A Plug-and-Play Semantic Execution Framework for Reusable Cross-Domain Automation in Digital Twins

Chien-Pu Huang[1] **and Shang-Hsien Hsieh**[1]

[1]Department of Civil Engineering, National Taiwan University, Taipei, Taiwan

chienpu@gmail.com, shhsieh@ntu.edu.tw

Abstract -

This research proposes a plug-and-play *semantic execution framework* that enables Digital Twin–oriented automation to be configured through reusable domain knowledge rather than project-specific code. The framework is built on *Domain Packs*—modular semantic units encapsulating ontologies, data mappings, reasoning rules, workflow templates, and provenance policies—and a set of domain-agnostic agents operating over a shared knowledge graph through a unified Trigger–Issue–Action–Actor (TIAA) event model. The framework is demonstrated through a representative cross-domain Digital Twin–enabled scenario in a campus environment, where IoT-enabled building systems support proactive responses to equipment anomalies while coordinating maintenance and procurement actions under organisational constraints. The scenario highlights automation requirements that extend beyond data synchronisation, including semantic anomaly interpretation, impact assessment, cross-domain action coordination, and traceable execution. The results show that coordinated automation can emerge from composing reusable semantic capabilities within a domain-agnostic runtime, without predefined workflows or hard-coded integration logic. Overall, the findings indicate that introducing a reusable semantic execution layer enables scalable, explainable, and workflow-oriented interoperability for Digital Twin applications in the AECO domain.

Keywords -

Digital Twin automation; Semantic interoperability; Domain Packs; Multi-agent systems; Knowledge graphs; Built environment

1 Introduction

1.1 From Digitalisation to Smart Automation

Over the past decade, the AECO industry has progressed from data-centric digitalisation toward increasingly automated and intelligent operational systems [1]. Early Digital Twin initiatives primarily focused on data synchronisation between physical assets and virtual models. More recent efforts seek to embed decision logic and automation capabilities into these environments [2]. However, while data interoperability has improved, automation logic remains tightly coupled to project-specific scripts and platform-dependent implementations [3]. As a result, automation solutions are costly to configure, difficult to transfer across projects, and rarely reusable across domains.

This limitation reveals a critical shift in the automation bottleneck. The primary challenge is no longer data availability or model integration, but the absence of a reusable semantic execution layer capable of transforming heterogeneous domain knowledge into operational, explainable actions [4]. Without such a layer, Digital Twin systems remain data-rich but action-poor, unable to support scalable and sustainable automation across the built environment lifecycle. This limitation highlights the need for action-oriented interoperability rather than data integration alone.

1.2 The Executable Knowledge Gap

Despite widespread adoption of BIM, IoT platforms, and facility management systems, domain knowledge in the built environment remains fundamentally non-executable. Knowledge related to faults, maintenance logic, procurement rules, and organisational procedures is repeatedly reconstructed because it is embedded in static ontologies, documentation, or hard-coded automation scripts that cannot be reused or directly executed across systems [5]. Consequently, integration efforts prioritise data exchange rather than knowledge continuity, resulting in brittle automation, duplicated engineering effort, and limited traceability between sensing, reasoning, and action.

This structural disconnect constitutes an *executable knowledge gap*: domain expertise exists in semantic form but cannot function as a persistent, computable asset across domains or lifecycle stages. Addressing this gap requires rethinking automation not as a collection of scripts or workflows, but as a semantic execution problem—one in which domain knowledge must be packaged, interpreted, and enacted in a reusable and auditable manner.

1.3 Research Aim and Anchor Claim

This paper argues that the scalability of Digital Twin automation is limited not only by data integration challenges, but by the absence of a *plug-and-play semantic execution mechanism* that allows domain knowledge to be reused as an operational asset. To address this gap, the study proposes a semantic execution framework in which automation behaviour is configured through semantic composition rather than code modification.

In this framework, "plug-and-play" refers specifically to the runtime reusability of semantic execution logic through Domain Packs. The framework operates as a semantic execution layer within Digital Twin architectures, decoupling domain knowledge from execution logic and enabling automation behaviour to be determined dynamically and audibly at runtime.

1.4 Proposed Approach and Objectives

The proposed framework is centred on *Domain Packs*—modular semantic units that encapsulate ontologies, data mappings, executable reasoning rules, workflow templates, and provenance policies for a given domain. These Packs are interpreted by a reusable set of domain-agnostic agents operating over a shared knowledge graph. Central to this execution process is the Trigger–Issue–Action–Actor (TIAA) abstraction [6], which formalises how semantic reasoning outcomes are transformed into actionable, auditable events across domains. To realise this approach, the study pursues four objectives:

1. Define the structure and lifecycle of Domain Packs as reusable units of executable semantics.
2. Develop a domain-agnostic multi-agent semantic runtime capable of interpreting Pack-defined logic.
3. Formalise the TIAA model as a minimal, cross-domain semantic execution abstraction.
4. Demonstrate cross-domain reusability across heterogeneous operational domains without modifying agent implementations.

1.5 Paper Structure

Section 2 reviews related work on semantic modelling, Digital Twins, and agent-based automation. Section 3 presents the Domain Pack concept and the multi-agent semantic runtime. Section 4 demonstrates the framework through a representative cross-domain Digital Twin automation scenario. Section 5 evaluates reusability, cross-domain reasoning, and auditability. Section 6 discusses implications for automation scalability and Digital Twin ecosystems. Section 7 concludes the paper and outlines future work.

2 Background and Related Work

2.1 Semantic Modelling and Knowledge Graphs in AECO

While ontologies and knowledge graphs (KGs) are established tools for integrating BIM and IoT data in the AECO industry, they are primarily used as descriptive or query-oriented representational models [7]. Existing research largely focuses on formalizing static domain concepts and enabling semantic queries, but rarely encapsulates reasoning patterns, workflow behaviors, or governance policies as reusable, executable units. As a result, ontological knowledge can describe system states but cannot directly drive runtime automation or decision execution [2]. Moreover, most semantic models remain confined to single domains, limiting the semantic composition—such as linking maintenance alerts to procurement workflows—required for intelligent building operations. Addressing these limitations requires a shift from descriptive semantics toward executable semantic units that transform domain knowledge into operational logic.

2.2 Automation and Digital Twins

Digital twins (DTs) in the AECO domain aim to synchronise physical and virtual system states. However, while contemporary DT platforms excel at data sensing and simulation, their operational semantics are typically embedded in platform-specific scripts or procedural logic [8]. This tight coupling makes automation developed for one environment difficult to port or reuse, severely limiting scalability across systems and lifecycle stages. More critically, current DTs lack reusable semantic action patterns for multi-domain operational reasoning [2]; for example, a detected maintenance anomaly cannot automatically propagate into procurement or resource planning workflows. As a result, DTs often function as data mirrors rather than integrated automation platforms. Overcoming this limitation requires a semantic execution layer that decouples automation behavior from system-specific implementations, enabling reusable and coordinated automation across domains.

2.3 Multi-agent Systems and AI for Construction

Multi-agent systems (MAS) have demonstrated potential for distributed coordination and automated task planning in the AECO domain [9]. However, most existing agent-based implementations lack grounding in formal ontologies, limiting their ability to reason over structured domain semantics [10]. In addition, agent behaviors are typically handcrafted and tightly coupled to specific applications, rendering them non-portable across projects and lifecycle contexts. Without modular and reusable semantic units, agents cannot reliably share logic or exploit cross-domain knowledge. To address these limitations, this research decouples execution logic from domain knowledge by introducing a domain-agnostic semantic runtime, in which agents act as interpreters of modular Domain Packs (reusable semantic logic units). This architecture enables agents to interpret, execute, and audit complex domain semantics across heterogeneous AECO scenarios without requiring code-level modifications.

3 Domain Pack and Multi-Agent Semantic Runtime

This section presents a reusable, cross-domain semantic execution framework for digital twin applications in the AECO domain. As shown in Figure 1, the architecture separates physical reality, data representation, semantic capability provisioning, graph-based reasoning, and execution-oriented applications. Semantic automation is enabled by two tightly coupled elements: (1) modular *Domain Packs*, which encapsulate reusable semantic capability modules such as ontologies, reasoning rules, workflow templates, and governance constraints; and (2) a domain-agnostic multi-agent semantic runtime that interprets and enacts these capabilities over a shared knowledge graph serving as semantic execution memory.

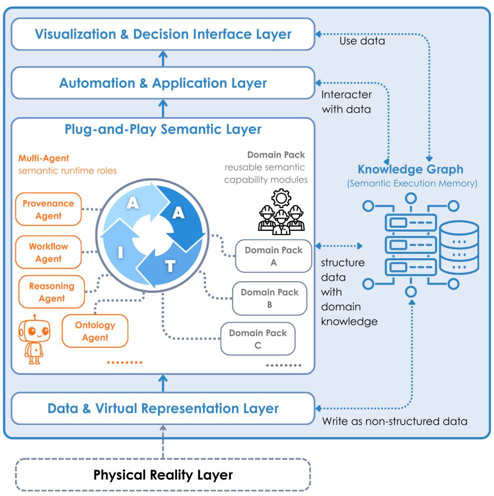

Figure 1. Reusable cross-domain digital twin architecture based on Domain Packs and a domain-agnostic multi-agent semantic runtime.

In this framework, agents denote logical semantic runtime roles rather than persistent system services. Each agent assumes specific functional responsibilities during execution, while coordination across agents and Domain Packs is mediated through the Trigger–Issue–Action–Actor (TIAA) abstraction, which standardises domain-specific reasoning outcomes into actionable events and enables interoperable execution across heterogeneous domains.

3.1 Domain Pack Concept

A Domain Pack is a self-contained semantic execution unit that encapsulates the domain knowledge and operational logic required to automate tasks within a specific AECO context. Unlike descriptive ontologies or configuration artefacts, which are typically passive and non-executable, a Domain Pack functions as an active semantic plug-in: when loaded into the semantic runtime, it configures agent behaviour through semantic specification alone, without modifying agent code. This elevates domain knowledge from static representation to an executable and reusable asset.

As shown in Figure 2, a Domain Pack encapsulates three categories of executable semantic assets: (i) semantic alignment assets for unifying heterogeneous data sources, (ii) executable semantic assets for generating TIAA-aligned events, and (iii) governance and provenance assets that ensure traceable and auditable automation.

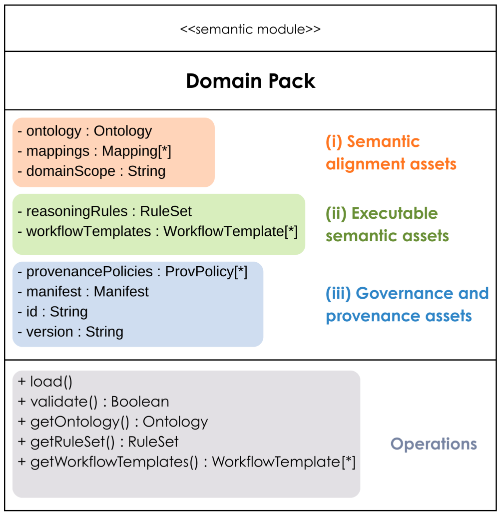

Figure 2. Conceptual structure of a Domain Pack as a modular semantic execution unit.

Although Domain Pack creation requires upfront semantic engineering effort, this effort is structured to allow domain experts to contribute executable knowledge without direct involvement in semantic web formalisms. Contributors focus on operational logic, data relevance, and decision rules, which are encoded as executable semantic assets by the framework.

In practice, Domain Packs are authored through a lightweight process: (i) identifying operational triggers and decisions, (ii) defining relevant domain entities and observable states, (iii) mapping existing BIM, IoT, or organisational data sources to semantic elements, (iv) specifying Trigger–Issue–Action–Actor (TIAA) rules, and (v) packaging these assets as a versioned Domain Pack.

Domain Packs evolve independently of the runtime through versioned updates, allowing domain logic to be refined without disrupting domain-agnostic agents and supporting scalable cross-domain automation across the building lifecycle.

3.2 Multi-Agent Semantic Runtime

Automation is executed by a set of domain-agnostic agents operating over a shared knowledge graph. Rather than embedding domain logic, these agents act as *semantic interpreters* that activate execution behaviours defined by loaded Domain Packs. This separation between domain knowledge (Domain Packs) and execution capabilities (agents) enables configurable, extensible, and cross-domain automation.

Figure 3 illustrates the semantic runtime architecture, highlighting how agents interact with the Knowledge Graph and coordinate execution through Trigger–Issue–Action–Actor (TIAA) events in an event-driven manner.

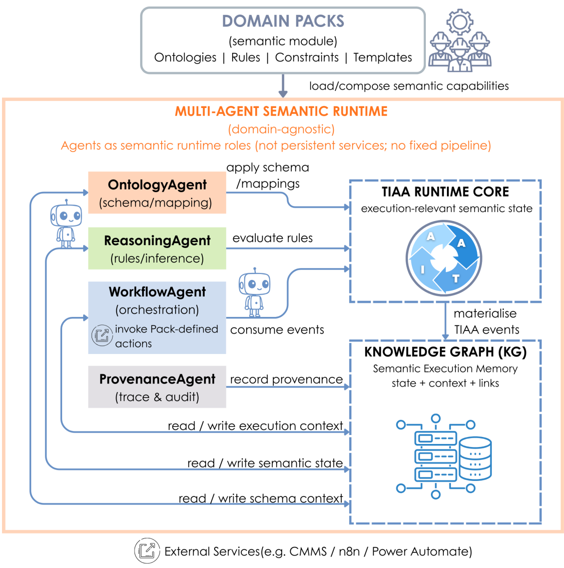

Figure 3. Multi-agent semantic runtime enabling reusable, cross-domain Digital Twin automation through Domain Pack composition, TIAA-based event abstraction, and a shared Knowledge Graph.

Within the runtime, agents assume complementary roles. The *Ontology Agent* applies ontologies and mappings from Domain Packs to establish a unified semantic layer. The *Reasoning Agent* evaluates executable semantic logic and emits standardised TIAA events. The *Workflow Agent* consumes these events to activate Pack-defined workflows or external services, while the *Provenance Agent* records event lineage and execution outcomes to ensure traceable and auditable automation across domains.

3.3 The TIAA Operational Semantic Model

This subsection defines the Trigger–Issue–Action–Actor (TIAA) model as a domain-agnostic operational semantic abstraction that enables action-oriented interoperability across heterogeneous domains. Rather than coupling semantic reasoning directly to domain-specific workflows, TIAA functions as a minimal *execution contract* that decouples reasoning outcomes from execution mechanisms [6].

Within the proposed framework, domain knowledge is externalised into *Domain Packs*, while execution is performed by domain-agnostic agents. Domain Packs do not directly invoke workflows or services; instead, domain-specific reasoning results are first normalised into a shared TIAA structure that can be interpreted uniformly at runtime. Through this indirection, heterogeneous semantic logic can be translated into actionable and traceable operational outcomes without introducing domain-specific integration logic.

At an abstract level, TIAA defines a consistent four-stage semantic chain:

1. **Trigger**: Represents the semantic ingestion of domain-specific conditions (e.g., IoT anomalies) as defined by Pack-level mappings.
2. **Issue**: Captures the semantic interpretation of a trigger as a higher-level operational or organisational concern, based on Pack-defined reasoning logic.
3. **Action**: Maps the identified issue to modular action templates, which may be composed across one or more Domain Packs (e.g., maintenance and procurement).
4. **Actor**: Assigns execution responsibility to an appropriate role, system, or human decision-maker, enabling both automated execution and human-in-the-loop workflows.

By adopting TIAA as a runtime intermediary, the Ontology, Reasoning, Workflow, and Provenance agents operate without embedded domain logic, and system behaviour is determined entirely through semantic composition. This separation enables a plug-and-play execution architecture in which the underlying runtime remains invariant while domain-specific requirements evolve across the facility lifecycle. A concrete cross-domain instantiation of the TIAA model and its execution dynamics is demonstrated in the following section.

4 Implementation scenario

This section demonstrates the proposed semantic execution framework through a single, representative cross-domain Digital Twin automation scenario. The scenario is intentionally designed to show how coordinated automation emerges from reusable semantic capabilities—rather than predefined workflows—by composing Domain Packs within a domain-agnostic semantic runtime.

4.1 Scenario context

The scenario considers a university campus in which IoT-enabled building systems are deployed across dormitory facilities. Facility managers aim to respond proactively to equipment anomalies while coordinating maintenance and procurement activities under organisational, contractual, and policy constraints. This context is representative of many campus-scale and institutional Digital Twin deployments, where operational sensing, asset management, and administrative decision-making intersect but are rarely automated in a coordinated manner.

4.2 Digital Twin automation requirements

From a Digital Twin perspective, the scenario reveals automation requirements that transcend simple data synchronization. These include the semantic detection of anomalies from sensors, the assessment of operational impacts, and the coordination of procurement and maintenance actions with full traceability. Given that these requirements span multiple organizational domains, they necessitate a framework capable of cross-domain semantic reasoning and action coordination.

4.3 Data sources and semantic alignment

All data sources are integrated within a single operational scenario to illustrate semantic heterogeneity rather than to conduct multiple case studies. The scenario incorporates heterogeneous data commonly found in campus environments, including time-series sensor observations from building management systems, equipment and asset descriptions derived from BIM models or asset inventories, and procurement records describing vendors, contracts, and component specifications. Rather than coupling automation logic directly to these heterogeneous sources, all inputs are aligned into a shared Knowledge Graph using ontologies and mappings provided by Domain Packs. This semantic alignment decouples domain knowledge from data formats and system-specific interfaces, ensuring that automation behaviour remains reusable and transferable across systems.

4.4 Domain Packs as semantic capability providers

Two Domain Packs provide complementary semantic capabilities for this scenario:

- **Predictive Maintenance (PdM) Pack**, which defines equipment- and sensor-level semantics, mappings for operational data, and executable rules for anomaly detection; and
- **Procurement Pack**, which defines procurement-related semantics, sourcing constraints, and action templates for procurement decisions.

Importantly, Domain Packs do not prescribe execution sequences. Instead, they supply reusable semantic assets that can be dynamically composed at runtime to support cross-domain automation.

4.5 Agent roles in the scenario

The domain-agnostic agents interpret and execute the composed semantic assets as follows:

- **Ontology Agent** integrates schemas and mappings from both Domain Packs into a unified Knowledge Graph.
- **Reasoning Agent** evaluates Pack-defined rules and produces TIAA events that encode cross-domain semantic dependencies for execution.
- **Workflow Agent** consumes TIAA events and instantiates coordinated workflows across facility management and procurement domains.
- **Provenance Agent** records end-to-end semantic traces linking data inputs, reasoning outcomes, and executed actions.

Each agent operates without embedded domain logic; its behaviour is configured entirely by the semantic assets provided by the loaded Domain Packs.

4.6 Execution outcomes and TIAA instantiation

This subsection addresses a core operational question in campus-scale Digital Twin automation:

> *How can a campus Digital Twin determine whether a sensor-detected equipment anomaly should trigger maintenance actions, procurement processes, or human intervention under organisational constraints?*

Rather than relying on predefined pipelines or hard-coded integrations, the proposed framework resolves this question through runtime semantic composition. Figure 4 illustrates how coordinated automation emerges through the instantiation and execution of a Trigger–Issue–Action–Actor (TIAA) event.

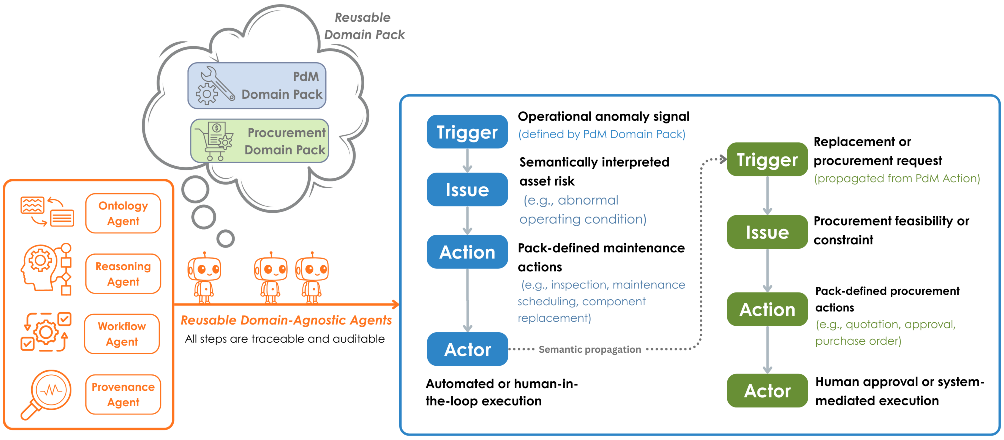

Figure 4. Cross-domain semantic execution through concurrent TIAA chains, where an Action from one Domain Pack is semantically propagated as a Trigger to another.

In this example, a sensor anomaly is first normalised as a *Trigger* according to Pack-defined semantic conditions and interpreted as an *Issue* representing an operational or asset risk. Based on this interpretation, candidate *Action* types are identified using modular templates provided by the active Predictive Maintenance and Procurement Domain Packs.

Execution responsibility is then assigned to an appropriate *Actor*, distinguishing between automated execution and human-in-the-loop decision-making. Crucially, the same TIAA abstraction is preserved across domains, allowing Actions generated in one Domain Pack to be semantically propagated as Triggers for another without modifying agent implementations.

This instantiation demonstrates that coordinated maintenance and procurement behaviour arises from semantic interpretation at runtime rather than predefined control flows. All reasoning outcomes and executions are recorded as provenance, ensuring traceable and auditable automation across organisational and domain boundaries.

5 Validation Approach

Validation focuses on architectural behaviour rather than numerical performance, evaluating whether the proposed framework enables action-oriented automation in complex AECO environments. This choice aligns with the study's objective of assessing reusability and semantic composability rather than optimisation performance. Nevertheless, the proposed framework explicitly exposes measurable runtime characteristics, including semantic reasoning latency, knowledge graph query time, and the computational overhead of concurrent agent execution, which are relevant to real-time Digital Twin operation. Accordingly, three criteria are examined: (i) reuse of execution logic across heterogeneous domains, (ii) cross-domain reasoning and action propagation without custom integration code, and (iii) traceability and auditability of automated decisions.

The objective is to demonstrate that reusable and auditable automation emerges through semantic composition, with execution behaviour governed by Domain Packs and the unified Trigger–Issue–Action–Actor (TIAA) abstraction rather than hard-coded workflows.

5.1 Reusability across Domain Packs

Across all evaluated scenarios, the same four domain-agnostic agents executed all automation tasks without modification to their source code, container images, or orchestration configurations. Only the active Domain Pack—or combination of Packs—was changed.

At runtime, the Ontology Agent loaded Pack-specific schemas and mappings, the Reasoning Agent evaluated Pack-defined semantic rules, and the Workflow Agent instantiated Pack-defined action templates. This confirms that execution behaviour is driven entirely by semantic configuration, validating that reusability resides in the Domain Pack layer rather than within agent implementations.

5.2 Cross-domain reasoning

The smart campus scenario demonstrates that Domain Packs can be composed without modifying the semantic runtime. Both the Predictive Maintenance (PdM) Pack and the Procurement Pack were activated concurrently, while the Reasoning Agent—running an unchanged code base—generated TIAA events spanning entities from both Packs.

Through the shared TIAA structure, dependencies such as procurement requirements affecting maintenance tasks, delays influencing timelines, and anomalies triggering coordinated actions were expressed. This confirms that cross-domain automation emerges from semantic composition and shared execution abstractions, rather than from custom cross-module scripting.

5.3 Auditable automation

To support governance and accountability, the Provenance Agent generated PROV-O activity–entity–agent traces for all reasoning and workflow steps. Each workflow instance was explicitly linked to its originating reasoning activity and Domain Pack version.

This provenance structure enables systematic inspection of why a decision occurred, which data and rules contributed to it, and how the resulting action was executed, ensuring transparent, traceable, and audit-ready automation.

6 Discussion

The validation results highlight a persistent limitation in current AECO systems: while data interoperability and digital twin platforms support information integration, they lack a semantic execution layer capable of driving reusable, cross-domain automation. This work operationalises such a layer by coupling Domain Packs with a domain-agnostic semantic runtime.

6.1 Domain Packs versus existing approaches

Ontologies. Existing AECO ontologies (e.g., IFC, Brick, SOSA/SSN) provide descriptive semantics but do not encode executable behaviour. Automation therefore relies on custom scripts or platform-specific logic. Domain Packs extend ontologies by bundling executable semantics—rules, workflow templates, mappings, and provenance policies—into portable, runtime-interpretable modules.

Machine-learning pipelines. Machine-learning approaches typically produce task-specific outputs bound to fixed schemas and lack explainable execution semantics. When transferred across domains, they require retraining or reintegration. Domain Packs instead externalise domain variability into explicit semantic artefacts, enabling reusable, explainable automation through a stable runtime and unified TIAA abstraction.

Digital twin platforms. Most digital twin platforms are asset- or geometry-centric and embed workflow logic through platform-specific connectors. The proposed framework complements such platforms by acting as a semantic control layer: the twin supplies data, while Domain Packs govern how data propagates into reasoning, action, and auditable execution.

6.2 Implications and limitations

The results indicate that scalable AECO automation benefits from an explicit semantic execution layer:

- **Scalability**: a single runtime supports multiple domains through reusable semantic Packs.
- **Standardisation**: Domain Packs encapsulate automation logic as structured, shareable semantic units.
- **Automation readiness**: TIAA links semantic reasoning to executable decisions with built-in traceability.

Practical limitations remain. Developing high-quality Domain Packs requires upfront semantic engineering effort, and Pack-defined workflows must align with heterogeneous enterprise systems (e.g., CMMS, ERP), typically through service-based interfaces exposed by the Workflow Agent (e.g., REST-based APIs). These conditions define when Domain Packs deliver maximal value rather than limiting the framework's applicability.

The architecture is also designed to accommodate learning-enabled reasoning as a future extension. Machine learning models may be integrated as data-driven Trigger providers or as auxiliary components within the Reasoning Agent to support Issue classification and Action prioritisation. Learning outputs are interpreted through the same TIAA abstraction rather than executed directly, preserving explainability, traceability, and governance.

Governance policies are essential in large-scale, multi-Pack deployments. Conflicting rules or Actions are treated as governed decision points, resolved through Pack-level priorities, policy constraints, or human approval via the Workflow Agent. All conflicts and resolution outcomes are recorded as provenance, ensuring transparent and auditable automation in high-stakes environments.

Overall, the framework advances *action-oriented interoperability*, in which semantic assets move beyond description to directly enable coordinated, traceable automation across domains in the built environment.

7 Conclusion

This study introduced a plug-and-play semantic automation framework for reusable and cross-domain execution in AECO workflows. By addressing fragmented data models and non-reusable automation logic, the framework advances AECO digitalisation through the formalisation of *Domain Packs* as modular units of executable semantics. Each Domain Pack encapsulates ontologies, data mappings, reasoning rules, workflow behaviours, and provenance policies, enabling domain knowledge to be packaged as a structured and lifecycle-aware operational asset.

To operationalise these semantic assets, a domain-agnostic multi-agent semantic runtime was developed over a shared knowledge-graph substrate. Agents remain free of embedded domain logic; instead, system behaviour is fully determined by the semantic content of the loaded Domain Packs. This decoupling allows automation behaviour to be reconfigured and generalised across heterogeneous operational contexts through semantic configuration alone, without code-level modification or platform-specific integration.

Cross-domain execution is enabled through the Trigger–Issue–Action–Actor (TIAA) model as a minimal semantic execution contract. TIAA allows reasoning outcomes defined in one Domain Pack to be interpreted as actionable input for workflows defined in another, achieving coordinated cross-domain execution without custom integration logic.

Validation through representative cross-domain scenarios confirms that the same agents, container images, and runtime configuration operate unchanged when different Domain Packs are activated or combined, demonstrating that reusability emerges from runtime semantic composition rather than tightly coupled workflows.

Beyond scenario-based demonstrations, the framework establishes a foundation for *action-oriented interoperability*: a semantic execution layer in which domain knowledge directly drives coordinated, traceable, and auditable actions across organisational and lifecycle boundaries. In this respect, the framework complements existing Digital Twin research by extending semantic Construction Digital Twin (CDT) directions [11], ISO 23247 [12], and CPS-oriented models [13] with executable semantics that bridge semantic reasoning and operational action.

Future work will extend Domain Pack authoring practices, integrate learning-enabled reasoning capabilities, and evaluate the framework at larger spatial and organisational scales. Ultimately, the framework establishes a reusable, explainable, and scalable *semantic execution* foundation for next-generation intelligent automation in the built environment.

8 Acknowledgements

Supported by NSTC projects “Development of a Digital Twin Application Platform for Net-Zero Circular and Healthy Campus” (NSTC 112-2221-E-002-078-MY3) and “Phase I Forward-looking Technology Research Program for Taiwan Spatial Sustainability Planning – Joint Demonstration Field Project” (NSTC 114-2119-M-002-034).

References

[1] E. Nuyts, M. Bonduel, and R. Verstraeten. Comparative analysis of approaches for automated compliance checking of construction data. *Advanced Engineering Informatics*, 60:102443, 2024.

[2] C.-P. Huang and S.-H. Hsieh. Semantic reasoning and integration for automating predictive maintenance in smart facility management. *Advanced Engineering Informatics*, 71:104240, 2026.

[3] C. Su, Y. Han, X. Tang, Q. Jiang, T. Wang, and Q. He. Knowledge-based digital twin system: Using a knowlege-driven approach for manufacturing process modeling. *Computers in Industry*, 159-160: 104101, 2024.

[4] A. Polenghi, I. Roda, M. Macchi, A. Pozzetti, and H. Panetto. Knowledge reuse for ontology modelling in Maintenance and Industrial Asset Management. *Journal of Industrial Information Integration*, 27: 100298, 2022.

[5] X. Huang, C. Yang, Y. Zhang, S. Lou, L. Kong, and H. Zhou. Ontology guided multi-level knowledge graph construction and its applications in blast furnace ironmaking process. *Advanced Engineering Informatics*, 62:102927, 2024.

[6] C.-P. Huang. *Graph-Native Semantic Action Management for Operations and Maintenance Automation in Smart Built Environments*. PhD thesis, National Taiwan University, Taiwan, 2025.

[7] D. Shkundalov and T. Vilutienė. Bibliometric analysis of Building Information Modeling, Geographic Information Systems and Web environment integration. *Automation in Construction*, 128:103757, 2021.

[8] S. Wu, G. Wang, J. Lu, Y. Yan, Y. Gong, M. Dong, and D. Kiritsis. Digital thread in engineering: Concept, state of art, and enabling framework. *Advanced Engineering Informatics*, 65:103258, 2025.

[9] L. Xiang, Y. Tan, G. Shen, and X. Jin. Applications of multi-agent systems from the perspective of construction management: A literature review. *Engineering, Construction and Architectural Management*, 29(9):3288–3310, 2022.

[10] Y. Zheng, S. Törmä, and O. Seppänen. A shared ontology suite for digital construction workflow. *Automation in Construction*, 132:103930, 2021.

[11] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui. Towards a semantic Construction Digital Twin: Directions for future research. *Automation in Construction*, 114:103179, 2020.

[12] International Organization for Standardization. Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles (iso 23247-1:2021), 2021. URL `https://www.iso.org/standard/75066.html`. ISO 23247-1:2021.

[13] S. Roy and S. Das, editors. *Principles of Cyber-Physical Systems: An Interdisciplinary Approach*. Cambridge University Press, Cambridge, UK, 2020.