RWTH Aachen University
Department of Architecture
Chair of Design Computation

Ontology-Based Harmonization of Disparate Datasets for Early-Phase Building Life Cycle Assessment

Master's Thesis in the M.Sc. Program Construction and Robotics

Georgi Tsakov

Univ.-Prof. Dr. Jakob Beetz
Prof. Dr. Linda Hildebrand

communicated by Univ.-Prof. Dr. Jakob Beetz

Aachen, 10. June 2025

Table of Contents

Abstract

The construction sector significantly contributes to global carbon emissions. However, early-stage design tools rarely provide reliable life cycle assessment feedback. Although many ILCD-compliant repositories exist, semantic fragmentation impedes automated integration.

This thesis addresses three research questions: (1) how to formally model ILCDx Data as an ontology, (2) how to enrich it through automated classification and inference, and (3) how to improve querying for environmental data in the early design phases. A four-stage pipeline, comprising data normalization, hybrid regex/RAG categorization, ontology-based graph construction, and prototype demonstration, was developed using LinkML, SKOS, SHACL, and SPARQL.

Applied to 1,102 ready-mix concrete instances from ÖKOBAUDAT, IBU-Categories, EPDNorge, and The International EPD System, the pipeline achieved alignment rates of 94% and 87% for the two most structured repositories. The retrieval-augmented generation classifier achieved 93% top-50 accuracy on 100 manually labeled instances; regex-based rules achieved 100% precision in a spot check of 100 entries. Each instance generated approximately 2,800 RDF triples without blank nodes, and the prototype answered multi-criteria SPARQL queries in under 0.3 seconds across a 400-instance graph.

This work introduces a reusable semantic integration pipeline linking ÖKOBAUDAT, DIN 276, and BKI with ILCDx Data. A novel graph-native similarity metric supports automated selection of placeholder datasets, reducing manual environmental data search in concept design.

Limitations include a focus on ready-mix concrete, commercial GPT APIs for final classification, and SHACL scalability. Future work will target the transport module integration, extension to other materials, and embedding in BIM environments.

Zusammenfassung

Der Bausektor trägt erheblich zu den weltweiten Kohlenstoffemissionen bei. Allerdings liefern die Werkzeuge für die frühe Entwurfsphase nur selten ein zuverlässiges Feedback zur Ökobilanz. Obwohl viele ILCD-konforme Repositories vorhanden sind, erschwert die semantische Fragmentierung die automatische Integration.

Diese Arbeit befasst sich mit drei Forschungsfragen: (1) Wie können ILCDx-Daten formal als Ontologie modelliert werden, (2) wie können sie durch automatische Klassifizierung und Inferenz angereichert werden, und (3) wie die Abfrage von Umweltdaten in den frühen Entwurfsphasen verbessert werden kann. Unter Verwendung von LinkML, SKOS, SHACL und SPARQL wurde eine vierstufige Pipeline entwickelt, die Datennormalisierung, hybride Regex/RAG-Kategorisierung, ontologiebasierte Graphenkonstruktion und Prototypdemonstration umfasst.

Angewandt auf 1.102 Transportbeton-Instanzen von ÖKOBAUDAT, IBU-Categories, EPDNorge und dem International EPD System, erreichte die Pipeline Übereinstimmungsraten von 94% und 87% für die beiden am besten strukturierten Repositories. Der Retrieval-unterstützte Generierungsklassifikator erreichte eine Top-50-Genauigkeit von 93% bei 100 manuell beschrifteten Instanzen; Regex-basierte Regeln erreichten eine Genauigkeit von 100% bei einer Stichprobenprüfung von 100 Einträgen. Jede Instanz generierte etwa 2.800 RDF-Tripel ohne leere Knoten, und der Prototyp beantwortete multikriterielle SPARQL-Anfragen in weniger als 0,3 Sekunden über einen 400-Instanzen-Graphen.

In dieser Arbeit wird eine wiederverwendbare semantische Integrationspipeline vorgestellt, die ÖKOBAUDAT, DIN 276 und BKI mit ILCDx-Daten verknüpft. Eine grafennative Ähnlichkeitsmetrik, die eine automatische Auswahl von Platzhalterdatensätzen ermöglicht und eine Grundlage für die Reduzierung der manuellen Umweltdatensuche in frühen Design-Workflows bietet.

Zu den Einschränkungen gehören der Fokus auf Transportbeton, kommerzielle GPT-APIs für die endgültige Klassifizierung und die Skalierbarkeit von SHACL. Zukünftige Arbeiten werden die Integration des Transportmoduls, die Erweiterung auf andere Materialien und die Einbettung in BIM-Umgebungen zum Ziel haben.

Abbreviations

AEC	Architecture, Engineering and Construction (sector)	**LCI**	Life Cycle Inventory
API	Application Programming Interface	**LCIA**	Life Cycle Impact Assessment
BIM	Building Information Modeling	**LD**	Linked Data
BKI	*Baukosteninformationszentrum Deutscher Architektenkammern* (German Building-Cost Information Centre)	**LinkML**	Linked Data Modeling Language
BNB	*Bewertungssystem Nachhaltiges Bauen* (German Assessment System for Sustainable Building)	**LLM**	Large Language Model
CEN	European Committee for Standardization	**OWL**	Web Ontology Language
CG	Cost Group (DIN 276)	**PCR**	Product Category Rule
DIN	*Deutsches Institut für Normung* (German Institute for Standardization)	**PENRT**	Primary Energy Non-Renewable Total
EPD	Environmental Product Declaration	**RAG**	Retrieval-Augmented Generation
FAISS	Facebook AI Similarity Search	**RDF**	Resource Description Framework
FN	False Negative	**RQ**	Research Question
FP	False Positive	**SHACL**	Shapes Constraint Language
F_1	Harmonic mean of precision and recall	**SKOS**	Simple Knowledge Organization System
GWP	Global Warming Potential	**SPARQL**	SPARQL Protocol and RDF Query Language
IFC	Industry Foundation Classes	**SQL**	Structured Query Language
ILCD	International Life Cycle Data system	**TP**	True Positive
ILCDx	ILCD schema + EPD extension (all ILCD-compatible data)	**TIES**	The International EPD System
ISO	International Organization for Standardization	**URI**	Uniform Resource Identifier
JRC	Joint Research Centre (European Commission)	**WP**	Work Package
JSON	JavaScript Object Notation	**XML**	Extensible Markup Language
LCA	Life Cycle Assessment	**YAML**	YAML Ain’t Markup Language

Glossary

Core ILCDx Terms

ILCDx Data A general term used throughout this thesis to refer to any dataset encoded in the standard ILCD schema or its ILCD+EPD extension. ILCDx Data encompasses verified Environmental Product Declarations (EPD Data) and unverified Life Cycle Assessment (LCA Data).[1]

[1] The "ILCDx" label is adopted to avoid confusion with the narrower term "ILCD Data" defined in the ILCD Handbook, which excludes the ILCD+EPD extension.

EPD Data A subset of ILCDx Data representing manufacturer-specific, third-party verified Environmental Product Declarations prepared according to ISO 14025 and EN 15804 (e.g., specific).

LCA Data A subset of ILCDx Data representing generic, average, representative, or template datasets (e.g., average).

ILCDx Dataset A source repository containing ILCDx Data, such as ÖKOBAUDAT, IBUCategories, EPDNorge, or The International EPD System.

ILCDx Data Instance An individual XML or JSON file within an ILCDx Dataset, representing a single EPD Data or LCA Data record.

Supplemental Terms

Building Information Modeling (BIM) A methodology and family of software tools that create and manage a shared digital representation of a building's physical and functional characteristics across its life cycle.

Baukosteninformationszentrum Deutscher Architektenkammern (BKI) The Building Cost Information Centre of the German Chambers of Architects. BKI provides standardized cost benchmarks and template building elements to support budgeting and cost planning in the AEC sector.

Closed-World Assumption A logical assumption in which any statement not known to be true is considered false. Common in engineering and data validation contexts, it underpins SHACL reasoning for deterministic classification and constraint enforcement.

DIN 276 Cost Classification (DIN 276) The German standard (DIN 276-1:2018) defines a hierarchical cost group (CG) system for building and infrastructure projects, widely used for early-phase budgeting and cost estimation.

Facebook AI Similarity Search (FAISS) An open-source library for efficient similarity search over vector embeddings.

Industry Foundation Classes (IFC) The ISO 16739-1 open, object-oriented data model used to exchange BIM information across software platforms.

Knowledge Graph A machine-readable data graph, typically based on RDF, that instantiates one or more ontologies to represent real-world entities and their relationships. Enables semantic querying, linking, and inference.

Linked Data Modeling Language (LinkML) A declarative schema language that compiles to RDF, SHACL, and other artefacts, enabling schema-driven data modeling and validation.

Ontology A formal schema that defines the classes, properties, and constraints used to describe a domain of knowledge. Ontologies provide the semantic backbone for knowledge graphs by enforcing structure and logical consistency.

openBIM A project-delivery approach that promotes vendor-neutral, standards-based workflows (e.g., IFC, BCF, mvdXML) to ensure long-term interoperability of BIM data.

Product Category Rule (PCR) A document under ISO 14025 and EN 15804 that defines calculation rules, scenarios, and data requirements to ensure consistent EPDs for comparable products.

Retrieval-Augmented Generation (RAG) A hybrid technique that first retrieves candidate information via vector search and then uses a language model to generate or select the final answer.

Semantic Alignment The process of mapping equivalent or related concepts across heterogeneous vocabularies or datasets to ensure consistent interpretation and integration.

Semantic Enrichment The process of enhancing raw data with additional structured information, such as classifications, inferred properties, or aligned vocabularies, to improve interoperability, queryability, and reasoning.

Shapes Constraint Language (SHACL) A W3C standard for validating RDF graphs and expressing rules over data structures.

Simple Knowledge Organization System (SKOS) A lightweight RDF vocabulary for modeling classification systems, taxonomies, and controlled vocabularies.

SPARQL Protocol and RDF Query Language (SPARQL) A declarative query language for RDF data. SPARQL enables flexible, multi-criteria retrieval over the knowledge graph, including filtering, aggregation, and statistical ranking.

Taxonomy A structured classification system that organizes concepts into hierarchical relationships, typically from general to specific.

Evaluation Metrics

True Positive (TP) A correct positive prediction. In this thesis, an ILCDx Data Instance is a true positive when the pipeline assigns the correct category, CG, or alignment link.

False Positive (FP) An incorrect positive prediction. A false positive occurs when the pipeline assigns a label or link that does not match the ground truth.

False Negative (FN) A missed positive case. A false negative occurs when the correct label or link exists, but the pipeline fails to assign it.

Precision The share of correct predictions among all positive predictions:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

High precision indicates few false positives.[2]

[2] For example, a precision of 1.00 means every predicted label was correct.

Recall The share of relevant items correctly predicted:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

High recall indicates few false negatives.[3]

[3] Recall is sometimes called "sensitivity" in classification contexts.

F_1**-Score** The harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

It balances both error types in a single metric.[4]

[4] An F_1 score of 1.00 indicates perfect precision and recall.

1 Introduction

1.1 Background

The construction and building sector remains one of Europe's most resource- and emission-intensive sectors. It accounts for approximately 40% of final energy consumption and 36% of energy-related greenhouse gas emissions, while 75% of buildings are still considered energy-inefficient under current standards (EU, 2024). While regulatory frameworks such as the European Union's Energy Performance of Buildings Directive have historically prioritized operational energy reductions, further climate gains increasingly depend on addressing the full life cycle of materials and structures (Röck et al., 2020). In highly energy-efficient buildings, embodied GHG emissions can account for 45–50% of life-cycle GHG emissions and may surpass 90% in extreme cases (Röck et al., 2020).

In response, national and international policy frameworks, including the EU's "Fit for 55" strategy and Germany's commitment to climate neutrality by 2045, establish the broader decarbonization context for the building sector (Bundesregierung, 2021; EU, 2025). Within this context, LCA provides a standardized methodological basis for measuring environmental impacts across product and building life cycles (CEN, 2011; ISO, 2006b).

To support consistent reporting and digital exchange, standardized data formats, such as the International Life Cycle Data system (ILCD) and its ILCD+EPD extension, have been developed (BMWSB, 2023; JRC, 2010). These provide the technical foundation for structuring and comparing environmental data across regions and applications, and form the common baseline for this thesis.

1.2 Problem Definition

Despite these advances, the integration of LCA into early-phase building design remains limited (Bahlau et al., 2024; Meex et al., 2018). At this stage, when core decisions about materials and systems are made, practitioners often lack access to structured environmental data in a form suitable for design tools (Bahlau et al., 2024). Instead, existing BIM-LCA workflows remain complex and time-consuming, with substantial manual effort and limited automation (Lambertz et al., 2020). In the repositories analyzed in this thesis, this limitation is reinforced by fragmented classification schemes, inconsistent metadata, and limited support for cross-repository querying. These issues reduce the timeliness and traceability of LCA results and diminish their influence on early design choices.

A key barrier is the semantic fragmentation of ILCD-compliant datasets. Although ILCDx Data repositories all follow the ILCD schema, they diverge significantly in internal taxonomies, language support, and metadata structure. As a result, critical attributes, such as material classification,

module coverage, or compressive strength, are often missing, inconsistently formatted, or not comparable across sources. These gaps complicate integration, reduce data interoperability, and hinder the deployment of automated classification and reasoning techniques.

1.3 Research Questions

The present work is structured around three core research questions (RQs):

1. **RQ1 – Ontology Modeling**
   How can disparate ILCDx Datasets be modeled as an ontology to support semantic interoperability?
2. **RQ2 – Semantic Enrichment**
   What automated methods can semantically enrich an ILCDx knowledge graph with ÖKOBAUDAT material categories, DIN 276 cost groups (CGs), BKI element classifications, and material property classes?
3. **RQ3 – ILCDx Data Querying**
   How can a semantically enriched ILCDx knowledge graph improve the retrieval of ILCDx Data in early-phase building design?

1.4 Research Design

To address these questions, this thesis employs a modular research design organized around a semantic modeling pipeline for ILCDx Data. The methodology consists of four work packages (WPs), each addressing a distinct stage of the harmonization process.

1. **WP1 – Data Collection and Analysis** (supports all RQs)
   WP1 analyzed four ILCDx Datasets: ÖKOBAUDAT, IBUCategories, EPDNorge, and The International EPD System (TIES), focusing on attribute availability, structural completeness, language coverage, and classification system heterogeneity. This diagnostic phase established the baseline conditions for semantic alignment and informed the design of downstream enrichment steps.
2. **WP2 – Automated Categorization** (addresses RQ2)
   WP2 developed a hybrid pipeline combining symbolic rule-based methods (regular expressions) and retrieval-augmented generation (RAG) using language models to address the lack of standardized material and cost-group classifications. This pipeline enabled consistent mapping of ILCDx Data Instances to ÖKOBAUDAT material categories and DIN 276 CGs. Attribute alignment and evaluation procedures ensured schema compatibility and traceability.
3. **WP3 – Ontology and Knowledge Graph Construction** (addresses RQ1 and RQ2)
   WP3 formalized the harmonized ILCDx Data into a modular ontol-

ogy using the LinkML modeling language. Simple Knowledge Organization System (SKOS) vocabularies were generated to represent external classification hierarchies, while Shapes Constraint Language (SHACL) rules were authored to infer material classes based on numeric properties such as compressive strength and bulk density. The result was a knowledge graph designed for queryability and semantic reasoning.

4. **WP4 – Prototype Implementation** (addresses RQ3)
   WP4 demonstrated the practical utility of the knowledge graph through a working prototype. An interactive Streamlit interface was developed to enable structured SPARQL querying, including multi-criteria filtering and statistical similarity-based recommendation. The prototype was evaluated in its support for early-phase design workflows and responsiveness under various query modes.

1.4.1 Technologies and Tools

The research made use of the following technologies, each selected for alignment with semantic modeling best practices and the thesis requirements:

- **LinkML** for schema-driven ontology modeling.
- **SKOS** for taxonomy alignment across heterogeneous classification systems.
- **SHACL** for closed-world validation and inference over material properties.
- **SPARQL** for declarative querying of the semantic graph.
- **Open-source embedding models, OpenAI GPT, and FAISS** for implementing the RAG classification pipeline.
- **Streamlit and Apache Jena Fuseki** for the prototype user interface and SPARQL endpoint.

1.5 Importance and Relevance of the Research

Building on the semantic pipeline outlined above, this thesis demonstrates how graph-based data models can enable sustainability feedback during early-stage building design. By transforming fragmented ILCDx Data into a harmonized knowledge graph, the approach bridges the gap between environmental repositories and planning abstractions such as material categories, CGs, and property filters.

The resulting prototype enables placeholder selection, semantic classification, and dynamic filtering, capabilities currently lacking in early-phase LCA tools despite being essential for rapid, sustainability-informed decisions. Its interface and automation features aim to reduce manual effort, improve traceability, and support informed choices when design flexibility is highest.

At a technical level, the contribution lies in combining symbolic and statistical classification methods with schema-driven ontology modeling. The implementation relies on open standards and widely adopted open-source tools. This foundation supports reproducible modeling, consistent classification, and standards-compliant querying.

While the implementation scope was limited to ready-mix concrete, the architecture was designed for generalizability and supports extension to additional material domains, classification systems, and design phases. The use of SPARQL for deterministic filtering and in-graph recommendation further showcases the expressive potential of knowledge graph architectures for environmental data applications.

This work aligns with ongoing digitalization and sustainability agendas in the architecture, engineering, and construction (AEC) sector. It contributes to openBIM workflows and promotes semantic technologies compatible with standardization frameworks such as DIN SPEC and buildingSMART. It further advances the integration of environmental reasoning into early design processes.

2 Literature Review

This chapter surveys the conceptual and technical landscape that frames the thesis. It is structured into four thematic areas, each corresponding to a WP and RQ. Each theme reviews the state of the art, identifies key limitations, and motivates the semantic strategies developed in this work.

- **§ 2.1 Conceptual Foundations** outlines the regulatory standards and technical vocabularies governing LCA and EPDs, including ISO 14040, EN 15804, ILCD, and ÖKOBAUDAT.
  Highlights standardization gaps and structural inconsistencies across repositories.
- **§ 2.2 BIM–LCA Integration** examines integration efforts in BIM workflows, particularly in Germany. Existing approaches remain limited to static file-based exchanges and late-phase certification contexts.
  Exposes the absence of flexible, early-phase connections between BIM and environmental datasets.
- **§ 2.3 Ontology-Based Semantic Interoperability** reviews how ILCDx Datasets can be modeled as Linked Data (LD) graphs. The section evaluates schema modeling tools, data transformation strategies, and semantic alignment techniques.
  Clarifies the trade-offs in modeling ILCDx Data as an interoperable ontology.
- **§ 2.4 Automated Semantic Categorization** surveys hybrid classification approaches that combine symbolic pattern matching with embedding-based retrieval and language model reasoning.
  Identifies scalable methods for enriching sparse metadata and structuring ILCDx Data Instances.

2.1 Conceptual Foundations

2.1.1 Life-Cycle Assessment and EPD Standards

LCA is a standardized methodology for quantifying environmental impacts across all stages of a product or service, from raw material extraction through manufacturing, use, and end-of-life disposal (ISO, 2006b). It is structured into four phases, as defined in ISO 14040 and ISO 14044: goal and scope definition, life cycle inventory (LCI), impact assessment (LCIA), and interpretation (ISO, 2006b, 2006c).

To harmonize data exchange and ensure methodological consistency, the European Commission's Joint Research Centre developed the ILCD, a structured data model for LCA datasets (JRC, 2010). Its extension, the ILCD+EPD format, supports Environmental Product Declarations that are prepared according to ISO 14025 and EN 15804 (BMWSB, 2023; CEN, 2012; ISO, 2006a). These standards provide the foundation for machine-readable, transparent, and comparable environmental data.

In the built environment, LCA assessments of buildings are further guided by EN 15978, which defines lifecycle modules ranging from product manufacturing (A1–A3) to end-of-life scenarios (C1–C4) and recovery (D) (CEN, 2011). This modular logic is mirrored in EPDs, which are third-party verified and follow predefined Product Category Rules (PCRs) to ensure consistency. The resulting declarations enable comparative analysis and support integration into design-phase tools and regulatory frameworks such as Germany's Assessment System for Sustainable Building (BNB) and Sustainable Building Certification (QNG) certification schemes (BNB, 2025; QNG, 2024).

ILCDx Data is disseminated through multiple national and international repositories, each with distinct technical structures, classification schemes, and access policies. ÖKOBAUDAT, maintained by the German Federal Ministry for Housing, Urban Development and Building (BMWSB), provides official datasets for regulatory LCA assessments under the BNB certification system (BBSR, 2024). The Institut Bauen und Umwelt (IBU) publishes over 300 third-party verified EPDs on behalf of manufacturers (IBU, 2024).[5] EPDNorge, Norway's national EPD program, participates in mutual recognition agreements across Europe (EPD Norge, 2025), while TIES, managed by EPD International AB, offers broad global coverage with data from over 400 organizations in nearly 50 countries (International EPD System, 2025).

[5] IBU's material category classification system used in the thesis as a reference to the IBU dataset is IBUCategories.

Although these platforms conform to the ILCD format, they diverge significantly in classification logic, metadata conventions, and technical implementations, creating substantial barriers to semantic interoperability. These discrepancies affect both regulatory compliance and practical implementation. For instance, ÖKOBAUDAT is the mandatory database for LCA calculations in the BNB assessment system (BBSR, 2024). In contrast, QNG permits both generic and product-specific ILCDx Data, under the condition that datasets originate from a specified ÖKOBAUDAT version or fulfill its inclusion principles, ensuring consistency, neutrality, and comparability (QNG, 2024). In practice, however, the information and data procurement required for building LCA remains unstructured, which hampers automation and delays early-phase environmental feedback (Bahlau et al., 2024).

Beyond these repository-level discrepancies, ILCDx Data Instances themselves exhibit significant metadata variability. Key fields differ in structure, language, and granularity, not only across datasets but even within a single source. Optional fields are often missing or inconsistently populated, especially in generic or programmatically generated entries. This heterogeneity hinders automated classification and downstream reasoning, as string-based heuristics fail to generalize across formats. Addressing these challenges requires robust semantic enrichment methods, such as embedding-based retrieval (§ 2.4), which can tolerate noisy inputs while enabling alignment with structured taxonomies.

2.1.2 Cost Classification in German Planning Practice

While ILCD standards provide a modular structure for environmental data, they are not aligned with the way design-phase decisions are typically made (Bahlau et al., 2024; Meex et al., 2018). In practice, LCA-based environmental assessment is still often not used to support or optimize design decisions during early design stages (Meex et al., 2018). To enable sustainability assessments that are actionable during early-phase design, environmental data must be mapped to familiar cost classification systems.

Architectural cost planning in Germany is formally structured by DIN 276, which organizes building costs into hierarchical CGs from site acquisition (CG 100) to financing (CG 800) (DIN, 2018). CG 300 (Structure — Construction works) plays a central role in early design, where material choices influence both cost and embodied carbon outcomes (Schneider-Marin et al., 2022). Linking ILCDx Data Instances to CG 300 categories allows planners to evaluate not just environmental impact but also budget relevance, even when quantity takeoff data is incomplete.

To support finer-grained assessments, early-phase design workflows often require realistic placeholder materials when material choices remain uncertain (Zong et al., 2022). The BKI Konstruktionsatlas KA2, published by the Baukosteninformationszentrum Deutscher Architektenkammern (BKI), provides component-level construction templates with layer structures, cost data, and environmental indicators (BKI, 2024). Mapping ILCDx Data to these elements enables reasoning at the assembly level, a level of abstraction more specific than DIN 276 CGs but still generalized enough for conceptual planning.

This thesis adopts a strategy that integrates DIN 276 CG 300 and selected BKI elements (WP3), enabling rule-based classification and structured querying (WP2, WP4). By aligning ILCDx Data with standard planning taxonomies, the resulting knowledge graph supports multi-criteria filtering, semantic enrichment, and early design feedback, core requirements for scalable BIM–LCA integration.

2.1.3 Semantic Web and Linked Data Fundamentals

Having established data standards and cost-planning vocabularies, the next step is to review the Semantic Web technologies used to integrate them. The Semantic Web offers a stack of technologies for expressing, linking, and validating heterogeneous data, addressing the interoperability challenges outlined in § 2.1.1 and § 2.1.2. At its foundation, the Resource Description Framework (RDF) models information as subject–predicate–object triples, enabling knowledge graphs that are both machine-readable and human-interpretable (Schreiber & Raimond, 2014). This structure is extended by SKOS, which provides mechanisms for defining concepts, hierarchical relationships, and multilingual labels (Miles & Bechhofer, 2009).

SPARQL serves as a declarative query language for traversing RDF graphs,

supporting flexible, federated access to distributed data sources (Harris & Seaborne, 2013). Of particular relevance in structured engineering contexts, SHACL supports constraint checking, enabling validation of both data structure and constraint consistency (Knublauch & Kontokostas, 2017).

The SHACL Advanced Features extension further introduces rule-based reasoning via *sh:SPARQLRule*, enabling inference directly within shape definitions (Knublauch et al., 2025). This allows declarative enrichment logic, such as material classification rules, to be co-located with schema constraints in a maintainable form.

These technologies are grounded in the LD principles formulated by (Berners-Lee, 2006): use URIs to identify entities, make those URIs dereferenceable, use RDF to describe them, and link to other URIs to provide context. As Bizer et al. (2023) emphasizes, this architecture promotes modular, scalable data integration across domains and systems.

Within the construction sector, Beetz et al. (2009, 2018) propose formalizing traditional building classifications (e.g., ISO 12006) using SKOS or OWL to enable semantic annotation in BIM workflows. While OWL-based reasoning lies outside the scope of this thesis, SKOS and SHACL offer a more lightweight and tractable framework for modeling domain vocabularies and validation constraints (Beetz et al., 2021).

Finally, newer schema languages such as LinkML provide a declarative, YAML-based framework that compiles to RDF and related schema formats (Moxon et al., 2021). This simplifies ontology engineering by allowing domain experts to model structured data without requiring deep knowledge of SPARQL or OWL, while still generating standards-compliant graph representations.

In summary, the Semantic Web stack provides a layered infrastructure for creating, validating, and querying distributed knowledge graphs. This thesis applies these principles to environmental data by modeling ILCDx Data Instances as interoperable graph-based entities.

2.2 BIM–LCA Integration

Building Information Modeling (BIM) is frequently presented as a centralized, digital source of truth encompassing geometrical, semantic data across a built structure's life cycle. When integrated with LCA, it offers the potential for environmental feedback to be accessed directly within the design environment, ideally alongside quantities and cost data. Achieving this integration requires BIM authoring tools to establish unambiguous links between model elements and environmental data (e.g., ILCDx Data), and maintain these links dynamically as the design evolves.

In research, five general BIM–LCA integration patterns are commonly identified (Lambertz et al., 2020): (a) use existing Industry Foundation

Classes (IFC) environmental-impact property sets, (b) introduce new custom property sets, (c) referencing external LCA files via multi-model containers, (d) *direct API-based linking* to ÖKOBAUDAT, and (e) embedding environmental data directly within authoring-tool objects. These approaches form the framework for assessing practical implementations.

2.2.1 Limitations of Late-Phase Workflows

Lambertz et al. (2020) evaluated the integration patterns above in the context of late-phase certification workflows. Among these, option (d), *direct API-based linking*, was considered technically promising, as it avoids embedding environmental data directly within IFC files. Instead, model elements store a UUID that resolves to ÖKOBAUDAT entries at runtime. However, the authors conclude that even this method remains oriented toward the later project phases where quantities are finalized and LCA results are prepared for certification (e.g., BNB reporting). For earlier design stages, where rapid iteration is essential, current workflows still rely on manual material mapping, introducing data drift and compromising traceability.

2.2.2 Early-Phase Priorities

The Digital Twin Footprint roadmap by Bahlau et al. (2024) shifts attention to early-phase design, outlining stakeholder priorities across a phased implementation timeline. Among the recommended short-term measures are improvements to IFC export quality to preserve BIM–LCA linkages, support for dynamic reference datasets, and enhancements to the ÖKOBAUDAT REST API, including the addition of versioning features. These steps effectively mirror the API-based integration approach (Option d) proposed by Lambertz et al. (2020) but refocus it on the demands of early-phase design.

Beyond these technical actions, the study identifies deeper infrastructural gaps. The absence of interoperable taxonomies, specifically, the need to restructure ÖKOBAUDAT classifications for seamless BIM integration, emerges as a primary barrier. Related concerns include the lack of standardized identifier policies (e.g., persistent UUIDs in IFC exports) and the absence of systematic feedback mechanisms to reintegrate LCA results into BIM environments and archive them for future traceability.

The authors conclude that environmental information cannot be effectively utilized during iterative design stages without unified identifiers and dynamic data linkage. These conclusions align with broader calls for semantic infrastructure that enables adaptable classification systems and traceable data management across the early design phases.

2.3 Ontology-Based Semantic Interoperability

This section reviews the state of the art in modeling environmental data as LD knowledge graphs. It frames RQ1 and underpins the ontology and graph work delivered in WP3.

2.3.1 Semantic Catalogs and Modular Ontologies for LCA

Early work on semantic integration of life LCI data focuses on constructing LD catalogs that expose common metadata elements across databases. Kuczenski et al. (2016) introduce a prototype schema that unifies U.S. LCI, GaBi, ELCD, and ecoinvent datasets using JSON-LD and a shared vocabulary of three core entities, Activity, Flow, and Flow Quantity and linking quantities to QUDT for unit semantics.

However, the approach purposefully makes only minimal ontological commitments. The schema lacks support for ILCD-specific modules, offers no modeling of methodological assumptions, and does not align with domain-specific vocabularies. As a result, while useful for cataloging, it falls short of enabling deeper integration into regulatory or planning workflows.

A similar concern is raised by Janowicz et al. (2015), who highlight that widely used LCA Data formats such as ILCD and Ecospold, despite their structural rigor, lack formal semantics. As a result, they suffer from inconsistent terminology usage, limited machine interpretability, and insufficient support for reproducible integration. To address this semantic gap, they propose a minimal ontology pattern for LCA, focusing on a shared conceptual core of Activity, Flow, Agent, and Reference Product entities. The pattern, while intentionally lightweight, is designed to promote semantic comparability across heterogeneous LCI datasets and to serve as a foundation for future extensions.

To address these limitations, later work has shifted toward modular ontologies as a strategy for more robust semantic integration and querying in Life Cycle Sustainability Assessment (LCSA). Ghose et al. (2022), for instance, use RDF and SPARQL to integrate EXIOBASE and the Yale Stocks and Flows Database, demonstrating how modular schemas support cross-domain environmental analysis. These models emphasize minimal, reusable class hierarchies that can be extended with domain-specific semantics as needed.

Spatiotemporal validity, often underrepresented in LCA datasets, has also been explicitly formalized through ontology design. Yan et al. (2015) introduce structured vocabularies for geographic scope and temporal coverage, improving the ability to filter and compare datasets in region-specific applications. This supports more transparent assessments in contexts where regulatory baselines or impact factors vary by location and time.

In the built environment, semantic technologies have been applied to

align BIM models with LCA datasets and sensor data. Boje et al. (2023) present a building-scale case study where RDF-based alignments enable real-time sustainability feedback. While their ontology does not fully model ILCD semantics, it illustrates the feasibility of embedding LCA into operational construction data environments using LD infrastructure.

A related effort in the construction domain is Radinger et al. (2013), who introduce BauDataWeb, a LD portal for the Austrian construction materials market. The project transforms Eurobau's relational database into RDF using a combination of GoodRelations, a newly developed FreeClassOWL ontology, and a utility vocabulary for modeling logistics constraints. The resulting knowledge graph exposes over 88 million triples and enables structured querying of products, suppliers, and attributes across eight languages and ten EU countries.

While the focus is commercial matchmaking rather than environmental assessment, BauDataWeb demonstrates key principles relevant to LCA integration: hierarchical taxonomy alignment, quantitative attribute filtering, and federated querying with external vocabularies. These principles anticipate many of the modeling decisions adopted in this thesis, but are here applied to ILCDx Data and sustainability-driven design use cases.

Across these efforts, recurring design principles include modular ontology structures, lightweight vocabularies, explicit context modeling, and declarative query interfaces. Nonetheless, most existing approaches stop short of modeling the full ILCD schema or directly integrating with classification systems used in planning workflows, such as DIN 276 and BKI. Bridging that gap requires not only conceptual modeling but also mappings that reflect regulatory practice.

2.3.2 Mapping Structured Data to RDF

Constructing RDF-based knowledge graphs from ILCDx Datasets requires explicit mappings between source fields and ontology terms. Several approaches address this challenge with different trade-offs in expressiveness, maintainability, and compatibility with nested data formats such as JSON. This section reviews three options, R2RML, SPARQL Anything, and LinkML, to contextualize the design choice made in this thesis.

R2RML (RDB to RDF Mapping Language) is a W3C standard for converting relational databases into RDF (Das et al., 2012). While robust for tabular data, it assumes SQL (Structured Query Language) access and does not natively support nested formats (Dimou et al., 2014). Applying it to ILCD JSON would require flattening the structure, adding complexity, and reducing schema transparency. A hybrid approach could extract only simple fields of interest into SQL views (e.g., using JSON functions in PostgreSQL (PostgreSQL Global Development Group, 2024)) and map them with R2RML, while delegating nested data structures to JSON-native tools such as SPARQL Anything (Warren et al., 2024) or JSON-supporting tools such as LinkML (Moxon et al., 2021). This avoids flattening the entire

structure but requires maintaining two separate pipelines.

SPARQL Anything generalizes SPARQL queries to operate on non-RDF sources such as CSV, JSON, and XML. It enables graph construction via *CONSTRUCT* queries and virtual views (Daga et al., 2021). This approach is flexible but tends to blur the separation between data mapping and domain logic, limiting reuse and complicating validation for complex schemas such as ILCD.

LinkML is a declarative, YAML-based schema language that compiles to RDF (Moxon et al., 2021). It separates schema design from transformation logic and supports modular, nested structures, making it well-suited for ILCD. LinkML also enables generation of SHACL constraints from the same schema, improving transparency and maintainability (LinkML Project, 2024). Given the need for a single, schema-driven source of truth and tight coupling with SHACL rules, this thesis adopts LinkML for end-to-end transformation rather than maintaining parallel R2RML and JSON pipelines.

While modular ontologies and structured mappings ensure that ILCDx Data can be modeled and validated as graph-based knowledge, they do not by themselves populate these models with high-quality semantic annotations. The next challenge lies in assigning appropriate classifications to ILCDx Data Instances, particularly in cases where labels are inconsistent, sparse, or absent. § 2.4 reviews automated enrichment methods that combine rule-based parsing with machine-learning techniques, including sentence embeddings, semantic retrieval, and large-language-model reasoning.

2.4 Semantic Categorization Techniques

2.4.1 Large Language Models

Recent advances in transformer-based models have led to the emergence of large language models (LLMs) capable of solving a wide range of tasks through in-context learning. These models, typically trained on web-scale corpora, learn to predict the next token in a sequence and generalize across domains without requiring task-specific fine-tuning (Bommasani et al., 2021; Brown et al., 2020).

LLMs are part of the broader field of Natural Language Processing (NLP), which develops computational methods for analyzing and generating human language. Modern LLMs contain billions of parameters and are designed for few-shot task adaptation via natural-language prompts (Brown et al., 2020). This “pre-train and prompt” paradigm shifts the burden of adaptation from supervised re-training to strategic input formatting, enabling flexible deployment across diverse applications. In structured data contexts, this allows LLMs to function as high-level semantic classifiers without requiring domain-specific retraining (Brown et al., 2020; Kojima et al., 2022).

These capabilities make LLMs well suited for workflows in environmental data classification, where the goal is to match sparse metadata to rich taxonomic structures using natural language inputs alone.

2.4.2 Retrieval-Augmented Generation and Semantic Similarity

While LLMs are highly flexible, they are not optimized for tasks that require selecting a correct entry from hundreds of possible options, such as classifying an ILCDx Data Instance into one of 326 ÖKOBAUDAT categories. To address this, RAG has become a standard technique. Rather than prompting the language model with all possible answers, RAG inserts a lightweight retrieval step that filters the search space in advance (Gao et al., 2023; Lewis et al., 2020).

This retrieval step depends on semantic similarity, a technique that compares how closely two pieces of text match in meaning. Semantic similarity tasks, typically evaluated under the umbrella of Semantic Textual Similarity (STS), assess how closely two text snippets align semantically (Agirre et al., 2016). Modern approaches often use sentence embeddings, numerical representations of text learned by transformer models such as Sentence-BERT (Reimers & Gurevych, 2019), to map text into a shared vector space. In this space, semantically related phrases are located near one another, allowing relevant candidates to be found by geometric comparison.

The most common way to measure this proximity is cosine similarity, a scoring method that evaluates how aligned two embeddings are in direction, regardless of length. This approach is both efficient and highly effective in identifying related terms, even when they differ in wording or structure (Manning et al., 2008).

In a typical RAG pipeline, the input (e.g., a product name or description from an ILCDx Data Instance) is encoded into an embedding. This is compared against precomputed embeddings (e.g., ÖKOBAUDAT category labels) using cosine similarity, and the top-matching candidates are retrieved. These are then passed to the LLM for final classification. This layered structure improves performance, reduces hallucination, and ensures that the model only considers contextually relevant options (Lewis et al., 2020), (Gao et al., 2023).

This retrieval-first approach is especially useful in environmental data contexts, where short and variable metadata fields must be linked to large taxonomies. By combining fast vector search with LLM reasoning, RAG offers a scalable method for semantic alignment in classification tasks.

2.4.3 Rule-Based Labeling from Structured Patterns

Prior to vector-based retrieval, symbolic filters offer a lightweight method to extract high-confidence matches, especially in domains such as con-

crete where naming conventions are standardized. Many ILCDx Data Instances encode identifiers directly in their titles, for example concrete strength (e.g., "C20/25") or exposure (e.g., "XC4") classes. A structured hierarchy of regex patterns can be applied to metadata fields to identify high-confidence subsets of records before applying more complex enrichment. This rule-based filtering provides reliable seed instances and reduces noise for downstream classification.

Such hybrid strategies, combining symbolic rules with statistical models, remain widely used in domain-specific NLP pipelines (Chiticariu et al., 2013; Keber et al., 2024). These methods have proven effective in contexts where short technical descriptions must be mapped to standardized vocabularies under conditions of sparse or inconsistent metadata.

2.5 Synthesis and Research Gaps

The literature review identifies five unresolved challenges that structure the research agenda across WPs:

1. **Early-phase BIM–LCA workflows lack automation.** Existing tools rely on finalized quantities and manual mappings. WP4 develops a knowledge-graph–driven prototype to support semantic filtering and recommendation during conceptual design.
2. **ILCDx Datasets lack schema-level interoperability.** JSON-based ILCDx records from ÖKOBAUDAT, EPDNorge, and others must be formalized into a shared ontology. WP3 provides this through a LinkML-defined model serialized as RDF.
3. **The application of LinkML to ILCDx Data is undocumented.** WP3 demonstrates LinkML's suitability for deeply nested, semi-structured data by modeling ILCDx Datasets and enabling SHACL-based validation.
4. **DIN 276, BKI, and ÖKOBAUDAT taxonomies are not semantically integrated.** WP3 interlinks them with ILCDx Data Instances, allowing semantic classification and integration across taxonomies.
5. **Metadata enrichment strategies for fragmented ILCDx records remain limited.** WP2 develops a hybrid pipeline combining regex parsing and RAG to improve semantic coverage.

These challenges shape the design logic and evaluation goals of the semantic integration pipeline implemented in WP1–WP4.

3 Methodology

Building on the five research gaps in § 2.5, Chapter 3 translates those needs into an executable four-stage semantic integration pipeline (Figure 3.1).

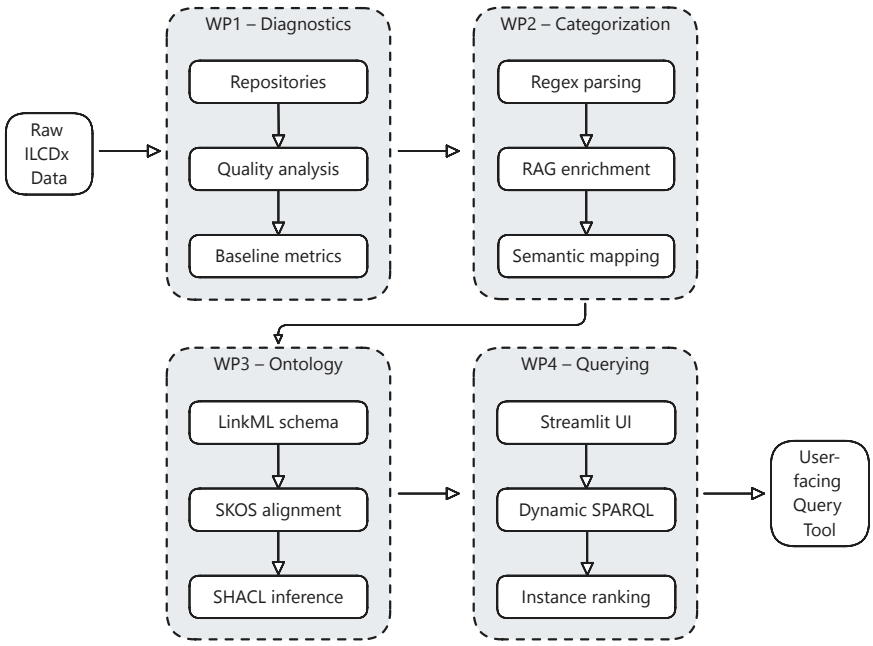


Figure 3.1: Semantic integration pipeline illustrating the four WPs and the flow of ILCDx Data from raw input to user-facing querying.

3.1 WP1 - Data Collection and Analysis

The primary objective of WP1 is to comprehensively document and assess the current state of raw ILCDx Data available from multiple ILCDx Datasets.[6] This diagnostic analysis aims to quantify key issues related to data completeness, consistency, and uniformity, thereby motivating the data harmonization and categorization tasks in subsequent WPs.

[6] The ILCDx Data used in this analysis were provided by the Chair of Design Computation at RWTH Aachen University.

3.1.1 ILCDx Dataset Overview and Diagnostic Setup

Each ILCDx Data Instance conforms to the ILCD schema, which structures environmental data into five sections: *Process Information*, *Modelling and Validation*, *Administrative Information*, *Inputs and Outputs*, and *LCIA Results* (JRC, 2010). The most relevant field for this study is the *Classification* entry in *Process Information*, used to tag datasets with material categorization taxonomies such as from ÖKOBAUDAT (Table 3.1).

The analysis in WP1 focuses on ILCDx Data Instances classified with material category "Ready mixed concrete" from ÖKOBAUDAT, with key attributes extracted for further processing: *Name*, *Modules*, *Scenario*, *Environmental Indicators*, *Lifecycle Indicators*, *Compressive Strength*, *Bulk Density*, *Country*, and *Dataset Type*.

Table 3.1: Example of classification levels in ÖKOBAUDAT

Level	Class ID	Value
0	1	Mineral building products
1	1.4	Mortar and Concrete
2	1.4.01	Ready mixed concrete

3.2 WP2 - Automated Categorization

WP2 addresses the categorization gap identified in WP1 by developing a two-stage enrichment pipeline. The first stage extracts and normalizes material-relevant attributes from ILCDx Data, such as compressive strength, bulk density, and declared lifecycle modules. The second stage assigns two complementary classifications to each ILCDx Data Instance: a three-level ÖKOBAUDAT material category and a set of DIN 276 CG codes. These steps provide the semantic backbone for downstream ontology modeling in WP3 and ensure that ILCDx Data can be queried meaningfully in WP4. Categorization is achieved through a hybrid method that combines deterministic pattern matching (regex) with RAG to balance coverage and interpretability.

3.2.1 Attribute-Alignment Strategy

Attributes already consistent across ILCDx Datasets, such as Environmental Impact and Life Cycle Indicators, Country, and Dataset Type, were carried forward without transformation. In contrast, others, such as Classification, Concrete Compressive Strength, and Concrete Bulk Density, required targeted extraction and enrichment (Table 3.2).

To ensure a uniform module structure, the three A-modules (A1, A2, A3) were collapsed into a synthetic A1–A3 entry; coverage patterns across datasets are detailed in § 4.1.5.

Additional alignment steps focused on material properties. Concrete compressive strength was extracted from ILCDx Data Instance names using pattern matching (e.g., detecting "C20/25" or "B25" entries). Bulk density values were derived by extracting mass and volume from the flow property entries (e.g., 2475 kg mass and 1 m^3 volume yield a bulk density of 2475 kg/m^3). Terminology was standardized by preferring the label gross density (as used in ÖKOBAUDAT) over bulk density.

Table 3.2: Summary of attribute-level discrepancies and their alignment implications

Attribute	Discrepancy Description	Alignment Implication
Classification	Number of classification levels varies (1–3); terms are generic or overly broad	Mapped to ÖKOBAUDAT via regex, RAG, and GPT-assisted categorization in WP2
Name	Naming conventions inconsistent; compressive strength or classification embedded in text	Parsed to extract missing attributes (e.g., compressive strength, classification)
Module	Inconsistent module coverage; missing A1–A3, A4, A5, B1, etc. across datasets	Normalized to a fixed set of modules (A1–A3, A4, A5, B1, C1–C4, D) in alignment with ÖKOBAUDAT
Scenario "Recycled"	Scenario flag not available in some datasets	Cannot be inferred from the data
Environmental Impact Indicator	Uniform naming across datasets	Directly mapped without transformation
Life Cycle Indicator	Mostly uniform. EPDNorge includes non-standard indicators (e.g., GWP-IOBC/GHG)	GWP-IOBC/GHG excluded
Concrete Compressive Strength	Structured values missing or inconsistently embedded in ILCDx Data Instance names	Extracted via string parsing
Concrete Bulk Density	Present in some; derived indirectly in others	Calculated from mass and volume flow properties if available
Country	Uniformly present	Used as-is
Dataset Type	Uniformly present	Used as-is

3.2.2 Categorization Pipeline

To achieve consistent categorization, a hybrid automated pipeline was developed comprising two complementary methods:

1. **Regular-Expression Parsing**:
    - Quickly identifies clearly labeled ready-mix concrete ILCDx Data Instances.
    - High precision and fast execution for straightforward cases.
2. **RAG Pipeline with GPT**:
    - Handles ambiguous cases that regex alone could not confidently resolve.
    - Employs embedding-based similarity scoring and GPT-based reasoning for accurate categorization.

ILCDx Data Attributes Used for Categorization Tasks

Both approaches required structured input derived from selected ILCDx Data attributes (Table 3.3).

Table 3.3: ILCDx Data attributes used for categorization tasks and their descriptions

Attribute	Description
name	The base product name, typically used for initial filtering and embedding-based retrieval.
referenceToLCAMethodDetails	Reference to the underlying LCA methodology or PCR document.
technologyDescriptionAndIncludedProcesses	Describes technological characteristics and background system processes included in the dataset.
technologicalApplicability	Brief explanation of the product's intended applications, helping differentiate between functionally distinct materials.
materialProperties	Encodes key physical characteristics (e.g., compressive strength, bulk density).
flowProperties	Describes technical aspects of the product flow (e.g., mass and volume).

Regular-Expression Parsing

The first step in the categorization pipeline applied a structured hierarchy of regular-expression-based rules to assign material categories to ILCDx Data Instances. This approach was designed to handle high-confidence cases, particularly for TIES, where product names and descriptions frequently follow recognizable industry conventions.

The classification rules combined textual patterns from multiple attributes, namely the product *name*, *technologyDescriptionAndIncludedProcesses*, and *technologicalApplicability*, with optional *flowProperties* metadata where available. The matching logic was implemented as a multi-tiered rule set, starting with triggers for category-specific entries (e.g., concrete admixtures, cement, aggregates) and progressing toward more generic fallback conditions (Figure C.1).

Key rule types included:

- **Concrete identification** based on characteristic patterns such as "C20/25" or exposure classes (e.g., "XC2", "XD3"), with optional reinforcement from flow properties (e.g., exactly 1 m^3 volume).
- **Cement detection** when the word "cement" appeared in both the product name and description or through CEM-type codes (e.g., "CEM II/A-LL").

- **Aggregate classification** based on size descriptors (e.g., "10/20 mm") co-occurring with keywords “aggregate” or “fill”.
- **Precast differentiation**, used to exclude reinforced or prefabricated concrete products from the “Ready mixed concrete” category.

To prevent false positives, additional filters were implemented for steel reinforcement products (e.g., "rebar", "mesh"), which commonly contain the word “concrete” but do not belong to the target category. The final classification output assigned either a three-level ÖKOBAUDAT-style material category or marked the instance as "Other" if no confident match was found.

For EPDNorge, no regex logic was applied due to inconsistent attribute values and poor pattern separability. These entries were instead classified exclusively via the RAG pipeline.

Retrieval-Augmented Generation

The RAG pipeline supplements rule-based categorization by embedding ILCDx Data Instances into a semantic vector space. Each instance is represented by embeddings generated from selected attributes metadata (Table 3.3) and compared against a pre-embedded vector store of ÖKOBAUDAT material category labels. A top-10 similarity search retrieves the most relevant ÖKOBAUDAT categories for each ILCDx Data Instance. These ten candidates are passed in full to the GPT model, which selects the final category using both semantic proximity and contextual cues from the original metadata. No direct selection is made at the similarity stage; categorization always proceeds via the GPT model.

This architecture was chosen to avoid inefficiencies: directly prompting the reasoning model with all 326 ÖKOBAUDAT material categories (~4,000 tokens) would significantly increase monetary cost, both due to larger prompt size and the increased reasoning effort required to select the correct category label.

Vector-Store Configuration

The vector store was implemented as an FAISS index using cosine similarity as the retrieval metric (Johnson et al., 2019). Each ÖKOBAUDAT material category was encoded as a flattened hierarchical label (e.g., "Mineral building products > Mortar and Concrete > Ready mixed concrete"). These embeddings formed the retrieval backend, supporting nearest-neighbor searches for ILCDx Data Instances.

Embedding Model Selection

The embedding stage transforms ILCDx Data attributes metadata and ÖKOBAUDAT material categories into dense vector representations. Models were shortlisted based on performance and practical constraints (Enevoldsen et al., 2025).[7] The final evaluation set included eleven candidates, four of which performed better than OpenAI's *text-embedding-3-large* (Table A.1).

[7] Top-ranked in the January 2025 MTEB English leaderboard with model size below 600 million parameters.

Each embedding model produced a separate FAISS index, allowing a direct comparison of retrieval accuracy. Embeddings were computed locally, ensuring full control over the embedding generation process without external dependencies.

The embedding models selected for the final pipeline were:

- English: *mxbai-embed-large*
- Multilingual (including English): *jina-embeddings-v3*

However, model selection is strictly tied to the data at hand. It was observed that depending on the data other models might perform better but these two showed consistent performance, always among the top.

GPT Model Selection

The second RAG stage employs a GPT model to finalize category assignments from the candidate shortlist generated by the similarity filtering step. This step demands robust contextual understanding and sensitivity to engineering-specific terminology.

During development, a range of open-source language models, including LLaMA, Falcon, Dolphin, Granite, Qwen, and Deepseek, were tested, limited to configurations with approximately 8 billion parameters due to local deployment constraints. Ultimately, OpenAI's GPT model (*o3-mini-high*) was adopted for its superior reasoning performance and manageable computational costs.

3.2.3 Embedding Model Evaluation Design

A subset of 100 ILCDx Data Instances was selected for manual categorization to assess the filtering step of the RAG pipeline. Sampling was stratified by the *referenceToLCAMethodDetails* attribute[8], which loosely corresponds to material type. Approximately 16% of entries lacked PCR and other metadata. A proportional number of such sparse records were included (Table A.3). Records without PCR were grouped as "Missing", and PCR groups with fewer than 20 entries were aggregated under "Other". The PCR groups were randomly sampled in proportion to their frequency to maintain representativeness.

[8] *referenceToLCAMethodDetails* is a proxy for the PCR.

Eight embedding-variant experiments were defined using various bun-

dles of ILCDx Data attributes to analyze the impact of different attribute combinations on retrieval accuracy (Table 3.4).

The defined attribute combinations were:

- **Variant A**: (*name*, *referenceToLCAMethodDetails*, *technologyDescriptionAndIncludedProcesses*, *technologicalApplicability*)
  Established a robust baseline integrating naming and rich descriptive contexts.
- **Variant B**: (*name*)
  Provided a minimal baseline, testing the capacity of surface-level naming features alone.
- **Variant C**: (*name*, *referenceToLCAMethodDetails*)
  Evaluated the added value of structural metadata (i.e., PCR) alongside the name attribute.
- **Variant D**: (*technologyDescriptionAndIncludedProcesses*, *technologicalApplicability*)
  Assessed the standalone semantic relevance of detailed functional descriptions without naming or PCR context.
- **Variant E**: (*materialProperties*, *flowProperties*)
  Explored the embedding model's capability to leverage numeric descriptors directly encoded as textual input.
- **Variant F**: (All attributes A–E)
  Served as a maximal context scenario to test whether increased input improved embedding effectiveness.
- **Variant G**: (*name*, *referenceToLCAMethodDetails*, *materialProperties*, *flowProperties*)
  Combined structural metadata with numeric attributes, assessing whether numeric data enhances categorization.
- **Variant H**: (*name*, *technologyDescriptionAndIncludedProcesses*, *technologicalApplicability*)
  Tested the effectiveness of descriptive usage context alongside naming, explicitly excluding PCR.

3.2.4 DIN 276 Cost-Group Assignment

A pilot enrichment step assigned DIN 276 CGs to 40 ready-mix concrete ILCDx Data Instances, ten each from ÖKOBAUDAT, IBUCategories, EPDNorge, and TIES. Each instance was passed to a GPT prompt that exposed the full hierarchy of CG 300 and required the model to return every selected parent code together with at least one child. This safeguard prevented structurally invalid combinations.

3.3 WP3 - Ontology Creation and Knowledge Graph Generation

With attributes normalized and categories assigned in WP2, WP3 formalizes the cleaned dataset into an ontology and query-ready knowledge graph. This graph links ILCD attributes explicitly to established external classification systems. The technical objectives of this WP are to:

- Design a modular ontology schema using LinkML, incorporating

core ILCD properties and specific extensions for concrete-related attributes.

- Align each harmonized ILCDx Data Instance with external vocabularies and classification schemes (e.g., ÖKOBAUDAT, DIN 276, BKI) using semantic web standards, SKOS and SHACL.
- Generate a robust knowledge graph accessible via a public SPARQL endpoint, facilitating subsequent querying and exploration in WP4's prototype.

3.3.1 Data Preparation

From WP2's categorization and normalization pipeline, Data Instances explicitly categorized under "Mineral building products > Mortar and Concrete > Ready mixed concrete" were retained for semantic enrichment. Each retained instance was fully aligned with mandatory attributes to ensure the completeness required for reliable semantic inference. The instance data was serialized in JSON.

3.3.2 Ontology Modeling with LinkML

Modular Schema Design

The LinkML schema comprises five modules aligned with the main ILCD sections: *processInformation*, *modellingAndValidation*, *exchanges*, *LCIAResults*, and shared types in *SharedDefinitions*.

Each ILCD section was modeled as an independent LinkML module. Module development followed an iterative process: schema definitions were first derived from the ILCD documentation and ILCDx Data, then incrementally tested for structural completeness and semantic consistency. This modular strategy enabled isolated debugging, structural reuse, and parallel development.

In each iteration, a JSON excerpt was translated into a minimal LinkML YAML instance, checked with *linkml-validator*, and exported to RDF/Turtle format via *RDFLibDumper* to verify that no structural information was lost.[9] A module was finalized only when all tested instances passed validation and the RDF output matched the original data structure.

After completing individual modules, the entire schema was applied to the full set of ready-mix concrete ILCDx Data Instances. This broader test surfaced a few problematic fields (e.g., unstructured values under *anies* and the numerical attribute *relativeStandardDeviation95In*) which were removed due to limited usage or low relevance for early-phase querying.[10] All remaining ILCD attributes were retained.

[9] LinkML requires a YAML input instance, a corresponding YAML schema file, and the compiled Python classes generated from the schema.

[10] One potentially useful attribute, *componentsAndMaterialsAndSubstances*, describing material composition, was excluded because it appeared only in IBUCategories.

Identifier Strategy and Preprocessing

To support stable RDF serialization, the original JSON-encoded ILCDx Data required preprocessing. As discussed in § 2.3.2, LinkML offers a schema-driven and modular approach well suited for modeling nested JSON structures such as ILCDx Data. Each nested element in the source JSON was assigned a unique identifiers, derived from the data path and the instance UUID. This ensured that every node in the resulting RDF graph had a predictable, URI-stable identifier suitable for querying and semantic alignment.

Before identifier injection, the JSON structure underwent schema-aware preprocessing to resolve naming conflicts and structural inconsistencies. These included clashes with reserved keywords (e.g., *uri*, *class*), ambiguities in *anies* arrays, and overlapping keys across nested entries (Table A.2).

As a result of the preprocessing step, the injected identifiers could follow a consistent namespace and hierarchy pattern, as demonstrated in Appendix B, Listing B.1.

This preprocessing and identifier strategy serves four purposes:

1. **RDF Graph Construction:** Enables each entity to be uniquely and reliably referenced.
2. **Ontology Modularity:** Allows schema validation and reuse across modules.
3. **Rule-Based Reasoning:** Facilitates SHACL inference and SKOS alignment.
4. **Traceability:** Preserves links between the RDF graph and the original JSON source.

Validated and identifier-enhanced instances were serialized to RDF using LinkML's RDFLib backend, forming the foundation for the semantic graph hosted in Fuseki (§ 3.3.5).

3.3.3 Semantic Alignment to External Vocabularies

To support structured querying and semantic reasoning, ILCDx Data Instances were aligned with external vocabularies using the SKOS (Miles & Bechhofer, 2009). Three vocabularies were integrated:

- **ÖKOBAUDAT Material Categories**: Extracted from official XML sources (German and English), canonicalized, and modeled as multilingual SKOS concepts under a shared *ConceptScheme*. All ILCDx *ClassificationEntry* nodes were mapped to this hierarchy where matches were found (Listing B.2).
- **DIN 276 Cost Groups**: CG 300 was encoded as a SKOS taxonomy (§ 2.1.1, § 4.3.3). ILCDx Data Instances previously categorized (§ 4.2.5)

were linked to the corresponding SKOS concepts using *din:hasDIN276CostGroup*.

- **BKI Elements**: Selected building components involving ready-mix concrete were modeled as *bki:BKIElement* nodes (§ 4.3.4). These were linked to DIN 276 CGs and enriched with layer-level material metadata.

This alignment ensures that ILCDx Data Instances are connected across material, cost, and building-component dimensions, supporting multi-criteria SPARQL queries in the WP4 prototype.

3.3.4 Rule-based Inference

ILCDx Data Instances were semantically enriched using SHACL constraints and SHACL-AF rules to support deterministic classification based on quantitative material properties (Knublauch & Kontokostas, 2017; Knublauch et al., 2025). SHACL rules were authored using *sh:SPARQLRule* and stored directly in the knowledge graph alongside the instance data. These rules assign custom SKOS concepts denoting strength and weight classes (e.g., *cc:MediumStrengthConcrete*, *cc:NormalWeightConcrete*) based on numeric attributes such as compressive strength and bulk density.

Six classification rules were defined in total. Each rule binds a classification concept to the instance using predicates such as *cc:hasStrengthClassification*. Where no classification exists, the SHACL rules instantiate and assign the appropriate category (§ 4.3.5).

Inference was executed using the pySHACL engine, configured for in-place enrichment. Post-processing included lightweight cleanup to remove blank nodes and redundant triples introduced by the engine. This step ensured semantic clarity and compactness of the final graph.

3.3.5 Knowledge-Graph Construction and Hosting

After modeling and semantic enrichment, the ontology was exported as RDF (Turtle) and deployed as a knowledge graph using Apache Fuseki, an open-source SPARQL endpoint server (Apache Software Foundation, 2021b). Fuseki was selected for its stability, standards compliance, and ease of integration into the development workflow. The resulting graph, comprising ILCDx Data Instances, aligned SKOS vocabularies, and SHACL-based inferences, was loaded into the triplestore to support querying and prototype interaction.

3.4 WP4 - Prototype Demonstration

The primary objective of WP4 was to design, implement, and validate a functional prototype demonstrating the practical applicability and querying capabilities of the ontology and knowledge graph developed in WP3. The prototype was intended to bridge the gap between advanced semantic modeling and real-world user interaction, addressing whether archi-

tects and engineers could intuitively query ILCDx Data Instances without direct use of SPARQL or prior experience with semantic web technologies.

The prototype was guided by the following key goals:

- **Accessible user interface:** Provide a clear, intuitive interface suitable for non-specialist users such as architects, engineers, or sustainability consultants.
- **Semantic interoperability:** Demonstrate dynamic, ontology-driven querying across multiple external classification systems (e.g., ÖKOBAUDAT, DIN 276, BKI).
- **Practical illustration:** Show that the prototype can support realistic early-phase LCA scenarios through representative examples.

3.4.1 Prototype Architecture Overview

The prototype was developed using a three-tiered architecture (Figure 3.2), consisting of:

- **Frontend (Streamlit UI):** An interactive web interface supporting intuitive user interaction
- **Middleware (Query Builder):** Dynamically constructs SPARQL queries based on user-selected filters and interface events
- **Backend (Fuseki Triplestore):** A semantic Jena Fuseki triplestore hosting the RDF-based ontology and knowledge graph developed in WP3.

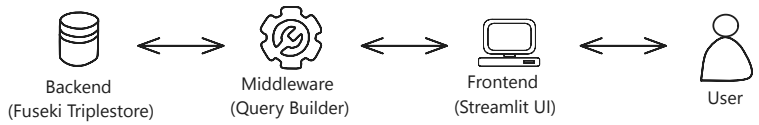


Figure 3.2: High-level client–server architecture of the prototype.

3.4.2 Interface Design

Streamlit was selected for its rapid prototyping capabilities and intuitive interface components (Streamlit Inc., 2025). The user interface consisted of the following components:

- **Sidebar Filters**, grouped thematically:
  - **Main Filters**: Material category, DIN 276 (CG), module, environmental and life cycle indicators, scenario (recycled)
  - **Concrete-Specific Filters**: Compressive strength and bulk density classification
  - **Dataset Filters**: Country of origin and dataset type (e.g., specific, average, representative)

- **Main Window**, composed of:
  - **Result Table**: Lists ILCDx Data Instances matching the selected criteria, with sortable columns for environmental indicators
  - **Highlight Option**: Marks the three datasets closest to the computed average, supporting representative selection in early-phase analysis
  - **Summary Panel**: Displays computed averages for selected indicators across the filtered dataset
  - **Context View**: Presents expandable BKI element descriptions linked via the selected DIN 276 CGs

A toggleable *Average EPD Mode* in the sidebar activates an alternative query configuration.

Interactive elements such as dropdown menus, multi-select boxes, checkboxes, and sliders were used throughout, complemented by contextual tooltips to guide users and support overall usability.

3.4.3 Dynamic Query Generation Logic

The prototype supports two distinct query execution modes, each defined by a different approach to filter processing and statistical evaluation. Both modes are implemented via dynamically generated SPARQL queries, translating user-selected filters into executable graph queries.

- **Standard Mode:** This default mode constructs a SPARQL query that directly reflects the selected filter parameters (e.g., material category, DIN 276 CGs, environmental indicator thresholds), but does not include embedded statistical operations. Statistical computations, such as calculating the mean environmental profile and ranking entries via squared Euclidean distance, are performed procedurally after query execution. This approach enables full access to the result set and supports exploratory analysis, including optional highlighting of instances closest to the computed average.
- **Average EPD Mode:** In contrast, this specialized mode moves statistical computation into the SPARQL layer. The query includes aggregation functions to compute average values across selected indicators, and *BIND* and *ORDER BY* clauses to calculate squared Euclidean distances and return the three most representative instances. All computation is performed within the graph query, avoiding procedural post-processing.

3.4.4 Implementation Details and Deployment

The system was deployed locally throughout development and evaluation. No containerization or cloud infrastructure was used. Given the scope of the thesis, the deployment architecture was deliberately kept minimal to support rapid iteration and maintain transparency. The cur-

rent setup remains easily portable and can be containerized or scaled in future work.

3.4.5 Evaluation Methodology

The prototype was evaluated informally through internal demonstrations with academic supervisors and research assistants. These sessions involved live walkthroughs and qualitative feedback, which informed iterative usability refinements during development.

Responsiveness testing was conducted using representative filter sets in both query modes, based on a subset of ILCDx Data Instances. To simulate larger-scale performance, a synthetic dataset of 400 ILCDx Data Instances was created by duplicating the original 40 entries nine times. This enabled a controlled comparison of query execution times without introducing additional variability from new data sources. The outcomes of these responsiveness tests are presented in § 4.4.2.

4 Results

The semantic integration pipeline described in Chapter 3 was implemented across the four WPs, each targeting a specific interoperability gap. Chapter 4 presents the results of this implementation, reporting quantitative baselines, enrichment performance, ontology metrics, and the prototype's practical functionality.

The source code is available at https://github.com/g3rezz/lca-data-harmonization-pipeline.

4.1 WP1 - Data Collection and Analysis

4.1.1 Dataset Inventory

ILCDx Datasets showed significant variation in entries (Figure 4.1). TIES had the most entries, followed by EPDNorge, ÖKOBAUDAT, and IBUCategories. About one-quarter of entries lacked explicit dataset attribution.

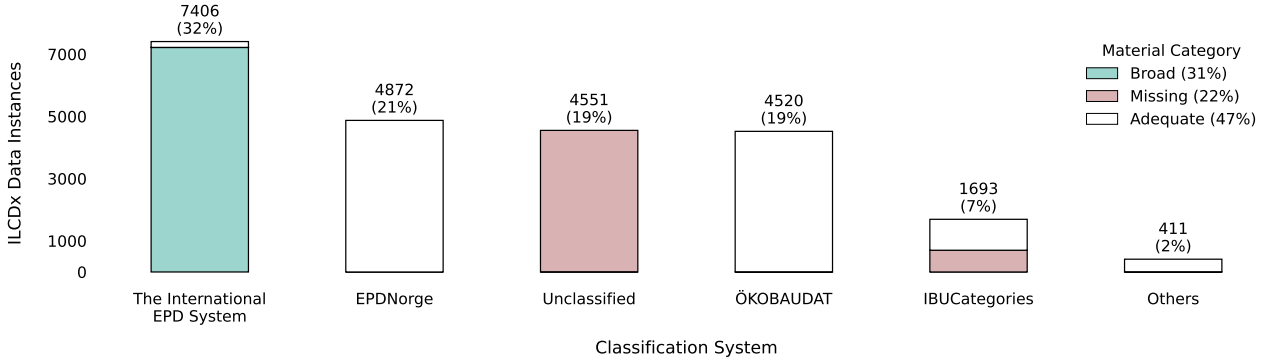


Figure 4.1: Count of ILCDx Data Instances per ILCDx Dataset included in the analysis

4.1.2 Material-Category Identification

In addition to the quantitative counts, qualitative differences emerged among the four ILCDx Datasets.

- **ÖKOBAUDAT and IBUCategories** exhibit high classification consistency, with ready-mix concrete typically aligned to a detailed three-level hierarchy.
- **EPDNorge** has four distinct concrete-related categories:
  - *Bygg* > *Betongvarer* ("Construction > Concrete products"): 370 instances (7.6%)
  - *Bygg* > *Ferdig betong* ("Construction > Ready-mix concrete"): 165 instances (3.4%)
  - *Byggevarer* > *Betongvarer* ("Construction products > Concrete products"): 4 instances (0.08%)
  - *Byggevarer* > *Ferdig betong* ("Construction products > Ready-mix concrete"): 1 instance (0.02%)

  This fragmentation complicates direct category-based filtering, despite the relatively clear semantic intent.

- **TIES** exhibits the greatest deficiencies in categorization, with 94% of its ILCDx Data Instances classified broadly under one level "Construction products, Infrastructure & buildings" or "Construction products" (Figure 4.2). However, most records include standardized concrete notations in the ILCDx Data, suggesting high compatibility with rule-based pattern extraction of ready-mixed concrete. These characteristics informed the decision to apply regular-expression-based parsing in WP2 as the primary categorization method for this dataset.

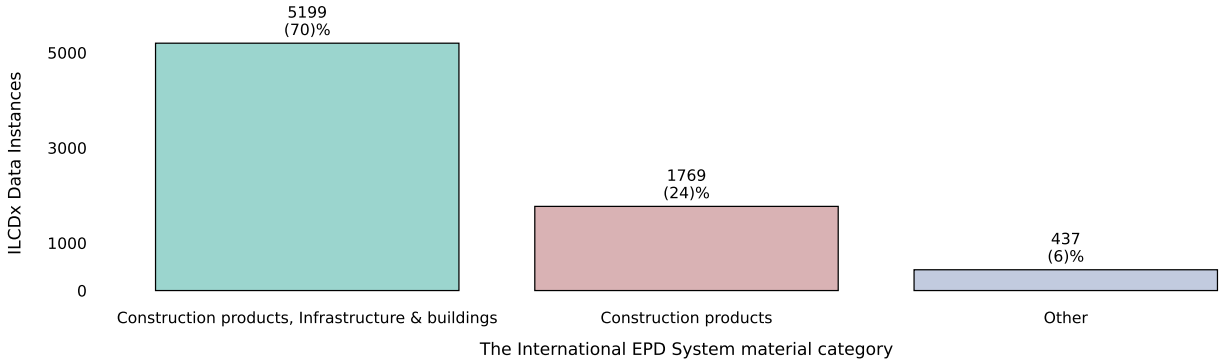


Figure 4.2: Too broad or missing ILCD material categories across ILCDx Data Instances from TIES

4.1.3 ÖKOBAUDAT Material-Category Alignment

To support downstream semantic classification and ontology generation, the ÖKOBAUDAT category hierarchy was aligned using the German source structure extracted from the official developer archive (BMWSB, 2023). This served as the canonical reference for correcting and completing the English labels distributed via the public API. The resulting vocabulary is fully ID-stable and bilingual.

Key observations:

- **Structural completeness:** The aligned hierarchy preserves all three classification levels and retains the original ÖKOBAUDAT category IDs. No nodes were removed, and every term was assigned a unique bilingual label.
- **Translation asymmetry:** Manual review identified 42 German leaf terms without direct English counterparts. These were typically domain-specific compound terms or categories only in the German source. Provisional translations were inserted to ensure completeness.
- **Spelling normalization:** 32 English labels were harmonized for consistency. Edits included standardizing hyphenation, aligning UK/US spellings, and removing non-essential parentheticals.
- **Provenance logging:** All manual interventions were documented via XML comments embedded in the vocabulary structure, ensuring full traceability and facilitating future updates.

4.1.4 Cross-Repository Classification Heterogeneity

To understand why ÖKOBAUDAT was chosen as the canonical backbone, its hierarchy with three other ILCD repositories was compared in terms of depth, language support, and breadth (Table 4.1). Only ÖKOBAUDAT offers a full three-level, robust, bilingual taxonomy; the others are shallower, monolingual, or both.

Table 4.1: Classification-system heterogeneity across ILCDx Datasets

ILCDx Dataset	Depth	Language(s)	Breadth	Example
ÖKOBAUDAT	3 levels	DE & EN	326 leaf classes	*Mineral building products > Mortar and Concrete > Ready mixed concrete*
IBUCategories	3 levels	DE & EN	113 leaf classes	*02 building products > normal/lightweight/autoclaved aerated concrete products > concrete components made of in-situ or ready-mixed concrete*
EPDNorge	2 levels	NO	56 leaf classes	*Bygg > Ferdig betong*
International EPD System	1–2 levels	EN	30 leaf classes	*Construction products > Concrete and concrete elements*

4.1.5 Attribute Completeness and Quality

Table 4.2: Attributes identified as relevant for the search of ILCDx Data Instances.

Attribute	ÖKOBAUDAT	IBUCategories	TIES	EPDNorge
Name	Detailed	Detailed	Less detailed	Less detailed
Classification	Detailed	Detailed	Less detailed	Less detailed
Module	Missing A1, A2, A3	Missing A1, A2, A3	Missing A4, A5	Missing A1–A3, B1
Scenario “Recycled”	Present	Present	Missing	Missing
Environmental Indicator	Uniform	Uniform	Uniform	Uniform
Life Cycle Indicator	Uniform	Uniform	Uniform	Uniform
Concrete Compressive Strength	Missing	Missing	Present partially, poorly inserted	Missing
Concrete Bulk Density	Present	Present	Present partially, poorly inserted	Missing
Country	Present	Present	Present	Present
Dataset Type	Present	Present	Present	Present

Attribute-level analysis revealed significant discrepancies across datasets (Table 4.2). These discrepancies informed later preprocessing and filtering strategies in WP2.

4.2 WP2 - Automated Categorization

WP2 evaluated the effectiveness of the automated categorization pipeline described in § 3.2, focusing on attribute normalization, category alignment, and classification performance across four ILCDx Datasets (Table A.6).

4.2.1 Attribute-Alignment Outcomes

Alignment was applied only to those ILCDx Data Instances that the following categorization steps had already labeled as ready-mix concrete (§§ 4.2.2–4.2.3). Within this domain-specific subset, an instance was counted as aligned only when all attributes mentioned in Table 3.2, could be normalized to a common schema. The dataset-level outcome is summarized in Table 4.3.

Table 4.3: Ready-mix instances successfully aligned to a common schema.

ILCDx Dataset	Identified	Aligned	Alignment rate
ÖKOBAUDAT	260	245	94%
IBUCategories	334	290	87%
EPDNorge	194	95	49%
TIES	1,225	472	39%
Totals	2,013	1,102	55%

As shown in Table 4.3, ÖKOBAUDAT and IBUCategories achieved the highest alignment rates, 94 % and 87 %, respectively, reflecting their consistent metadata and native use of ÖKOBAUDAT categories. TIES exhibited the lowest success rate, 39 %, primarily due to missing compressive strength and density values. However, due to the high volume of already aligned records, name-based parsing was not applied, which could have significantly improved the results. EPDNorge achieved a moderate alignment rate of 49 %, with most failures stemming from incomplete clues to compressive strength or missing volume and mass in the material properties. Overall, 1,102 fully normalized ready-mix concrete instances, formed the harmonized input to WP3. Instances lacking full alignment were excluded from downstream reasoning to preserve classification integrity.

4.2.2 Regular-Expression Outcomes

Targeted, regex-based categorization was applied to ILCDx Data Instances from TIES. Approximately 17.7% were categorized with high confidence

into ÖKOBAUDAT-aligned categories, predominantly “Ready mixed concrete” (Figure 4.3). Manual validation of 100 randomly selected entries confirmed accurate categorization in all cases. The remaining 9,692 entries lacked indicative patterns for concrete and were therefore excluded from further refinement.

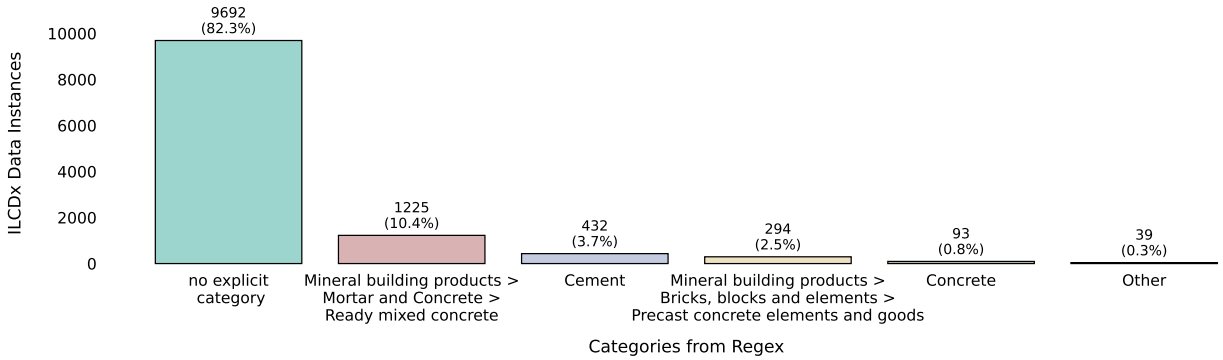


Figure 4.3: Distribution of regex-based categorization in TIES

The 93 ILCDx Data Instances initially labeled as “Concrete” were reprocessed through the RAG pipeline to achieve more granular classification (Figure 4.4). Only one instance was mapped to “Ready mixed concrete,” supporting the effectiveness of the regex step in capturing nearly all relevant cases. The remaining entries, shown in "Other", were distributed across a wide range of categories, including additives, insulation panels, dry mortar, and even transport components, highlighting the semantic ambiguity of the generic “Concrete” regex label.

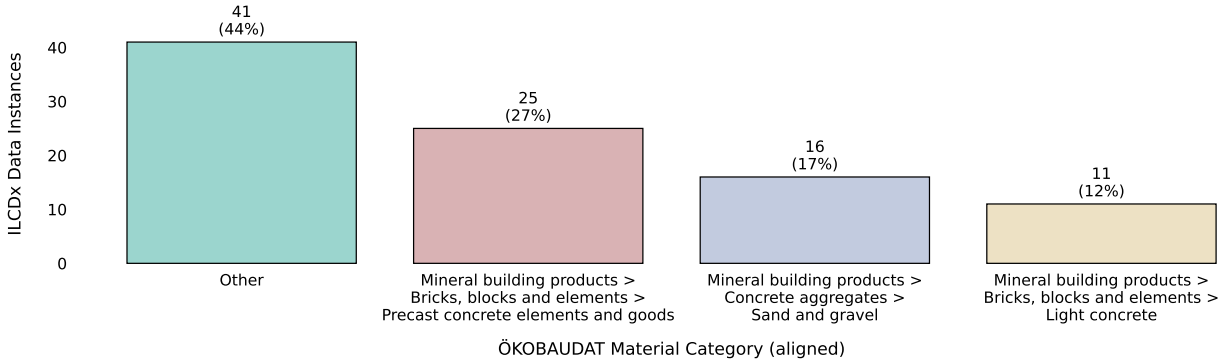


Figure 4.4: Distribution of RAG-based categorization of “Concrete” regex-identified entries from TIES

4.2.3 RAG Outcomes

The 540 ILCDx Data Instances from EPDNorge identified in WP1 as concrete-related were passed to the RAG pipeline for categorization against the ÖKOBAUDAT hierarchy (Figure 4.5). The resulting distribution demonstrates that the majority were successfully matched to concrete-relevant categories.

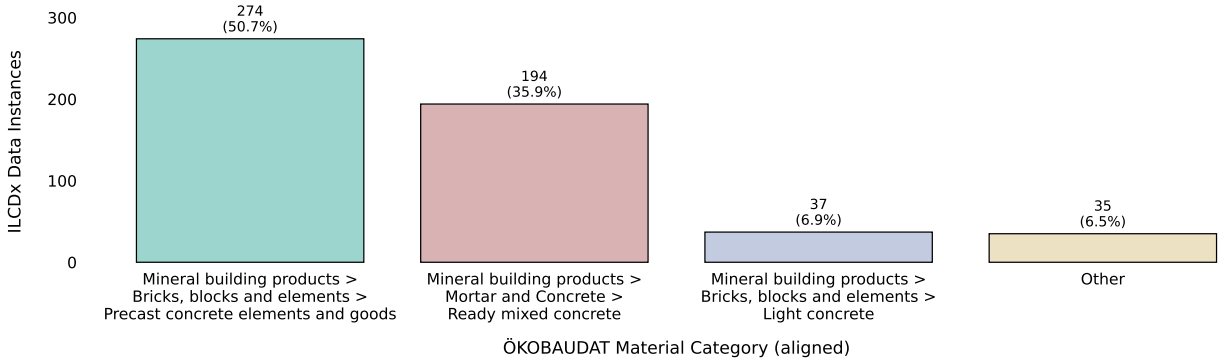


Figure 4.5: Distribution of RAG-based categorization in EPDNorge

The residual group labeled as "Other" comprised 35 ILCDx Data Instances, two originally categorized under ready-mixed concrete, and 33 under concrete products within the EPDNorge dataset. A detailed line-by-line evaluation of these instances, with exact EPD product names withheld, is documented in Table A.4. The key findings from this analysis are summarized below.

- **Incomplete retrieval rather than inference errors:** 32 out of the 33 "Concrete products" entries represent precast concrete elements, including hollow-core and voided slabs, semiprecast floor plates, rail sleepers, manhole rings, precast pipes, wing walls, ballast blocks, kerbs, terrazzo stairs, and cable trenches. These entries were misclassified by the RAG pipeline solely because the retrieval step did not propose the overarching ÖKOBAUDAT category *Mineral building products > Bricks, blocks and elements > Precast concrete elements and goods*. Consequently, the pipeline's ranking step defaulted to the next-best lexical matches, such as *Concrete roof tiles*, *Sewer pipes*, or *Masonry mortars*, artificially inflating the *Other* category. Introducing this missing parent category resolves these 32 cases, elevating the global categorization accuracy from **93.5% to 99.6%**.
- **Limited inference errors:** Only three genuine inference errors occurred:
  1. One foamed-concrete instance (No. 2) was incorrectly categorized as insulation panels instead of ready-mix concrete.
  2. Two rail sleeper instances (Nos. 22–23) were misclassified as renders and plasters despite the presence of indicative terms.

  These errors stem primarily from the semantic ambiguity of Scandinavian terminology rather than systemic biases in the RAG pipeline.

Quality of EPDNorge Source Categorization

The original EPDNorge taxonomy, limited to the two top-level tags "Concrete products" and "Ready-mixed concrete", is too coarse for reliable

automated mapping. Although many "Concrete products" entries include Norwegian or Danish terms that implicitly signal factory production, such as *hulldekke*, *element*, or *fabrikkprodusert*, these cues do not align with the English term precast used in the ÖKOBAUDAT hierarchy. As a result, the retrieval stage fails to identify the target class "Mineral building products > Bricks, blocks and elements > Precast concrete elements and goods", leaving the embedding model to select among semantically distant candidates and thereby producing the systematic errors documented in Table A.4.

4.2.4 Embedding Model Evaluation Outcomes

Table 4.4 reports how eleven pre-trained embedding models perform on the retrieval stage of the RAG pipeline (Variant A input, *k=50*) (§ 3.2.3).[11] The embedding models' performance was assessed in five metrics:

- **Lambda Multiplier** (Lambda): The trade-off between relevance and diversity in the retrieved items from the vector store; higher value prioritizes relevance.
- **k Value** (k): The depth at which the model looks to find the most relevant categories, fixed at 50.
- **Category Not Found** (CNF): Counts occurrences where the correct category was not retrieved within the top 50 results, highlighting gaps in retrieval effectiveness.
- **Mean Rank** (MR): The average rank of the correct category among retrieval results, where a lower value indicates better performance.
- **Top-k Accuracy** (Top-k): Measures the proportion of queries for which the correct category appeared within the top 50 retrieved results.

[11] Preliminary tests using six reranking models: *mxbai-rerank-large-v1*, *jina-reranker-v1-turbo-en*, *bge-reranker-large*, *jina-reranker-v2-base-multilingual*, *bge-reranker-v2-m3*, and *gte-multilingual-reranker-base*, did not yield measurable improvements in retrieval accuracy over the embedding models. As a result, reranking was not pursued further.

Table 4.4: Embedding model performance comparison (best in bold).

Embedding Model	Lambda	k	CNF	MR	Top-k
jina-embeddings-v3	**1.0**	**50**	**7**	**5.92**	**0.93**
snowflake-arctic-embed2	1.0	50	8	9.16	0.92
intfloat-multilingual-e5-large-instruct	0.9	50	11	7.64	0.89
bge-m3	1.0	50	11	9.53	0.89
mxbai-embed-large	0.9	50	15	9.55	0.85
KaLM-embedding-multilingual-mini-instruct-v1.5	1.0	50	16	7.37	0.84
granite-embedding-278m	0.7	50	16	10.60	0.84
gte-large-en-v1.5	0.8	50	16	12.19	0.84
bge-large	0.8	50	18	12.85	0.82
jina-embeddings-v2-base-de	1.0	50	20	10.03	0.80
paraphrase-multilingual	1.0	50	26	11.77	0.74

Attribute Variants Evaluation

The variant-based evaluation illustrates the contribution of different ILCDx Data attributes to embedding-based prediction of ÖKOBAUDAT material categories. Reported values correspond to the best-performing model, *jina-embeddings-v3*. Unless otherwise noted, the insights reflect consistent trends observed across all evaluated models (Figure 4.6). The corresponding attribute bundles and variant-specific metrics are summarized in Table 4.5.

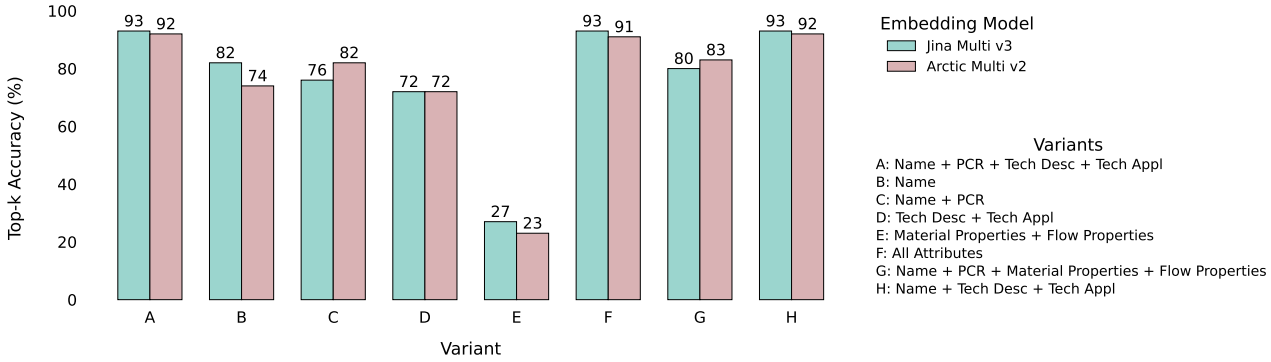


Figure 4.6: Top-k Accuracy per top-performing embedding model across variants.

Table 4.5: Summary of attribute-variant effects on retrieval performance (best in bold).

Variant	Top-k	MR	Interpretation
A	0.93	5.5	Rich product context yields best balance.
B	0.82	9.9	Name alone leads to a solid score.
C	0.82	9.8	Adding PCR helps only marginally if other context absent.
D	0.72	7.3	Usage context without Name is insufficient in isolation.
E	0.27	28.4	Numeric fields underperform due to weak semantic representation.
F	0.93	5.7	Full bundle performs similarly to A; excess fields add noise without benefit.
G	0.83	10.4	Numeric fields regain some utility when paired with structured metadata.
H	**0.93**	**5.2**	**Descriptive context alone matches full bundles, even without PCR.**

These findings confirm that not all attributes contribute equally to retrieval performance. Contextual and descriptive fields such as *name*, *technologyDescriptionAndIncludedProcesses*, and *technologicalApplicability* offer the greatest value. This insight informs future RAG designs, where attribute selection should prioritize semantically rich fields over exhaustive inclusion. Implications for attribute-selection strategy are explored in § 5.2.1.

4.2.5 DIN 276 Cost-Group Assignment (Pilot Evaluation)

Every ILCDx Data Instance received the foundational pair "320 (Foundations)" and "322 (Shallow foundations)" as well as "350 (Floors and ceilings)" and "351 (Floor components)", confirming that the model consistently captured the most common structural applications of ready-mix concrete (Table 4.6).[12]

[12] Evaluated qualitatively; no ground truth for precision or recall.

Table 4.6: Distribution of DIN 276 CGs assigned in the pilot evaluation

DIN 276 parent group	Example child	Frequency (n = 40)
320 Foundations	322	40
350 Floors and Ceilings	351	40
330 External walls	331	27
340 Internal walls	341	27
360 Roofs	361	24
370 Infrastructure systems	371	15
323 Deep foundations	—	8
343 Internal columns	—	7

Key observations:

- **Structural validity:** All assignments respected the DIN 276 hierarchy, every child code was correctly linked to its parent, and no invalid standalone entries were produced. The parent–child constraint embedded in the prompt was satisfied in every case.
- **Coverage breadth:** Each ILCDx Data Instance received at least four cost-group codes, with a median of eight (range: 4–14). This reflects the material's applicability across diverse structural domains in both building and infrastructure projects.
- **Contextual assignments:** Infrastructure-related groups CG 370 and CG 371 were returned in 38% of cases. While relevant for road or track-related planning, these codes can be optionally excluded in design contexts focused solely on architectural elements.

4.3 WP3 - Ontology Creation and Knowledge Graph Generation

4.3.1 Core Ontology

The RDF (Turtle) conversion of a single ILCDx Data Instance (i.e., *ProcessDataSet*) yields a knowledge graph comprising 2,833 triples, linking 706 unique resources via 115 distinct predicates (Table 4.7). The schema layer, defining the ontology's classes and properties, contains 40 distinct *ilcd:* classes and remains constant irrespective of the number of instances ingested. The data layer, in contrast, grows linearly, adding approximately 2,833 triples per additional instance.

See Figure C.3 for an Entity Relationship Diagram of the ILCD schema represented as a LinkML YAML schema forming the core ontology.

The graph strongly favors linked resources over isolated literals. Approximately two-thirds of all triple objects are IRIs (745), while about one-third (425) are literal values, resulting in a link-to-literal ratio of approximately 2 : 1. Importantly, no blank nodes appear in the graph, ensuring global uniqueness and direct referenceability of all resources. Structurally, the graph is fully connected, exhibiting an average node degree of 4.8 and a graph diameter of 9. These metrics confirm that the graph is densely linked yet shallow in depth, a topology that facilitates efficient traversal.

Table 4.7: Module Footprint per *ProcessDataSet* (top-level linking structure only)

Metric	Value
Total triples	2,833
ilcd: classes (schema)	40
Distinct subjects (nodes)	706
Distinct predicates	115
IRI objects	745
Literal objects	425
Blank nodes	0
Average node degree	4.8
Connected components	1
Graph diameter	9

The core ontology structure is organized across five distinct ILCD schema modules (Table 4.8). Each module contributes a fixed set of top-level linking triples per dataset (a total of 55 triples): *processInformation* (6 triples), *modellingAndValidation* (5 triples), *administrativeInformation* (4 triples), *exchanges* (20 triples), and *lciaResults* (20 triples). The largest modules, *exchanges* and *lciaResults*, reflect ILCD's core emphasis on life-cycle inventory flows and impact assessment outcomes. Because every module is anchored to the top-level *ProcessDataSet* node, the inter-module link structure scales predictably as additional data is added.

Table 4.8: Module footprint per *ProcessDataSet* (top-level linking structure only)

ILCD Module	Triples
processInformation	6
modellingAndValidation	5
administrativeInformation	4
exchanges	20
lciaResults	20

4.3.2 Material Category Alignment

The alignment procedure produced a unified graph structure representing material classification relationships (Figure 4.7). Source datasets employ two principal labels for ready-mix concrete: "Beton" (German), used in ÖKOBAUDAT and IBUCategories, and "Ready mixed concrete" (English), used in EPDNorge and TIES. After alignment, both labels converge on a single SKOS concept: *obd:Category_1_4_01* (i.e., "Ready mixed concrete"). As a result, the following uniform SPARQL pattern retrieves all ready-mix records, regardless of source language:

```
?ds obd:hasCanonicalCategory obd:Category_1_4_01 .
```

The alignment is additive: each *ClassificationEntry* retains its original literal ("Beton" or "Ready mixed concrete"), while the new *obd:hasCanonicalCategory* link provides the normalized semantics. No source data is overwritten.

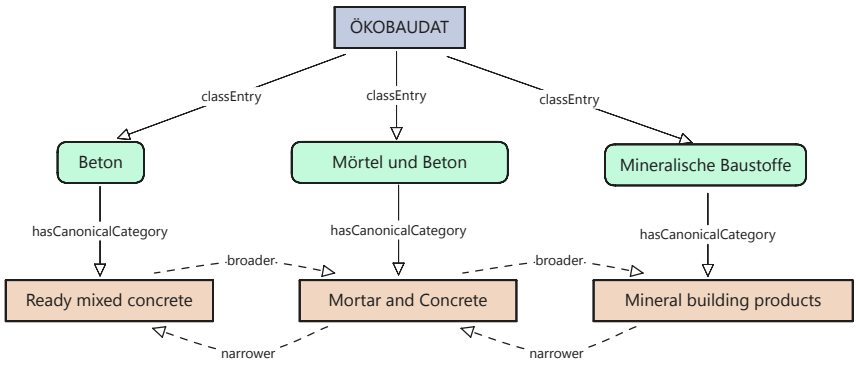


Figure 4.7: Mapping of the ÖKOBAUDAT German classifications to canonical ÖKOBAUDAT English SKOS categories (color legend: Table 4.9)

Table 4.9: Color legend for Figures 4.7–4.11[13]

Color / Shape	Semantic Role (Class)	Typical Label
Teal oval	ILCDx Data Instance (*ilcd:ProcessDataSet*)	*Instance 1*, *Instance 2*
Purple rectangle	Source dataset tag (*ilcd:Classification*)	*ÖKOBAUDAT*, *EPDNorge*
Mint rounded	Raw classification literal (*ilcd:ClassificationEntry*)	*Beton*, *Ready-mixed concrete*
Orange rectangle	Canonical ÖKOBAUDAT concept (*obd/skos:Concept*)	*Ready-mixed concrete*
Beige rectangle	DIN 276 CG (*din/skos:Concept*)	*320*, *322*, *331*
Green rounded	BKI element (*bki:BKIElement*)	*Fundamentplatte*
Red rounded	Inferred concrete class (*cc:ConcreteClassification*)	*MediumStrengthConcrete*
White oval	Layer objects / SKOS notes (*bki:Layer*, *skos:note*)	*Bewehrungsstahl*; *Density 2000–2600 kg/m³*

4.3.3 DIN 276 Cost Groups

[13] Simplified visualizations; unrelated edges and nodes are omitted for clarity.

As visualized below, the graph embeds DIN 276 CG for ready-mix concrete ILCDx Data Instances (Figure 4.8). The solid arrows show that multiple ILCDx Data Instances converge on shared CG nodes, enabling cost-centric aggregation without duplicate literals. Because each CG participates in the DIN hierarchy via *skos:broader*, a single transitive SPARQL pattern:

```
?epd din:hasDIN276CostGroup din:costgroup_320 .
```

retrieves all datasets belonging anywhere in the 320-series, supporting roll-up queries such as all instances belonging to foundations. Conversely, querying the narrower code 322 isolates shallow-foundation instances, demonstrating the granularity supported by the hierarchy.

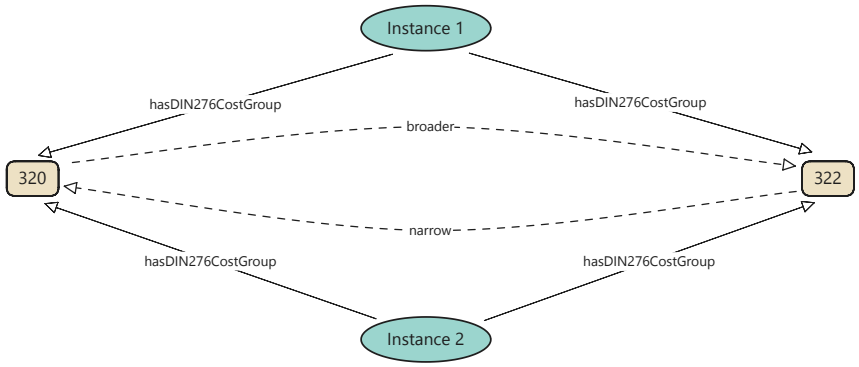


Figure 4.8: Two ILCDx Data Instances (purple) reference CGs 320 and 322 (beige) (color legend: Table 4.9)

4.3.4 BKI Elements Mapping

The final enrichment step integrates BKI element templates and ready-to-use building-component records into the knowledge graph. The resulting linkage pattern is shown below (Figure 4.9). Each BKI element is linked to one or more CGs via *din:hasDIN276CostGroup*, and to one or more material layers via *bki:hasLayer*. In the example, the foundation slab element is associated with CG 322 (*Shallow foundations and base slabs*). Since 322 is a *skos:narrower* of 320, the element can be retrieved using either CG in SPARQL queries.

The internal composition of the slab is modeled through *bki:Layer* nodes, here representing *Transportbeton C20/25* and *Bewehrungsstahl*.

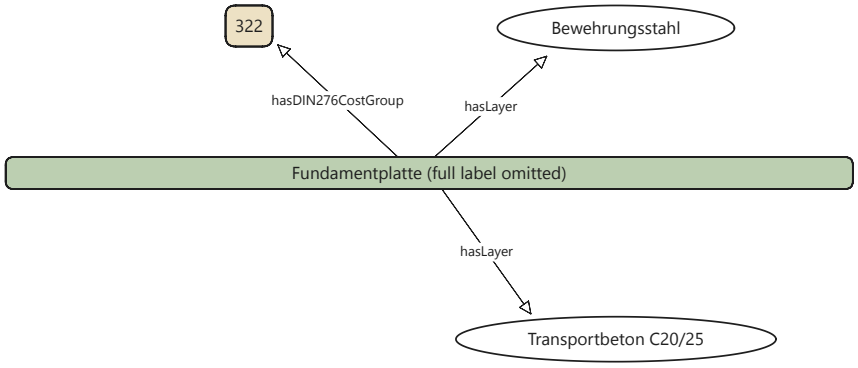


Figure 4.9: BKI element linked to DIN 276 CG 322 and two constituent layers (color legend: Table 4.9)

4.3.5 SHACL Inference Outcomes

The SHACL classification process produces a set of inferred links across ILCDx Datasets (Figure 4.10). Arrows represent the inferred classification links *cc:hasStrengthClassification* and *cc:hasWeightClassification*, as well as the *skos:note* triples that are automatically attached to newly inferred concepts not previously present in the graph.

Each ILCDx Data Instance is linked to shared SKOS nodes, enabling unified retrieval. For example, compressive strengths above 40 MPa were labeled as high-strength concrete (Listing B.3), and the following SPARQL query selects all medium-strength, normal-weight concretes:

```
1 ?ds  cc:hasStrengthClassification  cc:MediumStrengthConcrete ;
2      cc:hasWeightClassification    cc:NormalWeightConcrete .
```

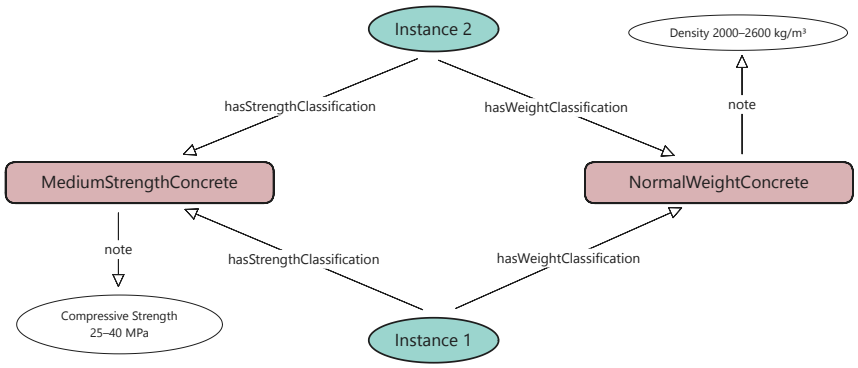


Figure 4.10: Strength- and weight-class assignments produced by the SHACL rules (color legend: Table 4.9)

4.3.6 Knowledge Graph Relationships

A consolidated view of all enrichment steps, material alignment, DIN 276 mapping, BKI element integration, and SHACL inference is visualized in Figure 4.11. Each ILCDx Data Instance is now connected along four independent semantic axes:

1. **Program provenance**, retained via *hasClassification* links to original source tags.
2. **Material identity**, normalized through *obd:hasCanonicalCategory*, bridging multilingual labels to a canonical ÖKOBAUDAT concept.
3. **Cost relevance**, expressed using *din:hasDIN276CostGroup*, with hierarchical structure preserved via *skos:broader/narrower*.
4. **Engineering properties**, classified via SHACL rules that assign *cc:hasStrengthClassification* and *cc:hasWeightClassification* properties.

A fifth axis emerges when a dataset is linked to a BKI element; the element inherits CG links and exposes layer-level material composition.

Because all enrichment paths terminate in shared SKOS or DIN nodes, multi-constraint retrieval becomes straightforward. For example, the following pattern returns every ready-mix concrete that belongs to CG 322 (*Shallow foundations*) and is classified as medium-strength (25–40 MPa):

```
1 ?ds  obd:hasCanonicalCategory       obd:Category_1_4_01 ;       #
      Ready-mixed concrete
2      din:hasDIN276CostGroup         din:costgroup_322 ;         #
           Shallow foundations
3      cc:hasStrengthClassification   cc:MediumStrengthConcrete .
```

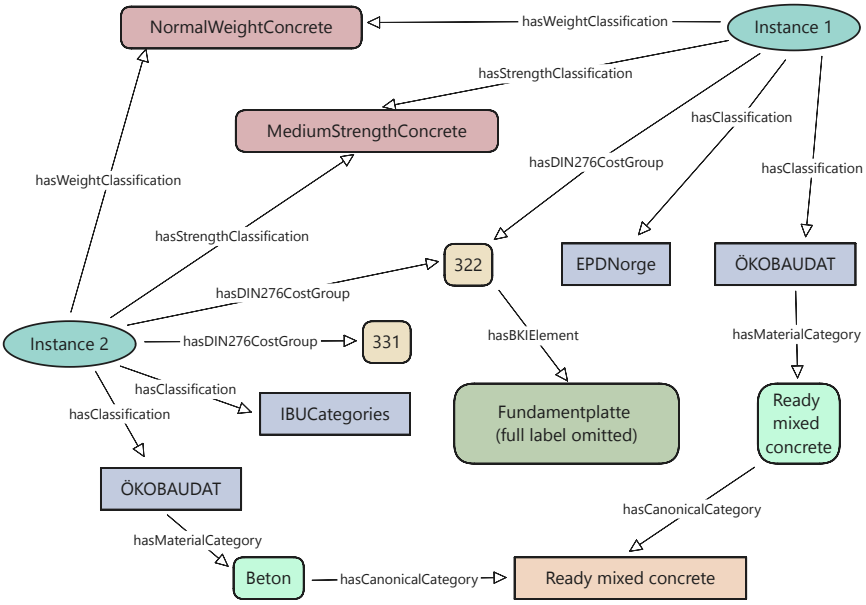


Figure 4.11: Knowledge graph excerpt showing the combined material, CG, BKI, and SHACL-inference relationships for two ILCDx Data Instances (color legend: Table 4.9)

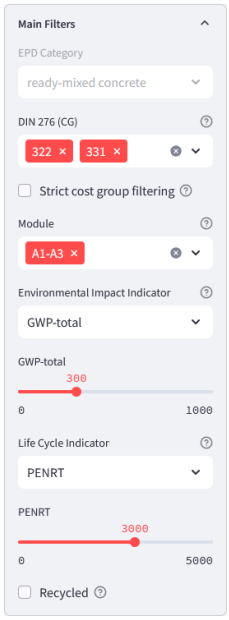
4.4 WP4 - Prototype Demonstration

4.4.1 Functional Capabilities

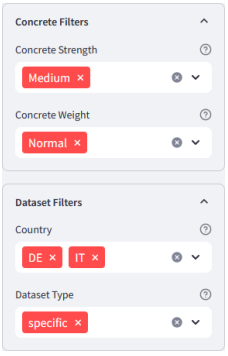
The developed prototype demonstrates three key functional capabilities that support intuitive, semantically enriched querying of ILCDx Data Instances. All identifiers and relationships originate from the SKOS- and SHACL-enriched knowledge graph described in WP3. The graph comprises 40 ILCDx Data Instances, with a total of approximately 125,000 triples (Figure C.2).

Multi-faceted Filtering

The interface supports comprehensive, multi-dimensional filtering. Users can apply criteria such as material category, DIN 276 CG, life cycle module, and environmental indicators, including slider-based thresholding (Figure 4.12a).



(a) Multi-faceted filter panel for category, CG, module, and indicator thresholds.



(b) Concrete-specific and metadata filters for strength, density, country, and dataset type.

Figure 4.12: Side panel filtering options.

Specialized filters tailored to concrete and metadata-specific attributes, such as country of origin or dataset type, offer additional refinement (Figure 4.12b).

Detailed Exploration and Result Analysis

In *Standard Mode*, all ILCDx Data Instances matching the selected filters are displayed in a sortable results table (Figure 4.13). Key indicators such as GWP-total (Global Warming Potential) and PENRT (Primary Energy Non-Renewable Total) are shown alongside visual ranking cues. Optional highlighting identifies the three entries closest to the computed average, supporting representative selection for early-phase decision-making.[14] Each entry is also linked to its associated DIN 276 CG(s) and BKI element(s), enabling contextual exploration of construction-specific metadata.

[14] Although the dataset retains its original *specific* designation, its proximity to the computed average allows it to function as a representative placeholder, effectively serving the role of *average* or *representative* LCA Data in early-phase design.

Statistical Ranking and Recommendation (Average EPD Mode)

In *Average EPD Mode* (Figure 4.14), the three ILCDx Data Instances whose environmental profiles lie closest to the average of the filtered set are shown. The ranking is calculated entirely inside a single SPARQL query (§ 3.4.3).

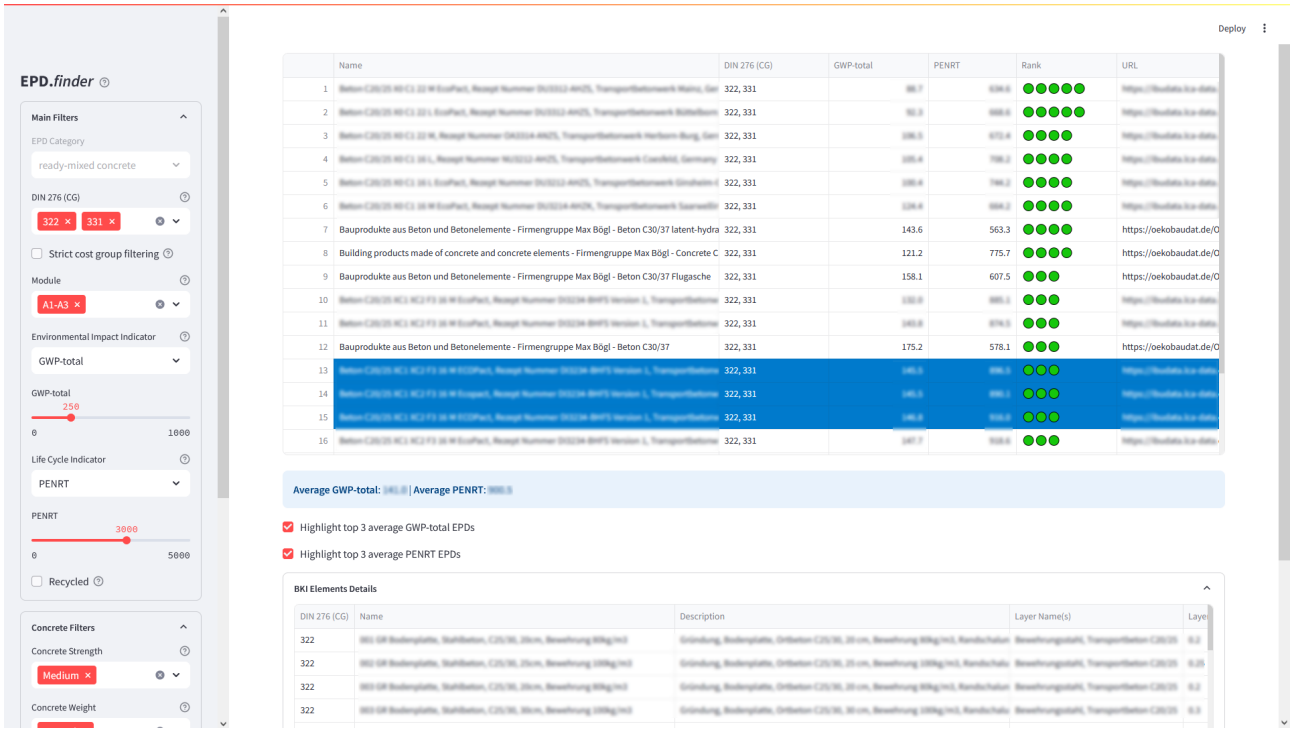


Figure 4.13: Results table with sorting, ranking highlights, and linked BKI details.

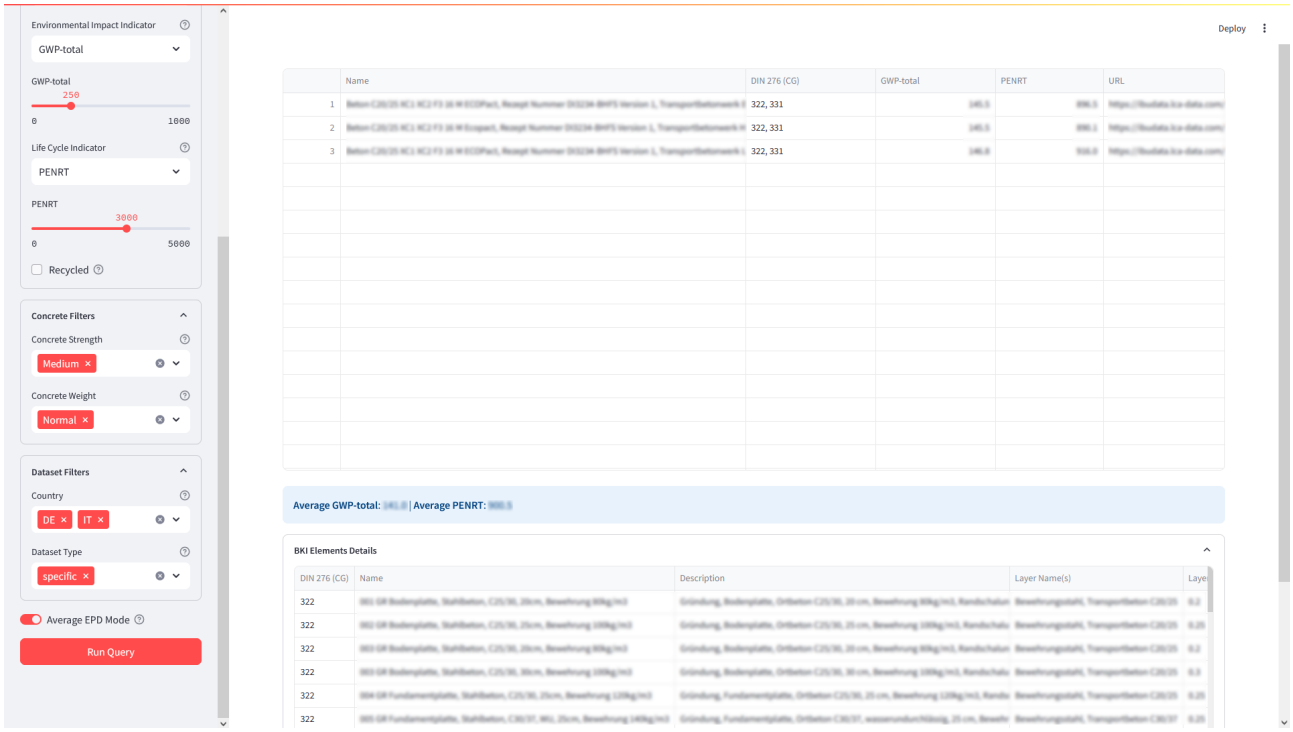


Figure 4.14: Top three ranked records in *Average EPD Mode*, computed entirely in SPARQL.

4.4.2 Performance Evaluation

Query performance was evaluated by measuring execution times across both query modes under varying filter complexity (Table 4.10). Tests ran locally on Apache Fuseki. The selected filters combined categorical and numeric constraints, reflecting a realistic early-phase LCA scenario:[15]

[15] See full SPARQL queries in Listings B.5 and B.6.

> ***Query 1:*** *"Return all German or Italian ready-mix concrete EPD Data instances classified in CGs 322 or 331 whose GWP $\leq$ 250 kg CO_2-eq and PENRT $\leq$ 3,000 MJ for lifecycle module A1–A3."*
> ***Query 2:*** *"As above, plus modules C3, C4, and D."*

All queries returned in under one second. However, *Average EPD Mode* was three to eight times slower than *Standard Mode* for identical filters. Both modes consistently selected the same three top-matching ILCDx Data Instances, confirming alignment in selection logic.

To assess scalability, both queries were repeated on a synthetic dataset of 400 ILCDx Data Instances. *Standard Mode* maintained near-constant runtimes, while *Average EPD Mode* increased substantially, exceeding 1.6 seconds in the second query. This confirms differing scaling behaviors across the two modes.

Table 4.10: Query runtime by mode, filter complexity, and dataset size.

Filter set	**Mode**	**Runtime (ms)** (40 LCDx Instances)	**Runtime (ms)** (400 ILCDx Instances)
Query 1	Standard	35	231
	Average EPD	108	791
Query 2	Standard	31	232
	Average EPD	268	1692

Query 1: DIN 276 (CG): 322, 331; Module: A1–A3; GWP: 250; PENRT: 3000; Country: DE, IT; Dataset Type: Specific.[16]

Query 2: DIN 276 (CG): 322, 331; Module: A1–A3, C3, C4, D; GWP: 250; PENRT: 3000; Country: DE, IT; Dataset Type: Specific.

[16] See Query 1 results in Table A.7 and Table A.8

5 Discussion

The preceding chapter reported the outcomes of each WP, from attribute normalization and automated categorization to ontology modeling and prototype evaluation. Chapter 5 reflects on these results, assessing their methodological implications, practical limitations, and alignment with the research questions introduced in Chapter 1. It also contrasts the strengths and constraints of the graph approach with conventional data-processing pipelines, and identifies opportunities for future improvement.[17]

[17] "graph" refers to the data model, while "semantic" refers to the enrichment and reasoning layers built on top of that model.

5.1 WP1 - Data Collection and Analysis

As shown in Table 4.3, core environmental indicators and life-cycle modules are generally well populated across repositories. However, material properties such as compressive strength and bulk density are frequently missing or inconsistently encoded (Table 4.2). These omissions pose a key bottleneck for automated classification, as they prevent reliable inference and categorization, which motivated the hybrid strategies developed in WP2.

5.1.1 Classification-System Heterogeneity and Alignment Rationale

As detailed in Results § 4.1.4, the four ILCDx Datasets differ significantly in classification depth, breadth, and language support. ÖKOBAUDAT's three-level hierarchy and bilingual coverage made it the most suitable target for canonical alignment, despite translation inconsistencies and incomplete English labels (Table 4.1).

Two broader insights emerge from this alignment:

- **Translation asymmetry:** The alignment process showed that even well-maintained repositories contain language gaps and inconsistencies, reflecting a broader issue in environmental datasets. These gaps reduce the reliability of rule-based matching and embedding-based retrieval in multilingual settings (§ 4.2.4).
- **Granularity mismatch:** Systems such as TIES use much coarser material categories than ÖKOBAUDAT, leading to irreversible semantic loss when aligning from fine-grained to coarse-grained structures. Anchoring the classification pipeline on ÖKOBAUDAT preserved semantic detail and enhanced interoperability across systems.

The alignment work underscores a key observation in § 2.1.1: although LCA repositories contain rich numerical data, they often lack semantic coherence across program operators. This semantic fragmentation necessitates alignment efforts to enable interoperability and consistent downstream analysis.

5.2 WP2 - Automated Categorization

5.2.1 Embedding-Based Categorization: Model and Input Design

The RAG pipeline implemented in WP2 follows a two-stage architecture: first, an embedding model ranks candidate categories based on similarity to input attributes; second, a language model selects the final category from the shortlist. This section evaluates both stages, focusing on how embedding performance varies with attribute design and how model choice affects classification robustness.

The attribute variant experiments conducted in § 4.2.4 revealed three key design patterns that shape retrieval effectiveness:

1. **Descriptive fields consistently improve accuracy.** Variants that included *technologyDescriptionAndIncludedProcesses* and *technologicalApplicability* yield substantially better results than those using only structural or name-based fields. These fields provide semantic context that helps the model distinguish between material types more reliably.
2. **PCR are model-sensitive.** Adding the *referenceToLCAMethodDetails* attribute (i.e., PCR) improved performance for some embedding models while degrading it for others. This suggests that technical codes can either enrich or confuse, depending on the model. Their inclusion should therefore be evaluated on a per-model basis.
3. **Raw numeric attributes degrade performance.** Attributes such as *materialProperties* and *flowProperties*, when used without textual context, consistently lowered retrieval accuracy. Their numerical content appears to dilute the semantic signal.

These results confirm that retrieval quality depends not just on the choice of embedding model, but on how ILCDx attributes are curated and formatted. Descriptive fields should always be included, PCR codes should be tested iteratively, and numeric attributes should be excluded or transformed into an interpretable form Table 3.2.

Downstream classification quality further depends on the reasoning model used to interpret the retrieved shortlist. As shown in § 4.2.4, open-source language models performed well in the retrieval stage but failed to consistently assign the correct ÖKOBAUDAT category. In contrast, commercial models such as OpenAI's *o3-mini-high* delivered accurate and stable outputs, confirming their superior handling of nuanced language tasks. This result reflects a practical trade-off in applied LLM pipelines: open-source language models can offer advantages in cost, throughput, and self-hosting potential (Ateia & Kruschwitz, 2024). However, reasoning over retrieved evidence remains challenging in RAG settings, especially when open-source LLMs must synthesize information across multiple retrieved passages (Islam et al., 2024). The final inference step therefore benefits from models capable of context-sensitive reasoning over retrieved evidence (Kojima et al., 2022).

These findings suggest that effective RAG pipelines for ILCDx classification require careful attribute selection, lightweight embedding models for filtering, and high-accuracy reasoning models for final decision-making, especially in multilingual or underspecified datasets.

5.3 WP3 - Ontology Creation and Knowledge Graph Generation

5.3.1 Graph versus Relational Modeling

One of the central contributions of WP3 is a comparative evaluation of graph and relational modeling approaches for structuring and enriching ILCDx Data. While relational databases dominate traditional LCA tooling, this thesis adopts a knowledge-graph-based model using ontologies, SKOS vocabularies, and SHACL rules. The goal is to assess trade-offs in schema flexibility, query expressiveness, and scalability for early-phase environmental assessment.

Flexibility and maintenance Graph modeling offers declarative extensibility: new SKOS concepts and SHACL shapes can be introduced without modifying the underlying schema or application logic. This modularity enables domain-specific enrichment (e.g., concrete strength classes) without brittle migration scripts. In contrast, relational approaches require structural modifications, such as adding columns, altering tables, or rewriting triggers, which cumulatively increase maintenance overhead and the likelihood of introducing errors or inconsistencies over time (Hellerstein & Stonebraker, 2005). Logic and schema are tightly coupled, reducing adaptability.

Query expressiveness SPARQL natively supports hierarchical queries (e.g., *skos:broader*) and can flexibly traverse linked vocabularies such as DIN 276 or BKI. For example, one SPARQL triple pattern can retrieve all elements belonging to the DIN 276 '320-series' without requiring multiple intermediate joins or queries. Inferred knowledge from SHACL rules becomes immediately queryable and integrated into the graph without preprocessing. Relational databases, by comparison, must emulate hierarchies via recursive Common Table Expressions or mapping tables and compute derived attributes externally or through procedural extensions, which must then be synchronized manually.

Performance and scale SPARQL queries incur higher overhead when embedding logic for ranking or statistical aggregation directly in the query layer, as seen in *Average EPD Mode*. SHACL-based inference was sufficient for prototyping, but would require optimization for large-scale use. [18] Relational systems, by contrast, remain more performant for aggregations and joins due to decades of engine optimization. However, their fixed schema reduces agility when aligning heterogeneous datasets or integrating external classification systems.

These trade-offs are summarized in Table 5.1.

[18] Measured locally using PySHACL v0.30.1 on an approximately 125,000-triple graph with six active SHACL-AF rules; 98 triples were inferred in 35 seconds.

Table 5.1: Comparison of graph versus relational approaches for ILCDx Data modeling.

Dimension	Graph Approach	Relational Approach
Rule Locality	Embedded directly within RDF shapes graph; rules travel with data.	Stored within database catalogs; rules cannot be exported easily.
Portability	W3C-compliant; rules are transferable across compliant triple-stores.	Vendor-specific PL/SQL dialects; limited cross-compatibility.[19]
Schema Evolution	Easy extension (add shapes or subclasses); no migration overhead.	Requires database schema migrations, ALTER TABLE commands, index rebuilds, and trigger maintenance (Hellerstein & Stonebraker, 2005).
Knowledge–Code Separation	Clear separation of domain logic from execution environment.	Logic intertwined with implementation-specific languages.
Standards Status	Standardized and broadly adopted W3C specification.	Triggers are SQL-standard but have vendor-specific behavior.
Performance	Moderate inference overhead during validation; inference runtimes depend on triple-store configuration and rule complexity.	Benefit from efficient row-level triggers, optimized for OLTP scenarios, where performance scales predictably with data volume (Hellerstein & Stonebraker, 2005).
Use-case Fit	Ideal for threshold-based and evolving classification logic.	Preferable for strict transactional processing and stable business rules.

In conclusion, the graph model adopted in this thesis prioritizes adaptability and transparency over raw performance. This choice aligns with the dynamic requirements of early-phase BIM-LCA workflows, in which taxonomies, properties, and data sources are frequently changing. While relational systems remain superior for fixed-scope computation, graph models provide a more future-proof foundation for integrating heterogeneous ILCDx Data at scale.

[19] An illustrative PL/pgSQL implementation is provided in Listing B.4.

5.4 WP4 - Prototype Demonstration

5.4.1 Procedural vs. Declarative Query Strategies

As introduced in § 3.4.3, the prototype implements two complementary query modes: *Standard Mode*, which combines SPARQL filtering with external post-processing, and *Average EPD Mode*, which performs filtering, scoring, and ranking entirely within SPARQL. These modes reflect two design philosophies: prioritizing speed and extensibility, and focusing on self-contained reproducibility. Table 5.2 summarizes their respective execution models, runtime characteristics, and implementation trade-offs.[20]

[20] For results see § 4.4.2 Performance Evaluation

Table 5.2: Comparison of Standard versus Average EPD query modes.

Aspect	Standard Mode (Procedural)	Average EPD Mode (Declarative)
Execution split	SPARQL and post-processing	SPARQL only
Data returned	All matching instances	Only the top 3 closest (`LIMIT 3`)
Statistical scope	Configurable externally	Fixed
Filter structure	Expressed once	Duplicated across subqueries
Runtime 40	31–35 ms	108–268 ms
Runtime 400	231–232 ms	791–1692 ms
Maintenance effort	Low (filters and logic separate)	Moderate (filter duplication)
Auditability and portability	External logic required	Fully self-contained in SPARQL

Standard Mode supports exploratory workflows where ranking strategies may evolve and comprehensive result sets are required for user inspection or export. Filters are applied through SPARQL, but scoring and ranking are handled procedurally, allowing greater flexibility and lower maintenance burden.

Average EPD Mode, by contrast, encapsulates the entire logic, including statistical aggregation and distance-based ranking, within SPARQL. This makes the query fully portable and auditable, particularly useful in automated environments or systems where post-processing is unavailable. While the declarative approach supports reproducibility, its performance limitations, analyzed further in § 5.5.1, highlight the need for targeted query optimization strategies.

5.5 General Discussion

5.5.1 Cross-Package Strengths and Limitations

Strengths

- **Compact embedding models, high retrieval performance**

  Open-source pre-trained embedding models with less than 600 million parameters achieved up to 93% top-50 accuracy in retrieving ÖKOBAUDAT material categories, demonstrating the feasibility of lightweight vector-based retrieval for environmental data (§ 4.2.4).

- **Attribute design insights for RAG**

  Systematic variant testing revealed that descriptive fields (e.g., *technologyDescriptionAndIncludedProcesses*) carry the strongest semantic signal. Numeric attributes performed poorly unless semantically structured. These findings inform future RAG input design (§ 4.2.4).

- **High precision regex layer**

  The symbolic regex layer yielded 100% precision in the TIES subset, with no false positives observed in a 100-instance spot check, confirming its value in well-structured datasets (§ 4.2.2).

- **Flexible and auditable retrieval pipeline**

  The architecture allows for the independent swapping of embedding models without causing disruption downstream. Its modular hand-offs (e.g., JSON, RDF, and SPARQL) enable the independent upgrading of embedding models or triplestore engines. The design also supports future deployment with self-hosted open-source language models, which reduces reliance on APIs.

- **Structured, performant graph topology**

  The RDF graph avoids blank nodes and maintains a 2:1 link-to-literal ratio. It exhibits a shallow, connected structure with graph diameter 9. These characteristics ensure efficient traversal and scalable reasoning (§ 4.3.1).

- **Predictable scaling and semantic consistency**

  Each *ProcessDataSet* generates a fixed number of 2,833 triples, supporting linear growth. SKOS and SHACL ensure robust, standards-aligned enrichment and enable multi-axis semantic linking (§ 4.3.6).

- **Accessible and expressive user interface**

  The prototype demonstrated that even complex semantic queries can be composed via intuitive filter panels. Declarative logic for ranking representative ILCDx Data Instances was fully embedded within SPARQL, requiring no external scripts (§ 4.4.1).

- **Positive early feedback**

  Internal reviewers highlighted the clarity and relevance of semantic filters, especially those based on DIN 276 CGs and SHACL-inferred material classes. The dual-mode architecture (Standard vs. Average EPD) supports exploration and reproducibility.

Limitations

- **Performance bottlenecks in declarative mode**

  Average EPD Mode exhibited poor scaling due to repeated evaluation of identical filter logic across nested subqueries. At a dataset size of 400 instances, execution time exceeded 1.6 seconds (Table 5.2), rendering the approach impractical for real-time applications without optimization. Several strategies could mitigate this issue:

  - **Two-step SPARQL execution**: Separating the computation of global averages from the subsequent distance-based filtering step reduces query repetition and improves runtime efficiency.
  - **Custom property functions**: Embedding reusable scoring functions directly within the triplestore engine (e.g., via Apache

Jena's *javascript* extension (Apache Software Foundation, 2021a)) enables more concise expressions and faster evaluation.

  - **Materialized summary graphs**: Caching statistical aggregates for frequently queried cohorts (e.g., by CG) allows rapid retrieval via named graphs or *SERVICE* clauses.

- **Attribute sparsity in source data**

  Classification quality remains heavily dependent on dataset completeness. ILCDx Datasets lacked key properties (e.g., compressive strength), limiting alignment success without further preprocessing (Tables 4.2 and 4.3).

- **Limited real-world validation**

  Functional evaluation was limited to internal reviewers. Broader validation with architects or sustainability consultants was outside the thesis scope but remains essential for generalizability. Future work should add an annotation interface to capture expert feedback and close an active-learning loop for continual improvement.

- **Incomplete automation**

  Certain datasets required manual alignment (e.g., translation gaps in ÖKOBAUDAT) or validation for category alignment (e.g., EPDNorge's *Concrete products*). This forced a hybrid strategy, transparent regex rules for well-structured data, RAG for ambiguous cases, after which SKOS alignment reconciles both outputs into a single taxonomy.

- **Lack of change-tracking and vocabulary governance**

  As source taxonomies evolve, automated systems must detect and adapt to these changes. Without regular monitoring, such as scheduled data updates and tools that highlight differences between versions, semantic mismatches may accumulate unnoticed, reducing the reliability of future analyses.

5.5.2 Implications for Early-Phase LCA Workflows

- **Programmatic access through URI-stable identifiers**

  Every ILCDx Data Instance and semantic node (e.g., material class, CG, SHACL classification) is globally addressable, supporting downstream integration with tools such as spreadsheets, LCA plugins, or parametric design environments without reliance on positional parsing or brittle data exports.

- **Efficient, reasoning-ready graph structure**

  The RDF graph maintains a shallow depth and high semantic link density, enabling fast SPARQL queries across material identity, cost classification, and engineering properties. The SHACL-enriched classification layer supports deterministic logic, such as querying all medium-strength concretes using a single pattern. [21]

[21] The study confirms that a SKOS and SHACL layer already supplies the hierarchy and deterministic inference needed for early-phase LCA; heavier RDFS/OWL tooling is optional.

- **ÖKOBAUDAT as semantic backbone**

  ÖKOBAUDAT's three-level, bilingual spine anchored all regex, vector search, RAG classification, SKOS mappings, and user filters, eliminating ad-hoc string matching across the pipeline. Its formalization as a SKOS vocabulary enabled reliable classification and filtering across WPs.

- **Support for domain-aligned placeholder logic**

  Both query modes support assigning ILCDx Data Instances as placeholders during early-phase design. In this manner, EPD Data can be a provisional substitute for missing average or representative data. This reflects domain practice, where credible stand-ins are often required before final material decisions are made (Schneider-Marin et al., 2022).

- **EPD Data for representativeness and traceability**

  Since over two-thirds of the ILCDx Data consists of EPD Data, users should prefer these richer records when computing statistical averages (Table A.5).

- **DIN 276 CG integration**

  Classification into DIN 276 CGs enables filtering and reasoning aligned with BIM workflows and spatial cost plans, bridging abstract design elements with environmental product data.

- **Template-driven extensibility across material domains**

  Because LinkML supports template-driven schema definitions (Moxon et al., 2021) and SKOS and SHACL provide standardized vocabularies (Miles & Bechhofer, 2009) and constraint frameworks (Knublauch & Kontokostas, 2017), the same pipeline can be adapted to steel, insulation, or timber with minimal adjustments.

Collectively, these features show that the graph is not merely a passive data store but an active knowledge service. It supports placeholder reasoning in ways that align with how architects and planners conceptualize materials and elements in early-phase BIM–LCA workflows.

6 Conclusion

Building on the critical analysis presented in Chapter 5, the final chapter consolidates the thesis contributions, revisits the research questions, and outlines directions for future work. It reflects on how the semantic pipeline advances early-phase LCA workflows and summarizes the broader implications for data interoperability in the AEC sector.

6.1 Summary of Contributions

This thesis addressed the persistent challenge of semantic fragmentation in ILCDx Data by proposing a modular, standards-aligned pipeline to harmonize and query ILCDx Datasets. The focus was placed on enabling early-phase LCA workflows in the AEC domain, where flexible data integration, category alignment, and property-based reasoning are critical, however, often absent.

To this end, four WPs were implemented, each targeting a distinct stage in the semantic modeling pipeline. WP1 delivered a systematic diagnosis of existing ILCDx Datasets, identifying inconsistencies in attribute availability, classification schemes, and language coverage. WP2 enabled scalable and partially automated categorization of ILCDx Data Instances, supporting alignment to both ÖKOBAUDAT material categories and DIN 276 CGs. This classification step formed a critical bridge between raw data and formal semantic modeling. WP3 formalized the enriched data into a modular ontology using LinkML, complemented by SKOS vocabularies for taxonomy alignment and SHACL rules for rule-based property classification. Finally, WP4 integrated the ontology into a prototype knowledge graph and query interface, demonstrating how harmonized ILCDx Data can be interactively queried along multiple semantic dimensions.

Together, these contributions establish a scalable and interpretable approach to data harmonization and classification, grounded in semantic web standards and validated on a diverse corpus of real-world ILCDx Data. The resulting framework supports fine-grained filtering, semantic enrichment, and downstream integration into LCA tools and BIM-based sustainability workflows.

6.2 Revisiting the Research Questions

The research questions guiding this thesis were each addressed through targeted technical contributions, evaluated using real-world ILCDx Data, and validated within the context of early-phase LCA. This section briefly reflects on each question and summarizes the corresponding findings.

1. **RQ1 – Ontology Modeling**

   How can disparate ILCDx Datasets be modeled as an ontology to support semantic interoperability?

This question was addressed by designing and implementing a modular ontology pipeline based on the LinkML modeling language. ILCDx Data Instances were first normalized and serialized in schema-conformant JSON, then enriched with deterministic identifiers, SKOS-aligned classifications, and SHACL-based material property inferences. The resulting knowledge graph preserves the ILCD schema structure while enabling SPARQL queries across harmonized datasets.

2. **RQ2 – Semantic Enrichment**

   What automated methods can semantically enrich an ILCDx knowledge graph with ÖKOBAUDAT material categories, DIN 276 CGs, BKI element classifications, and material property classes?

   A hybrid classification pipeline was developed, combining regular expressions and RAG to map ILCDx Data Instances to ÖKOBAUDAT and DIN 276 classifications. Rule-based methods provided high precision for well-structured entries, while embedding-based RAG methods increased coverage for ambiguous records. After this initial classification, subsequent enrichment was performed using SKOS and SHACL, which enabled semantic alignment across repositories and rule-based classification based on engineering-relevant numeric properties. This approach achieves scalable enrichment while maintaining transparency and interpretability for regulatory and planning workflows.

3. **RQ3 – ILCDx Data Querying:**

   How can a semantically enriched ILCDx knowledge graph improve the retrieval of ILCDx Data in early-phase building design?

   The prototype developed in WP4 demonstrates that structured, SPARQL-based querying over semantically enriched ILCDx Data enables flexible exploration of environmental and material attributes. The interface supports multi-criteria filtering across dimensions such as ÖKOBAUDAT material category, DIN 276 CG, lifecycle modules, environmental and life cycle indicators, scenario type, inferred material properties, country of origin, and dataset type. Two SPARQL strategies supported statistical similarity-based recommendation, one procedural, one declarative.

6.3 Limitations

This thesis was evaluated primarily on ILCDx Data Instances representing ready-mixed concrete, limiting generalizability to other material domains. Classification remains partially dependent on proprietary GPT APIs, as tested open-source models underperformed. SHACL-based inference was constrained to a small rule set and graph size, limiting scalability. Several steps, including category translation, hierarchy modeling, and validation, required manual input. These constraints reflect the current boundary of applicability and highlight opportunities for further automation, broader dataset testing, and performance optimization.

6.4 Outlook and Future Work

While this thesis provides a functional and extensible pipeline for harmonizing and querying ILCDx Data, several promising directions remain for future development.

One avenue involves fine-tuning open-source embedding and language models directly on ÖKOBAUDAT categories. Tailoring model behavior to construction-specific terminology could improve classification accuracy, resolve ambiguous cases more reliably, and reduce reliance on manually curated rules.

Another direction concerns enriching the knowledge graph with additional metadata extracted from PDF attachments in ILCDx Data Instances. Attributes such as product composition could enable new semantic queries, for example to identify waterproof concrete through compositional indicators.

Geographic data integration also presents an opportunity. Incorporating location-specific information could support transport-related impact estimation and improve the contextual relevance of retrieved ILCDx Data Instances for specific regions, such as cities or postal codes.

Maintaining alignment with evolving external vocabularies, such as ÖKOBAUDAT, requires semi-automated ingestion mechanisms and version control. These workflows could help maintain interoperability and prevent misalignment when source taxonomies change.

Another area of interest is improving SHACL-based reasoning performance. Evaluating high-throughput triple stores and hybrid rule engines could enable the scalable application of property-based classification rules in production environments.

Finally, system usability could be enhanced through natural language querying and contextual guidance. A dialogue-based interface would support early-phase planners in formulating sustainability queries without requiring prior knowledge of dataset structure or terminology. Integrating BIM authoring tools could embed sustainability logic directly into modeling environments.

Appendices

A Appendix: Tables

Table A.1: Overview of the evaluated embedding models according to MTEB ranking (English), January 2025.

Rank	Emebedding Model	Size (M)	Memory (GB)	Dimension	Tokens	Avg
28	jina-embeddings-v3	572	2.13	1024	8194	65.51
29	gte-large-en-v1.5	434	1.62	1024	8192	65.39
33	KaLM-embedding-multilingual-mini-instruct-v1.5	494	1.84	896	131072	64.94
36	mxbai-embed-large-v1	335	1.25	1024	512	64.68
39	text-embedding-3-large	–	–	3072	8191	64.59
46	bge-large-en-v1.5	335	1.25	1024	512	64.23
55	intfloat-multilingual-e5-large-instruct	560	2.09	1024	514	63.61
96	snowflake-arctic-embed-l-v2.0	568	2.12	1024	8194	60.50
142	granite-embedding-278m	278	1.04	768	514	56.87
161	paraphrase-multilingual-mpnet-base-v2	278	1.04	768	514	54.64
241	jina-embeddings-v2-base-de	161	0.60	768	8192	–
211	bge-m3	567	2.11	1024	8192	–

Table A.2: Renaming operations across key ILCD schema sections.

Section	Original key	Renamed key
All	*uri*	*refObjectUri*
All	*anies*	(pruned strings)
classification[].class	*class*	*classEntries*
modellingAndValidation.validation	*validation*	*validationInfo*
time.other.value	*value*	*timestampValue*
…other.anies[].value	*value*	*objectValue*
processInformation.dataSetInformation	*name*	*dataSetName*
exchanges.exchange[].flowProperties[].name	*name*	*nameFP*
exchanges.exchange[].classification.name	*name*	*nameClass*
modellingAndValidation.other	*other*	*otherMAV*
dataSourcesTreatmentAndRepresentativeness.other	*other*	*otherDSTAR*
publicationAndOwnership.other	*other*	*otherPAO*

Table A.3: TIES data sourced proportionally at random by PCR to maintain representativeness.

PCR group	Count	% of total	Instances to sample
Construction products (EN 15804 A2)	5659	45	45
Missing	1935	16	16
PCR ICMQ 3.0 rev. 3.0 – Products and services for construction	1039	8	8
2019:14–c–PCR–003 Concrete and concrete elements (EN 16757)	621	5	5
Other	620	5	5
2019:14–c–PCR–005 Thermal insulation products	439	4	4
2012:01 Construction products and construction services	244	2	2
NPCR Part A: Construction products and services, ver. 1.0 (March 2021)	216	2	2
2019:14–c–PCR–001 Cement and building lime (EN 16908)	192	2	2
2019:14–c–PCR–007 Windows and doors (EN 17213)	175	1	1
Floor coverings.pdf	142	1	1
ITB–PCR A (PCR based on PN–EN 15804)	134	1	1
2019:14–c–PCR–004 Resilient, textile and laminate floor coverings (EN 16810)	122	1	1
2019:14–c–PCR–006 Wood and wood-based products for use in construction (EN 16485)	119	1	1
EN 15804:2012 A1:2013, EPD Ireland PCR Part A	105	1	1
2019:14–c–PCR–014 Acoustical ceiling and wall solutions	91	1	1
PCR Part A: I.S. EN 15804:2012 A1 & A2, and CEN TR 16970:2016 in Ireland (05-03-2022), ver. 2.1	61	0	1
Bodenbeläge.pdf	61	0	1
NPCR 025:2017 Part B for Asphalt	55	0	1
2019:14–c–PCR–009 Flat glass products used in buildings and other construction works (EN 17074)	55	0	1
PCR Part A: I.S. EN 15804:2012 A1 & A2, and CEN TR 16970:2016 in Ireland (17-08-2021), ver. 2.0	52	0	0
2019:14–c–PCR–008 Lifts (elevators)	51	0	0
Beschichtungen mit organischen Bindemitteln.pdf	46	0	0
PCR: Basic module for construction products and services (PCR–mb001)	38	0	0
NPCR 020:2018 Part B for Concrete and concrete elements	37	0	0
Windows and doors.pdf	33	0	0
Product descriptions and scenarios are based on IBU PCR Part B for coatings with organic binders (also applies to inorganic coatings)	32	0	0
Mineralische Werkmörtel.pdf	31	0	0
Mineral insulating materials.pdf	28	0	0
Reaktionsharzprodukte.pdf	27	0	0
2019:14–c–PCR–016 Photovoltaic modules and parts thereof	20	0	0

Table A.4: Detailed manual audit of the 35 "Other" EPDNorge instances ("—" indicates identical to the entry above; exact product names are withheld due to reuse restrictions).

No.	RAG-chosen category	Correct ÖKOBAUDAT category	Key justification
1	Building service engineering > Electrical > Lighting	Building service engineering > Electrical > Lighting	Not a concrete product; RAG result already correct.
2	Insulation materials > Foamed concrete > Foamed concrete insulation panels	Mineral building products > Mortar and Concrete > Ready mixed concrete	Foamed concrete delivered and pumped as ready-mix.
3	Mineral building products > Mortar and Concrete > Screed dry mortar	Mineral building products > Mortar and Concrete > Screed dry mortar	Dry-spray repair mortar aligns with screed class.
4	Mineral building products > Bricks, blocks and elements > Tiles and cladding panels	Mineral building products > Bricks, blocks and elements > Precast concrete elements and goods	Exterior paving slabs, not façade tiles.
5	Mineral building products > Bricks, blocks and elements > Ceiling panel	—	Hollow-core floor slab.
6	—	—	—
7	Mineral building products > Bricks, blocks and elements > Concrete roof tiles	—	—
8	—	—	—
9	Mineral building products > Mortar and Concrete > Mortar (masonry)	—	Man-hole ring.
10	Plastics > Pipes > Sewer pipes	—	Concrete sewer pipe.
11	—	—	Reinforced concrete pipe.
12	Mineral building products > Bricks, blocks and elements > Concrete roof tiles	—	Pre-tensioned rail sleeper.
13	Mineral building products > Bricks, blocks and elements > Natural cut stone	—	Two-layer kerb stone (vibro-pressed concrete).
14	Mineral building products > Bricks, blocks and elements > Aerated concrete	—	Man-hole ring.
15	Mineral building products > Bricks, blocks and elements > Artificial stone	—	Vibro-pressed paving block.

Continued on next page

Table A.4—continued from previous page

No.	RAG-chosen category	Correct ÖKOBAUDAT category	Key justification
16	Mineral building products > Bricks, blocks and elements > Tiles and cladding panels	—	Pavement slabs, not façade tiles.
17	Mineral building products > Bricks, blocks and elements > Substrate	—	Semi-precast floor plate.
18	—	—	—
19	Mineral building products > Bricks, blocks and elements > Ceiling panel	—	Pre-stressed floor plate.
20	Mineral building products > Bricks, blocks and elements > Substrate	—	—
21	—	—	—
22	Mineral building products > Mortar and Concrete > Renders and plasters	—	Concrete rail sleeper.
23	—	—	—
24	Mineral building products > Concrete aggregates > Pozzolan	—	Precast column.
25	Mineral building products > Bricks, blocks and elements > Sand lime brick	—	Man-hole ring.
26	Mineral building products > Bricks, blocks and elements > Concrete roof tiles	—	Precast column.
27	Components for windows and curtain walls > Walling > Other walling	—	Pre-moulded wing wall for culverts.
28	Mineral building products > Mortar and Concrete > Mortar (masonry)	—	Standard concrete pipe.
29	—	—	Precast concrete saddle weight for subsea pipes.
30	Mineral building products > Bricks, blocks and elements > Artificial stone	—	Vibro-pressed paving and block suite.

Continued on next page

Table A.4—continued from previous page

No.	RAG-chosen category	Correct ÖKOBAUDAT category	Key justification
31	Mineral building products > Bricks, blocks and elements > Concrete roof tiles	—	Precast staircase.
32	Building service engineering > Electrical > Cable	—	Cable pull-box (concrete manhole), not an electrical cable.
33	Mineral building products > Bricks, blocks and elements > Substrate	—	Voided slab element.
34	Mineral building products > Mortar and Concrete > Mortar (masonry)	—	Bolt-free concrete pipe weight.
35	Building service engineering > Electrical > Cable	—	Factory-made trough section of reinforced concrete.

Table A.5: ILCDx Data Instance count by classification system and dataset type.

Classification system	Specific	Average	Generic	Representative	Template	Unknown
ÖKOBAUDAT	1,554	985	1,685	227	68	1
IBUCategories	894	751	0	40	8	0
EPDNorge	4,685	1	39	53	1	93
TIES	6,628	390	0	119	1	268
Unclassified	3,199	194	0	12	0	1,146
Total	16,960	2,321	1,724	451	78	1,508
Percentage	73.6%	10.1%	7.5%	2.0%	0.3%	6.5%

Table A.6: Performance metrics for every evaluated matching and alignment step.

Task	Dataset	Method	TP	FP	FN	Precision	Recall	F_1
Ready-mix detection	TIES	Regex	1 225	0	1	1.00	0.999	0.999
ÖKOBAUDAT category classification (v1)	EPDNorge	Embedding + GPT	505	35	0	0.94	1.00	0.97
ÖKOBAUDAT category classification (v2 + parent fix)	EPDNorge	Embedding + GPT	537	3	0	0.99	1.00	0.997
Attribute alignment	ÖKOBAUDAT	Rule set	245	0*	15	1.00*	0.94	0.97
Attribute alignment	IBUCategories	Rule set	290	0*	44	1.00*	0.87	0.93
Attribute alignment	TIES	Rule set	472	0*	753	1.00*	0.39	0.56
Attribute alignment	EPDNorge	Rule set	95	0*	99	1.00*	0.49	0.66

* FP = 0 by definition: an ILCDx Data Instance is counted as aligned only when all mandatory attributes are normalized; partially aligned or wrongly mapped instances therefore contribute to FN, not FP.

Precision, recall and F_1 are estimated from a *100-instance random spot-check per dataset*.

Table A.7: Query 1 results Standard Mode.

Name	CG	GWP	PENRT	Rank	Resource URL
IBU.data subset (11 row-level identifiers omitted in public version)	322, 331	*	*	5–4	ibudata.lca-data.com/…
Building products made of concrete and concrete elements – Firmengruppe Max Bögl – Concrete C30/37 blast furnace cement	322, 331	121.1900	775.6700	4	oekobaudat.de/…
Bauprodukte aus Beton und Betonelemente – Firmengruppe Max Bögl – Beton C30/37 Flugasche	322, 331	158.0700	607.5100	3	oekobaudat.de/…
Beton C20/25 XC1 XC2 F3 16 M EcoPact, Rezept Nummer DI3234-BHFS Version 1, Transportbetonwerk Frankfurt, Germany	322, 331	131.9739	885.0524	3	oekobaudat.de/…
Beton C20/25 XC1 XC2 F3 16 M EcoPact, Rezept Nummer DI3234-BHFS Version 1, Transportbetonwerk Dortmund-Schüren, Germany	322, 331	143.7997	874.4845	3	oekobaudat.de/…
Bauprodukte aus Beton und Betonelemente – Firmengruppe Max Bögl – Beton C30/37	322, 331	175.1500	578.1300	3	oekobaudat.de/…
The International EPD System subset (row-level identifiers omitted in public version)	322; 322, 331	*	*	1	data.environdec.com/…

Additional matching records from IBU.data and The International EPD System were returned by the query but are omitted from the public version due to source-specific reuse restrictions.

Table A.8: Query 1 results Average EPD Mode.

Name	CG	GWP	PENRT	Rank	Resource URL
IBU.data subset (row-level identifiers omitted in public version)	322, 331	*	*	3	ibudata.lca-data.com/…

Additional matching records from IBU.data and The International EPD System were returned by the query but are omitted from the public version due to source-specific reuse restrictions.

B Appendix: Code

Listing B.1: Identifier injection, before *top* and after *bottom*.

```
// JSON excerpt BEFORE deterministic ID injection
"processInformation": {
  "dataSetInformation": {
    "UUID": "00000000-0000-0000-0000-000000000000",
    "dataSetName": {
      "baseName": [
        {
          "value": "Product name",
          "lang": "en"
        }]}}
}
// JSON excerpt AFTER deterministic ID injection
"processInformation": {
  "id":
   ↪  "ilcd:00000000000000000000000000000000_processInformation",
  "dataSetInformation": {
    "id":
     ↪  "ilcd:00000000000000000000000000000000_dataSetInformation",
    "UUID": "00000000-0000-0000-0000-000000000000",
    "dataSetName": {
      "id":
       ↪  "ilcd:00000000000000000000000000000000_dataSetName",
      "baseName": [
        {
          "id":
           ↪  "ilcd:00000000000000000000000000000000_dataSetName_baseName_01",
          "value": "Product name",
          "lang": "en"
        }]}}
}
```

Listing B.2: Canonical SKOS concept triple.

```
ilcd:classificationEntry_XYZ obd:hasCanonicalCategory
 ↪  obd:Category_1_4_01 .
```

Listing B.3: SHACL shape that classifies high-strength concrete (compressive strength $>$ 40 MPa)

.

```
cc:HighStrengthConcreteShape
  a sh:NodeShape ;
  sh:targetClass ilcd:ProcessDataSet ;
  sh:rule [
    a sh:SPARQLRule ;
    sh:construct """
      CONSTRUCT {
        $this cc:hasStrengthClassification
         ↪  cc:HighStrengthConcrete .
      }
      WHERE {
        $this ilcd:exchanges ?exchanges .
        ?exchanges ilcd:exchange ?exchangeEntry .
        ?exchangeEntry ilcd:materialProperties ?mpEntry .
        ?mpEntry ilcd:name "compressive strength" ;
                 ilcd:value ?csValueStr .
        BIND(xsd:float(?csValueStr) AS ?csValue)
```

```
        FILTER(?csValue > 40)
      }
    """ ;
  ] .
```

Listing B.4: PL/pgSQL trigger for concrete-strength classification.

```
CREATE OR REPLACE FUNCTION epd_classify() RETURNS trigger AS $$
BEGIN
  IF NEW.compressive_strength_mpa IS NOT NULL THEN
    IF NEW.compressive_strength_mpa < 25 THEN
      NEW.strength_class := 'low';
    ELSIF NEW.compressive_strength_mpa <= 40 THEN
      NEW.strength_class := 'medium';
    ELSE
      NEW.strength_class := 'high';
    END IF;
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Listing B.5: “Standard” SPARQL query example.

```
PREFIX ilcd: <https://example.org/ilcd/>
PREFIX obd: <https://example.org/obd/>
PREFIX din:  <https://example.org/din276/>
PREFIX cc: <https://example.org/concreteclass/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
SELECT
  (SAMPLE(?nameLit) AS ?Name)
  (GROUP_CONCAT(DISTINCT ?notation ; separator=", ") AS ?DIN276CostGroupList)
  (COALESCE(?SumENV_) AS ?ENV)
  (COALESCE(?SumLC_) AS ?LC)
  (SAMPLE(?resourceURLValue) AS ?resourceURL)
WHERE {
  # (1) Get the filtered EPD set
  ?epd a ilcd:ProcessDataSet ;
       ilcd:processInformation ?pInfo ;
       ilcd:modellingAndValidation ?modVal .
  # -- Get Name
  ?pInfo ilcd:dataSetInformation ?dsi .
  ?dsi ilcd:dataSetName ?dsName .
  ?dsName ilcd:baseName ?baseName .
  ?baseName ilcd:value ?nameLit .
  # -- Get resourceURL
  ?modVal ilcd:dataSourcesTreatmentAndRepresentativeness ?dst .
  ?dst ilcd:otherDSTAR ?dstarRoot .
  ?dstarRoot ilcd:aniesDSTAR ?dstarEntry .
  ?dstarEntry ilcd:name "referenceToOriginalEPD" ;
              ilcd:valueDSTAR ?dstarRef .
  ?dstarRef a ilcd:DSTARReference ;
              ilcd:resourceURLs ?resourceURLValue .
    # -- Filter Category
    ?dsi ilcd:classificationInformation ?ci .
    ?ci ilcd:classification ?class .
    ?class ilcd:classEntries ?entry .
```

```
    ?entry a ilcd:ClassificationEntry ;
            ilcd:value ?entryValue ;
            obd:hasCanonicalCategory ?canon .
    ?canon skos:prefLabel ?canonLabel .
    FILTER(lcase(str(?canonLabel)) = "ready mixed concrete")
    # -- Filter Country
    ?pInfo ilcd:geography ?geo .
    ?geo ilcd:locationOfOperationSupplyOrProduction ?loc .
    ?loc ilcd:location ?country .
    FILTER(?country IN ("DE","IT"))
    # -- Filter dataset type (subType)
    ?epd ilcd:modellingAndValidation ?mVal .
    ?mVal ilcd:LCIMethodAndAllocation ?lciaMa .
    ?lciaMa ilcd:otherMAA ?maa .
    ?maa ilcd:anies ?subTypeNode .
    ?subTypeNode ilcd:name "subType" ;
                 ilcd:value ?subType .
    FILTER(?subType IN ("specific dataset"))
    # -- Filter Strength Classification
    ?epd cc:hasStrengthClassification ?strengthClass .
    ?strengthClass skos:prefLabel ?strengthLabel .
    FILTER(STR(?strengthLabel) IN ("Medium Strength
    ↪  Concrete"))
    # -- Filter Weight Classification
    ?epd cc:hasWeightClassification ?densityClass .
    ?densityClass skos:prefLabel ?densityLabel .
    FILTER(STR(?densityLabel) IN ("Normal Weight Concrete"))
    # -- Filter DIN 276 cost groups
    ?epd din:hasDIN276CostGroup ?cg .
    ?cg skos:notation ?notation .
    FILTER(?notation IN ("322","331"))
  # (2) Compute ENV per filtered EPD
  OPTIONAL {
    {
      SELECT ?epd (SUM(DISTINCT ?valEnv) AS ?SumENV)
      WHERE
      {
      OPTIONAL {
        ?epd a ilcd:ProcessDataSet ;
                ilcd:lciaResults ?lr .
        ?lr ilcd:LCIAResult ?lciaRes .
        ?lciaRes ilcd:referenceToLCIAMethodDataSet ?methodDs
        ↪  .
        ?methodDs ilcd:shortDescription ?methodName .
        ?methodName ilcd:value ?methodReg .
        FILTER(regex(?methodReg, "(GWP-total)", "i"))
        ?lciaRes ilcd:otherLCIA ?oLCIA .
        ?oLCIA ilcd:anies ?mValG .
        ?mValG ilcd:module ?modEnv ;
                ilcd:value ?valEnvStr .
        FILTER(?valEnvStr != "ND")
        BIND(xsd:float(?valEnvStr) AS ?valEnv)
        FILTER(?modEnv IN ("A1-A3"))
        }
        }
      GROUP BY ?epd
    }
    BIND(?SumENV AS ?SumENV_)
  }
  # (3) Compute LC per filtered EPD
  OPTIONAL {
    {
```

```
      SELECT ?epd (SUM(DISTINCT ?valLc) AS ?SumLC)
      WHERE
      {
      OPTIONAL{
        ?epd a ilcd:ProcessDataSet ;
               ilcd:exchanges ?ex .
        ?ex ilcd:exchange ?exEntry .
        ?exEntry ilcd:referenceToFlowDataSet ?flowDs .
        ?flowDs ilcd:shortDescription ?flowName .
        ?flowName ilcd:value ?flowReg .
        FILTER(regex(?flowReg, "(PENRT)", "i"))
        ?exEntry ilcd:otherEx ?oEx .
        ?oEx ilcd:anies ?mValLc .
        ?mValLc ilcd:module ?modLc ;
               ilcd:value ?valLcStr .
        FILTER(?valLcStr != "ND")
        BIND(xsd:float(?valLcStr) AS ?valLc)
        FILTER(?modLc IN ("A1-A3"))
        }
      }
      GROUP BY ?epd
    }
    BIND(?SumLC AS ?SumLC_)
  }
}
GROUP BY ?epd ?SumENV_ ?SumLC_
HAVING (
  COALESCE(?SumENV_) < 250
  &&
  COALESCE(?SumLC_) < 3000
)
```

Listing B.6: “Average-EPD mode” SPARQL query example.

```
PREFIX ilcd: <https://example.org/ilcd/>
PREFIX obd: <https://example.org/obd/>
PREFIX din:  <https://example.org/din276/>
PREFIX cc: <https://example.org/concreteclass/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
SELECT
  ?Name ?DIN276CostGroupList ?SumENV ?SumLC ?avgENV ?avgLC
  ↪ ?distSquared ?resourceURL
WHERE {
  # (1) Get the filtered EPD set with sums
  {
    SELECT ?epd
           (SAMPLE(?nameLit) AS ?Name)
           (GROUP_CONCAT(DISTINCT ?notation; separator=", ")
           ↪ AS ?DIN276CostGroupList)
           (SUM(DISTINCT ?valEnv) AS ?SumENV)
           (SUM(DISTINCT ?valLc) AS ?SumLC)
           (SAMPLE(?resourceURLValue) AS ?resourceURL)
    WHERE {
      # -- Get EPD name (English)
      ?epd a ilcd:ProcessDataSet ;
           ilcd:processInformation ?pInfo ;
           ilcd:modellingAndValidation ?modVal .
      # -- Get Name
      ?pInfo ilcd:dataSetInformation ?dsi .
      ?dsi ilcd:dataSetName ?dsName .
      ?dsName ilcd:baseName ?baseName .
```

```
    ?baseName ilcd:value ?nameLit .
    # -- Get resourceURL
    ?modVal ilcd:dataSourcesTreatmentAndRepresentativeness ?dst .
    ?dst ilcd:otherDSTAR ?dstarRoot .
    ?dstarRoot ilcd:aniesDSTAR ?dstarEntry .
    ?dstarEntry ilcd:name "referenceToOriginalEPD" ;
                ilcd:valueDSTAR ?dstarRef .
    ?dstarRef a ilcd:DSTARReference ;
                ilcd:resourceURLs ?resourceURLValue .
  # -- Filter Category
  ?dsi ilcd:classificationInformation ?ci .
  ?ci ilcd:classification ?class .
  ?class ilcd:classEntries ?entry .
  ?entry a ilcd:ClassificationEntry ;
         ilcd:value ?entryValue ;
         obd:hasCanonicalCategory ?canon .
  ?canon skos:prefLabel ?canonLabel .
  FILTER(lcase(str(?canonLabel)) = "ready mixed concrete")
  # -- Filter Country
  ?pInfo ilcd:geography ?geo .
  ?geo ilcd:locationOfOperationSupplyOrProduction ?loc .
  ?loc ilcd:location ?country .
  FILTER(?country IN ("DE","IT"))
  # -- Filter dataset type (subType)
  ?epd ilcd:modellingAndValidation ?mVal .
  ?mVal ilcd:LCIMethodAndAllocation ?lciaMa .
  ?lciaMa ilcd:otherMAA ?maa .
  ?maa ilcd:anies ?subTypeNode .
  ?subTypeNode ilcd:name "subType" ;
               ilcd:value ?subType .
  FILTER(?subType IN ("specific dataset"))
  # -- Filter Strength Classification
  ?epd cc:hasStrengthClassification ?strengthClass .
  ?strengthClass skos:prefLabel ?strengthLabel .
  FILTER(STR(?strengthLabel) IN ("Medium Strength Concrete"))
  # -- Filter Weight Classification
  ?epd cc:hasWeightClassification ?densityClass .
  ?densityClass skos:prefLabel ?densityLabel .
  FILTER(STR(?densityLabel) IN ("Normal Weight Concrete"))
  # -- Filter DIN 276 cost groups
  ?epd din:hasDIN276CostGroup ?cg .
  ?cg skos:notation ?notation .
  FILTER(?notation IN ("322","331"))
    # -- ENV pattern
    OPTIONAL {
      ?epd ilcd:lciaResults ?lr .
      ?lr ilcd:LCIAResult ?lciaRes .
      ?lciaRes ilcd:referenceToLCIAMethodDataSet ?methodDs .
      ?methodDs ilcd:shortDescription ?methodName .
      ?methodName ilcd:value ?methodReg .
      FILTER(regex(?methodReg, "(GWP-total)", "i"))
      ?lciaRes ilcd:otherLCIA ?oLCIA .
      ?oLCIA ilcd:anies ?mValG .
      ?mValG ilcd:module ?modEnv ;
             ilcd:value ?valEnvStr .
      FILTER(?valEnvStr != "ND")
      BIND(xsd:float(?valEnvStr) AS ?valEnv)
      FILTER(?modEnv IN ("A1-A3"))
    }
```

```
      # -- LC pattern
      OPTIONAL {
        ?epd ilcd:exchanges ?ex .
        ?ex ilcd:exchange ?exEntry .
        ?exEntry ilcd:referenceToFlowDataSet ?flowDs .
        ?flowDs ilcd:shortDescription ?flowName .
        ?flowName ilcd:value ?flowReg .
        FILTER(regex(?flowReg, "(PENRT)", "i"))
        ?exEntry ilcd:otherEx ?oEx .
        ?oEx ilcd:anies ?mValLc .
        ?mValLc ilcd:module ?modLc ;
                ilcd:value ?valLcStr .
        FILTER(?valLcStr != "ND")
        BIND(xsd:float(?valLcStr) AS ?valLc)
        FILTER(?modLc IN ("A1-A3"))
      }
    }
    GROUP BY ?epd
    HAVING (
      COALESCE(SUM(DISTINCT ?valEnv)) < 250
      &&
      COALESCE(SUM(DISTINCT ?valLc)) < 3000
    )
  }
  # (2) Compute overall average values over the SAME filtered
   ↪  set (by reusing the filtering logic from (1))
  {
    SELECT
      (AVG(?sumEnv)  AS ?avgENV)
      (AVG(?sumLc)  AS ?avgLC)
    WHERE {
      {
        SELECT ?epd (SUM(DISTINCT ?valEnv) AS ?sumEnv)
         ↪  (SUM(DISTINCT ?valLc) AS ?sumLc)
        WHERE {
          # -- EPD name (English) again to ensure we match
           ↪  the same set
          ?epd a ilcd:ProcessDataSet ;
              ilcd:processInformation ?pInfo2 ;
              ilcd:modellingAndValidation ?modVal2 .
          # -- Get Name
          ?pInfo2 ilcd:dataSetInformation ?dsi2 .
          ?dsi2 ilcd:dataSetName ?dsName2 .
          ?dsName2 ilcd:baseName ?baseName2 .
          ?baseName2 ilcd:value ?nameLit2 .
          # -- Get resourceURL
          ?modVal2
           ↪  ilcd:dataSourcesTreatmentAndRepresentativeness
           ↪  ?dst2 .
          ?dst2 ilcd:otherDSTAR ?dstarRoot2 .
          ?dstarRoot2 ilcd:aniesDSTAR ?dstarEntry2 .
          ?dstarEntry2 ilcd:name "referenceToOriginalEPD" ;
                       ilcd:valueDSTAR ?dstarRef2 .
          ?dstarRef2 a ilcd:DSTARReference ;
                       ilcd:resourceURLs ?resourceURLValue2 .
    # -- Filter Category
    ?dsi2 ilcd:classificationInformation ?ci2 .
    ?ci2 ilcd:classification ?class2 .
    ?class2 ilcd:classEntries ?entry2 .
    ?entry2 a ilcd:ClassificationEntry ;
            ilcd:value ?entryValue2 ;
            obd:hasCanonicalCategory ?canon2 .
```

```
    ?canon2 skos:prefLabel ?canonLabel2 .
    FILTER(lcase(str(?canonLabel2)) = "ready mixed concrete")
    # -- Filter Country
    ?pInfo2 ilcd:geography ?geo2 .
    ?geo2 ilcd:locationOfOperationSupplyOrProduction ?loc2 .
    ?loc2 ilcd:location ?country2 .
    FILTER(?country2 IN ("DE","IT"))
    # -- Filter dataset type (subType)
    ?epd ilcd:modellingAndValidation ?mVal2 .
    ?mVal2 ilcd:LCIMethodAndAllocation ?lciaMa2 .
    ?lciaMa2 ilcd:otherMAA ?maa2 .
    ?maa2 ilcd:anies ?subTypeNode2 .
    ?subTypeNode2 ilcd:name "subType" ;
                  ilcd:value ?subType2 .
    FILTER(?subType2 IN ("specific dataset"))
    # -- Filter Strength Classification
    ?epd cc:hasStrengthClassification ?strengthClass2 .
    ?strengthClass2 skos:prefLabel ?strengthLabel2 .
    FILTER(STR(?strengthLabel2) IN ("Medium Strength
    ↪  Concrete"))
    # -- Filter Weight Classification
    ?epd cc:hasWeightClassification ?densityClass2 .
    ?densityClass2 skos:prefLabel ?densityLabel2 .
    FILTER(STR(?densityLabel2) IN ("Normal Weight Concrete"))
    # -- Filter DIN 276 cost groups
    ?epd din:hasDIN276CostGroup ?cg2 .
    ?cg2 skos:notation ?notation2 .
    FILTER(?notation2 IN ("322","331"))
          # -- ENV pattern
          OPTIONAL {
            ?epd ilcd:lciaResults ?lr2 .
            ?lr2 ilcd:LCIAResult ?lciaRes2 .
            ?lciaRes2 ilcd:referenceToLCIAMethodDataSet
            ↪  ?methodDs2 .
            ?methodDs2 ilcd:shortDescription ?methodName2 .
            ?methodName2 ilcd:value ?methodReg2 .
            FILTER(regex(?methodReg2, "(GWP-total)", "i"))
            ?lciaRes2 ilcd:otherLCIA ?oLCIA2 .
            ?oLCIA2 ilcd:anies ?mValG2 .
            ?mValG2 ilcd:module ?modEnv2 ;
                    ilcd:value ?valEnvStr2 .
            FILTER(?valEnvStr2 != "ND")
            BIND(xsd:float(?valEnvStr2) AS ?valEnv)
        FILTER(?modEnv2 IN ("A1-A3"))
          }
          # -- LC pattern
          OPTIONAL {
            ?epd ilcd:exchanges ?ex2 .
            ?ex2 ilcd:exchange ?exEntry2 .
            ?exEntry2 ilcd:referenceToFlowDataSet ?flowDs2 .
            ?flowDs2 ilcd:shortDescription ?flowName2 .
            ?flowName2 ilcd:value ?flowReg2 .
            FILTER(regex(?flowReg2, "(PENRT)", "i"))
            ?exEntry2 ilcd:otherEx ?oEx2 .
            ?oEx2 ilcd:anies ?mValLc2 .
            ?mValLc2 ilcd:module ?modLc2 ;
                ilcd:value ?valLcStr2 .
            FILTER(?valLcStr2 != "ND")
            BIND(xsd:float(?valLcStr2) AS ?valLc)
        FILTER(?modLc2 IN ("A1-A3"))
          }
        }
```

```
        GROUP BY ?epd
        HAVING (
          COALESCE(SUM(DISTINCT ?valEnv)) < 250
          &&
          COALESCE(SUM(DISTINCT ?valLc)) < 3000
        )
      }
    }
  }
  # (3) Compute the squared Euclidean distance using the
   ↪  overall averages
  BIND(
    (
      (COALESCE(?SumENV) - ?avgENV) * (COALESCE(?SumENV) -
       ↪  ?avgENV)
    + (COALESCE(?SumLC) - ?avgLC) * (COALESCE(?SumLC) -
     ↪  ?avgLC)
    )
    AS ?distSquared
  )
}
ORDER BY ?distSquared
LIMIT 3
```

C Appendix: Images

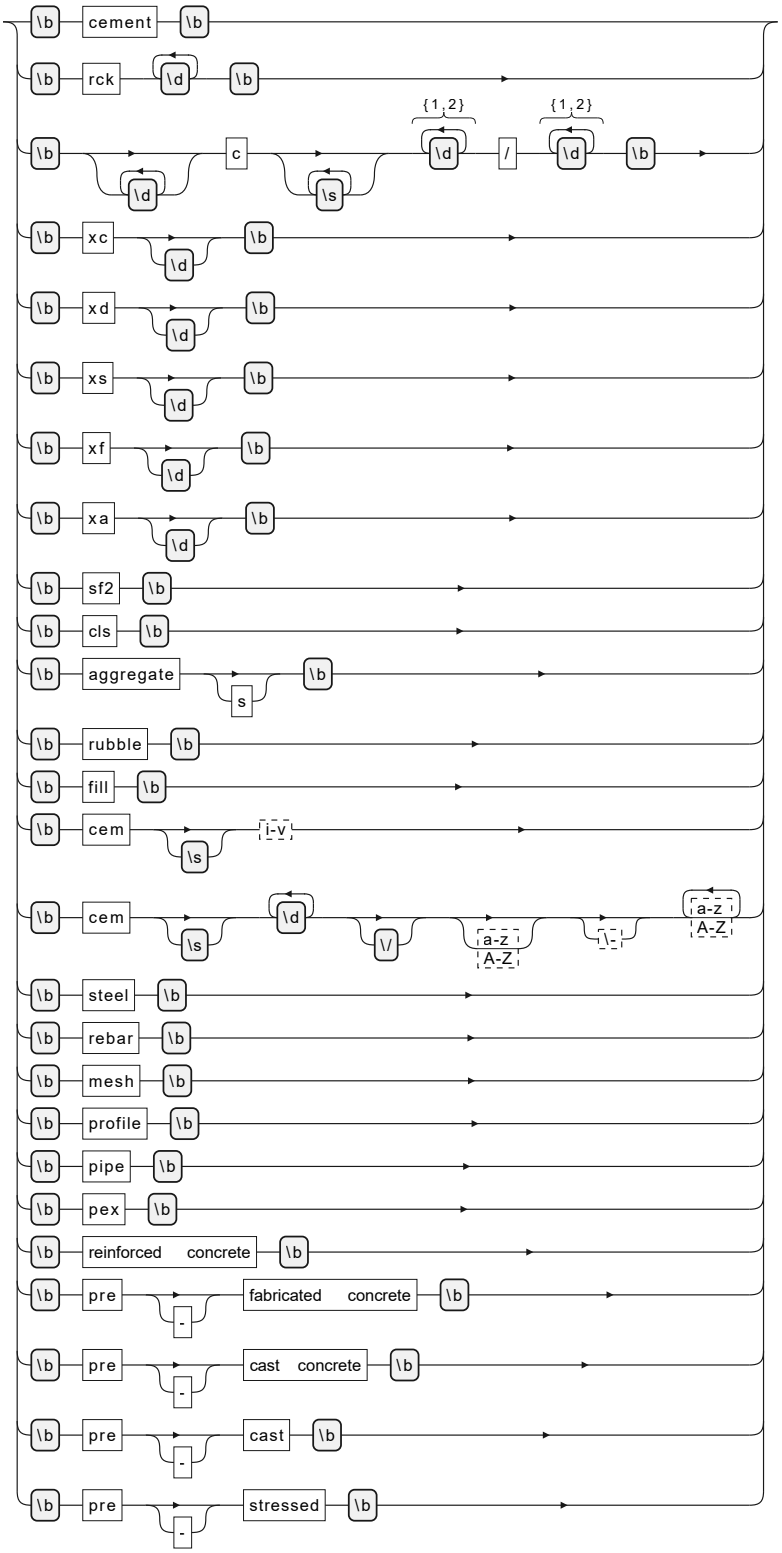


Figure C.1: “Alternation” regex diagram.

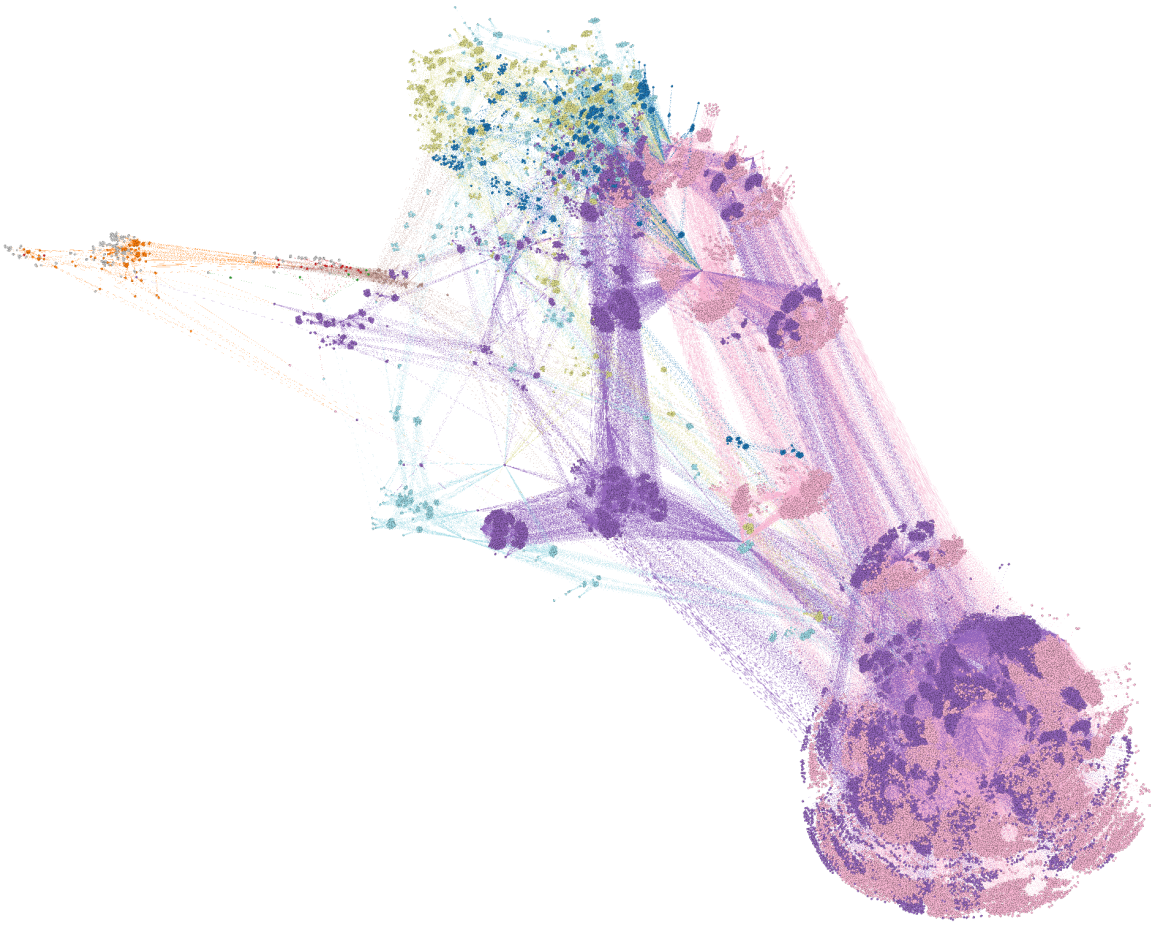


Figure C.2: Visualization of the knowledge graph consisting of 40 ILCDx Data Instances, approximately 125,000 triples.

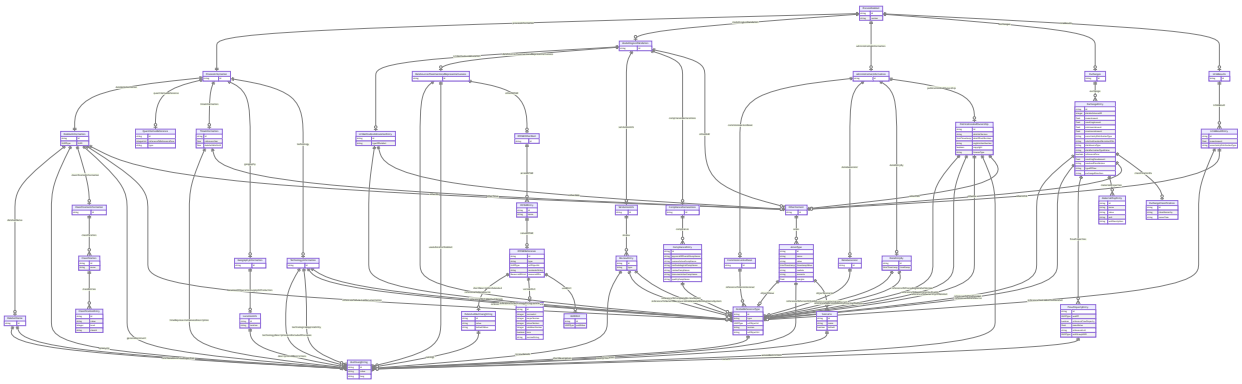


Figure C.3: Entity Relationship Diagram of the ILCD schema represented as a LinkML YAML schema forming the core ontology.

References

Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., & Wiebe, J. (2016). SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 497–511. https://doi.org/10.18653/v1/S16-1081

Apache Software Foundation. (2021a). Apache Jena - ARQ - JavaScript SPARQL Functions. Retrieved June 10, 2025, from https://jena.apache.org/documentation/query/javascript-functions.html#using-javascript-functions

Apache Software Foundation. (2021b). Apache Jena Fuseki. Retrieved June 10, 2025, from https://jena.apache.org/

Ateia, S., & Kruschwitz, U. (2024). Can Open-Source LLMs Compete with Commercial Models? Exploring the Few-Shot Performance of Current GPT Models in Biomedical Tasks. https://arxiv.org/pdf/2407.13511

Bahlau, S., Schumacher, R., Lambertz, M., Theißen, S., Höper, J., Borrmann, A., Forth, K., von Both, P., Ebertshäuser, S., & Horn, R. (2024). *Digital Twin Footprint - Erarbeitung eines ganzheitlichen Meilensteinplans mit Handlungsempfehlungen und notwendigen Forschungsbausteinen zur zielführenden Verknüpfung der Lebenszyklusanalyse (Gebäudeökobilanzierung) und BIM-Planungsprozesse mit einem Fokus auf den frühen Planungsphasen*. https://mediatum.ub.tum.de/1742632

BBSR. (2024). ÖKOBAUDAT. Retrieved June 10, 2025, from https://www.oekobaudat.de/en.html

Beetz, J., Borrmann, A., Koch, C., & König, M. (Eds.). (2018). *Building Information Modeling: Technology Foundations and Industry Practice* (1st ed. 2018). Springer International Publishing; Imprint: Springer.

Beetz, J., Pauwels, P., McGlinn, K., & Tormä, S. (2021). Linked Data im Bauwesen. *Building Information Modeling*, 223–242. https://doi.org/10.1007/978-3-658-33361-4_11

Beetz, J., van Leeuwen, J., & de Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *23*(1), 89–101. https://doi.org/10.1017/S0890060409000122

Berners-Lee, T. (2006). Linked Data - Design Issues. Retrieved June 10, 2025, from https://www.w3.org/DesignIssues/LinkedData.html

Bizer, C., Heath, T., & Berners-Lee, T. (2023). Linked Data - The Story So Far. In O. Seneviratne & J. Hendler (Eds.), *Linking the World's Information* (pp. 115–143). ACM. https://doi.org/10.1145/3591366.3591378

BKI. (2024). BKI Konstruktionsatlas KA2. Retrieved June 10, 2025, from https://bki.de/bki-konstruktionsatlas-ka2

BMWSB. (2023). Developer documentation ILCD+EPD v1.2 MR7 [ZIP archive]. Retrieved June 10, 2025, from https://www.oekobaudat.de/service/downloads.html

BNB. (2025). Assessment System - Bewertungssystem Nachhaltiges Bauen (BNB). Retrieved June 10, 2025, from https://www.bnb-nachhaltigesbauen.de/en/assessment-system/

Boje, C., Navarrete, T., Kubicki, S., & Beach, T. (2023). Linked data for the life cycle assessment of built assets. Retrieved June 10, 2025, from https://www.researchgate.net/publication/375379781_Linked_data_for_the_life_cycle_assessment_of_built_assets

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., … Liang, P. (2021). On the Opportunities and Risks of Foundation Models. https://arxiv.org/pdf/2108.07258

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. https://arxiv.org/pdf/2005.14165

Bundesregierung. (2021). Climate Change Act: climate neutrality by 2045. Retrieved June 10, 2025, from https://www.bundesregierung.de/breg-en/service/archive/climate-change-act-2021-1936846

CEN. (2011). Sustainability of construction works - Assessment of environmental performance of buildings - Calculation method. Retrieved June 10, 2025, from https://standards.globalspec.com/std/1406797/en-15978

CEN. (2012). Sustainability of construction works - Environmental product declarations - Core rules for the product category of construction products (includes Amendment :2019) (2012th ed.). Retrieved June 10, 2025, from https://standards.globalspec.com/std/14216904/din-en-15804

Chiticariu, L., Li, Y., & Reiss, F. (2013). Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems! *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 827–832. https://aclanthology.org/D13-1079/

Daga, E., Asprino, L., Mulholland, P., Gangemi, A., et al. (2021). Facade-X: an opinionated approach to SPARQL anything. *Studies on the Semantic Web*, *53*, 58–73.

Das, S., Sundara, S., & Cyganiak, R. (2012). R2RML: RDB to RDF Mapping Language. Retrieved June 10, 2025, from https://www.w3.org/TR/r2rml/

Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., & van de Walle, R. (2014). RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. *7th Workshop on Linked Data on the Web*, *1184*. https://ruben.verborgh.org/publications/dimou_ldow_2014/

DIN. (2018). Kosten im Bauwesen (12th ed.). Retrieved June 10, 2025, from https://dx.doi.org/10.31030/2873248

Enevoldsen, K., Chung, I., Kerboua, I., Kardos, M., Mathur, A., Stap, D., Gala, J., Siblini, W., Krzemiński, D., Winata, G. I., Sturua, S., Utpala, S., Ciancone, M., Schaeffer, M., Sequeira, G., Misra, D., Dhakal, S., Rystrøm, J., Solomatin, R., … Muennighoff, N. (2025). MMTEB: Massive Multilingual Text Embedding Benchmark. *arXiv preprint arXiv:2502.13595*. https://doi.org/10.48550/arXiv.2502.13595

EPD Norge. (2025). EPD Norge - Forsiden. Retrieved June 10, 2025, from https://www.epd-norge.no/

EU. (2024). Towards zero-emission buildings by 2050: Council adopts rules to improve energy performance [Council Directive PE-102-2023-INIT]. Retrieved June 10, 2025, from https://www.consilium.europa.eu/en/press/press-releases/2024/04/12/towards-zero-emission-buildings-by-2050-council-adopts-rules-to-improve-energy-performance/

EU. (2025). Fit for 55. Retrieved June 10, 2025, from https://www.consilium.europa.eu/en/policies/fit-for-55/

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. https://arxiv.org/pdf/2312.10997

Ghose, A., Lissandrini, M., Hansen, E. R., & Weidema, B. P. (2022). A core ontology for modeling life cycle sustainability assessment on the Semantic Web. *Journal of Industrial Ecology*, *26*(3), 731–747. https://doi.org/10.1111/jiec.13220

Harris, S., & Seaborne, A. (2013). SPARQL 1.1 Query Language. Retrieved June 10, 2025, from https://www.w3.org/TR/sparql11-query/

Hellerstein, J. M., & Stonebraker, M. (2005). *Readings in Database Systems: Fourth Edition*. The MIT Press.

IBU. (2024). Über den Verein | IBU - Institut Bauen und Umwelt e.V. Retrieved June 10, 2025, from https://ibu-epd.com/ibu/

International EPD System. (2025). International EPD System | EPD International. Retrieved June 10, 2025, from https://www.environdec.com/about-us/the-international-epd-system-about-the-system

Islam, S. B., Rahman, M. A., Hossain, K. S. M. T., Hoque, E., Joty, S., & Parvez, M. R. (2024). Open-RAG: Enhanced Retrieval-Augmented Reasoning with Open-Source Large Language Models. https://arxiv.org/pdf/2410.01782

ISO. (2006a). Environmental labels and declarations — Type III environmental declarations — Principles and procedures (2006th ed.). Retrieved June 10, 2025, from https://www.iso.org/standard/38131.html

ISO. (2006b). Environmental management — Life cycle assessment — Principles and framework (2006th ed.). Retrieved June 10, 2025, from https://www.iso.org/standard/37456.html

ISO. (2006c). Environmental management — Life cycle assessment — Requirements and guidelines. Retrieved June 10, 2025, from https://www.iso.org/standard/38498.html

Janowicz, K., Krisnadhi, A. A., Hu, Y., Suh, S., Weidema, B., Rivela, B., Tivander, J., Meyer, D., Berg-Cross, G., Hitzler, P., Ingwersen, W., Kuczenski, B., Vardeman, C., Ju, Y., & Cheatham, M. (2015). A Minimal Ontology Pattern for Life Cycle Assessment Data. *WOP*. https://www.semanticscholar.org/paper/A-Minimal-Ontology-Pattern-for-Life-Cycle-Data-Janowicz-Krisnadhi/09ed6ae8152d4db23edd9a2970b217f076dadcb6

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, *7*(3), 535–547.

JRC. (2010). *International Reference Life Cycle Data System (ILCD) Handbook – General guide for life cycle assessment – Detailed guidance*. Publications Office. https://doi.org/10.2788/38479

Keber, M., Grubišić, I., Barešić, A., & Jović, A. (2024). A Review on Neuro-symbolic AI Improvements to Natural Language Processing. *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, 66–72. https://doi.org/10.1109/MIPRO60963.2024.10569741

Knublauch, H., Allemang, D., & Steyskal, S. (2025). SHACL Advanced Features 1.1. Retrieved June 10, 2025, from https://w3c.github.io/shacl/shacl-af/

Knublauch, H., & Kontokostas, D. (2017). Shapes Constraint Language (SHACL). Retrieved June 10, 2025, from https://www.w3.org/TR/shacl/

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. https://arxiv.org/pdf/2205.11916

Kuczenski, B., Davis, C. B., Rivela, B., & Janowicz, K. (2016). Semantic catalogs for life cycle assessment data. *Journal of Cleaner Production*, *137*, 1109–1117. https://doi.org/10.1016/j.jclepro.2016.07.216

Lambertz, M., Wimmer, R., Theißen, S., Höper, J., Meins-Becker, A., & Zibell, M. (2020). Ressortforschung - Ökobilanzierung und BIM im Nachhaltigen Bauen (Endbericht). Retrieved June 10, 2025, from https://www.bbsr.bund.de/BBSR/DE/forschung/programme/zb/Auftragsforschung/2NachhaltigesBauenBauqualitaet/2019/oekobilanz-bim/01-start.html?nn=436654

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. https://arxiv.org/pdf/2005.11401

LinkML Project. (2024). SHACL generator: LinkML Documentation. Retrieved June 10, 2025, from https://linkml.io/linkml/generators/shacl.html

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Meex, E., Hollberg, A., Knapen, E., Hildebrand, L., & Verbeeck, G. (2018). Requirements for applying LCA-based environmental impact assessment tools in the early stages of building design. *Building and Environment*, *133*, 228–236. https://doi.org/10.1016/j.buildenv.2018.02.016

Miles, A., & Bechhofer, S. (2009). SKOS Simple Knowledge Organization System Reference. Retrieved June 10, 2025, from https://www.w3.org/TR/skos-reference/

Moxon, S. T., Solbrig, H. R., Unni, D. R., Jiao, D., Bruskiewich, R. M., Balhoff, J. P., Vaidya, G., Duncan, W. D., Hegde, H. B., Miller, M., Brush, M. H., Harris, N. L., Haendel, M. A., & Mungall, C. J. (2021). The Linked Data Modeling Language (LinkML): A General-Purpose Data Modeling Framework Grounded in Machine-Readable Semantics. *International Conference on Biomedical Ontology*. https://api.semanticscholar.org/CorpusID:246446353

PostgreSQL Global Development Group. (2024). PostgreSQL: JSON Functions and Operators. Retrieved June 10, 2025, from https://www.postgresql.org/docs/current/functions-json.html

QNG. (2024). Homepage | QNG. Retrieved June 10, 2025, from https://www.qng.info/en/

Radinger, A., Rodriguez-Castro, B., Stolz, A., & Hepp, M. (2013). BauDataWeb. In M. Sabou, E. Blomqvist, T. Di Noia, H. Sack, & T. Pellegrini (Eds.), *Proceedings of the 9th International Conference on Semantic Systems* (pp. 25–32). ACM. https://doi.org/10.1145/2506182.2506186

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. https://arxiv.org/pdf/1908.10084

Röck, M., Saade, M. R. M., Balouktsi, M., Rasmussen, F. N., Birgisdottir, H., Frischknecht, R., Habert, G., Lützkendorf, T., & Passer, A. (2020). Embodied GHG emissions of buildings – The hidden challenge for effective climate change mitigation. *Applied Energy*, *258*, 114107. https://doi.org/10.1016/j.apenergy.2019.114107

Schneider-Marin, P., Stocker, T., Abele, O., Margesin, M., Staudt, J., Abualdenien, J., & Lang, W. (2022). EarlyData knowledge base for material decisions in building design. *Advanced Engineering Informatics*, *54*, 101769. https://doi.org/10.1016/j.aei.2022.101769

Schreiber, G., & Raimond, Y. (2014). RDF 1.2 Primer. Retrieved June 10, 2025, from https://www.w3.org/TR/rdf12-primer/

Streamlit Inc. (2025). Streamlit Documentation. Retrieved June 10, 2025, from https://docs.streamlit.io/

Warren, P., Mulholland, P., Daga, E., & Asprino, L. (2024). Path-based and triplification approaches to mapping data into RDF: User behaviours and recommendations. *Semantic Web*, *15*(6), 2479–2505. https://doi.org/10.3233/SW-243585

Yan, B., Hu, Y., Kuczenski, B., Janowicz, K., Ballatore, A., Krisnadhi, A. A., Ju, Y., Hitzler, P., Suh, S., & Ingwersen, W. (2015). An ontology for specifying spatiotemporal scopes in life cycle assessment. *CEUR Workshop Proceedings*, *1501*, 25–30. https://scholar.ui.ac.id/en/publications/an-ontology-for-specifying-spatiotemporal-scopes-in-life-cycle-as

Zong, C., Margesin, M., Staudt, J., Deghim, F., & Lang, W. (2022). Decision-making under uncertainty in the early phase of building façade design based on multi-objective stochastic optimization. *Building and Environment*, *226*, 109729. https://doi.org/10.1016/j.buildenv.2022.109729