Object-Centric Query Assembly for Heterogeneous System-of-Systems Integration in Digital Transformation

Valeria Shvedenko[1], Zumrat Gaibnazarova[2], Shirinkhon Khabibullaeva[3] and Gulzira Juraeva[4]
[1]*T-INNOVATIK LLC, Novosmolenskaya Embankment 1, Lit. E, Office 33-N, 199397 Saint Petersburg, Russia*
[2]*University of Science and Technologies, Diydor Str. 71, 100208 Tashkent, Uzbekistan*
[3]*Tashkent State Transport University, Temiryolchilar Str. 1, 100167 Tashkent, Uzbekistan*
[4]*Kimyo International University in Tashkent, Shota Rustaveli Str. 156, 100121 Tashkent, Uzbekistan*
vv_shvddenko@mail.ru, z.gaibnazarova@usat.uz, shirinxonchik@gmail.com, g.jorayeva@kiut.uz

Keywords: System of Systems, Topology of Information Space, Invariant Structures, Relationship Objects, Object-Centric Querying.

Abstract: Digital transformation of heterogeneous enterprises and infrastructures is costly and error-prone without a unified representation of data and processes across interacting subsystems. This paper proposes a systemic approach grounded in the theory of polystructural systems and formulates a System-of-Systems (SoS) topology of information space as a central organizing mechanism for integrating heterogeneous information arrays. To ensure universality and reduce coupling to domain-specific schemas, we introduce a set of invariant information structures and rules for organizing collinear and named connections, enabling consistent representation of goals, objects, processes, and indicators. The key novelty is an object-centric query mechanism that replaces static database key/index logic with first-class information objects: "Relationship" and "Relationship between two Relationships". These objects act as dynamic data sources for constructing information requests and support rapid recomposition of queries when new tasks emerge. This property is particularly valuable under unforeseen conditions (e.g., natural disasters or technogenic incidents) where operational processes and data dependencies change abruptly. The paper details the construction workflow for the proposed topology, including goal-tree integration via a polymetric system, and illustrates the approach through representative diagrams that can be reused as design patterns for digital transformation tools in heterogeneous SoS environments.

1 INTRODUCTION

Digital transformation of heterogeneous socio-technical environments requires coordinated management of data, processes, and goals across multiple interacting systems. In practice, these environments often form a System of Systems (SoS), where constituent systems remain operationally and structurally independent while needing interoperability and emergent coordination [1]-[4]. A key difficulty is that information about object states, indicators, and interdependencies is produced with different semantics, time scales, and levels of reliability, which complicates consistent decision-making and forecasting under evolving operational requirements.

Classical integration approaches typically rely on fixed schemas, static identifiers, and database key/index logic, which increases coupling between subsystems and makes rapid adaptation to new tasks expensive. Architecture description standards highlight the importance of consistent viewpoints and traceable representations when integrating complex systems [5]. At the same time, digital transformation research emphasizes that scalable integration mechanisms must be flexible enough to incorporate new processes, assets, and data streams without continuous restructuring of the underlying information infrastructure [6].

To address these challenges, data integration research has long argued for abstraction layers that decouple query and analytics requirements from source-specific schemas [7]. Recent developments in graph databases and knowledge-graph engineering further demonstrate that relationship-centric representations can improve interoperability and enable richer cross-domain queries compared to purely relational, index-driven approaches [8]-[12]. These requirements are also characteristic of large socio-economic and transport/logistics domains where heterogeneous systems and their digital projections must be coordinated within a shared information environment [13]-[15]. Computational modeling of train-assembly operations further illustrates how heterogeneous mechanical, operational, and performance indicators can be transformed into structured analytical representations for system-level analysis [16]. Similarly, optimization studies of nonlinear traction power-supply systems demonstrate the need to coordinate mathematical models, infrastructure parameters, and system-level decision variables within integrated analytical environments [17].

Contributions and novelty. This study makes three specific contributions to the digital transformation of heterogeneous System-of-Systems environments. First, it formalizes a topology of the SoS information space that provides a shared spatio-temporal representation layer for heterogeneous constituent systems without requiring full schema unification. Second, it introduces a compact set of invariant information structures for goals, objects, processes, indicators, and aggregates, enabling stable cross-system representation and traceable aggregation. Third, and most importantly, it proposes an object-centric query assembly mechanism in which "Relationship" and "Relationship between two Relationships" are treated as first-class information objects and used as dynamic query sources. Unlike traditional schema/key-based integration, where query adaptation typically requires reconfiguration of joins, indexes, or source-side mappings, the proposed approach supports query recomposition by instantiating and combining relationship objects under new task constraints. This property is especially valuable in heterogeneous SoS environments characterized by rapidly changing goals, temporary process links, crisis-response conditions, and evolving interoperability requirements.

The remainder of the paper is organized as follows. Section 2 presents the topology of the SoS information space, invariant structures, and the relationship-object model. Section 3 demonstrates pilot application and adaptive query recomposition. Section 4 discusses applicability and limitations. Section 5 concludes the paper.

2 PROPOSED METHODOLOGY

This section formalizes the proposed systemic representation for heterogeneous SoS environments. The approach consists of (i) defining a topology of the SoS information space, (ii) specifying a compact set of invariant structures for goals/objects/processes/indicators, and (iii) introducing relationship objects as first-class entities for adaptive, object-centric querying.

2.1 Problem Setting and Information Space Topology

We consider an SoS composed of heterogeneous constituent systems that maintain independent internal schemas and operational processes. Let the SoS information space be a unified digital environment that integrates heterogeneous information arrays in a shared spatio-temporal coordinate framework. The basic elements of the topology are information objects, their elementary properties, and relationships between them. The topology acts as an integration layer that supports consistent structuring, transferring, and combining data across systems while preserving heterogeneity at the source level (see Fig. 1 and 2).

To support interoperability, the topology uses two types of connectivity: named connections, which explicitly label semantically meaningful links (e.g., "goal-indicator", "object-process stage"), and collinear connections, which enable reusable structural fragments to appear as stable aggregates within different information objects. These mechanisms reduce coupling to local schemas and enable consistent mapping of heterogeneous elements into a shared information space.

2.2 Invariant Structures for Goals, Objects, Processes, and Indicators

To ensure universality, we define a compact set of invariant structures that remain stable across domains and can be reused to represent heterogeneous systems. The core invariant structures include: goal trees, objects, object properties, processes (with stages), indicators, and aggregates that link objects and processes. The conceptual mapping between goal trees, objects, properties, and indicators is illustrated in Figure 3.

Invariant indicator description. Each indicator is represented by an invariant description that consolidates identification and provenance metadata, unit/type attributes, normative and actual value projections with deviations, temporal validity, and hierarchical role/status within the goal-process model. This invariance enables polymorphic grouping and unified processing methods for management tasks at both subsystem and SoS levels.

2.3 Metamodel of the Topology

The topology metamodel provides a unified projection of information objects and processes across heterogeneous systems. It comprises: (i) core entities (goals, objects, processes, indicators), (ii) attribute/property sets, (iii) decomposition structures (stages and sub-processes), (iv) aggregates linking objects and processes, and (v) relationship objects enabling cross-level connectivity (object-object, goal-function, indicator-indicator, and relationship-relationship). A representative projection of object-process interrelationships is shown in Figure 4.

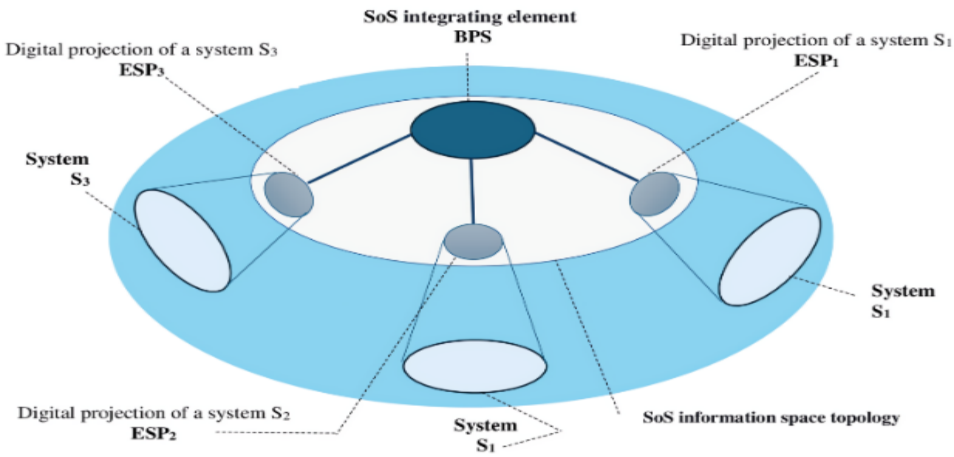

Figure 1: Diagram of multi-structural integration of SoS information resources.

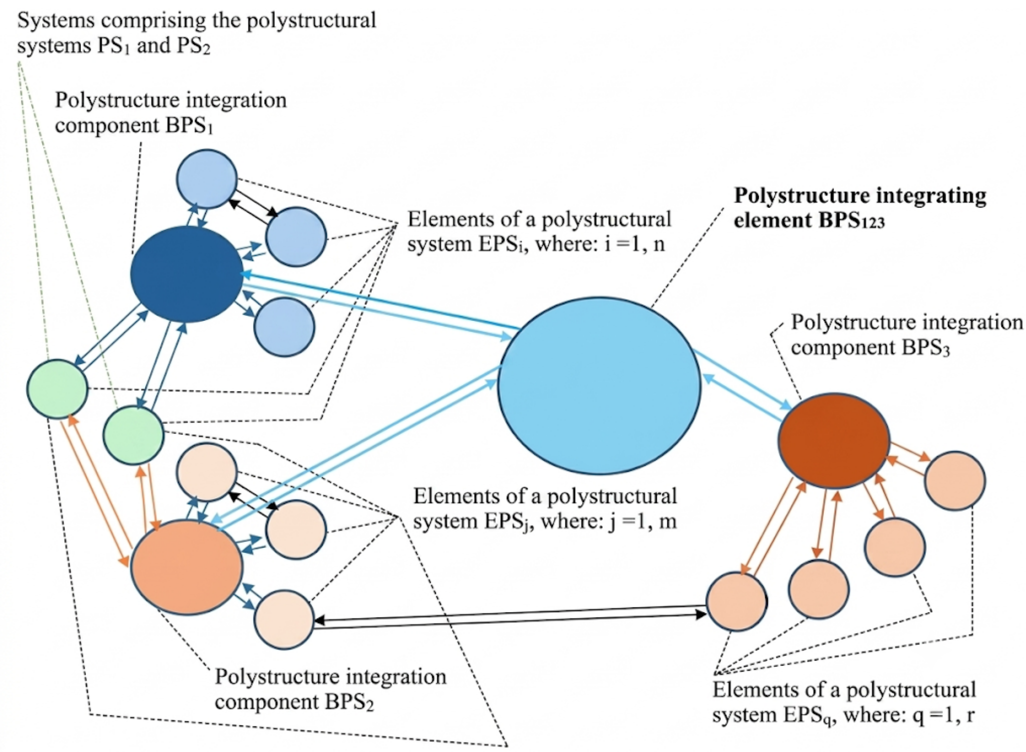

Figure 2: Example of the formed topology of the SoS information space.

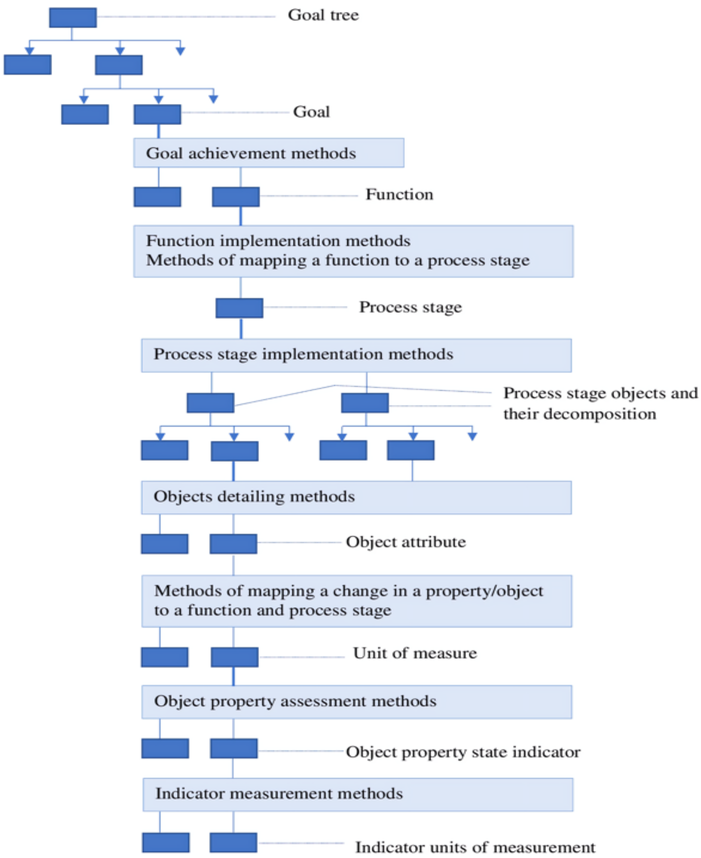

Figure 3: Relationships between the structures of the goal tree, object tree, object properties, and system indicators.

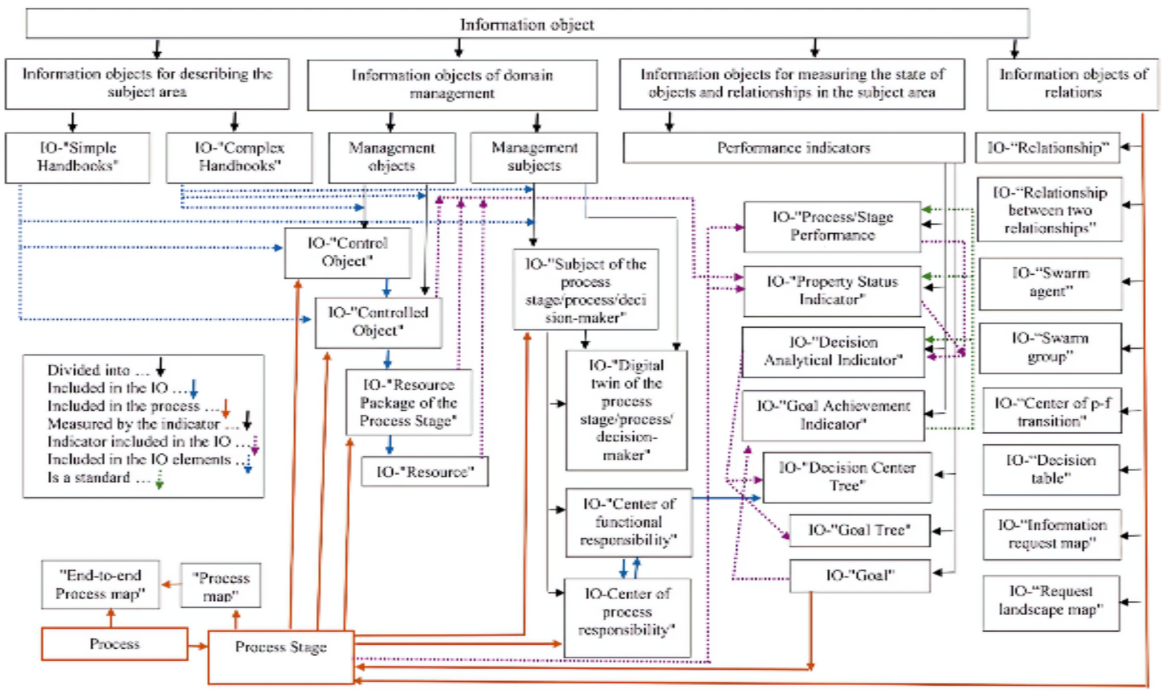

Figure 4: Projection of the relationship between information objects and processes.

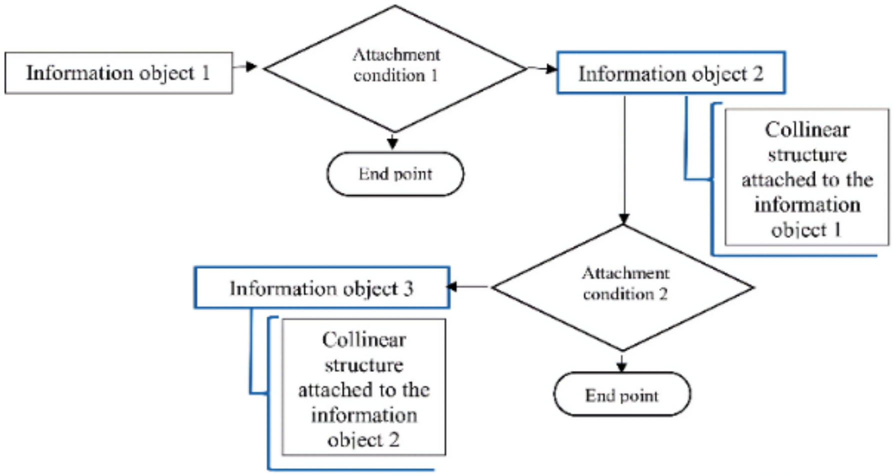

Figure 5: Example of a connected collinear structure.

Table 1: Comparison of query recomposition approaches.

Criterion	Static schema/key-based approach	Proposed relationship-object approach
Schema refactoring required	Yes	No
Support for new task constraints	Limited	High
Cross-system traceability	Partial	Explicit
Query recomposition without source change	No	Yes
Adaptation under unforeseen conditions	Low	High
Representative task change: new process stage linked to indicator and goal	Requires redesign of joins and source-side mappings	Requires instantiation of Relationship and Relationship between two Relationships objects
Provenance of assembled query	Reconstructed indirectly	Explicit relationship chain returned with result

Table 2: Qualitative configuration and validation characteristics of the pilot scenario.

Category	Characteristic	Description
Pilot scenario	System context	Heterogeneous System-of-Systems environment
Pilot scenario	Source organization	Operational data streams and indicator sets distributed across independent constituent systems
Integration basis	Information representation	Shared topology of the SoS information space
Integration basis	Main representational entities	Goals, objects, processes, indicators, aggregates, and relationship objects
Validation scope	Functional objective	Integrated view formation for monitoring, aggregation, and adaptive query recomposition
Validation scope	Monitoring capability	Detection of deviations from normative indicator values
Validation scope	Aggregation capability	Aggregation across time, goal achievement, management objects, and resources
Validation scope	Adaptation capability	Recomposition of information requests when tasks change
Integration property	Source-schema modification required	No
Integration property	Cross-system traceability	Explicit
Validation level	Type of evidence	Pilot-level functional validation
Validation level	Performance benchmarking	Not the primary objective at the current stage

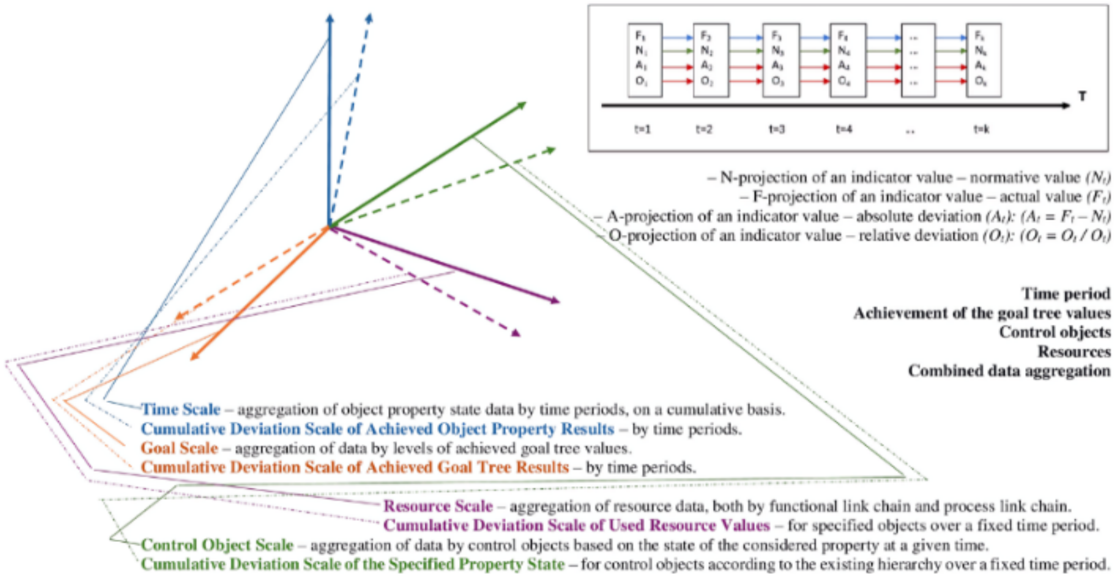

Figure 6: Structure of information flows and types of data aggregation.

When a new constituent system is included, integration does not require re-engineering of existing structures: invariant descriptions and relationship objects allow new stable or temporary links to be formed while preserving existing processing methods and data representations.

2.4 Collinear Structures as Reusable Aggregates

A collinear structure is a reusable fragment of an information object that can be embedded into or linked with other objects. Such fragments serve as stable aggregates that preserve consistency across the topology. They may be nested as long-term components or dynamically connected under specified conditions to form new composite structures. (see Fig. 5).

2.5 Relationship Objects for Object-Centric Querying (Key Novelty)

Traditional modeling notations (e.g., UML/IDEF-style links) represent relationships mainly as graphical connectors. In heterogeneous SoS environments subject to frequent changes, such representations are insufficient to support adaptive querying and integration. We therefore introduce first-class information objects: “Relationship” and “Relationship between two Relationships”.

```
Input:
S = {S1, S2, ..., Sn} - set of constituent systems;
O - mapped invariant objects;
I - mapped invariant indicators;
Rel - set of Relationship objects;
Rel2 - set of Relationship between two Relationships objects;
Q - task-driven query specification;
T - time window;
C - constraint set.
Output:
R - integrated result set with provenance and traceability metadata.
   Procedure:
 1) Parse Q and identify target object
    classes Oq, indicator projections Iq,
    and temporal scope T.
 2) Select from Rel all relationship
    instances connecting Oq, Iq, goals, and
    process stages under T.
 3) Apply rule constraints C to filter valid
    relationship instances.
 4) If composite dependencies are required,
    resolve Rel2 objects and propagate
    higher-order constraints.
 5) Construct the query graph Gq from the
    selected relationship objects.
 6) Retrieve source values from constituent
    systems through mapped invariant objects
    and indicators.
 7) Aggregate and align the retrieved values
    according to the required projection
    axes.
 8) Generate R together with provenance
    attributes, timestamps, and the
    relationship chain used for assembly.
 9) Return R.
```

Algorithm 1: Object-centric query assembly in a heterogeneous SoS.

A Relationship object stores identifiers of linked elements, initiation/termination timestamps, initiating function/process stage, and rule-based conditions for establishing and terminating the link. Crucially, in the proposed query model, a Relationship object becomes a direct source of query data, enabling information requests to be composed by referencing relationship types and conditions rather than by joining relational tables through static keys and indexes.

A Relationship between two Relationships object supports higher-order composition: it connects relationship instances to create structured “landscape maps” of information requests, where higher-level constraints can be propagated from goals to indicators across the polystructural SoS. This enables rapid recomposition of queries for newly assigned tasks and supports operation under unforeseen conditions without rebuilding schema-specific index structures.

Complexity note. The recomposition procedure changes the relationship configuration rather than the source schema, which reduces organizational overhead when new task constraints emerge.

2.6 Reproducible Workflow for Topology Construction and Query Execution

For reproducible implementation, each source system is profiled by schema, process stages, indicator sets, update frequency, and provenance attributes. Local entities are then mapped to invariant SoS structures, after which Relationship objects and, when needed, Relationship between two Relationships objects are instantiated with minimum provenance and rule attributes. The query engine then resolves the relevant relationship chain, assembles the integrated request, and returns the result with time-slice and traceability metadata.

3 RESULTS AND PILOT APPLICATION

To demonstrate the practical applicability of the proposed topology and relationship-object model, we consider a pilot heterogeneous SoS application scenario in which operational data streams and indicator sets are distributed across independent constituent systems. The objective is to assemble an integrated view that supports (i) monitoring deviations from normative indicator values, (ii) aggregating indicators across multiple axes (time, goal achievement, management objects, and resources), and (iii) recomposing information requests when tasks change without modifying source schemas. Table 1 summarizes the practical differences between static schema-dependent integration and the proposed relationship-object approach, while Table 2 presents the qualitative configuration and validation characteristics of the pilot scenario. The ratings in Table 1 are qualitative, expert-judgment assessments derived from the architectural analysis presented in this section (e.g., "No" for schema refactoring reflects that the relationship-object mechanism adds new connections without altering existing source schemas, while "Yes" for the static approach reflects the need to modify keys/indexes when new relationships are introduced) rather than outputs of a quantitative benchmarking study; a quantitative comparison against specific numerical criteria was not conducted and is identified as a direction for future empirical validation.

3.1 Prototype Configuration and Validation Setup

The pilot application considered in this study represents a heterogeneous System-of-Systems scenario in which operational data streams and indicator sets are distributed across independent constituent systems. The proposed topology is used as an integration layer for representing objects, processes, indicators, and their interdependencies in a shared spatio-temporal framework. The pilot validation focuses on topology-based organization of heterogeneous information, invariant indicator representation, adaptive query recomposition under changing tasks, and preservation of cross-system traceability without source-schema modification. The qualitative characteristics of the pilot configuration are summarized in Table 2.

3.2 Integrated Indicator Aggregation in the Information Space

Within the proposed topology, each constituent system provides indicator instances that are mapped to the invariant indicator structure. This enables consistent aggregation and cross-system comparison even when indicators are produced with different update frequencies and reliability levels. Figure 6 illustrates the aggregation workflow and the main projection axes used for analysis (time scale, goal scale, management object scale, resource-use scale, and their combinations). Such projections allow the system to identify actual or predicted deviations from reference benchmarks in near real time or according to a predefined schedule.

$$|s_i - s_i'(t)| < \varphi_i, i = 1, \dots, n. \quad (1)$$

where s_i is the normative state vector of system S_i, $s_i'(t)$ is the actual state over time window t, and φ_i is the stability threshold. Within the proposed framework, each φ_i is conceived as a per-indicator, domain-specific parameter set by the constituent-system owner or analyst (e.g., derived from normative operating ranges, historical variance, or regulatory tolerances for that indicator) rather than a single fixed constant applied uniformly across the SoS; it may also be defined as an adaptive value that is periodically recalibrated as operating conditions change. The present study does not empirically determine specific φ_i values or evaluate a fixed-vs-adaptive calibration policy, and this is identified as a direction for future work.

The pilot study is intended as a functional and architecture-level validation rather than a full performance benchmark. At this stage, it is sufficient to verify traceable integration, adaptive query recomposition, and schema-change avoidance in a heterogeneous SoS environment.

3.3 Homeostasis-Oriented Evaluation (Normative vs. Actual States)

The SoS "homeostasis" concept is expressed as compliance of actual indicator trajectories with normative benchmarks under allowable deviation thresholds: $|s_i - s_i'(t)| < \varphi_i$. This formulation enables uniform anomaly detection across heterogeneous subsystems through invariant indicator descriptions and standardized deviation handling.

3.4 Adaptive Query Recomposition Using Relationship Objects

A key result of the proposed approach is the ability to recompose information requests through Relationship objects without introducing new schema-level joins or rebuilding index structures. When a new task is assigned (e.g., linking a newly introduced process stage to indicators and relevant goals), the required query is formed by instantiating (i) Relationship objects between the corresponding elements and (ii) Relationship between two Relationships objects to express higher-order constraints. As a result, the query logic is expressed through relationship objects as dynamic sources, while the underlying constituent systems remain unchanged. This supports rapid adaptation under unforeseen conditions where new interactions emerge and must be represented immediately. The pilot case therefore validates the functional feasibility of the proposed object-centric query assembly workflow.

Illustrative mini-case. Consider a newly assigned task in which a previously unlinked process stage must be connected to a target object class, a relevant performance indicator, and a supervisory goal within a specified time window. In a static schema/key-based approach, such a task would typically require redesign of join paths and source-side mappings. In the proposed approach, the request is assembled by instantiating Relationship objects between the object class and the process stage, between the process stage and the indicator, and between the goal and the indicator, followed by a Relationship between two Relationships object that encodes the higher-order task constraint. The resulting query returns the affected objects, indicator values, deviation status, and provenance chain without modifying source schemas. This example illustrates how adaptive recomposition is achieved through relationship instantiation rather than schema refactoring.

Example of query assembly.

Input task: identify all objects affected by a newly introduced process-stage dependency, retrieve their current indicator values, and trace the corresponding supervisory goal within time window *T*.

Relationship assembly: $Rel(O, P_{new})$, $Rel(P_{new}, I)$, $Rel(G,I)$ and $Rel2(Rel(O, Pnew), Rel(Pnew,I))$.

Output: integrated result set containing the affected object subset, related indicator projections, deviation status, timestamps, provenance attributes, and the relationship chain used for query construction. The association between the newly introduced process stage and the objects it affects (i.e., the instantiation of Rel(O, P_new)) is intended to be established through the same relationship-object mechanism used elsewhere in the framework — that is, by explicitly instantiating the new relationship at the point the process-stage dependency is introduced, whether by an analyst/administrator or by an upstream system event that triggers relationship creation. The present study does not specify or evaluate an automated inference procedure for discovering such associations without explicit instantiation; this remains a direction for future work.

In practical terms, compared with a static schema/key-based approach, the proposed method reduces adaptation effort by replacing join-path redesign and source-side remapping with the instantiation of relationship objects, while preserving explicit provenance and cross-system traceability.

4 DISCUSSION

The proposed SoS information space topology provides a systematic way to integrate heterogeneous constituent systems while preserving their internal autonomy. Compared with fixed schema- and key-based integration, the object-centric representation treats relationships as first-class information objects, enabling query recomposition through the addition or modification of Relationship instances rather than restructuring source databases. In this respect, the approach is conceptually related to graph-database and knowledge-graph paradigms, but differs by emphasizing invariant structures for goals, processes, objects, and indicators, together with higher-order constructs ("Relationship between two Relationships") for expressing task-dependent constraints across SoS levels.

The framework is particularly relevant for environments in which heterogeneous systems must be coordinated under changing tasks and operational constraints. Its practical value lies in schema-change avoidance, explicit provenance and cross-system traceability, and adaptive recomposition of information requests without source-side refactoring. At the same time, effective application requires disciplined governance of invariant structures and metadata quality, including provenance, time-slice semantics, units, and status definitions.

The present study should therefore be interpreted as an architecture-level and functional validation of the proposed integration model rather than as a full-scale benchmark study. For the objectives of this work, such validation is sufficient because the main contribution is the formalization of the topology-driven integration model and executable query-assembly logic for heterogeneous SoS environments. Large-scale scalability and performance evaluation remain directions for future work.

5 LIMITATIONS

This study has several limitations that should be summarized explicitly. First, the proposed topology and query-assembly mechanism have been validated only at the architecture and pilot-scenario level; no software prototype has been implemented, and no large-scale scalability or performance benchmarking has been conducted against alternative integration approaches (including graph-database/knowledge-graph solutions such as Neo4j or RDF stores, which are mentioned in the literature review but not directly compared quantitatively). Second, the qualitative comparison in Table 1 reflects expert architectural judgment rather than measured criteria from an implemented system. Third, the framework does not currently specify empirically derived values or a calibration procedure for the stability thresholds φ_i, nor an automated mechanism for discovering newly relevant object-process associations without explicit relationship instantiation. Finally, the pilot validation described in Section 4 is a small-scale, illustrative configuration rather than an integration with real-world constituent systems; extending the framework to a real software prototype, benchmarking it against existing integration technologies, and validating it in production SoS environments remain directions for future work.

6 CONCLUSIONS

This paper addressed the challenge of coordinating data and process management in heterogeneous System-of-Systems (SoS) environments undergoing digital transformation, where constituent systems preserve their autonomy, data semantics remain heterogeneous, and operational tasks evolve dynamically. To address this problem, a topology-driven organization of the SoS information space was proposed, providing a shared spatio-temporal perspective and standardized connectivity patterns for integrating heterogeneous information arrays. A compact set of invariant structures-goals, objects and their elementary properties, processes and their stages, indicators, and object-process aggregates-was introduced to support consistent aggregation, monitoring, and deviation analysis across interconnected systems.

The main scientific and applied contribution of the study is the development of an object-centric query mechanism in which Relationship objects and Relationship between two Relationships objects are formalized as first-class information entities. Unlike conventional index- or key-based approaches, the proposed model supports adaptive recomposition of information requests without schema refactoring, while preserving explicit provenance and cross-system traceability.

Overall, the pilot results provide architecture-level evidence of feasibility and practical relevance. The framework can serve as an interoperability layer for decision-support and digital transformation toolchains in heterogeneous SoS settings. Large-scale benchmarking and prototype software implementation remain directions for future work.

REFERENCES

[1] C. E. Dridi, Z. Benzadri, and F. Belala, "System of systems modelling: Recent work review and a path forward," in Proc. 2020 Int. Conf. on Advanced Aspects of Software Engineering (ICAASE), 2020, pp. 1-8.

[2] Z. Fang, "System-of-systems architecture selection: A survey of issues, methods, and opportunities," IEEE Systems Journal, vol. 16, no. 3, pp. 4768-4779, 2022, [Online]. Available: https://doi.org/10.1109/JSYST.2021.3119294.

[3] P. Uday and K. Marais, "Designing resilient systems-of-systems: A survey of metrics, methods, and challenges," Systems Engineering, vol. 18, no. 5, pp. 491-510, 2015, [Online]. Available: https://doi.org/10.1002/sys.21325.

[4] M. W. Maier, "Architecting principles for systems-of-systems," Systems Engineering, vol. 1, no. 4, pp. 267-284, 1998, [Online]. Available: https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D.

[5] ISO/IEC/IEEE 42010:2011, Systems and Software Engineering - Architecture Description. ISO, 2011.

[6] G. Vial, "Understanding digital transformation: A review and a research agenda," The Journal of Strategic Information Systems, vol. 28, no. 2, pp. 118-144, 2019, [Online]. Available: https://doi.org/10.1016/j.jsis.2019.01.003.

[7] A. Halevy, A. Rajaraman, and J. Ordille, "Data integration: The teenage years," in Proc. 32nd Int. Conf. on Very Large Data Bases (VLDB), 2006, pp. 9-16.

[8] R. Angles and C. Gutierrez, "Survey of graph database models," ACM Computing Surveys, vol. 40, no. 1, Art. 1, 2008, [Online]. Available: https://doi.org/10.1145/1322432.1322433.

[9] A. Hogan et al., "Knowledge graphs," ACM Computing Surveys, vol. 54, no. 4, Art. 71, 2021, [Online]. Available: https://doi.org/10.1145/3447772.

[10] W3C, "RDF 1.1 Concepts and Abstract Syntax," W3C Recommendation, Feb. 25, 2014, [Online]. Available: https://www.w3.org/TR/rdf11-concepts/, [Accessed: Mar. 2026].

[11] W3C, "SPARQL 1.1 Query Language," W3C Recommendation, Mar. 21, 2013, [Online]. Available: https://www.w3.org/TR/sparql11-query/, [Accessed: Mar. 2026].

[12] I. Robinson, J. Webber, and E. Eifrem, Graph Databases: New Opportunities for Connected Data, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2015.

[13] V. Bakharev, B. B. Giyosidinov, T. Kirillova, N. Kozlova, V. Sheleiko, and T. Cherkesova, "State-Run food retail chains as a tool of food security: An analysis of existing practices," in Understanding the Digital Transformation of Socio-Economic-Technological Systems, T. C. Devezas, M. A. Berawi, S. E. Barykin, and T. Kudryavtseva, Eds. Cham, Switzerland: Springer, 2024, pp. 581-594, [Online]. Available: https://doi.org/10.1007/978-3-031-56677-6_45.

[14] S. A. Yuldasheva, A. J. Abdullayev, L. U. Akbarova, J. K. Igitov, and N. B. Shanazarova, "The vanguard experience and the prospect of sustainable development of the transport and logistics complex," in Technological Horizons of Decarbonization Based on Environmental Innovations, E. G. Popkova, Ed. Cham, Switzerland: Springer, 2025, [Online]. Available: https://doi.org/10.1007/978-3-031-82210-0_10.

[15] Val. Shvedenko and Vl. Shvedenko, "Ensuring a harmonious state of smart space when there is a conflict of interest of its elements," in Smart Spaces, Intelligent Data-Centric Systems. Amsterdam, The Netherlands: Elsevier, 2024, pp. 235-255, [Online]. Available: https://doi.org/10.1016/B978-0-443-13462-3.00005-4.

[16] M. M. Talipov, "Computational modeling and analysis of mechanical power consumption in train assemblers' work," Proc. Int. Conf. on Applied Innovation in IT, vol. 13, no. 2, pp. 419-426, 2025, [Online]. Available: https://doi.org/10.25673/120513.

[17] M. Yakubov, M. Talipov, and U. Nurullaev, "An optimized Lagrangian approach to reactive power compensation in nonlinear traction power-supply systems," Vibroengineering Procedia, vol. 60, pp. 724-730, Dec. 2025, [Online]. Available: https://doi.org/10.21595/vp.2025.25604.