2026 European Conference on Computing in Construction
Corfu, Greece
July 12–15, 2026

TOWARDS A MULTI-DOMAIN SEMANTIC ARCHITECTURE FOR GRAPH-BASED ORCHESTRATION OF DECENTRALISED ADDITIVE MANUFACTURING

Lukas Kirner, Alex Donkers, and Ekaterina Petrova
Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract

Construction-scale additive manufacturing (AM) is often executed in decentralised, multi-actor project environments, where production information is distributed across heterogeneous systems. This fragmentation hinders automation, coordination, and life-cycle traceability. This paper proposes a multi-domain semantic system architecture to support graph-based integration and orchestration of decentralised AM workflows. The architecture separates system topology, functional service interfaces, information requirements, and runtime provenance. Using a concrete 3D printing use case, the approach demonstrates how modular software components can be integrated through a shared digital-twin interface while maintaining end-to-end traceability of generated and consumed artefacts.

Introduction

Construction sectors in many developed countries share persistent structural challenges, including rising costs, long project durations, shortages of skilled labour, and increasing regulatory requirements related to sustainability and life-cycle assessment (Abdelalim et al., 2025; Cabeza et al., 2014). At the same time, the demand for affordable housing and resource-efficient construction continues to grow. These pressures challenge traditional construction methods and motivate the exploration of alternative production paradigms (Kedir and Hall, 2021).

Construction-scale additive manufacturing (AM) such as concrete 3D printing, has emerged as a promising approach to address some of these challenges (Lim et al., 2012; Buswell et al., 2018). By enabling digitally driven fabrication processes for building components, AM offers potential benefits including reduced labour dependency, shorter construction times, and material-efficient geometries (Hager et al., 2016). At the same time, the strong reliance on digital planning, parameterisation, and process control introduces complex information dependencies between design, material definition, and execution.

The implementation of AM on construction scale is usually integrated within temporary, multi-actor project environments. Design, material definition, machine control, and quality assurance are often handled by different stakeholders and supported by heterogeneous software systems. As a result, production-related information is generated across organisational and temporal boundaries, while increasing requirements for documentation and life-cycle assessment demand consistent traceability of design decisions and production-related information (Wu et al., 2016).

While practical challenges that relate to printing hardware and material behaviour are prominent, efficient data management and process coordination represent equally significant hurdles (Jaskuła et al., 2024). Information is frequently exchanged through isolated files, proprietary software tools, or manual handovers, resulting in limited interoperability and weak integration between design, execution, and as-built documentation. These limitations hinder automation, scalability, and the consistent availability of information required for life-cycle assessment (Cabeza et al., 2014).

Semantic models and linked data approaches have been proposed to address these challenges by enabling formal, machine-interpretable representations of entities, relationships, and dependencies across organisational and system boundaries. Graph-based representations are particularly well suited for decentralised settings in which no single system can act as a central source of truth (Pauwels et al., 2017).

Despite this potential, the practical use of semantic approaches in decentralised AM remains limited. Existing ontologies and data models typically address isolated aspects of the process chain, such as geometry, manufacturing activities, or sensor observations (Kirner et al., 2024a), while lacking a coherent architectural separation between system integration, functional interfaces, information dependencies, and execution traces. As a result, semantic models often become either difficult to operationalise or insufficient to support cross-system coordination and traceability.

This paper addresses this gap by proposing a multi-domain semantic system architecture for decentralised AM workflows. The architecture explicitly separates system topology, functional service descriptions, information requirements between modules, and runtime provenance. Using construction-scale concrete 3D printing as a representative use case, the approach demonstrates how existing semantic standards can be combined into an actionable architecture that supports integration, coordination, and end-to-end traceability across the AM process chain.

Related Work

Additive manufacturing

Construction-scale AM transfers layer-wise fabrication principles from industrial AM to the production of building components and, increasingly, full-scale construction, with early work on large-scale automated material deposition such as Contour Crafting and concrete printing systems (Khoshnevis, 2004; Lim et al., 2012). Over the past decade, concrete 3D printing (3DCP) has become the dominant form of construction-scale AM due to its poten-

tial to reduce formwork, enable material-efficient geometries, and increase automation (Buswell et al., 2018).

Typical 3DCP workflows span digital design, toolpath generation, material preparation, and layer-wise deposition with subsequent quality assurance. While geometry-to-machine translation resembles industrial CAM pipelines, construction-scale AM is strongly influenced by time-dependent material behaviour, such as buildability and inter-layer adhesion, which couples digital models, process parameters, and execution (Mechtcherine et al., 2020). Challenges in data management and process coordination include fragmented information flows, limited interoperability, and insufficient traceability of material definitions, parameters, and as-built outcomes (Bos et al., 2016; Jaskuła et al., 2024).

Semantic and ontology-based approaches in AM

To address gaps in data availability, data-centric frameworks in AM focus on the structured capture of process information. Deetman et al. introduce a database framework for 3D concrete printing to store cross-disciplinary process data for analysis and reproducibility focussing on storage and schema (Deetman et al., 2026).

For semantic integration across tools and domains, ontology-driven approaches have been suggested to structure and integrate knowledge in AM. Li and Petzold present a mixture-of-domain documentation framework that captures cross-domain knowledge in a shared knowledge graph to support semantic retrieval and question answering across robotics, materials, and BIM domains (Li and Petzold, 2025). Domain-specific semantic models have also been introduced, for example for wire arc additive manufacturing and composite extrusion, to formalise process knowledge and enable semantic interoperability across tools (Ahmad et al., 2025). Gouttebroze et al. propose a semantic standard and process ontology for additive manufacturing that aligns standard terminology with formal process descriptions (Gouttebroze et al., 2023).

System integration and system-of-systems approaches

The increasing distribution of digital tools in construction has motivated system-of-systems approaches for integrating heterogeneous services. Hagedorn et al. propose a system-of-systems architecture for deploying containerised construction IT services, focusing on modular integration and explicit service interfaces across independently managed components (Hagedorn et al., 2025). In a more general setting, Feng et al. present an ontology-based approach for modelling system-of-systems architectures and operational views to provide a shared semantic foundation for describing interacting systems (Feng et al., 2023).

These approaches address deployment and architectural modelling of distributed systems, but do not explicitly model information dependencies between tools or capture execution-time provenance needed for traceability across workflows.

Methodology

This work develops a multo-domain semantic system architecture to support graph-based coordination and traceability in decentralised construction-scale AM. The methodology to achieve this follows four steps: (I) use-case driven examination of an exemplary AM process chain and involved software systems, (II) identification of coordination and traceability concerns across system boundaries, (III) structuring of these concerns into distinct semantic domains, and (IV) alignment of each domain with existing standards and ontologies to form an extensible architectural framework.

I Use-case driven system examination

To ground the methodology in practice, the approach starts from a representative decentralised AM workflow derived from the EU-funded project "Additive To Predictive Manufacturing" (AM2PM)[1]. The focus is placed on a possible design–material–optimisation sub-workflow that supports predictive and sustainability-driven decision-making, while abstracting from the physical printing process itself. This scoping reflects the intention to examine information dependencies between software modules rather than machine control or sensing.

The considered sub-workflow couples material design, structural optimisation, and life-cycle assessment. A concrete mix design module produces a mix specification that includes, among others, cement content and material properties such as compressive strength. The cement component of the mix is linked to environmental product declaration (EPD) data, enabling the association of embodied CO_2 emissions with the mix composition. These material properties are required inputs for a shape optimisation module, which aims to minimise material usage for a slab structure by exploiting the mechanical performance of both the printed lost formwork and the subsequently poured concrete. Equipment type, staff, material quantities and other practical parameters are assessed in a simulation module. The involved modules and their interactions are illustrated in a simplified manner in Figure 1, where modules interact with a decentralized graph-based integration domain. Brown arrows represent two-way interaction between a module and the integration domain, and contain data conversion. Solid green arrows represent semantic relationships between concepts in a module, whereas dotted green arrows represent mappings between such semantic concepts. An optimization module that feeds a control module is using the integration domain. The latter is not the primary focus of this work.

Based on the material parameters and structural requirements, the optimisation module determines an efficient geometry, which directly influences the amount of material to be printed and cast. Whether material quantities are later measured during printing or estimated through predictive simulation is outside the scope of this work. In-

[1] https://www.am2pm-project.eu/

stead, attention is placed on how the required information is made available and propagated across modules. Given access to both the optimised geometry and the concrete mix specification, a life-cycle assessment (LCA) module can estimate the environmental impact of the slab, dominated by cement-related emissions.

In such a setting, changes in concrete composition affect structural behaviour, geometric optimisation results, material quantities, and downstream environmental assessments. These interdependencies are realised across multiple software modules that operate independently but rely on shared information artefacts.

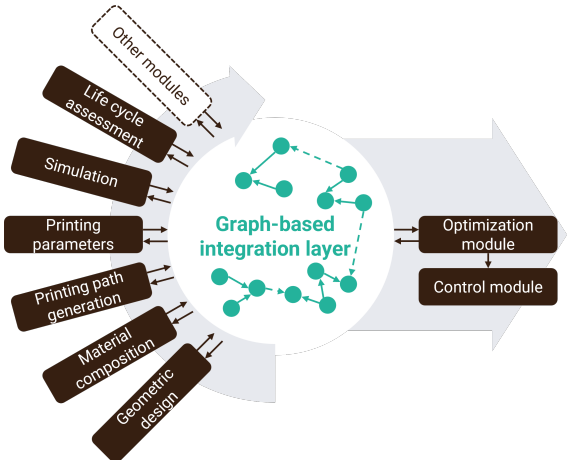

Figure 1: Simplified AM2PM module interaction.

II Coordination and traceability concerns

From the use-case driven system examination, a set of recurring and characteristic coordination and traceability concerns emerge. These concerns arise from the fact that multiple independent software systems contribute to a shared production outcome, while relying on different representations, responsibilities, and temporal perspectives on the same information artefacts.

A first class of concerns relates to information dependencies across system boundaries. Design artefacts, material definitions, optimisation results, and environmental data are produced and consumed by different modules, often at different points in time. Dependencies between these artefacts are typically implicit, making it difficult to determine whether all required inputs are available, which upstream changes necessitate recomputation, or which downstream results are affected by modifications in design or material parameters. Such dependencies are relevant at design time and influence validation, scheduling, and coordination decisions before any execution takes place.

A second class of concerns concerns traceability across execution and decision contexts. Results generated during execution, simulation, or optimisation need to be traceable back to the design-time inputs, assumptions, and configurations that led to them. This traceability extends beyond runtime logging and includes the ability to associate outputs with the specific versions of input artefacts and system configurations that were in effect when they were produced. Without explicit representation of these relations, it becomes difficult to support reproducibility, accountability, and life-cycle assessment in workflows where responsibilities and data are distributed across multiple systems.

Addressing these issues requires a structuring of system-related knowledge that separates stable system descriptions, information dependencies, and execution-time records.

III Derivation of the semantic architecture

The identified coordination and traceability concerns cannot be handled well within a single, undifferentiated semantic model. Design-time dependencies, stable system descriptions, and execution-time records differ in how long they remain valid, who is responsible for them, and how often they change. Treating these aspects in the same way risks either mixing volatile execution data into long-lived system descriptions or failing to represent dependencies that are needed for coordination before execution.

To address this, the proposed architecture separates semantic responsibilities into four domains with distinct scope, temporality, and mutability (see Table 1). The domains distinguish between system structure (*D1*), functional interfaces between systems (*D2*), information requirements and dependencies (*D3*), and provenance records of concrete executions (*D4*). This separation follows patterns observed in the use-case driven examination rather than assumptions imposed by a specific ontology.

By separating these concerns, each domain can be modelled at an appropriate level of abstraction. System topology and interfaces change slowly and support integration, information requirements capture dependencies that must be known before execution, and provenance records document what actually happened. This allows the domains to evolve independently while remaining linked, which is essential in decentralised workflows with multiple interacting systems.

IV Ontology selection and integration strategy

The proposed architecture is instantiated using a deliberately small set of established ontologies, selected for their conceptual fit with individual domains rather than for exhaustive domain coverage. This choice defines a minimal semantic starting point. Fig. 2 shows the integration of these ontologies.

System topology (D1)

At the system topology domain, structural relationships between software components are represented using *SAREF for System* (SAREF4SYST, prefix `s4syst`) (Lefrançois, 2019). It provides explicit concepts for systems, connection points, and connections, making it suitable for modelling decentralised software modules and their integration while focussing on a relatively stable system structure that aligns with the slow-changing nature of topology descriptions.

Table 1: Domain separation and according to scope, temporality and mutability (non-exhaustive).

Domain	Scope	Temporality	Mutability
D1: System topology	system structure	mostly static	slow
D2: Functional description	interface	mostly static	slow
D3: Information requirements	workflow dependency	logical	medium
D4: Provenance and traceability	execution record	historical	immutable

Functional service description (D2)
Functional interfaces are described using the W3C *Web of Things (WoT) Thing Description* (prefix `td`) (Charpenay et al., 2023). Thing Descriptions provide a machine-interpretable, declarative description of interaction affordances exposed by a system, including *properties*, *actions*, and *events*. In this architecture, the Digital Twin interface is represented as a `td:Thing` whose affordances describe access to shared project artefacts (e.g., concrete mix specifications, optimisation results) without prescribing orchestration logic or encoding execution-time provenance. This makes WoT TD suitable for complex, decentralised module ecosystems while keeping the functional contract separated from design-time information dependencies (*D3*) and runtime traces (*D4*).

Information requirements and dependencies (D3)
Design-time information dependencies are captured in the information requirement domain using the IoC Process Ontology (prefix `ioc`) (Kirner et al., 2024b) in alignment with the *WoT* (Charpenay et al., 2023). An `ioc:Process` is aligned with a `td:Thing`, while specific operations are modelled as WoT affordances. Information requirements are specified at the level of these affordances using `ioc:hasInputInformation` and `ioc:hasOutputInformation` to indicate which `ioc:Information` artefacts are required or produced by a given action, independent of their runtime access or generation (e.g. a topology optimisation action requiring a concrete mix specification). Information artefacts are treated as logical coordination objects supporting dependency reasoning, completeness checks, and change impact analysis prior to execution. Artefacts may be described using domain ontologies, but their internal semantics are not modelled at this domain. The separation of D3 from execution-time provenance enables validation and orchestration without reliance on historical execution data. In practice, explicitly modelled information requirements can be validated using SHACL shapes (Knublauch and Kontokostas, 2017). The domain itself may require future extensions by more formal requirement or constraint ontologies.

Provenance and traceability (D4)
Execution-time behaviour and traceability are represented using the W3C Provenance Ontology *PROV-O* (prefix `prov:`) (Lebo et al., 2013). PROV-O provides a standard model for representing activities, entities, and agents, and for expressing how information artefacts are generated, used, and derived over time. In the proposed architecture, provenance is used exclusively to record immutable execution histories, linking concrete runs of distributed modules to the specific input artefacts and system configurations that produced them. This makes PROV-O well suited for auditability, reproducibility, and life-cycle relevant traceability, while remaining strictly separated from coordination and dependency modelling.

Table 2: Overview of used ontologies for the proposed semantic architecture.

Ontology	Main classes and purpose
D1: `s4syst:`	*s4syst:System*, *s4syst:Connection/Point* System topology: software modules and connections (Lefrançois, 2019)
D2: `td:`	*td:Thing*, *td:PropertyAffordance*, *td:ActionAffordance*, *td:EventAffordance* Functional interfaces: interaction affordances and specific endpoints (Charpenay et al., 2023)
D3: `ioc:`	*ioc:Process*, *ioc:Information* Information requirement: Expresses design-time input/output dependencies (Kirner et al., 2024b) validated using SHACL (Knublauch and Kontokostas, 2017)
D4: `prov:`	*prov:Entity*, *prov:Activity*, *prov:Agent* Immutable execution history: Branches with used/generated artefacts and responsible agents for traceability (Lebo et al., 2013)

Cross-domain integration principles
The architecture deliberately avoids merging ontologies or introducing inheritance relations across domains. Each domain defines its own relations and responsibilities, while instances may be referenced across domains when appropriate (e.g. a software system described at the topology domain acting as a provenance agent at runtime). Relations that describe execution-time behaviour are expressed exclusively at the provenance domain. This separation prevents semantic overlap between design-time coordination and runtime history, while allowing referenced domain ontologies to evolve independently.
The selected vocabularies are not assumed to be sufficient in the long term. Instead, the architecture is designed to allow extensions and the integration of additional domain-specific ontologies as requirements evolve. Table 2 summarises the ontologies and standards used at each domain of the proposed architecture and their respective roles.

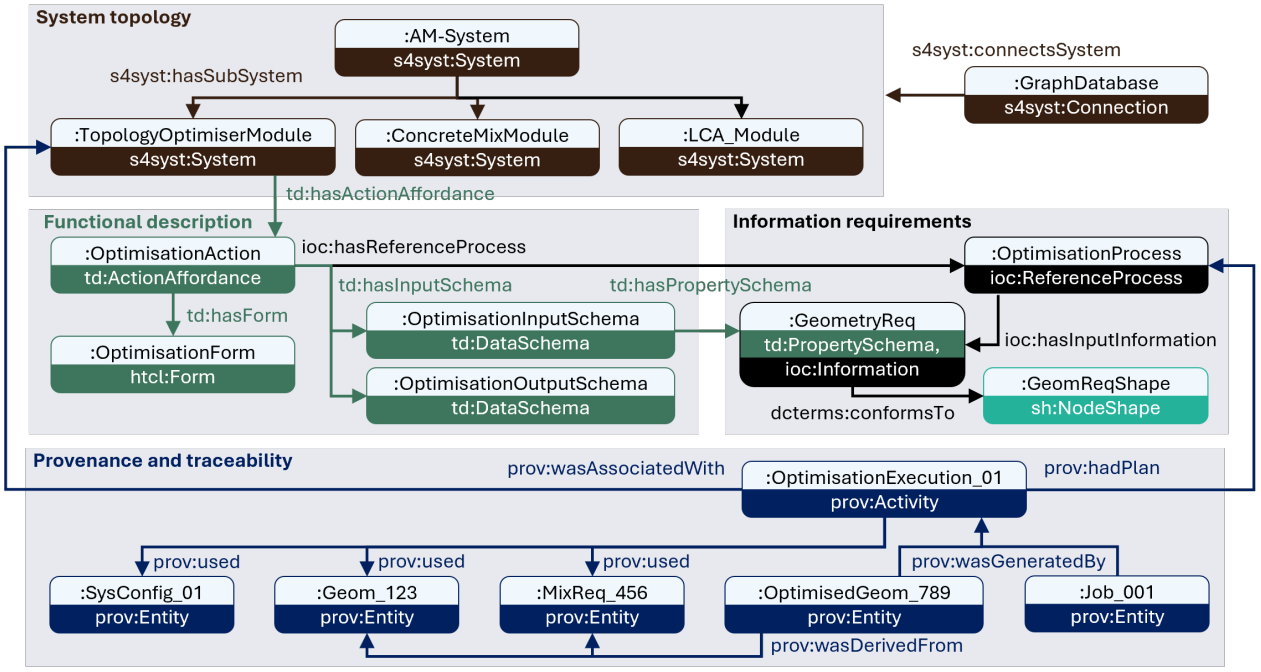

Figure 2: System integration using proposed ontologies, including a demonstrator instantiation

Evaluation and Demonstrator

Having defined the semantic architecture and its ontology alignment, the evaluation demonstrates its instantiation in the AM2PM use case through RDF instances, exemplary SPARQL queries, and a visualised provenance-aware information flow, and presents the semantic model together with its implementation in a platform prototype.

Instance-level semantic model for the AM2PM use case

The system and topology domain (D1) instantiates the Digital Twin Orchestrator as a `s4syst:System` and `td:Thing`. Exemplarily, the material design, topology optimisation, and LCA modules are attached via `s4syst:hasSubSystem`. Interactions between these components are mediated through an explicit `s4syst:Connection`, enabling the model to describe both structural composition and communication paths without mixing functional roles. Each module is additionally a `prov:Agent`, allowing execution responsibilities to be linked directly to provenance records.

The interaction domain (D2) is realised by a `td:ActionAffordance` bound to an HTTP endpoint via `hctl:hasTarget` and `htv:methodName`. The action is linked through `wp4-ext:hasReferenceProcess`, connecting the API operation to the abstract process description. Its payload is constrained by `td:hasInputSchema` and `td:hasOutputSchema`, where inputs such as *:geometry* and *:mixRequirements* are defined as IRIs, enforcing a by-reference data exchange pattern. This separates interface invocation from data ownership and allows the same service interface to operate over distributed or externally managed artefacts.

The information domain (D3) specifies required and produced artefacts independently of execution. The reference process declares its contract using `ioc:hasInputInformation` and `ioc:hasOutputInformation`, linking, for example, to *inst:GeometryInfo*, *inst:ConcreteMixReqInfo*, and *inst:OptimisedGeometryInfo*. Each information object is annotated with `dcterms:conformsTo` to reference a SHACL shape, enabling validation of the referenced resources without embedding the data into the invocation payload. As a result, information completeness and structural validity can be checked before execution and reused consistently.

The provenance domain (D4) models specific process executions as `prov:Activity` and links them to the reference process via `prov:hadPlan`. Used inputs are instantiated with `prov:used`, generated results with `prov:wasGeneratedBy`, and input-result dependencies with `prov:wasDerivedFrom`, making the full derivation chain between input versions and outputs explicitly queryable. Input and output artefacts are represented as `prov:Entity`. As example, revising a concrete mix parameter such as its compressive strength results in a new mix entity that is `prov:used` in a subsequent optimisation run. This leads to a modified optimised geometry, which in turn affects the life-cycle assessment results derived from that geometry. The corresponding result entities can be linked via `prov:wasRevisionOf`, while preserving `prov:wasDerivedFrom` relations to the exact input versions, enabling causal tracing of how changes in material parameters propagate through optimisation and LCA.

Query-based semantic analysis

To demonstrate the usability of the proposed approach, a set of simple questions covering all four domains is answered using SPARQL queries. The queries reflect typical information needs of users and software agents interacting with the Digital Twin system. Listing 1 shows the additional prefixes used.

Listing 1: Additional prefixes for subsequent SPARQL queries.

```
dcterms:<http://purl.org/dc/terms/>
hctl: <https://www.w3.org/2019/wot/hypermedia#>
htv: <http://www.w3.org/2011/http#>
omg: <https://w3id.org/omg#>
sh: <http://www.w3.org/ns/shacl#>
```

The first question addresses the system topology (D1): *Which modules exist and how are they connected?* The corresponding SPARQL query (Listing 2) returns all connected modules with their labels and descriptions and serves as an entry point for discovering the current system structure.

Listing 2: Query for retrieving all connected modules (D1).

```
SELECT ?subsystem ?label ?description
WHERE {
  ?syst s4syst:hasSubSystem ?subsystem;
        s4syst:connectedThrough ?connection.
  ?subsystem rdfs:label ?label;
  dcterms:description ?description;
  s4syst:connectedThrough ?connection .}
```

The second question focuses on the available interaction methods, such as callable API endpoints (D2). Using one of the modules identified in Listing 2, the question is: *Which actions can be invoked on the TopologyOptimiser module and what input do they expect?* The query returns the available actions, for example *inst:RunOptimisationForm*, an HTTP POST form that triggers the topology optimisation. Labels and `dcterms:description` support not only invocation but also discovery of these endpoints, for instance by software agents or agent-based toolchains.

Listing 3: Query to the form for a specific affordance (D2).

```
SELECT ?action ?actionl ?actionDesc ?method ?target
     ?contentT
WHERE {
  inst:TopologyOptimiserModule td:hasActionAffordance
       ?action.
  OPTIONAL {?action rdfs:label ?actionl.}
  OPTIONAL {?action dcterms:description ?actionDesc.}

  ?action td:hasForm ?form.
  OPTIONAL {?form htv:methodName ?method.}
  OPTIONAL {?form hctl:hasTarget ?target.}
  OPTIONAL {?form hctl:forContentType ?contentT.}}
```

The third question addresses information requirements and constraints (D3): *What is the needed content of infromation the input needs to perform the requested action? How must it be sturctured?* Starting from the action identified in Listing 3, the query follows its link to the associated `ioc:ReferenceProcess` (aligned to `prov:Plan`). Using `ioc:hasInputInformation`, it retrieves the strcture of the required information artefacts together with their SHACL-based constraints. The use of D3 and the required domain ontologies to concisely describe the expected data structures is subject to future work.

Listing 4: Query to retrieve information requirements.

```
SELECT ?info ?fieldName ?dataT ?shape
WHERE {
  inst:OptimisationAction ioc:hasReferenceProcess ?
      proc.
  ?proc ioc:hasInputInformation ?info.
  OPTIONAL {?info rdfs:label ?fieldName.}
  OPTIONAL {?info td:datatype ?datatT.}
  OPTIONAL {?info dcterms:conformsTo ?shape.}}
```

The fourth question addresses provenance after a process execution (D4): *Which executions produced which results, and which data influenced these outcomes?* In the example query, the resulting `omg:Geometry` is used as an entry point to retrieve the activity that generated it and the associated execution context. This includes the used input data, the reference process or plan that was followed, and the responsible module (`prov:Agent`), in this case `inst:TopologyOptimiser`. For simulation workflows, this supports comparing outputs produced under the same reference process but with different inputs, enabling systematic analysis of how parameter changes affect the results e.g. for additive manufacturing.

Listing 5: Query to retrieve provenance information after a sucessfull process execution.

```
SELECT ?activityl ?inputs ?result ?agentl
WHERE {
  ?result a omg:Geometry.
  ?result prov:wasGeneratedBy ?activity.
  ?activity prov:used ?inputs;
    prov:hadPlan ?reference;
    prov:wasAssociatedWith ?agent.
  OPTIONAL { ?activity rdfs:label ?activityl.}
  OPTIONAL { ?agent rdfs:label ?agentl.}}
```

The presented questions can be asked independently, for example by different systems or software agents with distinct responsibilities. They can also be applied in the given sequence, progressing from a general understanding of the system, to concrete instructions for its use, to its information requirements and constraints, and finally to the interpretation of its outputs. This progression is essential for supporting simulation workflows and, ultimately, can enable future predictive manufacturing applications.

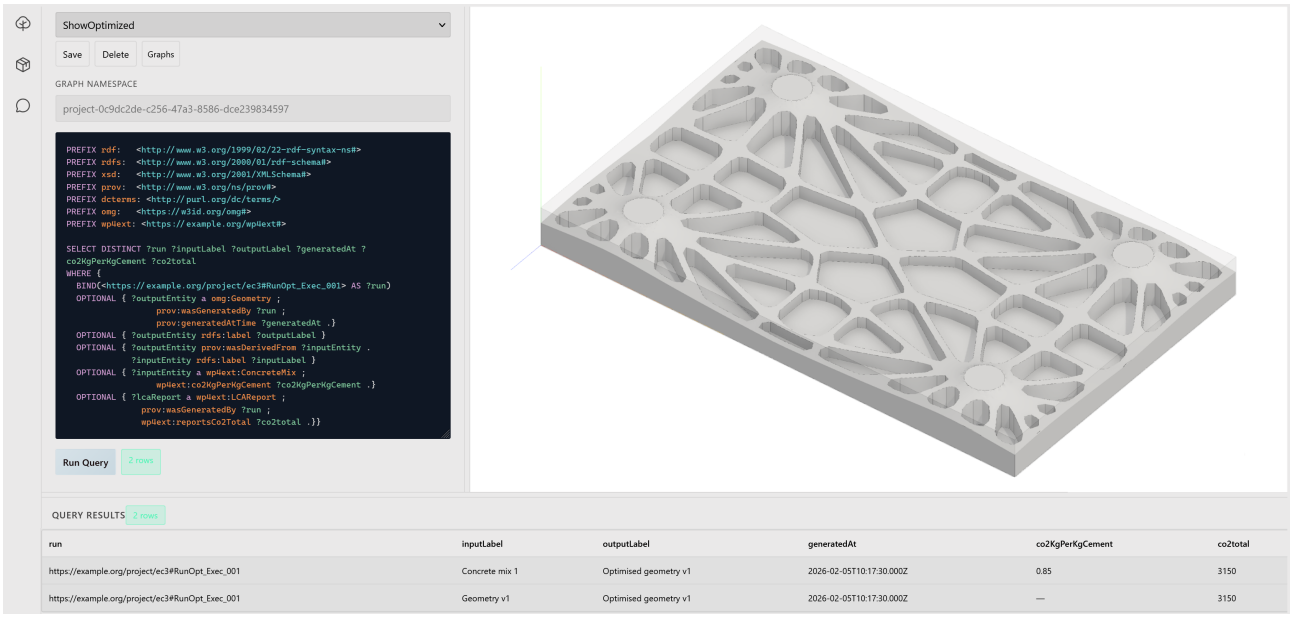

Figure 3: Web platform linking original and optimised geometry to multi-domain semantic data.

Platform-based visualisation and comparison

To demonstrate what the multi-domain semantic architecture can achieve for user-facing applications, the presented exemplary A-Box is stored in a triple store of a linked data–based web platform. The application exposes a query endpoint that can be controlled by its viewer component (see Fig. 3). On the left, a SPARQL query is used to retrieve the output of an optimisation run involving the concrete mixing, topology optimisation, and LCA modules. Here, we focus on the query logic of D3 and D4. The 3D viewer shows in transparent (as hint) the original, non-optimised geometry and highlights the results of the optimised run. Below, the query results are shown in a table, so users can access relevant information about the completed optimisation run, such as its IRI, the used cement mix, the time the geometry was generated and resulting CO_2-equivalent.

Results & Discussion

The evaluation shows that the semantic integration of distributed modules, information requirements, and provenance records enables the comparison of two variants that satisfy the same functional and structural requirements while differing in material usage and CO_2-equivalent emissions, and that a simple web-based 3D viewer can make these differences accessible to users.

Beyond this specific example, queries across domains illustrate how the approach supports typical coordination and analysis tasks, such as discovering available modules, checking required information, and tracing how results were produced. This indicates that the architecture is not limited to a single workflow, but can serve as a general integration pattern to enable system-of-systems approaches. At the same time, the study also reveals clear limitations. The current evaluation is restricted in scale, while the modelling effort is significant, especially for defining information requirements and maintaining alignments between vocabularies. Constraints are currently limited to basic validation and need to be refined to remain both precise and usable in practice. In addition, issues such as governance, trust, and security were not addressed, but may become critical in cross-organisational settings. In short, the approach can be a viable basis for coordination and traceability in decentralised AM workflows, but its practical adoption depends on reducing modelling effort, strengthening constraint definitions, and validating the approach in larger and more heterogeneous settings.

Conclusion

This paper presented a multi-domain semantic architecture for decentralised additive manufacturing workflows. The approach separates system structure, interfaces, information requirements, and provenance. The evaluation shows that this separation supports integration, comparison of results, and traceability across tools. The current work is limited to a prototype and a small use case. Future work will address larger and more distributed system landscapes in AM, where coordination may also involve agentic AI components. Building on this, simulation-based workflows that rely on explicit system and information modelling will be developed, with the goal of supporting predictive manufacturing (PM).

Acknowledgments

This work has received funding from the European Union's Horizon Europe research and innovation programme under the Pathfinder project "AM2PM" (Grant Agreement No. 101162318).

References

Abdelalim, A. M. et al. (2025). An analysis of factors contributing to cost overruns in the global construction industry. Buildings, 15(1):18.

Ahmad, M. E., Peralta Abadía, J. J., Chmelnizkij, A., Reimann, J., Hildebrand, J., Bergmann, J. P., and Smarsly, K. (2025). Ontology-based knowledge representation for wire arc additive manufacturing and composite extrusion modeling. In EG-ICE 2025, Glasgow, United Kingdom. University of Strathclyde Publishing.

Bos, F. P., Wolfs, R. J. M., Ahmed, Z. Y., and Salet, T. A. M. (2016). Additive manufacturing of concrete in construction: potentials and challenges of 3d concrete printing. Virtual and Physical Prototyping, 11(3):209–225.

Buswell, R. A., Leal da Silva, W. R., Jones, S. Z., and Dirrenberger, J. (2018). 3d printing using concrete extrusion: A roadmap for research. Cement and Concrete Research, 112:37–49.

Cabeza, L. F., Rincón, L., Vilariño, V., Pérez, G., and Castell, A. (2014). Life cycle assessment (lca) and life cycle energy analysis (lcea) of buildings and the building sector: A review. Renewable and Sustainable Energy Reviews, 29:394–416.

Charpenay, V., Lefrançois, M., Villalón, M. P., and Käbisch, S. (2023). Web of Things (WoT) Thing Description (TD) Ontology. W3C Editor's Draft. Editor's Draft, RDF vocabulary.

Deetman, A., Bos, D., Lucas, S. S., Salet, T., and Wolfs, R. (2026). A database framework for 3d concrete printing. Results in Engineering, 29:108669.

Feng, Y., Zou, Q., Zhou, C., Liu, Y., and Peng, Q. (2023). Ontology-based architecture process of system-of-systems: From capability development to operational modeling. Applied Sciences, 13(9):5419.

Gouttebroze, S., Friis, J., Hovig, E. W., and Boivie, K. (2023). Toward semantic standard and process ontology for additive manufacturing. IOP Conference Series: Materials Science and Engineering, 1281(1):012014.

Hagedorn, P., Petrova, E., and König, M. (2025). A system-of-systems approach for deploying containerized construction digital twins using linked data. In Advances in Information Technology in Civil and Building Engineering - Proceedings of ICCCBE 2024, Lecture Notes in Civil Engineering, pages 478–491.

Hager, I., Golonka, A., and Putanowicz, R. (2016). 3d printing of buildings and building components as the future of sustainable construction? Procedia Engineering, 151:292–299.

Jaskuła, K., Kifokeris, D., Papadonikolaki, E., and Rovas, D. (2024). Common data environments in construction: state-of-the-art and challenges for practical implementation. Construction Innovation.

Kedir, F. and Hall, D. M. (2021). Resource efficiency in industrialized housing construction – a systematic review of current performance and future opportunities. Journal of Cleaner Production, 286:125443.

Khoshnevis, B. (2004). Automated construction by contour crafting—related robotics and information technologies. Automation in Construction, 13(1):5–19.

Kirner, L., Jung, V., Oraskari, J., and Brell-Cokcan, S. (2024a). Enhancing robotic steel prefabrication with semantic digital twins driven by established industry standards. Automation in Construction, 167:105699.

Kirner, L., Oraskari, J., Wildemann, P., and Brell-Cokcan, S. (2024b). Internet of Construction Process Ontology (ioc) v 0.5.

Knublauch, H. and Kontokostas, D. (2017). Shapes Constraint Language (SHACL). W3C Recommendation.

Lebo, T., Sahoo, S., and McGuinness, D. (2013). PROV-O: The PROV Ontology. W3C Recommendation. W3C Recommendation, 30 April 2013.

Lefrançois, M. (2019). SAREF4SYST: An Extension of SAREF for Typology of Systems and Their Interconnections. ETSI Technical Specification TS 103 548. Version 1.1.2, ETSI Smart Applications REFerence ontology extension.

Li, C. and Petzold, F. (2025). Ontology-driven mixture-of-domain documentation: A backbone approach enabling question answering for additive construction. Buildings, 15(1):133.

Lim, S., Buswell, R. A., Le, T. T., Austin, S. A., Gibb, A. G. F., and Thorpe, T. (2012). Developments in construction-scale additive manufacturing processes. Automation in Construction, 21:262–268.

Mechtcherine, V., Bos, F. P., Perrot, A., Leal da Silva, W. R., Nerella, V. N., Fataei, S., Wolfs, R. J. M., Sonebi, M., and Roussel, N. (2020). Extrusion-based additive manufacturing with cement-based materials—production steps, processes, and their underlying physics: A review. Cement and Concrete Research, 132:106037.

Pauwels, P., Zhang, S., and Lee, Y. (2017). Semantic web technologies in aec industry: a literature review. Automation in Construction, 73:145–165.

Wu, P., Wang, J., and Wang, X. (2016). A critical review of the use of 3-d printing in the construction industry. Automation in Construction, 68:21–31.