not present in individual components [4].

1.2. The Challenge of Complex Infrastructure Management

Infrastructure management can be understood, from a systems engineering perspective, as a complex socio-technical system in which technological components, organizational processes, human decision-makers, and environmental factors coevolve and interact across multiple scales [2,13]. In large facilities such as hotels, which constitute the focus of this case study, operators must continuously balance multiple and often competing objectives, including occupancy and revenue maximization, operational cost reduction (e.g., energy, water, and waste management), compliance with sustainability targets, and the preservation of service quality [14,15].

Traditional approaches to this multi-objective optimization problem exhibit several fundamental limitations:

- **Reactive decision-making:** Facility managers typically respond to problems after they manifest, rather than anticipating issues through predictive analytics [16,17].
- **Data fragmentation:** In many infrastructure contexts, operational data frequently reside in heterogeneous and weakly integrated systems, creating interoperability challenges and constraining holistic analysis from a systems engineering perspective [1].
- **Technical barriers:** Leveraging advanced analytics and simulation tools requires specialized expertise, creating a disconnect between domain experts (facility managers) and technical capabilities [4].
- **Static configuration:** Conventional simulation and forecasting approaches are often characterized by limited adaptability and lack of real-time responsiveness to evolving operational contexts [5].

These limitations underscore the need for a paradigm shift: from isolated tools toward integrated intelligent systems that combine predictive modeling, real-time adaptation, and intuitive human-machine interaction within a coherent systems architecture.

The proposed approach is illustrated through a case study in hotel infrastructure management, a representative complex system characterized by occupancy dynamics; at this stage, the validation focuses exclusively on occupancy estimation, while multi-resource consumption and waste generation are considered future extensions.

1.3. Theoretical Background

1.3.1. Digital Twins in Infrastructure: State of the Art

Digital twins enable infrastructure operators to create virtual representations of physical assets that synchronize with real-world states through continuous data streams from IoT sensors [18,19]. Key applications include:

- **Predictive maintenance:** AI-driven DTs analyze sensor data (temperature, vibration, operational hours) to forecast equipment failures before they occur, reducing downtime by up to 25% [16,20].
- **Urban planning:** City-scale DTs, such as Singapore's Virtual Singapore and Australia's Digital Twin Victoria, integrate multi-source data for disaster management, infrastructure planning, and policy simulation [21].
- **Construction lifecycle management:** DTs facilitate design optimization, construction monitoring, and facility operations across the entire building lifecycle [5,6].

Machine learning (ML) models constitute the predictive core of contemporary DTs. Time-series forecasting algorithms (ARIMA, SARIMA) capture temporal patterns and seasonality, while ensemble methods (Random Forest, XGBoost) model complex multivariate relationships [22,23].

However, a critical gap persists: most DT implementations assume technically proficient users capable of configuring simulation parameters, interpreting model outputs, and translating insights into operational decisions. This assumption creates a usability barrier that limits DT adoption, particularly in domains where facility managers lack data science expertise [2].

1.3.2. Evolution from Static Blueprints to Agent-Managed Digital Twins

The adoption of Digital Twins (DT) in the hospitality and tourism sectors has transitioned from conceptual visualization toward integrated decision-making systems. A foundational benchmark in this domain is the five-layer architectural framework proposed by Keertana and Kumar (2025) [24], which defines the structural requirements for a DT Predictive Analytics Engine (Layer 3) and its role in Resource Optimization (Layer 5). While their work provides a peer-reviewed "blueprint" for destination and hotel management, it remains primarily conceptual, relying on secondary data and Structural Equation Modeling (SEM) to theorize potential gains.

Our work advances the state of the art by providing empirical validation and agentic intelligence absent in existing frameworks. We propose a digital twin that allows users to upload datasets, configure models dynamically, and interact via natural language, enabling adaptive modeling. This transforms the system from a passive monitoring tool into an interactive operational asset.

1.3.3. Conversational AI: Democratizing Human-AI Interaction

Conversational AI agents, defined as systems that interpret natural language inputs, maintain dialogue context, and generate human-like responses, have evolved rapidly with the advent of large language models (LLMs). These agents employ natural language

processing (NLP) techniques including intent recognition, named entity recognition (NER), and semantic understanding to facilitate intuitive human-computer interaction [9,10].

Recent trends in conversational AI include:

- **Autonomous agency:** Modern conversational agents exhibit goal-directed behavior, executing multi-step tasks with minimal human intervention [25,26].
- **Multimodal interfaces:** Integration of text, voice, and visual inputs enhances accessibility and user engagement [27].
- **Tool integration:** Conversational agents increasingly orchestrate external services, APIs, and computational tools, functioning as cognitive interfaces to complex systems [28].

Despite recent advances in conversational AI, its application in technical domains such as infrastructure management remains limited compared with its widespread use in areas focused on customer service, information retrieval, and task automation. A contemporary review of AI-powered virtual conversational agents highlights that most research to date has concentrated on general-purpose implementations and user interaction design, and identifies gaps and open challenges that motivate future work in more complex application contexts [29]. Accordingly, the potential for conversational agents to serve as configuration interfaces for complex simulation systems—enabling non-technical users to adjust parameters and interpret results through natural dialogue—has received relatively little attention in the literature.

1.3.4. The Integration Gap: A Systems Engineering Perspective

Integrating AI with a digital twin creates a unique system-of-systems challenge. While the digital twin delivers evolving predictive modeling, the AI agent must orchestrate multiple models, data pipelines, and user intents—a complex coordination task. Together, they produce new capabilities—such as parameter configuration exploration and adaptive preprocessing and forecasting—not achievable by either alone, combining simulations, human-computer interaction, and dynamic orchestration. From a systems perspective, this integration represents a system-of-systems (SoS) problem characterized by [3,12]:

- **Operational independence:** The conversational agent and the digital twin function as autonomous subsystems with distinct purposes.
- **Managerial independence:** Each subsystem may employ different technologies, development frameworks, and operational paradigms.
- **Emergent behavior:** The integrated system exhibits capabilities such as adaptive, dialogue-driven predictive simulation that neither component possesses individually.
- **Evolutionary development:** Both subsystems continue to evolve, requiring architectural flexibility to accommodate technological advances.

The current literature lacks comprehensive frameworks for designing, implementing, and validating such integrated systems. Existing work tends to address either DT development [6] or conversational AI design [9], but rarely examines their synergistic integration through a systems engineering methodology.

1.3.5. Research Contribution and Objectives

This paper addresses the identified gap by presenting a systems engineering approach to integrating conversational AI agents with a user-adaptive, data-driven digital twin for complex infrastructure management. Specifically, we contribute:

1. **A systems architecture** that positions a conversational AI agent as an orchestration layer and human-facing configuration interface, and a digital twin as a persistent, context-specific simulation and prediction engine, with well-defined interfaces and data flows.

2. **A functional validation** of this integrated system at Technology Readiness Level (TRL) 4, demonstrating technical feasibility in a controlled laboratory environment using a case study of hotel management, including user-driven data ingestion and model configuration workflows.
3. **Empirical evidence** that natural language-based configuration substantially lowers barriers to using predictive tools, enabling non-technical stakeholders to leverage advanced analytics and explore different data processing and predictive models.
4. **Methodological insights** on designing socio-technical systems where AI mediates the interaction between human decision-makers and computational models, acting not merely as a conversational interface but as an intelligent coordinator of data integration, model training, and query execution, advancing the theory and practice of systems engineering.

The case study focuses on hotel infrastructure management, a representative complex system involving occupancy forecasting, multi-resource consumption (electricity, gas, water), and waste generation prediction. The conversational agent, powered by Google's Gemini 2.5 Flash Lite LLM [30], interprets user requests in natural language, modifies configuration parameters, and orchestrates the execution of ML models, including Linear Regression, Decision Tree, Random Forest, XGBoost, ARIMA, SARIMA, and Moving Average, within the digital twin. The results are delivered through inline graphs (using the matplotlib 3.10.3. library), allowing users to review predictions and make informed decisions based on the model outputs.

The remainder of this paper is organized as follows: Section 2 describes the systems engineering methodology, detailing the architecture, subsystem designs, and integration strategy. Section 3 presents validation results from controlled experiments, including user interaction examples, model performance metrics, and visualization outputs. Section 4 discusses the implications of this integration for systems engineering theory and infrastructure management practice, analyzes limitations, and compares our approach with existing work. Section 5 concludes by summarizing the main contributions and outlining future research directions toward higher TRL deployment and expanded real-world validation with operational datasets.

2. Materials and Methods

This section presents the systems engineering methodology used to design, implement and validate the integrated conversational AI agent and digital twin system. We adopt a modular architecture perspective, describing first the overall system design, then detailing each major subsystem, their integration mechanisms, and finally the validation approach consistent with the Technology Readiness Level (TRL) 4 requirements.

2.1. Systems Architecture and Design Philosophy

The integrated system is architected as a system of systems (SoS) comprising two primary subsystems: (1) a **Conversational AI Agent** that serves as the human-machine interface, enabling natural language configuration and query capabilities; and (2) a **Digital Twin** that encapsulates the computational intelligence for predictive modeling. Both were programmed using Python 3.12.7. Figure 1 illustrates the high-level architecture and information flows.

Architectural design adheres to several systems engineering principles:

- **Separation of concerns:** Each subsystem maintains distinct responsibilities, with the agent handling user interaction and configuration management, while the digital twin executes the computational simulations.

- **Modularity:** Components are designed as loosely coupled modules communicating through well-defined interfaces (configuration files, API calls), facilitating independent development and testing.
- **Scalability:** The use of containerization (Docker) and cloud services (AWS Fargate, AWS RDS) positions the system for future expansion to production environments.
- **Human-centeredness:** Natural language interaction eliminates technical barriers, enabling facility managers without data science expertise to leverage advanced predictive analytics.

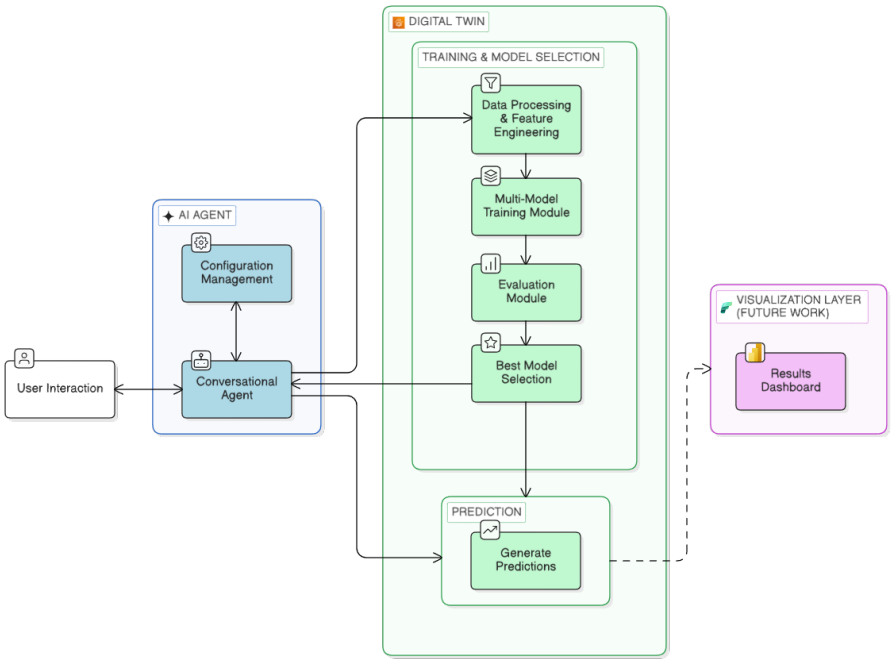

Figure 1. High-level architecture of the system, showing the integration of the Conversational AI Agent (dialogue and configuration modules) with the Digital Twin (training, evaluation, and prediction) and the proposed visualization layer. Arrows indicate the main data and control flows.

The system workflow proceeds as follows: (1) A user interacts with the conversational agent using natural language, either to specify configuration changes or to request predictions based on provided data. (2) The agent interprets the intent, updates the configuration file (JSON/YAML), and triggers the digital twin. (3) The digital twin receives data, performs preprocessing and feature engineering based on the specified configuration, trains and validates multiple ML models, selects the optimal model, and generates predictions. (4) The results are returned to the agent and presented through interactive visualizations (matplotlib charts in this case, but other technologies such as Power BI dashboards [31] could be added in the future).

A closed loop is achieved by incorporating the user's decision-making process into the system lifecycle. Specifically, the user interprets the model outputs to inform real-world actions or operational adjustments. These actions generate new data, which are subsequently captured and will be reintroduced into the digital twin by the user as updated inputs. This continuous feedback cycle enables iterative refinement of both the model and its predictions, thereby ensuring that the system evolves over time and remains aligned with the underlying dynamics of the real-world process. As future work, this loop could be further streamlined by reducing reliance on manual user input, for instance by integrating automated data sources directly into the system rather than depending exclusively on data provided through the conversational agent.

2.2. Subsystem 1: The Digital Twin

The digital twin constitutes the predictive intelligence core of the integrated system. It is implemented as an independent service that manages the full lifecycle of machine learning models: data preprocessing, feature engineering, model training, validation, and prediction generation.

2.2.1. Digital Twin Architecture

The digital twin is deployed as a cloud-based service designed to support scalable execution and flexible integration. It provides two main functional capabilities. First, it enables data preprocessing and model training based on historical datasets and configurable parameters, returning the best-performing model together with validation metrics such as MAE, MAPE, and RMSE. Second, it enables prediction by generating future estimates either over a specified time horizon for time-series models or based on user-provided input data for regression models.

This service-oriented architecture decouples the digital twin from the conversational agent, allowing independent scaling, version control, and potential reuse across different application contexts.

2.2.2. Data Model and Feature Engineering

The digital twin operates on structured time-series data representing hotel operational variables. The dataset is flexible and configurable, allowing adaptation to different sources and formats. Key aspects of the data configuration include:

- **Target variable:** Occupancy rate, representing the proportion of occupied rooms, which serves as the prediction target.
- **Numerical features:** Configurable numeric columns capturing operational metrics; these may vary depending on the dataset and use case.
- **Temporal information:** Date and/or time columns, with configurable formats, granularity, and indexing to support time-series analysis.
- **Grouping or context columns:** Optional columns to distinguish multiple entities or subseries within the data.

Preprocessing and feature engineering are controlled by configurable parameters to ensure adaptability:

- **Data aggregation:** Configurable methods (e.g., last, first, mean, sum) to summarize raw observations.
- **Missing value handling:** Flexible strategies such as linear interpolation, forward fill, seasonal average, or dropping missing values.
- **Outlier detection:** Optional identification and replacement of anomalous observations.
- **Lagged features:** Inclusion of previous time steps as features, configurable in the number of lags.
- **Correlation and multicollinearity control:** Optional removal of highly correlated features or analysis via Variance Inflation Factor (VIF).
- **Feature scaling:** Configurable scaling of input features and target variable (none, standard, min-max).
- **Temporal resampling:** Configurable frequency for aggregation or indexing (e.g., “30 min”, “H”, “D”), with start and end times for the time window.

These configurable preprocessing and feature engineering steps allow the digital twin to flexibly handle diverse datasets while capturing essential temporal patterns and operational relationships for accurate occupancy prediction.

2.2.3. Machine Learning Models

The digital twin implements a diverse ensemble of supervised learning algorithms, categorized into time-series and tabular regression models:

Time-Series Models:

- **Moving Average (MA):** A baseline model computing predictions as the mean of the last n observations (configurable window).
- **ARIMA (AutoRegressive Integrated Moving Average):** A classical statistical model capturing linear dependencies, trends, and seasonality. Implemented using the statsmodels library (version 0.14.4) with automatic parameter selection via grid search over (p, d, q) orders.
- **SARIMA (Seasonal ARIMA):** An extension of ARIMA incorporating explicit seasonal components (P, D, Q, s) to model periodic patterns (e.g., weekly, monthly).

Tabular Regression Models:

- **Linear Regression:** A simple interpretable baseline using ordinary least squares (OLS). Implemented via scikit-learn 1.7.0.
- **Decision Tree Regressor:** A non-parametric model partitioning the feature space into decision rules [32]. This model was included to the variety of the tested models, but it was not expected to perform better than Random Forest.
- **Random Forest Regressor:** An ensemble of decision trees trained on bootstrap samples, reducing variance and improving generalization [33].
- **XGBoost (Extreme Gradient Boosting):** A gradient boosting framework employing regularized objective functions and advanced tree construction algorithms [34].

Each model is trained independently on preprocessed data using time-based train-validation splits (e.g., 80%/20%) to preserve temporal ordering and prevent data leakage. Hyperparameters are tuned via cross-validation where applicable.

2.2.4. Model Selection and Validation

The digital twin automatically evaluates all trained models using three standard regression metrics:

- **Mean Absolute Error (MAE):** $\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- **Mean Absolute Percentage Error (MAPE):** $\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$
- **Root Mean Squared Error (RMSE):** $\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$

Here, y_i denotes actual values, $\hat{y}_i$ denotes predictions, and n is the number of validation samples. The model with the lowest MAPE is designated as the "best model" and persisted for subsequent prediction requests. This selection criterion prioritizes generalization performance on unseen data, aligning with predictive maintenance objectives [16].

For ARIMA/SARIMA models, a lightweight model container stores only the optimal (p, d, q) or (p, d, q, P, D, Q, s) structure. During prediction, a fresh model instance is instantiated with these parameters and retrained on 100% of historical data to maximize predictive accuracy.

2.3. Subsystem 2: The Conversational AI Agent

The conversational AI agent constitutes the interaction layer of the system, mediating all exchanges between users and the digital twin. Its architecture can be described in

terms of three functional layers: user interface, conversational coordination, and execution support services.

2.3.1. User Interface

The user interface provides a chat-based environment through which users express queries and commands in natural language. It presents the ongoing dialogue, the current system configuration, and the outputs of predictive analyses. Results may include graphical visualizations, thereby supporting informed decision-making within an intuitive interaction paradigm.

2.3.2. Conversational Coordination Layer

Conversational logic is managed through Chainlit 2.6.2, a Python framework specialized for building conversational AI applications [35]. In this system, Chainlit is responsible for maintaining contextual continuity across multi-step dialogues and coordinating the interaction between the user interface, the language model, and backend services.

This layer enables users to iteratively refine configurations, upload data when necessary, and initiate analytical processes directly through natural language. By preserving conversational state, it allows progressive specification of analytical tasks without requiring users to re-enter previously defined parameters.

2.3.3. Language Model Integration

The cognitive core of the agent is Gemini 2.5 Flash Lite, chosen for its lightweight deployment, low-latency inference, and compatibility with on-premise or laboratory environments typical of TRL4 validation. Compared to newer models such as GPT-5.4 mini or Gemini 3.1 Flash Lite, Gemini 2.5 Flash Lite provides sufficient natural language understanding and contextual reasoning for configuration orchestration while requiring significantly fewer computational resources, making it practical for integration in proof-of-concept systems. Its advantages include rapid inference, reduced memory footprint, and stable performance on structured tasks, which are critical for reliably translating user instructions into system actions within the digital twin. Future work may explore higher-capacity models for expanded reasoning and multi-turn dialogue capabilities in production-level deployments.

2.4. Integration Mechanisms and Data Flows

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

2.4.1. Configuration File as Interface Contract

A JSON/YAML configuration file serves as the "interface contract" between subsystems (see an example in Appendix A.2). The agent modifies this file based on user inputs, and the digital twin reads it to determine training parameters. This file-based coupling ensures loose coupling, allowing each subsystem to evolve independently provided the configuration schema remains stable.

2.4.2. RESTful API Communication

For real-time operations (training, prediction), the agent invokes the digital twin's API endpoints. Responses are structured JSON objects containing model performance metrics, prediction results, and metadata. This synchronous communication pattern enables immediate feedback to users while maintaining subsystem autonomy.

2.5. Visualization Layer

The proposed design aims to present the prediction results using two complementary visualisation methods:

- **Embedded charts (implemented):** The agent generates matplotlib-based line plots showing historical data and predicted values. These images are displayed directly within the conversational interface, providing immediate visual feedback.
- **Interactive dashboards (future work):** Provide the user with Microsoft Power BI dashboards for deeper exploration. These dashboards would integrate data from a database and offer interactive filtering, drill-down capabilities, and comparative analysis across multiple time periods or scenarios [31].

This dual-visualization approach would balance conversational flow (quick inline charts) with exploratory analysis (rich dashboards).

2.6. Validation Methodology: TRL 4 Experimental Design

Technology Readiness Level (TRL) 4 corresponds to "component and/or breadboard validation in laboratory environment". At this stage, the objective is to demonstrate that the integrated system's basic components function together in a controlled setting, without requirements for production-scale performance, security hardening, or real-world deployment.

2.6.1. Test Scenarios and Acceptance Criteria

Three sets of functional tests were designed to evaluate the main capabilities of the system:

Test Set 1: Configuration Parameter Modification

Objective: Ensure that the conversational agent can accurately interpret user instructions and update configuration parameters accordingly.

Details: The test will involve a series of queries intended to modify specific configuration settings. Each query will verify that the requested changes are correctly applied in the system configuration.

Acceptance criterion: At least 90% of parameter modifications are correctly executed, as verified by comparing the resulting configuration against the requested changes.

Test Set 2: Model Training and Selection

Objective: Evaluate the digital twin's ability to train predictive models, assess their performance, and select the optimal model.

Details: The test will cover multiple model combinations. Each pair of models will be trained on the provided dataset, evaluated using standard performance metrics, and the best-performing model will be identified for further use.

Acceptance criterion: The selected model achieves a MAPE below 15% on validation data.

Test Set 3: Prediction Generation

Objective: Verify that the system can generate accurate predictions and present them through accessible visualizations.

Details: The tests will explore different prediction scenarios and assess the system's ability to display results through time-series plots and interactive dashboards for user inspection.

Acceptance criterion: All predictions are numerically consistent with the underlying models, and all visualizations render correctly without missing data.

2.6.2. Execution and Metrics Collection

Each test case was executed manually by a researcher simulating the role of a hotel facility manager. Interactions were logged, and the following metrics were recorded:

- **Intent recognition accuracy:** Percentage of user requests correctly interpreted by the agent.
- **Configuration update correctness:** Whether the configuration file reflected intended changes.
- **Model training success rate:** Percentage of training runs completing without errors.
- **Prediction accuracy:** MAE, MAPE, RMSE for each model on validation data.
- **Visualization quality:** Subjective assessment (correct/incorrect) of chart rendering and data accuracy.
- **System response time:** Time from user input to agent response.

3. Results

This section presents the TRL 4 validation of the conversational AI agent and digital twin system under controlled laboratory conditions. Tests covered intent recognition and configuration, predictive model training, and prediction visualization, using a 12-month synthetic hotel dataset with Occupancy as the target. Synthetic data enabled controlled validation of system functionality and data flow while avoiding real-data constraints. Although hotel data was used for realism, the system supports arbitrary datasets, variables, and configurations.

3.1. Intent Recognition and Configuration Management

The conversational agent's ability to interpret user intentions and manage the digital twin's configuration was evaluated through a test corpus of 100 utterances. This dataset was designed to span four intent categories while testing different levels of linguistic complexity: technical queries, informal natural language, and out-of-scope interactions.

Table 1 summarizes the results, providing a breakdown of the corpus and a brief analysis of the four non-compliant cases identified.

Table 1. Test Corpus Breakdown and Intent Recognition Performance ($N = 100$).

Intent Category	Utterance Type	Test Cases	Non-Compliance	Accuracy (%)
Parameter Modification	Technical & Informal	38	2	94.7
Model Training	Technical/Structured	21	0	100.0
Prediction Generation	Technical/Structured	6	0	100.0
System Robustness & Usability	Informal & Out of scope	35	2	94.3
Overall		**100**	**4**	**96.0**

The system achieved an overall intent recognition accuracy of 96.0% across 100 diverse functional test cases, exceeding the predefined acceptance criterion of 90%. A detailed analysis of the four non-compliant cases identified the following root causes:

- **Linguistic Ambiguity & Performance:** Two failures occurred within the Parameter Modification category due to the system's inability to parse short technical abbreviations (e.g., "rf", "arimas") and an isolated instance of system timeout during rapid bursts of status queries.
- **Functional Constraints:** Two failures within the Robustness & Usability category stemmed from current architectural limitations. These included the inability to generate real-time graphical visualizations and constraints regarding direct Excel file uploads, as the system is currently optimized for CSV formats.

Results demonstrate that the agent correctly identified and processed the vast majority of evaluated intents, while non-compliant cases have been identified as specific areas for future iterative enhancement.

3.1.1. Configuration Modification Examples

To illustrate the agent's operational capabilities, we present three representative examples of successful configuration modifications.

In the example in Figure 2, the agent correctly identified a single-parameter modification intent and updated the corresponding configuration flag by enabling variance inflation factor (VIF) analysis. The request was processed successfully, and the configuration change was applied without errors, with a recorded response latency of 1.83 s.

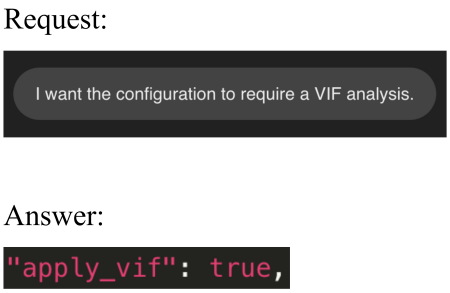

Figure 2. Example of a single-parameter modification.

The agent successfully interprets multi-parameter modification requests as the one shown in Figure 3, simultaneously disabling the variance inflation factor (VIF) analysis and updating the forecasting horizon to 21 periods. Both configuration changes were applied correctly within a single interaction, with a recorded response latency of 3.76 s.

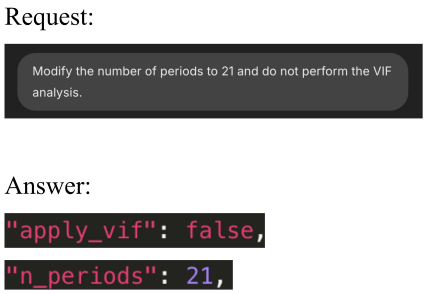

Figure 3. Example of a multi-parameter modification.

In Figure 4, the agent handled a contextual configuration update by first validating the user-provided aggregation method and identifying an invalid parameter value. Upon detecting the error, the agent returned an informative message listing the supported aggregation options. After the user provided a valid value, the agent successfully applied the configuration update, demonstrating the system's ability to maintain conversational context across multiple turns. The interaction was completed with response latencies of 1.47 s for the validation step and 3.74 s for the final configuration update.

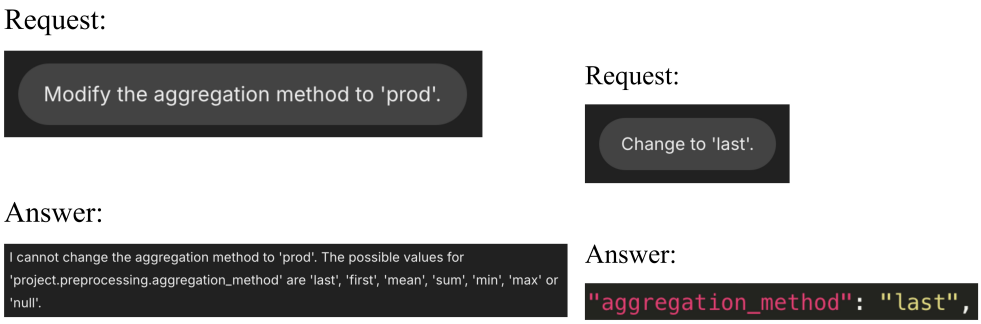

Figure 4. Example of a contextual dialogue.

3.1.2. Configuration Correctness Validation

Beyond intent recognition, we validated the correctness of configuration modifications. Across all test cases involving diverse modification patterns (single-parameter, multi-parameter, and conditional logic), the agent achieved 100% configuration correctness, with zero instances of:

- Type mismatches (e.g., string values assigned to numeric fields);
- Referential integrity violations (e.g., specifying non-existent variable values).

This result underscores the effectiveness of Pydantic-based schema validation (version 2.12.3) and the structured prompt engineering employed to guide Gemini 2.5 Flash Lite's configuration generation logic.

3.2. Predictive Model Training and Performance

The digital twin's model training subsystem was evaluated by 21 experiments, corresponding to all pairwise combinations of the available models. As the digital twin requires at least two models to be trained simultaneously, each experiment involved selecting the best-performing model based on validation metrics.

The evaluated models include:

- Statsmodels ARIMA;
- Statsmodels SARIMA;
- Moving Average;
- XGBRegressor;
- RandomForestRegressor;
- DecisionTreeRegressor;
- LinearRegression.

3.2.1. Model Performance Comparison

Model performance was evaluated using standard regression metrics, including MAE, RMSE, and MAPE, calculated on validation data. In addition, the computational cost of each experiment was assessed by recording the training time required to fit each model under the evaluated configuration.

All candidate models were trained within a single pipeline and evaluated simultaneously on the same dataset, ensuring consistent comparison conditions. The results indicate that model effectiveness varies according to the training scenario, with no single model family consistently outperforming the others. Consequently, different techniques emerged as optimal across the various experiments conducted. These findings highlight the importance of automated model comparison rather than relying on a single modeling approach [36].

Key Findings:

The evaluation results indicate that no single model consistently outperformed all others across the evaluated cases, and model selection was therefore determined based on validation metrics. Linear Regression was the most frequently selected model, followed by Random Forest and XGBoost, demonstrating that both linear and non-linear approaches can effectively support occupancy forecasting under the evaluated conditions.

Across the 21 comparative tests, the system achieved a mean MAPE of 9.98%, with the lowest observed error being 8.77%, obtained using Linear Regression. The majority of the evaluated configurations satisfied the predefined acceptance threshold of 15%, with 95% of experiments (20 out of 21) yielding MAPE values below this limit. These results confirm that the predictive model training and selection criteria were successfully met, validating the adequacy of the training data and the implemented feature engineering and automated model comparison strategy under controlled TRL 4 validation conditions.

Comparatively poorer performance was observed for the Statsmodels ARIMA model, whose higher prediction errors reflect the multifactorial nature of occupancy dynamics, which are not fully captured by classical univariate time-series approaches.

3.2.2. Training Success Rate and Computational Performance

The training success rate was evaluated across the 21 model training experiments conducted during the PoC. In all cases, the training processes completed successfully, resulting in a 100% success rate, with no observed execution failures, crashes, or data integrity issues.

Computational performance was assessed by observing the relative time required to complete each training request. Training durations varied depending on the selected model and configuration, reflecting the different computational complexities of the evaluated approaches when including only some of the model families, ranging between 2.89 and 34.68 s in total. Despite this variability, all training processes were completed within timeframes compatible with interactive analytical workflows, supporting the feasibility of the proposed approach at TRL 4.

More computationally demanding models, such as ARIMA-based approaches, exhibited longer execution times, while simpler or feature-based models completed training more quickly, as expected.

3.3. Prediction Generation and Visualization

To validate the prediction generation and visualization capabilities of the system, five representative prediction tests were conducted using different modeling approaches under the evaluated configuration. In all cases, prediction requests were executed successfully and the generated outputs were correctly processed and visualized, confirming the robustness of the prediction pipeline under TRL 4 laboratory conditions.

For clarity and conciseness, only representative examples are presented in the following subsections, illustrating both time-series-based and machine learning-based prediction approaches.

3.3.1. Time-Series Predictions with Confidence Intervals

Figure 5 illustrates a short-term time-series forecast generated using the ARIMA model. The prediction extends the historical pattern over a limited forecast horizon, providing a conservative assessment of near-term behavior under controlled validation conditions.

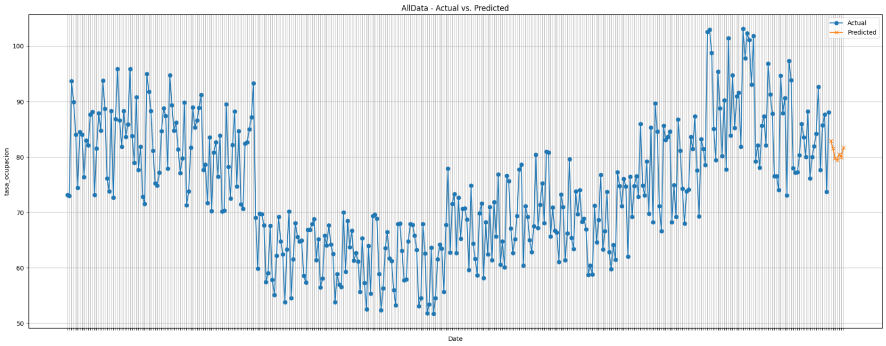

Figure 5. Seven-day occupancy time-series forecast with ARIMA model. The blue line represents daily historical data from 1 January to 31 December 2023, while the orange line shows predicted values for the first week of January 2024 (1–7 January).

3.3.2. Scenario-Based Predictions with Exogenous Variables

For non-time-series models, such as Random Forest, the system supports scenario-based prediction by accepting user-provided CSV files containing future values of exogenous variables. Figure 6 illustrates a representative occupancy prediction generated using the Random Forest model under this approach.

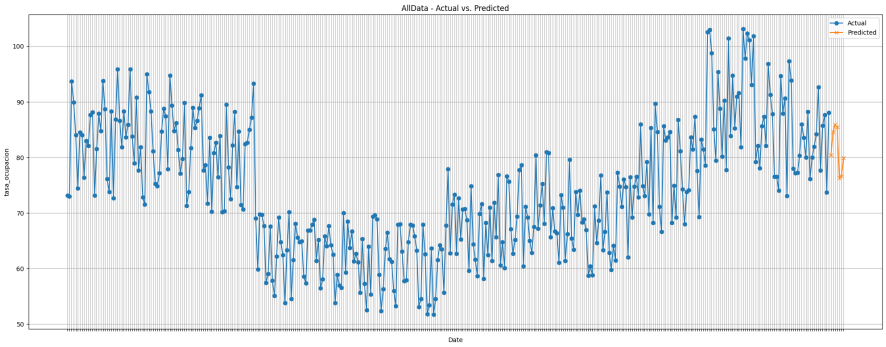

Figure 6. Occupancy predictions using the Random Forest model. The model was trained on daily historical data from 1 January to 31 December 2023, with predicted values shown for 1–7 January 2024.

The ability to generate what-if scenarios enables facility managers to evaluate the impact of operational decisions (e.g., promotional campaigns affecting occupancy) on resource requirements, supporting proactive capacity planning and cost optimization.

3.4. System Performance and Latency

System performance was evaluated by analyzing end-to-end response times of the conversational agent across the different types of interactions conducted during the PoC, including parameter modification, model training, and prediction generation requests. Table 2 summarizes the average end-to-end response times observed for the different interaction types evaluated during the PoC.

Table 2. Average End-to-End Latency by Interaction Type.

Interaction Type	Average Response Time (s)
Parameter modification	2.39
Model training	18.44
Prediction generation	15.44
System Robustness & Usability	<3

Parameter modification requests handled by the conversational agent exhibited an average response time of 2.39 s, supporting real-time interactive configuration. The majority of these interactions were completed in under 3 s, with only a small number of responses exhibiting slightly higher latency.

Furthermore, interactions focused on system robustness and usability, such as general inquiries, status checks, and interface navigation, demonstrated the highest efficiency based on the average response time of 2.39 s. This immediate feedback loop is critical for maintaining user engagement and ensures that the conversational flow remains fluid even during complex session management.

Model training requests required longer execution times due to the computational cost of fitting and validating predictive models. The average latency for training operations was 18.44 s. As expected, more computationally demanding models, such as ARIMA-based approaches, exhibited longer training times compared to simpler or feature-based models. Prediction generation requests showed a mean response time of 15.44 s, remaining within acceptable limits for interactive analytical workflows.

Overall, the observed response times across all interaction types fall within acceptable limits for a TRL 4 validation prototype, confirming the suitability of the system for conversational interaction, model training, and prediction generation under controlled laboratory conditions. As the system progresses toward higher maturity levels (TRL 5–6), further optimization strategies, including model serving optimization, asynchronous execution, and infrastructure scaling, will be explored to reduce latency and support deployment in production environments.

3.5. Validation Against Acceptance Criteria

Table 3 summarizes the system's performance against the predefined acceptance criteria established in the validation methodology.

Table 3. Validation Results Against Acceptance Criteria.

Acceptance Criterion	Target	Result
Configuration correctness	90%	100%
Model training success rate	90%	100%
Predictive accuracy (MAPE $<$ 15)	70%	95%
Intent Recognition	90%	96%
Visualization completeness	100%	100%

The validation results confirm that the integrated system meets or exceeds the predefined functional and performance criteria required for a TRL 4 proof of concept. The system demonstrated high-level performance in intent recognition and achieved full compliance in configuration correctness, model training robustness, and visualization completeness, while demonstrating adequate predictive accuracy through the successful identification of models, meeting the defined MAPE threshold.

Overall, these results provide a solid empirical foundation for advancing the prototype toward TRL 5 (validation in a relevant environment) and, subsequently, TRL 6 (demonstration in an operational environment).

4. Discussion

4.1. A Holistic Systems Engineering Perspective

The validation results presented in Section 3 demonstrate that integrating advanced computational components into a coherent architecture yields system-level capabilities beyond the isolated functionality of individual subsystems. This holistic integration, which

includes natural language interaction, predictive simulation models, and interpretative outputs, exemplifies the principles of *Digital Twin Systems Engineering (DTSE)* [37], where information flows and interactions among subsystems are organized to achieve complex system behavior.

Our approach addresses two key challenges commonly highlighted in digital twin research [38,39]:

1. **Heterogeneous model integration:** bridging the semantic gap between natural language user inputs and structured configuration schemas through NLP-based parameter extraction.
2. **Bi-directional synchronization:** enabling real-time feedback loops where human operators can query the digital twin's state and adjust parameters dynamically.

The conversational interface lowers the technical barrier for non-specialist users (e.g., operations managers) to interact with sophisticated predictive models [40,41]. By encapsulating complexity behind confirmation dialogues, users with minimal technical training can deploy machine learning models with 100% model training success rate (Table 3), which represents a particularly strong result for a conversationally driven workflow. This highlights a central Industry 5.0 principle: technology should serve human needs rather than humans adapting to technology [42].

From a systems engineering perspective, the proposed architecture emphasizes robustness, fault tolerance, and modularity. Resilience to potentially invalid or hallucinated outputs from the LLM is ensured through strict validation of all configuration inputs and tool interfaces using Pydantic schemas, preventing the execution of parameters that violate predefined constraints. Additionally, the agent has retry mechanisms to adapt its outputs to the required format and recover from minor inconsistencies, while logging mechanisms may be incorporated to enable traceability and support continuous system refinement. The architecture is composed of loosely coupled components, where the digital twin operates as an independent API that continuously awaits requests, allowing asynchronous interaction between the agent and the training module. This decoupling improves robustness to latency and communication failures, as components can operate and recover independently; however, it currently limits concurrency, since training processes may introduce bottlenecks. Although latency is higher during model training due to its computational cost, communication between components remains stable and predictable, relying mainly on internal services, and can be further optimized to reduce response times.

Regarding predictive performance, the models successfully met the predefined acceptance criterion of a MAPE < 15% for at least 70% of the tests, confirming the feasibility of the predictive pipeline within the digital twin architecture. While this level of accuracy is adequate for decision-support scenarios at the TRL 4 stage, it also indicates that further improvements may be possible through the inclusion of richer operational datasets and more advanced modeling strategies.

Two design implications emerge from this work:

1. **Human-Centeredness:** The system prioritizes usability and natural interaction, enabling domain experts to leverage advanced analytics without requiring data science expertise.
2. **Modularity and Composability:** Clear separation between the interaction layer (conversational agent), simulation layer (digital twin), and decision support layer (visualization) allows incremental enhancement or technology substitution without redesigning the entire system.

4.2. Limitations of TRL 4 Validation

As a TRL 4 validation, this work exhibits several limitations inherent to the controlled proof-of-concept stage. These limitations are acceptable at this maturity level, where the primary goal is to demonstrate technical feasibility rather than operational readiness. Nonetheless, they must be acknowledged to guide future work toward TRL 5–6 deployments.

4.2.1. Data Acquisition and Realism

The dataset used for validation primarily consisted of 12 months of hotel operation data. While it captures representative patterns such as weekly seasonality, it lacks the noise and non-stationarity characteristic of real-world sensor data. Additionally, preliminary tests were conducted using occupancy data from a real hotel obtained from Kaggle [43]; however, these yielded high error metrics. This indicates that the current training mechanism requires further refinement through deeper hyperparameter optimization. Furthermore, future iterations should expand the agent's capabilities to identify and propose additional explanatory variables to enhance predictive accuracy. Consequently, future work must validate the system against live data streams, addressing challenges such as missing readings, calibration drift, and operational uncertainties.

4.2.2. Scalability and Performance

Model training and prediction generation times were within acceptable limits for an experimental prototype, with average response times of approximately 18.44 s for training and 15.44 s for prediction generation. While these results demonstrate the operational feasibility of the architecture at the proof-of-concept level, they also reveal potential performance bottlenecks if the system is scaled to larger datasets or multi-property deployments. Scaling the system to large-scale hospitality chains (hundreds of properties, millions of records) will require distributed computing infrastructure, incremental learning algorithms, and careful attention to the bias-variance tradeoff inherent in model selection across heterogeneous facilities [38].

4.2.3. Human Factors and Trust

Even at TRL 4, the system must consider human factors in mission-critical contexts. Misconfigurations, though infrequent in the proof of concept, could have significant operational consequences. Future research should investigate, for example, explainability mechanisms such as visualizing the agent's reasoning process, and develop fail-safe protocols to enhance user trust and reliability.

4.2.4. Dashboard Integration and Visualization

A conceptual visualization layer based on Power BI was designed as a future extension of the system to support the interpretation and use of predictive outputs in decision-making contexts. A standalone dashboard prototype has been developed using a separate dataset and database connection (Figure 7); however, it is not integrated with the conversational agent or the digital twin core and has not been formally validated. At the current stage, validation relies on embedded charts, which are sufficient for TRL4 functional testing. Full integration with enterprise BI tools, along with extensions to additional target variables such as electricity or water consumption, are planned for higher maturity levels (TRL 5–6).

4.3. Ethical Considerations

Although this study did not involve human subjects or personally identifiable information, we adhered to responsible AI principles:

- **Data privacy:** All operational data were simulated or anonymized. No real guest information was used.
- **Human oversight:** The system is designed for decision support rather than autonomous action; operational decisions remain with human facility managers.
- **Error awareness and uncertainty communication:** Model predictions are accompanied by quantitative error metrics (e.g., MAPE), allowing users to assess the reliability of the outputs. During validation, the predefined accuracy criterion (MAPE < 15% for at least 70% of the tests) was satisfied, providing an empirical basis for supporting informed decision-making.
- **Non-autonomous operation:** The role of the agent is limited to analysis and configuration support and it does not execute real-world actions or modify external systems.

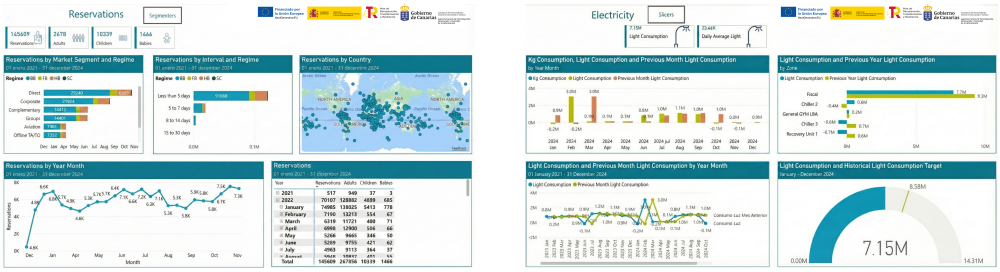

Figure 7. Conceptual design of a Power BI dashboard showing KPIs predicted by the digital twin. All data is fictitious and is provided solely to illustrate what a hotel data dashboard might look like.

4.4. Toward TRL 5–6: Roadmap for Operational Validation

Advancing this system to TRL 5 (validation in relevant environment) and TRL 6 (demonstration in operational environment) will require addressing the following technical and organizational challenges, particularly those related to scalability, real-world data variability, and system integration beyond the controlled validation environment.

4.4.1. Real-World Pilots

Deploy the system in 2–3 operational hotels with diverse characteristics (boutique vs. resort, urban vs. rural) to evaluate performance under realistic conditions. Metrics should include not only predictive accuracy but also decision impact—e.g., reduction in energy costs, improvement in waste diversion rates, and user satisfaction as measured by task completion time and perceived utility.

4.4.2. Continuous Learning and Model Drift

Implement online learning mechanisms that allow models to adapt to changing operational patterns (e.g., seasonal demand shifts, renovations, new sustainability initiatives) without requiring full retraining. This may involve ensemble methods that blend static models (trained on historical data) with dynamic models (updated incrementally via streaming data).

4.4.3. Multi-Stakeholder Governance

Infrastructure management involves diverse stakeholders (facility managers, sustainability officers, procurement teams, senior management). The conversational interface must evolve to support role-based interaction patterns, e.g., allowing sustainability officers to set carbon reduction targets that automatically propagate to model configuration, or enabling procurement teams to query cost-optimal resource allocation scenarios.

4.4.4. Prescriptive Analytics

The current system provides predictive insights ("electricity consumption will be X kWh next month"). The natural evolution is toward prescriptive analytics, recommending specific actions ("reduce HVAC setpoint by 2 °C in low-occupancy zones to save 12% energy while maintaining comfort"). This requires integrating optimization algorithms (e.g., mixed-integer programming, reinforcement learning) with the digital twin's predictive models [41].

5. Conclusions

This paper has presented the design, implementation, and TRL 4 validation of an integrated conversational AI agent and digital twin system for complex infrastructure management. The system achieves an overall intent recognition accuracy of 96% across 100 diverse functional test cases, 100% configuration correctness, and predictive model performance meeting the predefined acceptance threshold in 95% of evaluated configurations (mean MAPE 9.98%, lowest observed 8.77%), confirming the technical feasibility of the proposed architecture.

Beyond these quantitative results, the work was evaluated to demonstrate three qualitative contributions to systems engineering practice:

1. **Lowering Technical Barriers:** By encapsulating complex machine learning workflows behind a conversational interface, the system is designed to make advanced predictive analytics accessible to non-specialist users. The system design was informed through consultation with hotel management experts during the planning and development phases, aligning with human-centered design principles as defined in ISO 9241-210 [44], which emphasize early involvement of domain experts to improve usability and adoption. Preliminary evaluation through representative interaction scenarios indicates that users can configure models and define simulation parameters using natural language, without requiring knowledge of underlying ML processes. These results provide initial evidence of reduced technical barriers; however, validation with real end-users remains necessary.
2. **Emergent System-Level Capabilities:** The integration of conversational AI and digital twins creates a synergistic capability in the form of natural language-driven parametric simulation. This capability could not be achieved by either subsystem independently and reflects the systems engineering principle of emergence.
3. **Methodological Insights for TRL 4 Validation:** The structured validation methodology, including intent recognition, model performance, and traceability assessment, provides a replicable framework for evaluating similar AI-integrated systems.

A key limitation of this study is the absence of a large-scale, formal usability evaluation. While the research design was informed by continuous consultation with tourism industry experts and included a preliminary walkthrough with a non-technical hotel executive, these qualitative insights do not replace a larger-scale user study. Further research is necessary to confirm whether end-users can effectively integrate the proposed design into their daily operational workflows and to validate the system's utility across a broader range of professional profiles.

Looking ahead, the path to TRL 5–6 deployment requires addressing data realism, scalability, and human factors. The validated functional integration and interaction model are specifically designed to lower technical barriers, establishing a foundation for conversational, user-adaptive digital twins. However, confirming that this design objective translates into a measurable improvement for daily management remains a critical next step, requiring future validation in live operational contexts with a larger sample of end-users.

Author Contributions: Conceptualization, E.S.-O. and E.W.-S.; methodology, A.C.-R., M.Á.G.-E. and P.V.-M.; software, S.S.-G.; validation, M.Á.G.-E., P.V.-M. and C.V.-R.; writing—original draft preparation, E.S.-O., P.V.-M. and S.S.-G. 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.

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 following appendix presents the system prompt used to guide the behavior of the conversational agent. This prompt defines the agent's role, available actions, and interaction constraints, ensuring consistent and structured operation within the proposed architecture.

Listing A1: Agent System Prompt Definition.

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

You can help users:
- Check or display the initial or current configuration (tool: view
actual config)
- Update a configuration parameter (tool: change config value)
- Update multiple configuration parameters (tool: change config value)
- View the configuration file structure
- Save changes (tool: save config changes)
- Train models (tool: train model tool)
- Make predictions using a trained model (tool: predict model tool)

Instructions:
- Use the tools as needed and explain the changes you make.
- Display the current configuration settings when requested.
- Report any changes made and ask to save before exiting.
- Save changes only if the user requests it.
- Report any errors received by the tools.
- Training: Use 'train model tool'. Respond with model name, ID, and
metrics.
- CSV Updates: Ask for new CSV and modify 'project.data.path'.
- Predictions: Ask for Model ID. For ARIMA/SARIMA, ask for steps. For
others, require a CSV. Response consists solely of predictions.

Configuration structure:
{config structure}
```

Appendix A.2. System Configuration File

The following section presents the configuration file used within the system for data preprocessing and model training. This configuration defines the parameters that control feature selection, transformation processes, and training settings.

Listing A2: YAML Configuration Structure.

```yaml
project:
  name: Example
  data:
    path: "path/to/file.csv"
    split_ratio: 0.8

    # --- CSV Format Settings ---
    csv_sep: ","
    decimal_sep: "."
    encoding: "latin1"

    # --- Column Definitions ---
    date_column: "date_column_name"
    date_format: "%Y-%m-%d"
    target_variable: "target_column_name"
    numerical_columns:
      [
        "column_name_1",
        "column_name_2",
        "column_name_3",
      ]

  preprocessing:
    aggregation_method: "last"
    missing_value_method: none
    apply_outlier: true
    lagged_features: [1, 2]
    apply_correlation: true
    apply_vif: true
    x_scaling: none
    y_scaling: none
    frequency_options: "D"
    n_periods: 30

  results:
    output_dir: "path/to/results/"
    filename: "filename.csv"

  models:
    estimators:
      default:
        - MovingAverageModel
        - LinearRegression
        - DecisionTreeRegressor
        - RandomForestRegressor
        - XGBRegressor
        - StatsmodelsARIMAModel
        - StatsmodelsSARIMAModel
```

Appendix B. Examples of Interactions with the Digital Twin Configuration Agent

This appendix provides a series of documented interactions between human operators and the proposed AI agent. These cases illustrate the system's robustness in translating natural language instructions into precise configuration parameters for industrial Digital Twins. The following examples demonstrate the agent's capability to handle linguistic variability, maintain contextual continuity, and interpret suboptimal inputs without compromising the integrity of the configuration process.

Appendix B.1. Syntactic Invariance and Semantic Mapping

This section illustrates the agent's ability to process syntactically diverse inputs that convey identical functional requirements. The provided screenshots (Figures A1 and A2) demonstrate that whether an operator uses passive voice, a short, direct order or a hesitant tone, the agent consistently identifies the underlying intent. This ensures that the configuration logic remains decoupled from specific phrasing, allowing for a flexible user interface that accommodates various communicative styles.

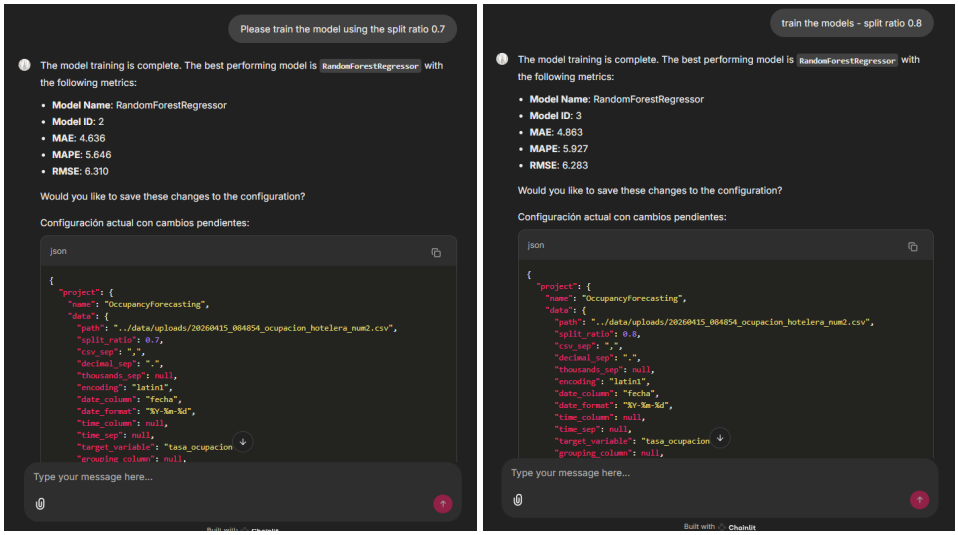

Figure A1. Examples of interactions with syntactically different inputs (1).

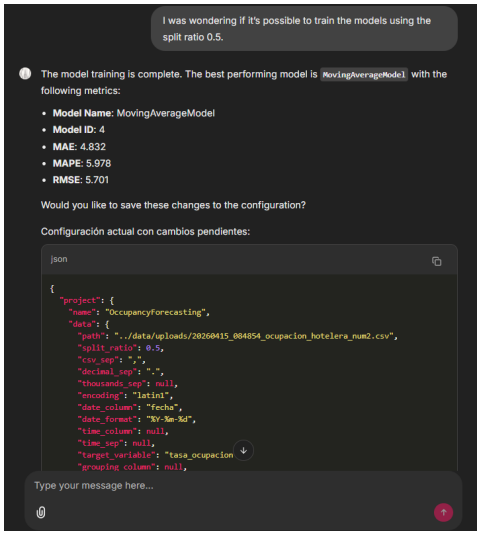

Figure A2. Examples f interactions with syntactically different inputs (2).

Appendix B.2. Contextual Persistence in Multi-Turn Dialogues

The following interactions highlight the agent's proficiency in managing multi-turn conversations where subsequent instructions depend on prior state information. In these scenarios, as shown in the example at Figure A3 the agent maintains a persistent internal representation of the session, allowing it to resolve anaphoric references (e.g., "use the configuration to run the training") and execute incremental adjustments. This capability is critical for complex setup tasks that cannot be effectively articulated in a single, isolated command.

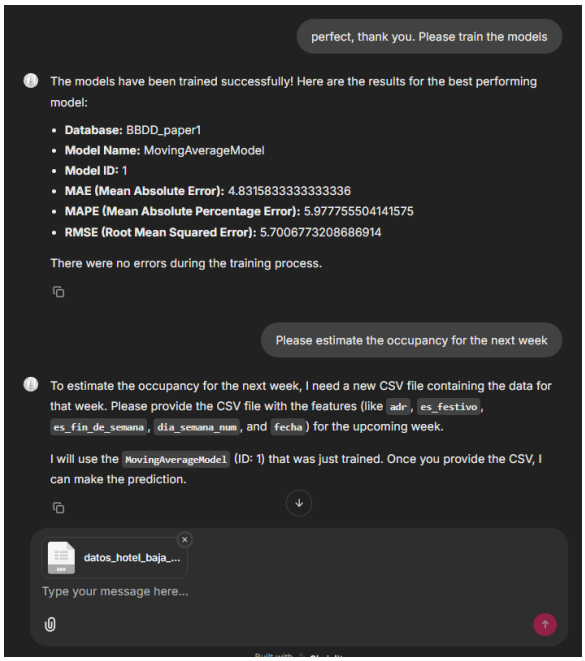

Figure A3. Examples of a multi-turn interaction.

Appendix B.3. Heuristic Processing of Informal Language and Typographical Errors

Figure A4 presents an English example of the agent's resilience when faced with informal register, colloquialisms, and common typographical mistakes, similar to those validated during the research. The system correctly infers the operator's intent despite "noisy" input, thereby reducing the need for repetitive manual corrections.

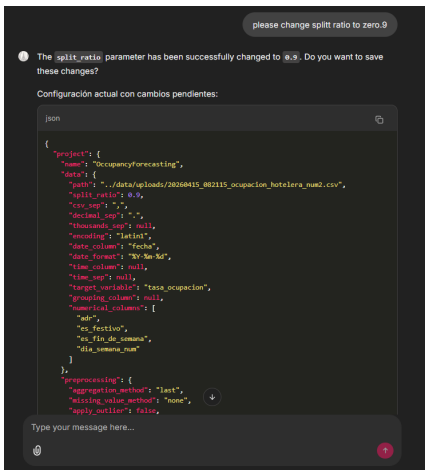

Figure A4. Example of an interaction with informal language.

References

1. Wodak, D.; Ou, C.; Akkermans, H.; de Rouw, L.-P. Scalability of digital twins for physical infrastructure management: A dynamic governance approach. In *International Conference on Information Systems (ICIS 2024)*; Association for Information Systems: Atlanta, GA, USA, 2024.
2. Norman, D.A. *Design for a Better World: Meaningful, Sustainable, Humanity-Centered*; MIT Press: Cambridge, MA, USA, 2023.
3. Santos, R.; Constantinou, E.; Antonino, P.; Bosch, J. Software engineering for systems-of-systems and software ecosystems. *Inf. Softw. Technol.* **2024**, *165*, 107335. [CrossRef]
4. Falegnami, A.; Tomassi, A.; Corbelli, G.; Romano, E. Managing complexity in socio-technical systems by mimicking emergent simplicities in nature: A brief communication. *Biomimetics* **2024**, *9*, 322. [CrossRef]
5. Moshood, T.D.; Rotimi, J.O.; Shahzad, W.; Bamgbade, J.A. Infrastructure digital twin technology: A new paradigm for future construction industry. *Technol. Soc.* **2024**, *77*, 102519. [CrossRef]
6. Broo, D.G.; Schooling, J. Digital twins in infrastructure: Definitions, current practices, challenges and strategies. *Int. J. Constr. Manag.* **2023**, *23*, 1254–1263. [CrossRef]
7. Grieves, M.; Vickers, J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary Perspectives on Complex Systems*; Springer: Cham, Switzerland, 2023.
8. 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]
9. Kusal, S.; Patil, S.; Choudrie, J.; Kotecha, K.; Mishra, S.; Abraham, A. AI-based conversational agents: A scoping review from technologies to future directions. *IEEE Access* **2022**, *10*, 92337–92356. [CrossRef]
10. Acikgoz, E.C.; Hakkani-Tur, D.; Tur, G. The Rise of Conversational AI Agents with Large Language Models, 2024. Available online: https://emrecanacikgoz.github.io/Conversational-Agents/ (accessed on 11 February 2026).
11. Merzouki, R.; Lakhal, O.; Aïtouche, A. (Eds.) 18th International Conference on System of Systems Engineering (SoSE 2023): AI and Autonomous Robotics in System of Systems, University of Lille, Polytech Lille, France, 14–16 June 2023. Available online: http://www.sosengineering.org/2023/ (accessed on 11 February 2026).
12. IEEE Systems Council, System of Systems Technical Committee. 2024. Available online: https://www.ieeesmc.org/technical-activities/systems-science-and-engineering/system-of-systems/ (accessed on 11 February 2026).
13. Mumford, E. A socio-technical approach to systems design. *Requir. Eng.* **2000**, *5*, 125–133. [CrossRef]
14. McKinsey & Company. Digital Twins: Boosting ROI of Government Infrastructure Investments, 3 July 2025. Available online: https://www.mckinsey.com/industries/public-sector/our-insights/digital-twins-boosting-roi-of-government-infrastructure-investments (accessed on 11 February 2026).
15. McKinsey & Company. What Is Digital-Twin Technology? 26 August 2024. Available online: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-digital-twin-technology (accessed on 11 February 2026).
16. Zhong, D.; Xia, Z.; Zhu, Y.; Duan, J. Overview of predictive maintenance based on digital twin technology. *Heliyon* **2023**, *9*, e14534. [CrossRef]
17. Dinter, R.V.; Tekinerdogan, B.; Catal, C. Predictive maintenance using digital twins: A systematic literature review. *Inf. Softw. Technol.* **2022**, *151*, 107008. [CrossRef]
18. Alibrandi, U. Risk-informed digital twin of buildings and infrastructures for sustainable and resilient urban communities. *ASCE-ASME J. Risk Uncert. Eng. Syst. Part A Civil Eng.* **2022**, *8*, 04022032. [CrossRef]
19. Kaewunruen, S.; Sresakoolchai, J.; Ma, W.; Phil-Ebosie, O. Digital twin aided vulnerability assessment and risk-based maintenance planning of bridge infrastructures exposed to extreme conditions. *Sustainability* **2021**, *13*, 2051. [CrossRef]
20. Kerkeni, R.; Khlif, S.; Mhalla, A.; Bouzrara, K. Digital twin applied to predictive maintenance for Industry 4.0. *J. Nondestruct. Eval. Diagn. Progn. Eng. Syst.* **2024**, *7*, 041008. [CrossRef]
21. Zhu, M.; Jin, J. Data-driven urban digital twins and critical infrastructure under climate change: A review of frameworks and applications. *Urban Plan.* **2025**, *10*, 10109. [CrossRef]
22. Chen, C.; Fu, H.; Zheng, Y.; Tao, F.; Liu, Y. The advance of digital twin for predictive maintenance: The role and function of machine learning. *J. Manuf. Syst.* **2023**, *71*, 581–594. [CrossRef]
23. Feng, K.; Ji, J.C.; Zhang, Y.; Ni, Q.; Liu, Z.; Beer, M. Digital twin-driven intelligent assessment of gear surface degradation. *Mech. Syst. Signal Process.* **2023**, *186*, 109896. [CrossRef]
24. Keertana, M.; Vijay Vishnu Kumar, C. Digital Twin Applications in Tourism Destination Management: Predictive Modelling for Tourist Flow and Resource Optimization. *Asian Rev. Soc. Sci.* **2025**, *14*, 43–49. [CrossRef]
25. Yao, S.; Shinn, N.; Razavi, P.; Narasimhan, K. τ-Bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv* **2024**, arXiv:2406.12045.
26. Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; et al. AgentBench: Evaluating LLMs as agents. *arXiv* **2023**, arXiv:2308.03688. [CrossRef]

27. Bravo, L.; Rodriguez, C.; Hidalgo, P.; Angulo, C. A systematic review on artificial intelligence-based multimodal dialogue systems capable of emotion recognition. *Multimodal Technol. Interact.* **2025**, *9*, 28. [CrossRef]
28. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv* **2023**, arXiv:2308.08155. [CrossRef]
29. Casheekar, A.; Lahiri, A.; Rath, K.; Prabhakar, K.S.; Srinivasan, K. A contemporary review on chatbots, AI-powered virtual conversational agents, ChatGPT: Applications, open challenges and future research directions. *Comput. Sci. Rev.* **2024**, *52*, 100632. [CrossRef]
30. Google AI. Gemini Models API. Available online: https://ai.google.dev/gemini-api (accessed on 11 February 2026).
31. Microsoft Corporation. Power BI Documentation. Available online: https://docs.microsoft.com/en-us/power-bi/ (accessed on 11 February 2026).
32. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; CRC Press: Boca Raton, FL, USA, 1984.
33. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
34. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016*; ACM: New York, NY, USA, 2016; pp. 785–794. [CrossRef]
35. Chainlit Documentation. Chainlit: Documentation for Building Conversational AI Applications. Available online: https://docs.chainlit.io/ (accessed on 11 February 2026).
36. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
37. Zhang, H.; Li, Y.; Zhang, S.; Song, L.; Tao, F. Artificial intelligence-enhanced digital twin systems engineering towards the industrial metaverse in the era of Industry 5.0. *Chin. J. Mech. Eng.* **2025**, *38*, 40. [CrossRef]
38. Michael, J.; Pfeiffer, J.; Rumpe, B.; Wortmann, A. Integration challenges for digital twin systems-of-systems. In *Proceedings of the 10th IEEE/ACM International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems (SESoS)*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 9–12. [CrossRef]
39. Dihan, M.S.; Akash, A.I.; Tasneem, Z.; Das, S.K.; Islam, M.R. Digital twin: Data exploration, architecture, implementation and future. *Heliyon* **2024**, *10*, e26503. [CrossRef]
40. Rasheed, A.; San, O.; Kvamsdal, T. Digital Twin: Values, Challenges and Enablers from a Modeling Perspective. *IEEE Access* **2020**, *8*, 21980–22012. [CrossRef]
41. Liu, G.-P. Control Strategies for Digital Twin Systems. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 170–180. [CrossRef]
42. Dhilipan, J.; Saravanan, V.; Agusthiyar, R. (Eds.) *Human Machine Interaction in the Digital Era: Towards Conversational Artificial Intelligence*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2024. [CrossRef]
43. Mojtaba, M. Hotel Booking Demand Dataset. Kaggle, 2019. Available online: https://www.kaggle.com/datasets/mojtaba142/hotel-booking (accessed on 9 April 2026).
44. *ISO 9241-210:2019*; Ergonomics of Human-System Interaction—Part 210: Human-Centred Design for Interactive Systems. International Organization for Standardization: Geneva, Switzerland, 2019.