Knowledge graph-governed enrichment for adaptive conceptual modeling environments: A Scene2Model platform extension

Alexander Voelz[1,*], Wilfrid Utz[1]

[1]OMiLAB NPO, Lützowufer 1, Berlin, 10785, Germany

Abstract

Scene2Model has recently been proposed as a platform for transforming co-creative design artifacts into adaptive conceptual models and machine-processable knowledge representations. However, adaptive conceptual modeling environments require mechanisms for dynamically extending semantic vocabularies, contextual relations, and reusable modeling assets while preserving semantic consistency and traceability. At the same time, large language model (LLM)-based extraction provides a flexible mechanism for acquiring candidate knowledge, while the resulting outputs require explicit governance before they can be integrated into project-specific vocabularies and relations. This paper presents a knowledge graph-governed enrichment framework for adaptive conceptual modeling environments in the context of the Scene2Model platform. The proposed framework combines LLM-based concept extraction with knowledge graph-informed concept resolution, application-level relation validation, rule-based relation suggestion, and targeted human review. The project knowledge graph serves as the semantic layer governing concept reuse, integration, and project knowledge evolution alongside the generation of reusable visual representations. The corresponding artifact is implemented as a modular service architecture using RDF-based graph management, curated relation rules, and versioned persistence. A construction-based scenario demonstrates the feasibility of the implemented workflow and illustrates the transition from a domain description to governed project knowledge and reusable Scene2Model resources.

Keywords

knowledge graphs, conceptual modeling, large language models, Scene2Model, digital design thinking

CEUR-WS.org/Vol-4313/kgmdse-paper3.pdf

1. Introduction

Conceptual modeling plays a central role in information systems engineering by enabling stakeholders to externalize domain knowledge into structured and machine-processable representations. Historically, conceptual models have been used to support communication, understanding, and analysis. In recent years, however, the increasing complexity of digital ecosystems has shifted their role towards reusable knowledge structures that support contextualization and reasoning. Consequently, knowledge graphs and ontology-based representations have gained importance in model-driven systems that require contextual reasoning and adaptive knowledge management. This is particularly relevant for conceptual modeling environments, where modeling constructs must evolve in response to changing requirements. Meanwhile, advances in artificial intelligence (AI), especially large language models (LLMs), have increased the need for structured representations that support downstream activities.

Within this context, Scene2Model has recently been introduced as a platform-oriented approach for transforming co-creative design artifacts into reusable knowledge representations in the form of conceptual models (cf. [1]). By enabling the transformation of informal design thinking artifacts into machine-processable conceptual structures, Scene2Model bridges human-centered design practices with model-driven development. The platform also supports the runtime adaptation of modeling constructs, allowing context-sensitive conceptualization across heterogeneous application domains and contributing to digital twins of design thinking.

Knowledge Graphs and Model-driven Systems Engineering (KGMDSE), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy

*Corresponding author.

alexander.voelz@omilab.org (A. Voelz); wilfrid.utz@omilab.org (W. Utz)

0000-0003-4580-1445 (A. Voelz); 0000-0002-6661-4668 (W. Utz)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-05

Beyond the transformation of artifacts into digital representations, adaptive modeling requires mechanisms that continuously integrate new concepts and relations while preserving semantic consistency and domain vocabularies over time. Existing modeling environments often rely on manually curated concept libraries and static vocabularies that are difficult to maintain in rapidly evolving domains. LLM-supported extraction can provide candidate concepts and contextual information from natural-language input, but these outputs are produced independently of the project-specific vocabulary required for controlled integration. These challenges become particularly critical when conceptual models are intended to function as reusable knowledge structures supporting interrelated operations (cf. [2]).

This paper addresses these challenges by presenting a knowledge graph-governed enrichment framework for adaptive conceptual modeling environments. A project-specific knowledge graph provides the evolving project representation used to resolve and integrate LLM-derived candidates. Following the neuro-symbolic tradition described in [3], LLMs support knowledge acquisition, while existing graph knowledge informs concept resolution and rule-based suggestions. Human review is incorporated at designated interaction points. The framework is implemented as a modular service-oriented extension of the Scene2Model platform, with the versioned project knowledge graph providing the central representation of integrated project concepts, relations, and modeling-resource associations.

Based on the adopted Design Science Research (DSR) approach, the remainder of this paper is structured as follows. Section 2 introduces the problem context and derives the design objectives, before Section 3 outlines the research methodology. Section 4 presents the proposed knowledge graph-governed enrichment framework together with its architectural realization. Subsequently, Section 5 demonstrates the artifact using a construction-based scenario and discusses limitations as well as directions for future research. Finally, Section 6 concludes the paper.

2. Background and research context

The proposed enrichment framework builds on recent advances in adaptive conceptual modeling, knowledge graph technologies, and AI-assisted information systems engineering. This section first motivates the need to adapt the resources of conceptual modeling environments, then discusses the relevant research foundations, and derives the design problem addressed by this work.

2.1. Problem context and motivation

Adaptive conceptual modeling environments should evolve alongside the application domains they represent. As new perspectives and requirements emerge, the underlying conceptual vocabulary must be extended without compromising consistency or interpretability [4]. The challenge thus concerns individual model adaptations and the reusable modeling resources on which the environment relies.

Recent developments in digital design thinking demonstrate how co-creative design artifacts can be transformed into machine-processable conceptual models that connect human-centered design practices with model-driven engineering [5]. However, while the resulting models can evolve, the resources underlying the modeling environment, such as concept libraries and project-specific visual assets, often remain largely static. Extending these resources requires manual engineering, asset curation, and synchronization across multiple representations, which limits adaptation in dynamic domains.

In the proposed setting, LLMs support the extraction of candidate concepts and contextual relations from textual input. These candidates are generated independently of the current project knowledge graph and are thus not automatically treated as project knowledge. Explicit concept-resolution steps are required to align candidates with the existing project knowledge, and extracted relation candidates must satisfy the supported predicate and category policies before integration.

These challenges motivate an enrichment process that combines the adaptability of LLM-assisted knowledge acquisition with an explicit, project-specific knowledge representation. The project knowledge graph provides the vocabulary and project context used for concept reuse and resolution, while explicit application-level policies check the admissibility of extracted relations before integration. Hu-

man review is incorporated at selected decision points, including document-based concept selection and the confirmation of rule-generated relation candidates.

2.2. Related work and theoretical foundations

This paper builds upon research in adaptive conceptual modeling, model-driven engineering, and semantic technologies. The following summarizes the foundations most relevant to the presented artifact, beginning with the underlying Scene2Model platform before discussing knowledge graphs and ontology-driven conceptual modeling as the semantic basis for adaptive enrichment.

Scene2Model and adaptive conceptual modeling Conceptual modeling commonly extends beyond its foundational purpose of supporting system understanding to the representation of machine-processable knowledge that supports subsequent system engineering activities [6]. At the same time, human-centered approaches such as haptic Design Thinking have become established for engaging diverse stakeholders within collaborative and co-creative workshops [7]. The resulting artifacts capture domain knowledge paired with a collective understanding of a given problem domain in explicit, physical form. However, such results typically remain isolated from subsequent model-driven development processes and require substantial manual effort to transform into formal conceptual models [8].

Scene2Model addresses this translation gap by enabling the transformation of haptic design artifacts into adaptive conceptual models within a model-driven engineering environment [5, 9]. Building on the ADOxx metamodeling platform, the approach captures physical workshop artifacts, such as storyboard-based design representations. It automatically transforms them into machine-processable conceptual models that can subsequently be refined and adapted to domain-specific requirements [10]. Scene2Model thereby establishes an adaptive modeling environment in which concepts, attributes, and representations can evolve throughout the modeling lifecycle.

Most recently, Scene2Model has been positioned as a user-centric platform that manages collaborative design knowledge across the preparation, runtime, and exploitation phases described in [1]. Across these phases, stakeholder knowledge is transformed into reusable conceptual representations that combine human creativity with computational processing. The continuous evolution of Scene2Model has further extended this perspective towards the notion of *Digital Twins for Haptic Design Thinking* [11]. Building on broader interpretations of digital twins as machine-processable representations that combine models, data traces, and purpose-specific services, Scene2Model captures knowledge externalized during collaborative workshops as reusable conceptual models, while providing services that support their semantic processing and refinement.

Knowledge graphs, ontologies, and model-driven systems engineering Knowledge graphs have emerged as a prominent paradigm for representing and managing structured knowledge by explicitly capturing concepts, entities, and their relationships in graph-based representations. Although the term is frequently used interchangeably with related notions such as ontologies or knowledge bases, it encompasses more than the representation of semantic structures alone [12]. An ontology provides a formal specification of a domain by defining concepts, relations, and axioms using semantic representation languages. A knowledge graph, in turn, combines such vocabulary with explicit entity and relation assertions and may be coupled with reasoning or validation mechanisms depending on the employed technology. In this work, the ontology provides the formal vocabulary underlying the representation, whereas the project knowledge graph contains the evolving project-specific concepts, relations, rules, and resource associations used during enrichment.

The combination of formal semantics and automated reasoning has motivated the integration of knowledge graphs into conceptual modeling and model-driven engineering [13]. Work on ontology-driven conceptual modeling demonstrated how ontologies can formalize modeling languages and improve semantic interoperability [14]. Building on these principles, subsequent research proposed mechanisms for transforming conceptual models into RDF-based knowledge representations to enable

querying, reasoning, and interoperability across modeling platforms [15]. Related efforts further extend conceptual models with linked semantic data to improve knowledge integration and reuse [16].

Beyond knowledge representation, recent work increasingly positions ontologies and knowledge graphs as distinct but complementary components of model-driven engineering processes [17, 18]. This development reflects a transition towards knowledge-centric engineering processes in which formal knowledge models support reasoning and continuous feedback throughout the system lifecycle [19, 20].

2.3. Problem statement

The previous sections have shown that adaptive conceptual modeling environments increasingly combine AI-assisted methods with ontology and knowledge graph technologies. Yet the modeling resources that define these environments, particularly their concept libraries, often remain static and require manual maintenance. Extending concept libraries thus continues to require manual effort, limiting the adaptability of conceptual modeling environments in evolving application domains.

For the design considered in this work, LLM-based extraction provides candidate concepts and contextual information from natural-language descriptions, while their subsequent integration remains a separate governance task. However, without explicit governance, their integration may introduce redundant concepts, inconsistent vocabularies, ambiguous relations, and limited traceability. For the present design problem, it is additionally important to distinguish relations extracted from source material from relation candidates proposed through explicit project rules.

From a DSR perspective, the design problem is to govern the continuous evolution of adaptive conceptual modeling environments through project-specific knowledge graphs. In addition to representing project knowledge, the graph governs concept integration, reuse, rule execution, and the generation of reusable modeling assets. Following the DSR formulation template proposed in [21], the resulting design problem can be summarized as:

> Improve the evolution of adaptive conceptual modeling environments (**problem context**)
> ... by introducing a knowledge graph-governed enrichment framework (**artifact**)
> ... that combines knowledge graph-informed concept resolution, asset reuse, procedural relation validation, and rule-based relation suggestion with LLM-assisted knowledge acquisition and targeted human review (**requirements**)
> ... to integrate new project knowledge while preserving consistency and both human and machine interpretability (**goal**).

2.4. Solution overview and research objectives

To address the design problem, the proposed framework extends the Scene2Model platform with adaptive capabilities for project-specific concepts and asset libraries. A project knowledge graph forms the central architectural element used to resolve new candidates against existing project knowledge, reuse established resources, and record integrated concepts, relations, and asset associations. LLM-based methods support candidate acquisition and mapping, while relation integration is governed through explicit predicate and category-compatibility policies.

Figure 1 illustrates how the proposed framework is integrated into the Scene2Model platform. Positioned primarily in the *Preparation* phase and also available during *Runtime* (cf. [9]), the extension aims to continuously enrich a project's modeling resources. Dedicated agents process knowledge from natural-language descriptions and supporting documents, while the project knowledge graph governs concept resolution, reuse, and controlled extension. The resulting project-specific Scene2Model library, containing concepts and associated visual assets, remains available for import into the Scene2Model environment throughout the project phases.

Following the DSR methodology, the objective is to realize a framework that enables controlled evolution of adaptive conceptual modeling environments. Accordingly, the proposed artifact aims to address the following research objectives:

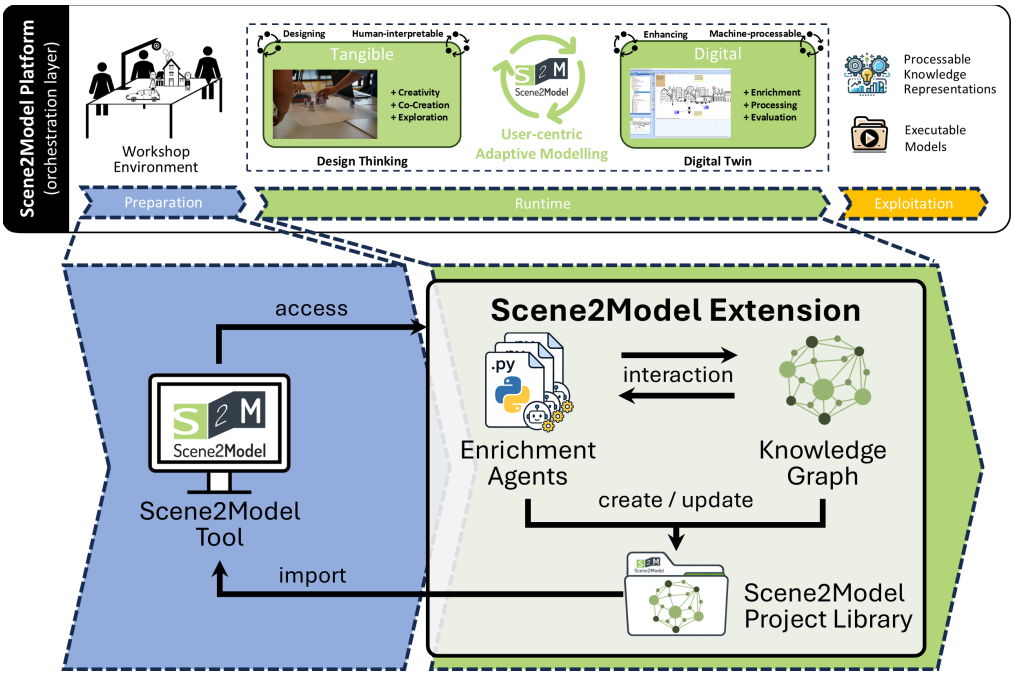


Figure 1: Conceptual overview of the proposed knowledge graph-governed enrichment framework, showing how it extends the adaptive conceptual modeling platform Scene2Model to support both the *Preparation* and *Runtime* phases (the Scene2Model Platform View is based on [9, p.70]).

- **Knowledge graph-governed:** each project maintains a project-specific knowledge graph that records project knowledge and governs enrichment decisions.
- **Adaptive:** newly acquired concepts, relation assertions, and project-specific modeling assets can be integrated as requirements change.
- **Vocabulary-grounded:** concept resolution reuses labels and aliases represented in the project knowledge graph before unresolved candidates are mapped against the available project vocabulary, while relation integration follows explicit predicate and category-compatibility policies.
- **Reusable:** integrated concepts and modeling assets are made available to Scene2Model for subsequent modeling activities.
- **Human-centered:** permanent changes require a transparent rationale and human validation.

The design principles derived from these objectives form the basis for the proposed knowledge graph-governed enrichment framework and its proof-of-concept implementation presented in Section 4.

3. Research method

This research follows the DSR methodology, which provides a structured approach for designing and demonstrating artifacts that address relevant design problems [21]. DSR is well-suited to this work, as the identified challenge concerns the design of a novel artifact that aims to enable the enrichment of adaptive conceptual modeling environments in the context of the Scene2Model platform.

According to the DSR process model [22], the research first identifies limitations of existing adaptive conceptual modeling environments (cf. Section 2.3) and corresponding design objectives (cf. Section 2.4). Based on these objectives, the artifact is designed as the proposed enrichment framework and realized as a Scene2Model platform extension that establishes the project-specific knowledge graph as the governance layer for all enrichment activities. The resulting artifact is subsequently demonstrated in a representative scenario and discussed with respect to controlled enrichment and machine interpretability.

Consequently, the remainder of this paper follows the underlying DSR process. Section 4 presents the framework as the proposed artifact, together with its design principles and architectural realization, while Section 5 provides its demonstration and a discussion of limitations and future research directions.

Table 1
Design principles derived from the research objectives in Section 2.4 and their realization within the proposed knowledge graph-governed enrichment framework.

Design Principle	Addresses Objective	Framework Realization
Knowledge Graph Governance	Knowledge graph-governed	Project knowledge graphs provide the versioned project state used for concept resolution and reuse, relation integration, rule application, and associations between concepts and modeling assets.
Controlled Project Knowledge Evolution	Adaptive; Reusable	Existing concepts and modeling assets are reused whenever possible, while new project concepts and relation integration are introduced through versioned project updates.
LLM-assisted Knowledge Acquisition and Vocabulary-Grounded Integration	Vocabulary-grounded; Adaptive	LLMs extract candidate concepts and contextual relations; candidate constructs are resolved against the project knowledge graph, while unresolved concepts are mapped against the project vocabulary.
Transparent, Human-Centered Enrichment	Human-centered	Versioned project states distinguish persisted relation assertions from non-persisted rule-generated candidates, while human review and confirmation are provided at designated workflow stages.
Versioned Consistency	Reusable	Project knowledge and reusable modeling assets are maintained through versioned persistence and explicit graph-library consistency operations.
Project-Specific Isolation	Adaptive; Knowledge graph-governed	Each project maintains its own knowledge graph, which is initialized from the shared Scene2Model base ontology and linked to its modeling assets.
Modular Platform Extension	Reusable	The enrichment framework is realized as an independent Scene2Model platform extension that can evolve without modifying the platform core.

4. Design and realization of the knowledge graph-governed enrichment framework

This section presents the knowledge graph-governed enrichment framework as the design artifact derived from the problem statement and research objectives in Sections 2.3 and 2.4. The framework extends Scene2Model with controlled evolution of project-specific concepts, relations, and reusable modeling assets. LLM-supported services acquire candidate knowledge, while a dedicated image-generation service produces visual assets. The project knowledge graph provides the project state used for concept resolution and integration, while explicit policies govern relation validation.

4.1. Design principles

The research objectives introduced in Section 2.4 are translated into the design principles summarized in Table 1. They define the architectural properties that guide the realization of the artifact.

These design principles separate LLM-assisted knowledge acquisition from the project-level decisions through which acquired candidates are reused, integrated, or transformed into modeling resources. Existing concepts and assets are reused where possible, while new project knowledge extends the versioned project knowledge graph and associated library resources. Relation integration is subject to explicit procedural checks, whereas rule-generated relations remain candidates until confirmation. Versioned persistence and explicit graph-library consistency operations support iterative project evolution, while project-specific isolation enables domain adaptation without modifying shared platform resources. Together, the principles in Table 1 establish a modular enrichment framework in which acquisition, governance, and resource generation remain explicitly separated.

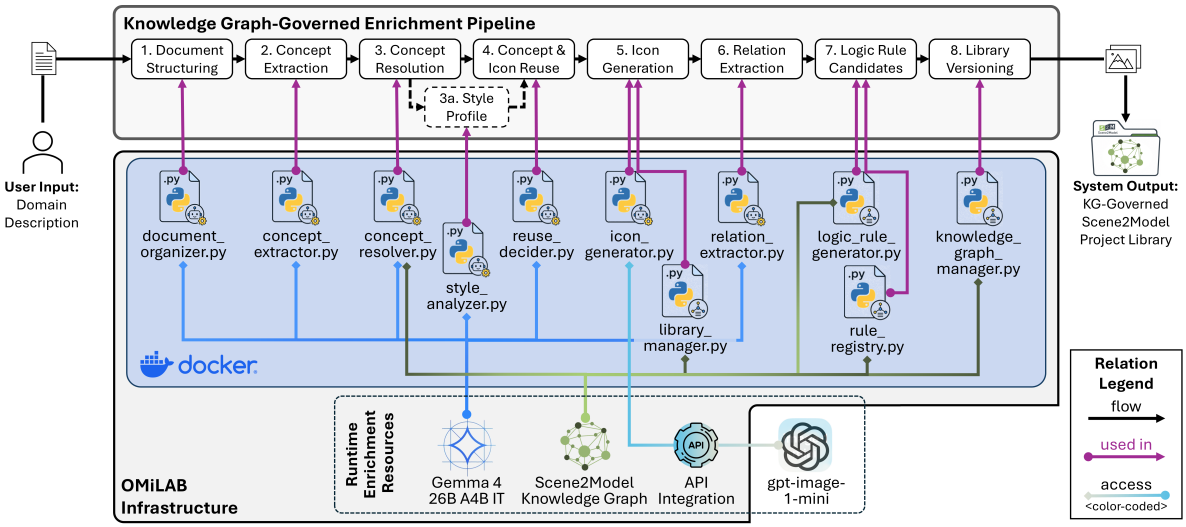


Figure 2: Architecture of the knowledge graph-governed framework, showing the interaction among modular agents, runtime resources, and the enrichment pipeline that produces project-specific Scene2Model libraries.

4.2. Architectural design

Figure 2 presents the architecture through which domain descriptions are transformed into project-specific Scene2Model resources. In accordance with the design principles (cf. Table 1), the architecture separates LLM-supported acquisition and generation from project-level resolution and application-level relation validation. These governance activities use the project knowledge graph as shared project state.

The upper part of Figure 2 presents the knowledge graph-governed pipeline that transforms domain descriptions into Scene2Model libraries. Document structuring first prepares the source material, after which concept extraction produces a list of candidate phrases, and resolution applies a matching strategy against the labels of project concepts. Candidates without an exact match are submitted to the language model together with the library context and project vocabulary. The response is a structured mapping record including a proposed concept label and category, with the label used to determine if an existing resource can be reused. If a new concept is integrated, the auxiliary style profile provides optional visual context for subsequent icon generation. LLM-based relation extraction and deterministic rule-based relation suggestion then produce candidates for governed project integration, after which persistent project resources are maintained through versioned graph and library management.

In the RDF representation, a resolved concept is persisted as an individual of `s2m:Concept` with an `rdfs:label` and an `s2m:belongsToCategory` link to a project category. Visual resources are represented separately as `s2m:Icon` individuals linked to their concepts through `s2m:representsConcept`. Relations are represented as explicit `s2m:RelationAssertion` resources that reference their subject, predicate, and object together with assertion metadata.

Relations are processed in two phases. First, relations explicitly expressed in the source text are extracted into subject-predicate-object candidates. Before persistence, the application checks that both endpoints are known project concepts, that the predicate belongs to the supported relation vocabulary, that the endpoint categories are compatible with the predicate, that the relation is non-reflexive, and that an equivalent assertion does not already exist. These checks are implemented procedurally at the application level. They do not interpret `rdfs:domain` or `rdfs:range` as closed-world validation constraints, and no OWL/RDFS reasoning or SHACL validation is performed. Second, curated rules stored in the project knowledge graph are applied to concept pairs whose categories match the rule conditions. Suggested relations remain non-persistent candidates until explicitly confirmed.

Each pipeline stage is implemented by an agent with a defined responsibility. This modular organization separates concerns such as document processing, concept acquisition and resolution, knowledge graph management, rule execution, and visual asset generation. Individual services can therefore be modified or replaced without changing the overall workflow or the Scene2Model platform core.

The agents access a shared set of runtime resources. Each project knowledge graph is initialized from the Scene2Model base ontology and combines the shared schema declarations with project-specific concepts, categories, relation assertions, rule records, icon associations, and version state. LLM-based services support acquisition tasks such as document structuring, concept extraction, candidate mapping, and relation extraction, but RDF persistence is performed by the application services. Most enrichment procedures are executed within the OMiLAB infrastructure (cf. [6]) using a locally deployed language model. Image synthesis is delegated to a dedicated external image generation service.

Overall, the architecture realizes the proposed design principles by combining LLM-supported acquisition services with knowledge graph-governed project management. Neural components propose candidate knowledge and visual resources, whereas existing project knowledge determines concept reuse and provides the context for subsequent integration. Persistent concepts, relation assertions, and resource associations are represented in the versioned project knowledge graph, while explicit application-level policies govern relation validation and rule-based candidate generation.

4.3. Proof-of-concept implementation

The framework was implemented as an independent Scene2Model platform extension using a modular service-oriented architecture. It is deployed as a containerized FastAPI application within the OMiLAB infrastructure. Each enrichment activity is encapsulated in an independent agent and exposed through a unified project API. Every project runtime maintains isolated resources, including its knowledge graph, generated modeling assets, metadata, and version history. The source code for the locally deployable proof-of-concept implementation is publicly available.[1]

The enrichment pipeline combines RDF-based knowledge management with task-specific LLM instructions. The text-processing services use *Gemma 4 26B A4B IT* through the OMiLAB-hosted chat-completions endpoint. In the prototype, the model is used for document structuring, concept extraction, candidate mapping, relation extraction, and conversational assistance. Each task uses a dedicated prompt and output-token limit.[2] The application does not explicitly transmit sampling parameters such as temperature, top-p, or a thinking-mode setting. Structured results are requested through task-specific prompt instructions, and the returned text is parsed by the application before subsequent processing. For example, concept extraction expects noun-phrase strings, whereas concept mapping expects records containing the source term, normalized term, proposed concept label, mapping status, and category. RDF resources and relation assertions are subsequently constructed by the application, so the language model does not generate RDF or modify the project knowledge graph directly.

The project knowledge graph records project concepts, relation assertions, rule records, and links to generated modeling assets, while versioned persistence supports their incremental evolution across enrichment runs. Visual asset generation is the only task delegated to an external service.[3]

Relation processing follows a two-phase pipeline. In the first phase, the language model extracts relations explicitly stated in the input text as structured triples. These records are normalized and checked for known concept endpoints, predefined category compatibility, self-relations, and duplicates before being persisted as asserted relations. The compatibility checks are implemented procedurally rather than through RDF/OWL entailment. In the second phase, curated rule records are applied deterministically to category-compatible concept pairs. The resulting relation suggestions are checked against the current project state but remain non-persistent candidates until explicitly confirmed.

The proof-of-concept implementation is integrated through Scene2Model's modular extension mechanisms and shared platform services without modifying the platform core. It therefore demonstrates the technical feasibility of implementing the knowledge graph-governed enrichment framework as a modular Scene2Model extension while preserving the underlying platform architecture.

[1] https://code.omilab.org/scene2model/development/scene2model-dynamic-library

[2] For the principal enrichment tasks, the configured output limits are 4,096 tokens for concept extraction and candidate mapping, 1,024 tokens for textual relation extraction, and up to 8,192 tokens for document structuring.

[3] For the demonstration, the GPT Image API with the *gpt-image-1-mini* model was selected because it consistently reproduced user-defined icon styles and supported transparent backgrounds relevant for integration into the given Scene2Model library.

Domain Description.

A family plans the construction of a house with a garage on an empty plot. The project involves the mother, father, child, and family EV, together with an architect, site manager, structural engineer, building inspector, construction workers, suppliers, and municipality representatives.

Planning and execution require a construction plan, the foundation, an actual construction site, tool warehouses, material warehouses, and a municipality's office.

Materials and equipment include brick pallets, concrete, beams, roof tiles, safety barriers, solar panels, an EV charging station, construction tools, robots, robotic arms, and inspection drones.

Smartphones, laptops, weather stations, and surveillance stations support the work. Relevant vehicles include concrete mixers, cranes, and material trucks, as well as tool vans and excavators.

Progress may be affected by safety risks, unavailable tools, material shortages, heavy rain, regulations, required inspections, or general requirements. Relevant weather conditions are thunderstorms and sunny weather.

Information and material flows coordinate activities.

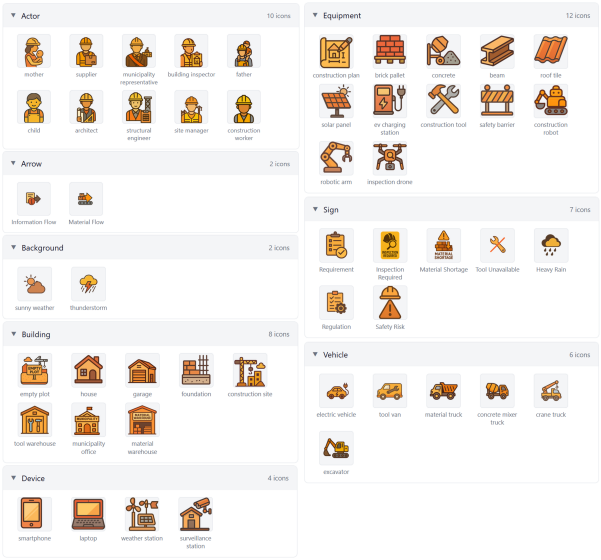
(a) Initial natural-language domain description for the *Construction-as-a-Service* scenario.



(b) Document-preview interface exposing the 51 extracted concept candidates for human review.

```
10 s2m:CR001 a s2m:CommonsenseRule,
11         s2m:LogicRule ;
12     s2m:ruleConfidence 0.8 ;
13     s2m:ruleId "CR001" ;
14     s2m:ruleObjectCategory "Vehicle" ;
15     s2m:rulePredicate s2m:drives ;
16     s2m:ruleRationale "actors often drive vehicles" ;
17     s2m:ruleSubjectCategory "Actor" .
   # ...
584 s2m:ConstructionWorkerConcept a s2m:Concept ;
585     rdfs:label "Construction Worker" ;
586     s2m:belongsToCategory s2m:Actor .
587
588 s2m:ExcavatorConcept a s2m:Concept ;
589     rdfs:label "Excavator" ;
590     s2m:belongsToCategory s2m:Vehicle .
   # ...
826 s2m:rel_00001 a s2m:RelationAssertion ;
827     s2m:confidence 0.8 ;
828     s2m:derivedFromRule s2m:CR001 ;
829     s2m:hasObject s2m:ExcavatorConcept ;
830     s2m:hasPredicate s2m:drives ;
831     s2m:hasSubject s2m:ConstructionWorkerConcept ;
832     s2m:sourceType "logic" ;
833     s2m:status "asserted" .
```

(c) Excerpt of the project knowledge graph showing the representative *construction worker drives excavator* relation, its governing rule, and the participating concepts.



(d) Resulting Scene2Model project library containing the 51 accepted concepts and their visual resources.

Figure 3: End-to-end demonstration of the knowledge graph-governed enrichment workflow for the *Construction-as-a-Service* scenario: (a) domain description, (b) human review of extracted concept candidates, (c) excerpt of the knowledge graph state and rule-generated relation, and (d) resulting project-specific Scene2Model library.

5. Demonstration and discussion

5.1. Artifact demonstration

To demonstrate its feasibility, the framework was applied during the 2026 edition of the annual NEMO Innovation Camp, which continues the educational Design Thinking setting established within the OMiLAB innovation ecosystem (cf. [23, 24]). A four-member OMiLAB team developed a scenario within the same one-week period allocated to the participating student teams. This provided a time-constrained setting in which the implemented enrichment workflow could be exercised.

The selected demonstration addressed a *Construction-as-a-Service* scenario. Figure 3 traces the initial enrichment run from the natural-language domain description through the intermediate review and governance stages to the resulting project-specific Scene2Model library. The document preview in Figure 3b displays 51 candidate concepts and presents their proposed categories, confidence values, and descriptions for review. The four-member team manually reviewed the complete candidate set and accepted all 51 before initiating their integration and visual asset generation.

At the start of this run, the project-specific library contained no domain-specific concepts for the demonstrated scenario. Consequently, the initial demonstration primarily exercised the extension branch of concept resolution in which no corresponding existing project concept was found for the extracted domain concepts, and new `s2m:Concept` resources were therefore introduced with their assigned categories. For example, *construction worker* and *excavator* were integrated as new concepts in the *Actor* and *Vehicle* categories, respectively, and corresponding visual resources were generated for the project library. The same concepts illustrate the subsequent relation-governance step shown in Figure 3c. The curated project rule specifies *drives* as a candidate relation for an *Actor-Vehicle* category combination and assigns a fixed confidence value of 0.8. Because *construction worker* and *excavator* satisfy these category conditions, rule application produced the candidate relation *construction worker drives excavator*. The candidate remained non-persistent until explicitly confirmed by the user.

After concept integration, the framework associated each accepted concept with a visual resource. Icons were generated based on the resolved concept label, category, and the specified visual theme and subsequently stored in the project library with a link to the corresponding concept in the project knowledge graph. The resulting library, shown in Figure 3d, contained the 51 accepted concepts and was imported into Scene2Model through the dedicated platform extension for use in the subsequent Design Thinking activities. The demonstration further included incremental enrichment during workshop execution. In contrast to the initially empty project state, these subsequent runs checked new terms against previously integrated concepts before introducing additional resources. Updated project libraries were versioned and made available to Scene2Model through the same extension mechanism.

The scenario serves as a feasibility-oriented artifact demonstration and does not provide an effectiveness evaluation. It was conducted within one construction-related setting and did not employ a formal evaluation instrument or experimental baseline for measuring preparation effort, usability, or performance relative to the previous manual process.

5.2. Discussion

The demonstration provides evidence of the technical feasibility of the proposed framework in the investigated setting. Starting from the domain description, the workflow produced 51 concept candidates that were manually reviewed and accepted. The resulting project concepts and visual resources were integrated into the project library, which was then made available within Scene2Model. The implementation further demonstrates that these capabilities can be realized as a modular Scene2Model extension without modifying the platform core. These observations establish end-to-end artifact operation in the demonstrated scenario and show that the framework automates selected preparation activities that would otherwise require manual identification or creation of suitable modeling assets. Because the demonstration did not employ formal measurements, it does not establish a validated reduction in overall preparation effort.

Nevertheless, the current implementation remains an initial proof of concept. The demonstration was limited to one scenario conducted by a four-member OMiLAB team and did not employ a formal evaluation instrument or comparative baseline. Consequently, the results do not establish scalability across domains or collaborative settings, nor do they quantify improvements in preparation effort or usability. Several limitations also arise from the current technical realization. Concept resolution returns a single mapping and does not yet expose ranked alternatives or an explicit ambiguity state for human disambiguation. Relation compatibility is checked through application-level predicate and category policies, but the prototype does not currently perform OWL/RDFS reasoning or SHACL-based constraint validation. Furthermore, knowledge graph evolution remains isolated to individual projects, and alignment between independently evolving project knowledge graphs is not yet supported. Visual asset generation currently depends on a commercial external service. The observed cost applies only to the presented scenario, and a single demonstration does not establish general cost efficiency.[4]

[4]During the demonstration, generating 51 icons using the theme *"Minimal flat vector icons with warm industrial colors and a construction-site aesthetic"* consumed 7,343 text input tokens and 17,464 image output tokens, corresponding to a total API cost of approximately $0.1544 under the pricing applicable at that time.

Future work will evaluate the framework across additional application domains, larger collaborative settings, and independent user groups. Such studies should employ explicit evaluation instruments and comparative baselines to assess preparation effort, resolution quality, and the usefulness of the generated modeling resources. On the technical level, future extensions will investigate ambiguity management in concept resolution, including ranked mapping alternatives and human-assisted resolution. The semantic layer can further be extended with OWL/RDFS reasoning where inferred type or hierarchy information can support enrichment decisions, and with SHACL or equivalent constraint mechanisms where explicit validation of graph structures is required. In addition, ontology matching and alignment techniques will be investigated to support the exchange of reusable knowledge between independently evolving project knowledge graphs (cf. [20]). Further interaction mechanisms may support more detailed review and provenance capture for acquired and rule-generated knowledge.

6. Conclusion

This paper presented a knowledge graph-governed enrichment framework for adaptive conceptual modeling environments that extends the Scene2Model platform with dynamically managed modeling libraries. Following a DSR approach, the artifact combines LLM-assisted knowledge acquisition with concept resolution, rule-based relation suggestion, and controlled integration through a project knowledge graph. It generates and manages reusable modeling assets, including icon libraries, while maintaining versioned project states. The proof-of-concept implementation demonstrates the technical feasibility of the proposed framework within an adaptive conceptual modeling workflow.

The contribution represents an initial proof-of-concept implementation of the proposed framework. The NEMO demonstration establishes the feasibility of executing the proposed enrichment workflow end-to-end within one educational construction scenario, but it does not constitute a comparative evaluation. Broader empirical studies with independent participants, explicit evaluation instruments, and additional application domains are therefore required. Interoperability between independently evolving project knowledge graphs likewise remains outside the scope of the present implementation. Future research will investigate ontology matching and alignment to support both interoperability between project knowledge graphs and ambiguity handling during concept resolution.

A further technical direction is the integration of OWL/RDFS reasoning for enrichment together with dedicated constraint-validation mechanisms where stronger formal governance is required. The central research result is the deliberate separation between neural services that provide adaptability through knowledge acquisition and the project knowledge graph that governs how the modeling environment evolves. In this regard, the enrichment framework lays the foundation for further extending the Scene2Model platform toward knowledge-driven conceptual modeling environments that combine the strengths of generative AI methods with explicitly governed knowledge representations to adapt continuously alongside their application domains.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5.6 Sol and Grammarly in order to: Check grammar and spelling; reformulate for clarity and word count reduction. Further, the authors used GPT-5.6 Sol for Figure 1 and Figure 2 in order to: Generate the Python-based agent file icon used within both figures. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Voelz, I. Vaidian, W. Utz, Scene2model as a platform for digital design thinking, in: Proceedings of the 34th International Conference on Information Systems Development (ISD 2026), Prague, Czech Republic, 2026. Forthcoming.

[2] R. Buchmann, J. Eder, H.-G. Fill, U. Frank, D. Karagiannis, E. Laurenzi, J. Mylopoulos, D. Plexousakis, M. Y. Santos, Large language models: Expectations for semantics-driven systems engineering, Data & Knowledge Engineering 152 (2024) 102324. doi:10.1016/j.datak.2024.102324.

[3] M. Hilario, An Overview of Strategies for Neurosymbolic Integration, Psychology Press, New York, 1997, pp. 13–35. doi:10.4324/9780203763667.

[4] A. Voelz, C. Muck, D. M. Amlashi, D. Karagiannis, Citizen-centric design of consumable services for smart cities, Digit. Gov.: Res. Pract. 4 (2023) 17. doi:10.1145/3597420.

[5] C. Muck, S. Palkovits-Rauter, Conceptualizing design thinking artefacts: The scene2model storyboard approach, in: D. Karagiannis, M. Lee, K. Hinkelmann, W. Utz (Eds.), Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools, Springer, Cham, 2022, pp. 567–587. doi:10.1007/978-3-030-93547-4_25.

[6] D. Karagiannis, R. A. Buchmann, W. Utz, The omilab digital innovation environment: Agile conceptual models to bridge business value with digital and physical twins for product-service systems development, Computers in Industry 138 (2022) 103631. doi:10.1016/j.compind.2022.103631.

[7] M. Pande, S. V. Bharathi, Theoretical foundations of design thinking – a constructivism learning approach to design thinking, Thinking Skills and Creativity 36 (2020) 100637. doi:10.1016/j.tsc.2020.100637.

[8] C. van der Lelie, The value of storyboards in the product design process, Personal and Ubiquitous Computing 10 (2006) 159–162. doi:10.1007/s00779-005-0026-7.

[9] C. Muck, A Dynamic Adaptation Framework for Haptic Modelling Methods: Creating Digital Twins from Design Thinking, Ph.D. thesis, University of Vienna, 2024. doi:10.25365/thesis.76296.

[10] A. Völz, C. Muck, W. Utz, Digital twins for haptic design thinking: An innovative prototype, in: C. M. Flath, G. Gust, F. Thiesse, A. Winkelmann (Eds.), Shaping the Digital Future Through Innovation and Practice, Springer, Cham, 2026, pp. 161–169. doi:10.1007/978-3-032-08489-7_12.

[11] D. Karagiannis, How digital twins for design thinking support innovative business models, The Kyoto Economic Review 198 (2024) 123–146. doi:10.57475/keizaironso.198.S9.

[12] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs, in: M. Martin, M. Cuquet, E. Folmer (Eds.), Joint Proceedings of the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), volume 1695, CEUR-WS.org, 2016.

[13] M. Verdonck, F. Gailly, S. de Cesare, G. Poels, Ontology-driven conceptual modeling: A systematic literature mapping and review, Applied Ontology 10 (2015) 197–227. doi:10.3233/ao-150154.

[14] G. Guizzardi, G. Wagner, J. P. A. Almeida, R. S. Guizzardi, Towards ontological foundations for conceptual modeling: The unified foundational ontology (ufo) story, Applied Ontology 10 (2015) 259–271. doi:10.3233/ao-150157.

[15] M. Smajevic, S. J. Ali, D. Bork, Cm2kgcloud – an open web-based platform to transform conceptual models into knowledge graphs, Science of Computer Programming 231 (2024) 103007. doi:10.1016/j.scico.2023.103007.

[16] D. Karagiannis, R. A. Buchmann, Linked open models: Extending linked open data with conceptual model information, Information Systems 56 (2016) 174–197. doi:10.1016/j.is.2015.10.001.

[17] E. Laurenzi, J. Allan, N. Campos, S. Stoller, An ontology-based meta-modelling approach for semantic-driven building management systems, in: Advanced Information Systems Engineering Workshops, Springer, Cham, 2024, pp. 200–211. doi:10.1007/978-3-031-61003-5_18.

[18] A. Chiș, A.-M. Ghiran, R. A. Buchmann, Graphxx: Dsml engineering for knowledge graph building and streamlining with graphrag, Software and Systems Modeling (2025). doi:10.1007/s10270-025-01297-y.

[19] R. A. Buchmann, Semantics-driven systems engineering: Requirements and prerequisites for a new flavor of model-driven engineering, in: H.-G. Fill, H. Kühn (Eds.), Metamodeling: Applications and Trajectories to the Future, Springer, Cham, 2024, pp. 19–34. doi:10.1007/978-3-031-56862-6_2.

[20] D. M. Amlashi, A. Voelz, J. Song, Dynamic adaption of metamodels based on knowledge graphs, in: J. Grabis, Y. Wautelet (Eds.), Advanced Information Systems Engineering Workshops, Springer, Cham, 2025, pp. 18–29. doi:10.1007/978-3-031-94931-9_2.

[21] R. J. Wieringa, Design Science Methodology for Information Systems and Software Engineering, Springer, Berlin, Heidelberg, 2014. doi:10.1007/978-3-662-43839-8.

[22] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, Journal of Management Information Systems 24 (2007) 45–77. doi:10.2753/mis0742-1222240302.

[23] D. Karagiannis, R. A. Buchmann, X. Boucher, S. Cavalieri, A. Florea, D. Kiritsis, M. Lee, Omilab: A smart innovation environment for digital engineers, in: L. M. Camarinha-Matos, H. Afsarmanesh, A. Ortiz (Eds.), Boosting Collaborative Networks 4.0, Springer, Cham, 2020, pp. 273–282. doi:10.1007/978-3-030-62412-5_23.

[24] A. Völz, I. Vaidian, Digital transformation through conceptual modeling: The nemo summer school use case, in: Modellierung 2024, Gesellschaft für Informatik e.V., Bonn, 2024, pp. 139–156. doi:10.18420/modellierung2024_014.