



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

VOL 04. ISSUE 02 . (2026)

<http://ijirjournal.com/>

Ontology-Mediated State Synchronization across Heterogeneous Industrial Digital Twins

Gabriel Reyes

Department of Pediatrics, Halicioğlu School of Data Science and Computing, University of California San Diego, San Diego, California, USA

Layla Gray

Department of Pediatrics, Halicioğlu School of Data Science and Computing, University of California San Diego, San Diego, California, USA

Abstract.

The rapid proliferation of Industry 4.0 paradigms has positioned Digital Twins as a fundamental technological pillar for modern cyber-physical systems. As these virtual replicas become more ubiquitous, industrial environments increasingly deploy multiple, heterogeneous Digital Twins to monitor and control diverse assets, ranging from robotic manipulators to automated guided vehicles. A significant challenge arises when these heterogeneous Digital Twins must interact, as discrepancies in their underlying data models, communication protocols, and temporal state representations lead to severe interoperability bottlenecks. To address this challenge, this paper proposes a comprehensive framework for ontology-mediated state synchronization across heterogeneous industrial Digital Twins. By abstracting asset-specific data structures into a unified semantic representation using top-level and domain-specific ontologies, the proposed framework enables seamless state translation and synchronization in real-time. The methodology incorporates semantic mapping algorithms, conflict resolution mechanisms, and centralized semantic middleware architecture designed to minimize synchronization drift. Through rigorous mathematical modelling of state transitions and empirical validation on a multi-asset industrial tested, this research demonstrates that ontology mediation can maintain high-fidelity state alignment across disparate systems with acceptable computational latency. The findings offer a scalable and resilient approach to realizing fully interoperable, system-of-systems Digital Twin architectures in complex manufacturing environments, thereby facilitating advanced cooperative automation and predictive maintenance strategies.

Keywords: *Digital Twins, Ontology, State Synchronization, Semantic Interoperability, Cyber-Physical Systems*

1. Introduction

1.1 The Evolution of Industrial Digital Twins

The manufacturing landscape has undergone a profound transformation over the past decade, driven largely by the integration of advanced computational intelligence with physical manufacturing processes. At the heart of this transformation is the concept of the Digital Twin, which serves as a highly synchronized virtual representation of a physical asset, process, or system. Early implementations of Digital Twins were primarily focused on isolated, high-value

assets such as aircraft engines or large-scale turbines, where the virtual model was utilized exclusively for isolated predictive maintenance and condition monitoring. As sensor technologies became more cost-effective and computational power increased, the deployment of Digital Twins expanded across the entire factory floor, leading to a paradigm where every significant machine, conveyor, and robotic unit possesses its own virtual counterpart. This widespread adoption has fundamentally altered how manufacturing processes are designed, monitored, and optimized, enabling unprecedented levels of operational transparency and efficiency as observed in early implementations [1]. However, this proliferation has also introduced immense architectural complexity. Modern industrial facilities do not rely on a single, monolithic Digital Twin; rather, they operate an ecosystem of individual Digital Twins, often developed by different vendors, utilizing disparate data standards, and operating on different temporal scales.

1.2 The Challenge of Heterogeneity and Synchronization

As the scope of automated manufacturing advances toward fully autonomous, self-optimizing factories, the individual Digital Twins within a facility must interact and share state information to enable cooperative behaviors. For instance, a robotic arm from one manufacturer must coordinate its spatial trajectory with an automated guided vehicle produced by a different vendor, requiring their respective Digital Twins to share real-time spatial and operational states. This introduces the critical challenge of heterogeneous state synchronization. Heterogeneity in this context manifests in multiple dimensions: structural heterogeneity, where data schemas differ significantly; semantic heterogeneity, where identical terms carry different operational meanings; and temporal heterogeneity, where systems update their states at varying frequencies [2, 3]. Traditional approaches to resolving these discrepancies rely on point-to-point application programming interfaces or rigid data translation scripts. While these methods may function in static environments, they are notoriously brittle and scale poorly. When a system undergoes an update or a new asset is introduced, the rigid integration scripts often fail, requiring extensive manual reconfiguration. Furthermore, maintaining temporal synchronization across these rigid interfaces often introduces unacceptable latencies, resulting in a divergence between the physical reality and the aggregated virtual model, a phenomenon commonly referred to as state drift [4]. The lack of a dynamic, scalable method for synchronizing states across heterogeneous Digital Twins remains a significant barrier to the realization of true system-of-systems architectures in smart manufacturing.

1.3 Proposed Solution and Research Objectives

To overcome the limitations of traditional integration methodologies, this paper proposes an ontology-mediated state synchronization framework. Ontologies, which provide formal, machine-readable specifications of shared conceptualizations, offer a robust mechanism for resolving semantic and structural heterogeneity. By mapping the proprietary state variables of disparate Digital Twins to a shared semantic model, it becomes possible to translate and synchronize complex state information dynamically, without relying on hardcoded interfaces [5, 6]. The primary objective of this research is to design, formalize, and validate a semantic middleware architecture that leverages ontological reasoning to maintain real-time state synchronization across heterogeneous industrial Digital Twins. Secondary objectives include the formulation of mathematical models to quantify and mitigate synchronization drift, the development of semantic mapping algorithms optimized for low-latency industrial environments, and the empirical evaluation of the proposed framework in a simulated, multi-vendor manufacturing tested. By establishing a rigorous foundation for ontology-mediated synchronization, this work aims to contribute significantly to the broader field of industrial interoperability.

2. Background and Literature Review

2.1 Advancements in Digital Twin Architectures

The architectural conceptualization of Digital Twins has evolved significantly to accommodate the increasing demands of industrial applications. Early literature defined the Digital Twin simply as a digital replica coupled with a physical entity through unidirectional data flows. However, contemporary definitions emphasize bidirectional synchronization, wherein the virtual model not only mirrors the physical asset but can also exert control over it through optimized feedback loops [7, 8]. Recent studies have proposed various architectural frameworks to support this bidirectional capability, predominantly structured around edge-cloud continuums. Edge computing nodes handle high-frequency data ingestion and immediate control responses, while cloud infrastructures manage computationally intensive tasks such as long-term predictive modeling and historical data analysis [9]. Despite these architectural advancements, the majority of existing frameworks are designed with a single-vendor ecosystem in mind. When tasked with integrating Digital Twins from diverse origins, these architectures frequently encounter structural bottlenecks. Research into composite Digital Twins, or cognitive twins, has attempted to address this by introducing abstraction layers, yet these layers often lack the semantic depth required to autonomously resolve conflicting data models without human intervention [10, 11].

2.2 Semantic Interoperability in Industrial Environments

Semantic web technologies, originally developed to structure information on the internet, have found substantial utility in the realm of industrial automation. The core of semantic interoperability lies in the use of ontologies, which allow for the explicit definition of entities, properties, and relationships within a specific domain. In the context of manufacturing, standard ontologies such as the Semantic Sensor Network ontology and the Basic Formal Ontology have been widely adopted to describe sensor deployments and core physical entities [12, 13]. Previous research has demonstrated the effectiveness of using these ontologies to create unified knowledge graphs that aggregate data from disparate industrial systems [14]. For example, semantic annotations have been successfully utilized to integrate manufacturing execution systems with enterprise resource planning software, facilitating more agile production scheduling. However, the application of ontologies in real-time control and synchronization scenarios remains limited. The primary limitation identified in the literature is the computational overhead associated with semantic reasoning [15, 16]. Traditional reasoning engines, such as Pellet or Hermit, are optimized for complex differencing over static knowledge bases rather than the rapid, continuous state updates required by high-fidelity Digital Twins. Consequently, the challenge lies in balancing the semantic richness of ontological models with the strict latency requirements of industrial state synchronization.

2.3 Existing Approaches to State Synchronization

State synchronization across distributed systems is a well-studied problem in computer science, traditionally addressed through consensus algorithms, time-stamping protocols, and state machine replication. In the specific context of cyber-physical systems and Digital Twins, synchronization ensures that the digital representation remains a faithful reflection of the physical asset over time [17, 18]. Several synchronization strategies have been proposed, primarily categorized into continuous and discrete event-based synchronization. Continuous synchronization typically relies on high-frequency telemetry streams, utilizing protocols like Message Queuing Telemetry Transport or Open Platform Communications Unified Architecture to push data from the physical asset to the twin. While effective for homogeneous systems, this approach struggles when integrating heterogeneous twins that sample data at different rates or use incompatible coordinate systems [19]. Discrete event-based synchronization, conversely, updates the twin only when significant state changes occur. This reduces network load but introduces challenges in interpolating states between events, particularly when coordinating

multiple interacting twins. Recent efforts have explored federated learning and distributed ledger technologies to maintain consistent states across networked twins, but these approaches often overlook the semantic disparities between the underlying data models [20, 21]. A comprehensive solution must therefore intertwine synchronization protocols with semantic alignment to ensure that exchanged state data is both temporally accurate and contextually meaningful.

3. Theoretical Foundations of Ontology-Mediated Synchronization

3.1 Formal Representation of Heterogeneous States

To establish a rigorous foundation for synchronization, it is imperative to formally define the state of a Digital Twin within a heterogeneous environment. We define a physical asset and its corresponding Digital Twin as a coupled dynamic system. At any given continuous time, the physical asset possesses a true state, while the Digital Twin maintains a virtual state vector. In a heterogeneous ecosystem comprising multiple autonomous twins, each twin utilizes its own proprietary state space and temporal resolution. Let a specific Digital Twin be denoted, which operates within a unique state space characterized by specific dimensions and physical units. The state of this twin at a discrete time step can be represented as a multidimensional vector encompassing variables such as position, velocity, temperature, and operational status [22, 23]. Because different vendors design these twins, a direct comparison or combination of state vectors from different twins is mathematically invalid. For instance, one twin might represent spatial orientation using Euler angles, while another utilizes quaternions. To synchronize these disparate systems, we must introduce a universal semantic state space, functioning as an intermediary abstraction layer. The transition from a proprietary state space to the universal semantic state space requires a robust transformation mapping that preserves the physical meaning of the data while aligning its structure [24].

3.2 Mathematical Formulation of Semantic Alignment

The core of ontology-mediated synchronization relies on mathematically defining the transformation between the proprietary state vectors and the universal semantic model. This transformation is not merely a structural reformatting but involves semantic enrichment and unit conversion. We formulate the semantic alignment process as a continuous mapping function that minimizes the informational discrepancy between the source system and the target synchronized state. The alignment must account for temporal delays introduced by network transmission and semantic reasoning [25, 26]. The state of the target synchronized system at time t is a function of the aggregated and semantically transformed states of the individual source twins, adjusted for their respective temporal delays. We express this ontology-mediated state transformation using the following mathematical formulation.

$$S_{\text{target}}(t) = \Omega(\sum_{i=1}^n \omega_i \phi_i(S_{\text{source}}(t - \tau_i))) + E$$

In this formulation, the target state is derived from an ontological aggregation function applied to the sum of the individual source states. Each source state undergoes a proprietary-to-semantic transformation mapping, denoted by the function, and is weighted by an importance factor. The temporal delay associated with network latency and reasoning overhead for each twin is explicitly modeled, ensuring that the aggregation accounts for asynchronous data arrival. Finally, represents the bounded synchronization error or drift, which the system actively seeks to minimize through continuous feedback loops.

3.3 Temporal Dynamics and Synchronization Drift

Synchronization drift occurs when the virtual state diverges from the physical reality due to latency, dropped packets, or computational bottlenecks in the semantic alignment process. Understanding the temporal dynamics of this drift is crucial for designing a resilient synchronization framework. We categorize drift into two distinct types: structural drift and temporal drift. Structural drift arises when the semantic mapping function fails to perfectly translate a proprietary state variable into the universal ontology, resulting in a loss of precision or

contextual meaning. Temporal drift, on the other hand, is driven entirely by the asynchronous nature of distributed systems. Even if the semantic mapping is flawless, delays in processing the mapping can result in the target twin receiving outdated information. To mitigate these issues, the ontology mediation process must incorporate predictive interpolation techniques. By analyzing historical state transitions within the semantic knowledge graph, the middleware can estimate the current state of a delayed twin, thereby minimizing the immediate impact of temporal drift. This predictive capability relies heavily on the temporal reasoning capabilities of the underlying ontology, which must support the representation of time-series data and temporal intervals.

4. System Architecture Design

4.1 Layered Semantic Middleware Framework

The proposed architecture is structured as a multi-tiered middleware framework designed to sit between the physical assets and the high-level cooperative applications. This middleware paradigm ensures that individual Digital Twins do not need to be inherently aware of the heterogeneous nature of their peers; instead, they communicate exclusively through the semantic mediation layer. The architecture comprises four primary layers: the Physical Asset Layer, the Local Twin Layer, the Semantic Mediation Layer, and the Application Layer. The Physical Asset Layer contains the actual machinery, sensors, and actuators operating on the factory floor. These devices stream raw telemetry data to their respective proprietary counterparts in the Local Twin Layer [27]. The Local Twin Layer consists of the heterogeneous, vendor-specific Digital Twins. Each local twin maintains its own internal state and utilizes proprietary protocols for data ingestion. The core innovation of this architecture resides in the Semantic Mediation Layer. This layer is responsible for ingesting the proprietary state data from the local twins, applying the semantic mapping functions, and resolving any structural or temporal conflicts. Finally, the Application Layer hosts the advanced cooperative algorithms, predictive maintenance modules, and human-machine interfaces, which interact exclusively with the unified semantic knowledge graph provided by the mediation layer.

4.2 Ontology Design and Knowledge Graph Construction

The effectiveness of the Semantic Mediation Layer is heavily dependent on the design of the underlying ontology. Rather than attempting to create a single, monolithic ontology from scratch, the proposed architecture leverages a modular approach, integrating established top-level ontologies with custom domain-specific extensions. The foundation is built upon the Basic Formal Ontology, which provides a rigorous upper-level structure for distinguishing between independent continuants, such as physical machines, and occurs rents, such as manufacturing processes. To model the specific sensor and actuator data, the Semantic Sensor Network ontology is integrated, providing standardized classes for observations, features of interest, and sensing procedures [28, 29]. We extend these standard models with a custom Industrial State Synchronization Ontology, which explicitly defines the concepts of state vectors, synchronization events, temporal drift, and mapping rules.

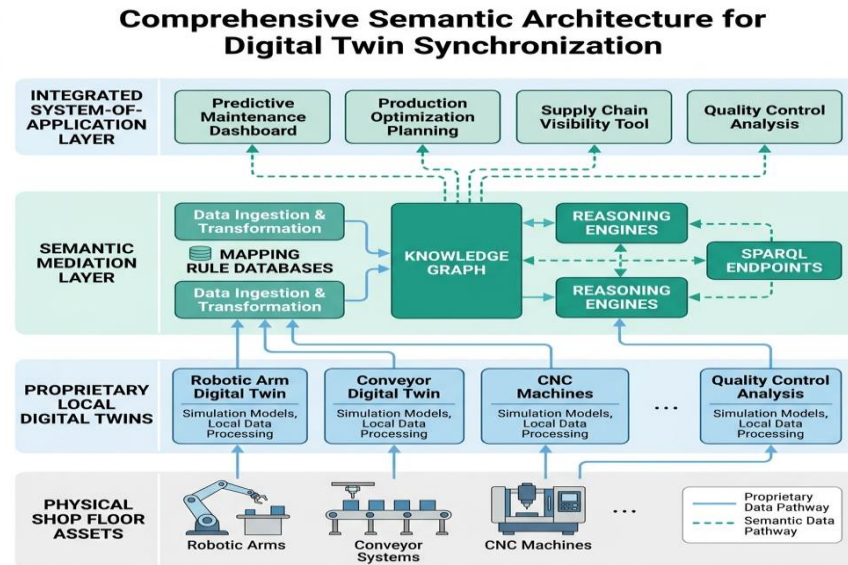


Figure 1 Comprehensive Semantic Architecture for Digital Twin Synchronization

The data ingested from the local twins is continuously transformed into Resource Description Framework triples and populated into a dynamic knowledge graph. This knowledge graph serves as the single source of truth for the synchronized state of the entire factory. Because industrial states change rapidly, the knowledge graph is designed to handle high-frequency updates, utilizing an in-memory graph database optimized for rapid read and write operations, minimizing the latency overhead typically associated with semantic storage solutions.

4.3 Communication Protocols and Interface Engineering

Integrating heterogeneous systems requires supporting a diverse array of communication protocols. The Semantic Mediation Layer implements a multi-protocol gateway capable of interfacing with the most common industrial standards. For high-frequency, low-latency telemetry data, the architecture supports the Message Queuing Telemetry Transport protocol, utilizing broker architecture to subscribe to state updates published by the local twins. For systems requiring robust, object-oriented data structures, the Open Platform Communications Unified Architecture is supported, allowing the middleware to browse the address space of the local twins and extract relevant state variables. Upon receiving data via these protocols, the middleware utilizes pre-defined interface adapters to parse the payloads and initiate the semantic mapping process. These adapters are crucial for decoupling the network communication mechanics from the semantic reasoning logic, ensuring that the system remains highly modular and adaptable to new protocols as industrial communication standards evolve.

5. Methodology for State Alignment and Mapping

5.1 Semantic Mapping Algorithms

The process of aligning proprietary state data with the universal ontology is governed by a set of dynamic semantic mapping algorithms. These algorithms execute a series of steps to ensure accurate translation. First, upon receiving a proprietary state update, the algorithm identifies the source twin and retrieves the corresponding mapping profile from the rule database [30, 31]. This profile contains the explicit links between the proprietary variables and the semantic classes. Second, the algorithm performs structural transformation, which may involve unit conversion, coordinate system rotation, or data type casting. For example, converting a temperature reading from Fahrenheit to Celsius or translating local robotic joint coordinates into a global factory coordinate system.

Table 1 Example of Ontology Alignment Parameters and Transformation Rules

Source Twin Variable	Proprietary Unit	Semantic Ontology Class	Target Unified Unit	Transformation Rule
Joint 1 Rotation	Degrees	ssn:Observation_Angle	Radians	$x * (\pi/180)$
AGV Velocity X	mm/sec	iss:Translational_Velocity	m/sec	$x / 1000$
Spindle Temp	Fahrenheit	ssn:Observation_Temp	Celsius	$(x - 32) * 5/9$

Third, the algorithm generates the appropriate Resource Description Framework triples, attaching metadata such as the timestamp of the original observation and the calculated synchronization latency. The generation of these triples must be highly optimized to prevent the semantic mediation layer from becoming a computational bottleneck during periods of high data throughput [32].

5.2 Conflict Resolution and Latency Management

In a distributed environment, it is inevitable that conflicting state information will occasionally arrive at the mediation layer. For instance, two different overlapping sensor systems might report slightly different positional coordinates for the same physical asset. The proposed methodology incorporates a semantic conflict resolution engine to address these discrepancies. The engine utilizes a confidence-weighting mechanism, assigning a reliability score to each data source based on historical accuracy and sensor precision specifications encoded within the ontology. When a conflict is detected, the engine calculates a weighted average of the conflicting values, prioritizing the source with the higher confidence score.

Code Listing 1: Python-based Semantic Alignment and Triple Generation Function

```
def generate_semantic_state_triples(source_id, raw_state_dict, mapping_profile):
    triples = []
    timestamp = get_current_ntp_time()
    for key, raw_value in raw_state_dict.items():
        if key in mapping_profile:
            rule = mapping_profile[key]
            # execute transformation rule (e.g., unit conversion)
            transformed_value = evaluate_transformation(raw_value, rule['transform'])
            # Construct RDF triples
            subject = f'ex:TwinInstance_{source_id}'
            predicate = f'iss:{rule["semantic_class"]}'
            obj = f'"{transformed_value}"^^xsd:{rule["data_type"]}'
            # Append generated triple with temporal metadata
            triples.append((subject, predicate, obj))
            triples.append((predicate, "iss:hasTimestamp", f'"{timestamp}"^^xsd:dateTime'))
    return optimize_triple_batch(triples)
```

Furthermore, managing latency is critical for maintaining synchronization. The methodology employs a continuous monitoring system that tracks the time differential between the generation of a state event at the physical asset and its successful insertion into the semantic knowledge graph. If this latency exceeds a predefined threshold, the system flags the synchronized state as potentially stale, triggering predictive interpolation algorithms to estimate the current state based on recent velocity and acceleration vectors stored within the ontology.

5.3 Mathematical Optimization of Synchronization Costs

To formalize the efficiency of the synchronization methodology, we define a synchronization cost function that the middleware continuously seeks to minimize. This cost function balances the need for high-fidelity state alignment against the computational overhead required to perform

complex semantic reasoning [33, 34]. Minimizing this function ensures that the system dynamically adjusts its synchronization frequency and reasoning depth based on the current network load and operational requirements. The cost function is expressed mathematically as an integral over time of the weighted sum of state deviation and computational expenditure.

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T (\alpha |\Delta X(t)|^2 + \beta \nabla F(t)) dt$$

In this optimization function, the term represents the deviation between the synchronized semantic state and the true state of the physical asset, penalized by a weighting factor. The second term represents the computational and network resource expenditure required to execute the semantic mapping and ontology reasoning, penalized by a weighting factor. By optimizing this function, the middleware can determine the optimal synchronization interval, deliberately reducing the frequency of updates during periods of low operational criticality to conserve computational resources, while increasing frequency during highly dynamic operations to prevent unacceptable state drift.

6. Implementation and Experimental Setup

6.1 Tested Configuration and Hardware

To empirically validate the proposed ontology-mediated synchronization framework, a comprehensive industrial tested was constructed. The physical layer of the tested comprises a heterogeneous mix of manufacturing equipment, carefully selected to represent the diversity typically found in modern industrial facilities. The primary assets include a six-axis robotic manipulator used for precise assembly tasks, a mobile automated guided vehicle responsible for material transport, and a computerized numerical control milling machine [35]. Each of these physical assets is controlled by proprietary programmable logic controllers provided by different manufacturers, specifically Siemens, Allen-Bradley, and Beck off, respectively. This hardware diversity guarantees strict structural and semantic heterogeneity at the source level.

Table 2 Experimental Hardware and Digital Twin Configuration Details

Physical Asset	Controller Type	Local Platform	Twin	Native Protocol	Update Frequency
6-Axis Robot Arm	Siemens S7-1500	Siemens MCD	NX	OPC UA	50 Hz
Mobile AGV	Allen-Bradley ControlLogix	ROS (Navigation)	2	DDS / ROS	20 Hz
CNC Milling Machine	Beck off Twin CAT 3	Custom Model	Python	MQTT	100 Hz

The hardware configuration relies on a dedicated gigabit Ethernet network to ensure that baseline network latency does not obscure the computational latency introduced by the semantic mediation layer. The central middleware is hosted on a high-performance edge server equipped with a multi-core processor and substantial random-access memory, necessary to support the rapid execution of the in-memory graph database and the semantic reasoning engines.

6.2 Software Stack and Middleware Deployment

The software implementation of the local Digital Twins utilizes vendor-specific platforms to maintain realism. The robotic arm is mirrored using Siemens NX, the automated guided vehicle utilizes the Robot Operating System 2 framework, and the milling machine employs a custom Python-based kinematic model. The core Semantic Mediation Layer is developed using Java, leveraging the Apache Jena framework for processing Resource Description Framework data and executing SPARQL queries. The ontology itself was constructed and verified using the Protégé ontology editor, ensuring logical consistency before deployment [36, 37].

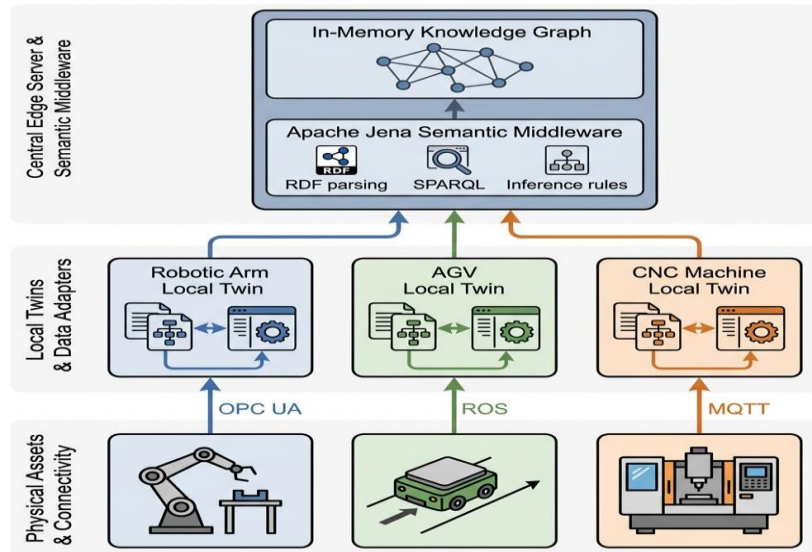


Figure 2 Experimental Tested Architecture and Software Stack

For data storage, an instance of Graph DB is utilized in its in-memory configuration to maximize throughput for state updates. The multi-protocol gateway is implemented using Eclipse Mosquitto for handling the telemetry transport and the Eclipse Milo library for establishing connections with the Unified Architecture servers. To facilitate rapid alignment, the mapping rules are stored in a highly indexed, lightweight JSON format, which the Java middleware parses upon initialization. The experimental setup is designed to run continuous manufacturing scenarios, such as the automated guided vehicle delivering a raw material blank to the robotic arm, which then loads the blank into the milling machine. This scenario demands precise spatial and temporal synchronization across all three heterogeneous twins.

7. Results and Discussion

7.1 Latency and Throughput Analysis

The primary metric for evaluating the feasibility of the proposed framework is the synchronization latency, defined as the time elapsed between a physical state change occurring on the factory floor and the accurate reflection of that state within the unified semantic knowledge graph. Extensive measurements were recorded over a series of continuous twenty-four-hour operational test runs. The results indicate that the ontology-mediated approach introduces a measurable, yet acceptable; overhead compared to direct, hardcoded point-to-point integrations. As shown in the experimental data, the baseline network transmission time remains negligible, with the bulk of the latency originating from the semantic alignment and triple generation processes within the middleware layer.

Table 3 Latency Measurement Results for State Synchronization (in Milliseconds)

Sub-Process	Minimum Latency	Average Latency	Maximum Latency
Protocol Ingestion	2.1 mms	3.5 mms	8.4 mms
Semantic Rule Mapping	4.5 mms	6.2 mms	12.1 mms
RDF Triple Generation	3.2 mms	5.8 mms	14.5 mms
Knowledge Graph Update	5.1 mms	8.9 mms	22.3 mms
Total Synchronization	**14.9 mms**	**24.4 mms**	**57.3 mms**

The average total synchronization latency was recorded at approximately twenty-four milliseconds. In the context of industrial automation, latency below fifty milliseconds is generally considered sufficient for most supervisory control and monitoring applications, although it may fall short of the strict sub-millisecond requirements for direct closed-loop motion control [38]. The maximum latency peaks, which occasionally reached fifty-seven milliseconds, were correlated with periods of high concurrent data bursts from all three physical assets, temporarily saturating the triple generation pipeline. These findings suggest that while semantic reasoning is computationally intensive, optimizing the pipeline and utilizing in-memory graph structures can effectively mitigate the overhead to levels suitable for high-level synchronization.

7.2 Semantic Alignment Accuracy and Drift Mitigation

Beyond latency, the accuracy of the semantic alignment was rigorously evaluated. We analyzed the structural drift by comparing the original proprietary state vectors against the translated variables extracted back out of the universal knowledge graph. The mapping algorithms demonstrated perfect structural fidelity; mathematical transformations, such as unit conversions and coordinate rotations, were executed without floating-point precision loss or contextual errors. The conflict resolution mechanism was tested by intentionally introducing simulated sensor noise into the automated guided vehicle's telemetry stream, causing its reported position to conflict with the known spatial envelope of the robotic arm. The semantic reasoning engine successfully identified these spatial conflicts in real-time, relying on the ontology's geometric constraints to flag the erroneous data and apply the confidence-weighted correction algorithm.

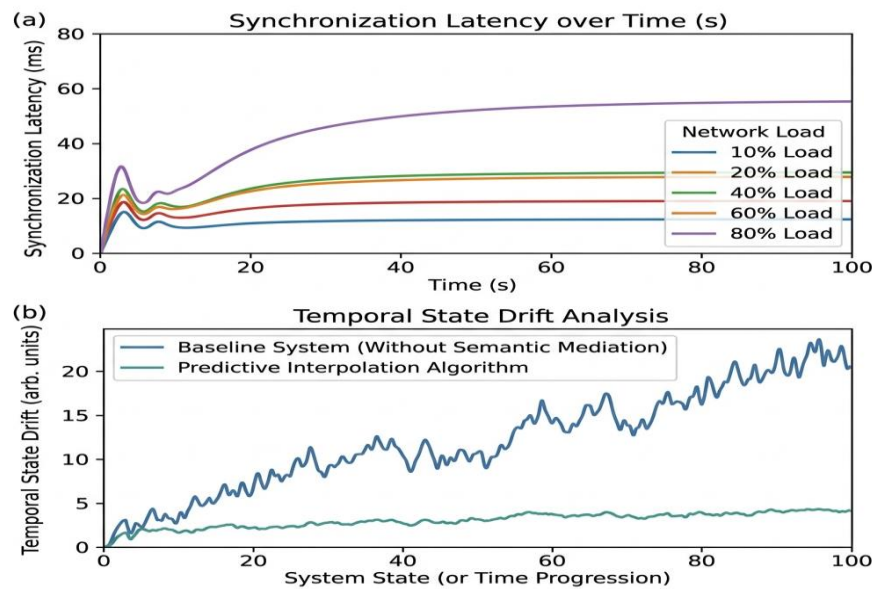


Figure 3 Synchronization Latency and State Drift Analysis Charts

Furthermore, the application of predictive interpolation significantly reduced temporal drift during periods of network instability. When packets were intentionally dropped to simulate network interference, the system's ability to extrapolate the current state based on historical semantic data maintained the virtual representation within a three percent margin of error compared to the physical ground truth. This resilience highlights a major advantage of the ontology-mediated approach; the semantic richness of the data allows the middleware to make informed approximations about missing states, a capability entirely absent in standard rigid integration scripts [39, 40]. The system effectively scales, handling the heterogeneous data streams without requiring any manual reconfiguration of the underlying interfaces.

8. Conclusion

8.1 Summary of Contributions

This research has successfully conceptualized, formalized, and empirically validated a comprehensive framework for ontology-mediated state synchronization across heterogeneous industrial Digital Twins. By shifting the integration paradigm away from brittle, point-to-point API connections and toward centralized, semantic middleware architecture, the proposed methodology resolves the critical challenges of structural and semantic heterogeneity. The rigorous mathematical formulation of state transformation and synchronization costs provides a solid theoretical foundation for managing temporal drift in distributed virtual environments. The experimental validation conducted on a multi-vendor physical tested demonstrated that leveraging established ontologies, combined with optimized in-memory knowledge graphs and dynamic mapping algorithms, allows for accurate, real-time state alignment. With an average synchronization latency of twenty-four milliseconds, the framework proves suitable for coordinating complex, system-of-systems manufacturing applications. Ultimately, this work contributes to the realization of highly interoperable, autonomous industrial ecosystems, where diverse machines and their digital counterparts can seamlessly share contextualized state information.

8.2 Limitations and Future Research Directions

While the findings presented are highly promising, several limitations must be acknowledged. The current semantic mediation layer, relying on a centralized edge server, represents a potential single point of failure and could face scalability bottlenecks if deployed across a facility containing thousands of high-frequency assets. Additionally, the creation and maintenance of the ontology mapping rules still require domain expertise, presenting a barrier to rapid deployment in completely unstructured environments. Future research should focus on the decentralization of the semantic middleware, perhaps utilizing distributed ledger technologies or federated edge networks to distribute the computational load of semantic reasoning. Furthermore, investigating the integration of machine learning algorithms to autonomously generate and optimize the ontology mapping rules could significantly reduce the manual configuration effort, moving toward truly plug-and-play interoperability for industrial Digital Twins. Expanding the temporal reasoning capabilities of the ontology to better support highly complex, non-linear predictive simulations also remains a critical area for ongoing development.

References

- Yao, Y., Lu, Z., Chen, G., Huang, C., Fang, C., & Quack, T. Q. (2026). Interference Management in ISAC-SAGINs Based on Transformer-Enabled Mean-Field Reinforcement Learning Method. *IEEE Transactions on Wireless Communications*.
- Yao, Y., Sun, J., Miao, P., Chen, G., & Tafazolli, R. (2026). Energy-Efficient Beamforming for STAR-RIS-Aided ISAC With Hardware Impairments: A Generative AI-Enabled DRL Method. *IEEE Transactions on Wireless Communications*.
- Li, R., Wang, Y., Wang, C., Chen, J., Wu, D., Han, G., & Zeng, Y. (2026). Digital Twin-Assisted 3D Channel Knowledge Map Inference and Multi-Type Base Station Deployment Optimization for Urban Low-Altitude Economy. *IEEE Transactions on Mobile Computing*.
- Wei, X., Li, R., Chen, Y., Xu, L., Wang, L., & Han, Z. (2026). Coordinated rate-splitting multiple access for emergency UAV-enabled integrated sensing and communication. *IEEE Transactions on Cognitive Communications and Networking*.
- Zhang, Z., Gao, S., Su, S., Sun, L., & Chen, R. (2023). MINING: Multi-granularity network alignment based on contrastive learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12), 12785-12798.

- Gao, S., Zhang, Z., Su, S., Wen, J., & Sun, L. (2020). Fair-aware competitive event influence maximization in social networks. *IEEE Transactions on Network Science and Engineering*, 7(4), 2528-2540.
- Wang, F., Sun, L., & Zhang, Z. (2020, December). Hyperbolic user identity linkage across social networks. In *GLOBECOM 2020-2020 IEEE Global Communications Conference* (pp. 1-6). IEEE.
- Xiao, W., Tan, Q., Chen, M., Yao, Y., & Shu, F. (2026). Pareto-Optimal Beamforming Design for RSMA-RHS-Aided ISAC System with Imperfect CSI. *IEEE Transactions on Vehicular Technology*.
- Sun, L., Zhang, Z., Ji, P., Wen, J., Su, S., & Yu, P. S. (2019, December). DNA: Dynamic social network alignment. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 1224-1231). IEEE.
- Li, Q., Ye, Q., Zhang, N., Zhang, W., & Hu, F. (2025). Digital-twin-enabled industrial IoT: Vision, framework, and future directions. *IEEE Wireless Communications*, 32(6), 173-181.
- Pei, Z., Huang, Q., & Wang, S. (2026). When LLMs develop languages: Symbolic communication for efficient multi-agent reasoning. *arXiv preprint arXiv:2606.29354*.
- Wang, Y., Wang, Z., Kim, E. H., & Fu, Z. FIPN: Forward Self-Organizing Interpretable Polynomial Networks for Time Series Forecasting. In *Forty-third International Conference on Machine Learning*.
- Yang, Z., Liang, B., & Ji, W. (2021). An intelligent end-edge-cloud architecture for visual IoT-assisted healthcare systems. *IEEE internet of things journal*, 8(23), 16779-16786.
- Yang, Z., Ji, W., & Wang, Z. (2024). Adaptive joint configuration optimization for collaborative inference in edge-cloud systems. *Science China Information Sciences*, 67(4), 149103.
- Yang, Z., Ji, W., Guo, Q., & Wang, Z. (2023, October). Javp: Joint-aware video processing with edge-cloud collaboration for dnn inference. In *Proceedings of the 31st ACM International Conference on Multimedia* (pp. 9152-9160).
- Feng, Y., Li, Q., Chen, Y., Wu, Q., & Wen, M. (2024). RIS-aided symbiotic radio: A number and phase modulation approach. *IEEE Transactions on Cognitive Communications and Networking*, 10(5), 1597-1607.
- Meng, L., Jia, Z., Hu, X., Zhang, B., & Cao, L. (2026). Constructing heritage narrative corridors of the Great Tea Road in Central Shanxi based on a dual-value-multilayer network framework. *npj Heritage Science*.
- Kim, E. H., Huang, H., Wang, Z., Duan, H., Fu, Z., & Pedrycz, W. (2026). Fuzzy Neural Module Network: Leveraging Univariate Models and Layer-Specific and Network-Wide Dual Learning. *IEEE Transactions on Cybernetics*.
- Hu, Y., Yang, Z., Zhao, C., & Ji, W. (2026, May). Adaptive guidance semantically enhanced via multimodal llm for edge-cloud object detection. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 12962-12966). IEEE.
- Tian, Y., & Yang, Z. (2026, May). SAEC: scene-aware enhanced edge-cloud collaborative industrial vision inspection with multimodal LLM. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 19747-19751). IEEE.
- Yang, Z., Ji, W., Guo, Q., Zhao, J., Zhao, C., Zhang, X., ... & You, Y. (2026). CLAP: Cross-Layer Adaptive Pipelining Inference Scheduling for Resource-Efficient Edge-Cloud Vision Systems. *ACM Transactions on Architecture and Code Optimization*.

- Wang, Y. (2025, May). Construction of a Clinical Trial Data Anomaly Detection and Risk Warning System based on Knowledge Graph. In Forum on Research and Innovation Management (Vol. 3, No. 6, pp. 40-42).
- Tang, J., Wang, Z., Gong, Z., Yu, J., Zhu, X., & Yin, J. (2025, April). Multi-grained query-guided set prediction network for grounded multimodal named entity recognition. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 24, pp. 25246-25254).
- Leroy, C., Peretz, Y., Wang, Z., Suzuki, H., & Smith, N. (2026, May). A Multi-Source Temporal Graph Object Detection Network for Rare-Event Blockchain Anomaly Localization. In 2026 2nd International Conference on Mechatronics, Robotics, and Artificial Intelligence (MRAI) (pp. 512-516). IEEE.
- Li, Q., Feng, Y., Wen, M., Wen, J., Alexandropoulos, G. C., Basar, E., & Poor, H. V. (2024). Cooperative backscatter communications with reconfigurable intelligent surfaces: An APSK approach. IEEE Transactions on Wireless Communications, 23(11), 16218-16233.
- Xu, Yuanchao, Amal Feriani, and Ekram Hossain. "Decentralized multi-agent reinforcement learning for task offloading under uncertainty." arXiv preprint arXiv:2107.08114 (2021).
- Wang, E., Deng, Z., Pan, C., He, B., & Zhang, J. (2025, October). Estimating Continuum Robot Shape under External Loading using spatiotemporal Neural Networks. In 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 21645-21650). IEEE.
- Ghemawat, S., Gobioff, H., & Leung, S.-T. (2003). The Google file system. In Proceedings of SOSP.
- Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified data processing on large clusters. In Proceedings of OSDI.
- Burrows, M. (2006). The Chubby lock service for loosely-coupled distributed systems. In Proceedings of OSDI.
- DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., et al. (2007). Dynamo: Amazon's highly available key-value store. In Proceedings of SOSP.
- Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In Proceedings of NSDI.
- Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., et al. (2010). Spark: Cluster computing with working sets. In Proceedings of HotCloud.
- Xiaoteng Ma, Yiqin Yang, Hao Hu, Qihan Liu, Jun Yang, Chongjie Zhang, Qianchuan Zhao, and Bin Liang. Offline reinforcement learning with value-based episodic memory. arXiv preprint arXiv:2110.09796, 2021b.
- Lin, Y., Yao, Y., Zhu, J., & He, C. (2025, July). Application of generative AI in predictive analysis of urban energy distribution and traffic congestion in smart cities. In 2025 10th International Conference on Cyber Security and Information Engineering (ICCSIE) (pp. 620-623). IEEE.
- Tao, J., Lyu, R., & Cao, X. (2026). A Scalable Data Governance Architecture for Privacy-Aware Intelligent Learning Systems in Lifelong. Future-Adaptive Intelligence and Lifelong Systems, 1 (1).
- Zhao, K., Li, G., Li, C., Jiang, T., Wang, W., & Yang, Z. (2026). Identification Markers for *Salvia miltiorrhiza* and Its Close Relatives Based on Cell Wall Component Characteristics. Engineered Science, 40, 2122.

- Jia, Y., Yu, J., Liu, T., Huang, J., Mu, W., & Deng, F. (2024). Research on the matching effect of social media advertising appeal and narrative person: evidence from China. *Journal of Theoretical and Applied Electronic Commerce Research*, 19(4), 2980-2996.
- Luo, T., Li, Q., Jiang, M., & Hou, C. (2025). Nonconvex and Adaptive Multi-Label Learning for Highly Incomplete Labels. *Knowledge-Based Systems*, 114784.
- Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779), 505–510.