



Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Javaherian Pour, E;Atazadeh, B;Rajabifard, A;Sabri, S;Norris, D

Title:

A Graph Data Model for CityGML Utility Network ADE: A Case Study on Water Utilities

Date:

2025

Citation:

Javaherian Pour, E., Atazadeh, B., Rajabifard, A., Sabri, S. & Norris, D. (2025). A Graph Data Model for CityGML Utility Network ADE: A Case Study on Water Utilities. ISPRS International Journal of Geo-Information, 14 (12), pp.1-29. <https://doi.org/10.3390/ijgi14120493>.

Persistent Link:

<https://hdl.handle.net/11343/362895>

License:

CC BY

Article

A Graph Data Model for CityGML Utility Network ADE: A Case Study on Water Utilities

Ensiyeh Javaherian Pour ^{1,*}, Behnam Atazadeh ¹, Abbas Rajabifard ¹, Soheil Sabri ² and David Norris ³

¹ The Centre for Spatial Data Infrastructure and Land Administration, Department of Infrastructure Engineering, The University of Melbourne, Melbourne, VIC 3010, Australia

² Urban Digital Twin Lab, School of Modelling Simulation and Training, University of Central Florida, Orlando, FL 32816, USA

³ South East Water, Frankston, VIC 3199, Australia

* Correspondence: ensiyehj@student.unimelb.edu.au

Abstract

Modelling connectivity in utility networks is essential for operational management, maintenance planning, and resilience analysis. The CityGML Utility Network Application Domain Extension (UNADE) provides a detailed conceptual framework for representing utility networks; however, most existing implementations rely on relational databases, where connectivity must be reconstructed through joins rather than represented as explicit relationships. This creates challenges when managing densely connected network structures. This study introduces the UNADE-Labelled Property Graph (UNADE-LPG) model, a graph-based representation that maps the classes, relationships, and constraints defined in the UNADE Unified Modelling Language (UML) schema into nodes, edges, and properties. A conversion pipeline is developed to generate UNADE-LPG instances directly from CityGML UNADE datasets encoded in GML, enabling the population of graph databases while maintaining semantic alignment with the original schema. The approach is demonstrated through two case studies: a schematic network and a real-world water system from Frankston, Melbourne. Validation procedures, covering structural checks, topological continuity, classification behaviour, and descriptive graph statistics, confirm that the resulting graph preserves the semantic structure of the UNADE schema and accurately represents the physical connectivity of the network. An analytical path-finding query is also implemented to illustrate how the UNADE-LPG structure supports practical network-analysis tasks, such as identifying connected pipeline sequences. Overall, the findings show that the UNADE-LPG model provides a clear, standards-aligned, and operationally practical foundation for representing utility networks within graph environments, supporting future integration into digital-twin and network-analytics applications.

Keywords: graph database; CityGML; utility networks; relational database; labelled property graph



Academic Editors: Wolfgang Kainz and Dev Raj Paudyal

Received: 28 September 2025

Revised: 7 December 2025

Accepted: 9 December 2025

Published: 11 December 2025

Citation: Javaherian Pour, E.; Atazadeh, B.; Rajabifard, A.; Sabri, S.; Norris, D. A Graph Data Model for CityGML Utility Network ADE: A Case Study on Water Utilities. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 493. <https://doi.org/10.3390/ijgi14120493>

Copyright: © 2025 by the authors.

Published by MDPI on behalf of the International Society for

Photogrammetry and Remote Sensing.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Utility networks are critical infrastructures that provide essential services such as water, electricity, and gas. Modelling connectivity in these networks is essential for their effective management and operation, particularly in the context of growing urban complexity. A variety of spatial data models such as Industry Foundation Classes (IFC), LandInfra, ArcGIS Pro (Version 3.x) with ArcGIS Utility Network Version 7, Infrastructure for Spatial

Information in the European Community (INSPIRE), and PipelineML have been developed to support data management in utility networks.

Among these, the CityGML Utility Network Application Domain Extension (UNADE), developed under the Open Geospatial Consortium (OGC) framework, offers a semantically rich approach for city-scale utility modelling. It supports multiple utility types, a hierarchical network structure, and integration with 3D geospatial data for urban elements such as buildings, tunnels, and bridges [1–3]. The UNADE is defined using the Unified Modelling Language (UML), a conceptual framework that represents entities, attributes, and relationships independently of any specific implementation or storage technology.

In practice, this UML-based schema is commonly translated into relational structures following the relational data model, enabling deployment in relational databases such as 3DCityDB [4]. Through this transformation, classes in the UML schema are mapped to tables, attributes to columns, and associations to foreign keys or join tables [5]. This approach facilitates the use of relational databases and supports compatibility with OGC-compliant data exchange formats such as Geography Markup Language (GML).

Relational databases based on the UNADE UML schema are effective for storing structured, attribute-rich information; however, they are less suited to representing the complex and highly interconnected nature of utility networks. While primary–foreign key constraints support referential integrity, connectivity is not encoded as an explicit structural element and must instead be reconstructed at query time through join operations. In dense utility networks, this join-based reconstruction introduces substantial computational overhead and complicates semantic reasoning, particularly when queries involve multi-level or deeply nested relationships. These limitations are made worse in dense urban environments with highly interconnected utility network systems [6–8]. Consequently, there is growing interest in alternative data structures and storage representations particularly graph-based implementations that provide greater flexibility, semantic richness, and topological awareness [9].

One such alternative is the graph data model, which structures interconnected entities as nodes and relationships and treats connectivity as an explicit component of the data structure. Unlike the relational model, which stores information in tables and reconstructs connectivity through join operations, the graph model encodes relationships directly, providing a natural foundation for representing highly connected utility networks [10]. Graph databases support spatial and semantic relationships, real-time traversal, hierarchical analysis, and flow tracing, enabling operational applications such as leak detection, dependency analysis, failure impact assessment, and integration with digital twin environments [11–13].

Despite these advantages, existing graph-based implementations of CityGML often remain fragmented and module-specific. Many approaches transform only selected parts of the CityGML schema or treat UML relationship types such as association, aggregation, and composition, as generic edges, resulting in partial or inconsistent semantic preservation. Others do not fully support complex ADEs such as UNADE, limiting their applicability for utility network analysis [14,15]. These gaps highlight the need for a consistent, reusable, and standards-aligned graph data model that fully reflects the conceptual structure of CityGML UNADE.

Developing a universal graph data model that complies with the CityGML UNADE specification is therefore scientifically and practically important. Scientifically, it enables the semantic structure, relationship hierarchy, and modelling constraints defined in the UML schema to be represented faithfully within a graph environment, maintaining alignment with OGC standards [1,2]. Practically, a UNADE graph model supports consistent modelling across diverse datasets, enabling water, sewer, and other utility networks to be analysed using the same structural rules. This promotes interoperability, reproducibility,

and a robust analytical foundation for connectivity, flow, and network-level operations. To address this gap, this paper proposes the UNADE-LPG graph data model, a labelled property graph representation specifically designed for the CityGML UNADE version 3.0. The proposed UNADE-LPG graph data model establishes a systematic mapping of UNADE's core features, relationships, and constraints to LPG components (nodes, edges, and properties), forming a generalisable and reproducible framework that preserves both semantic and topological structures. This model is not tailored to a specific dataset but is designed to support a wide range of utility network types defined within the UNADE specification.

Building on the UNADE-LPG graph data model, this study develops a conversion pipeline that transforms CityGML UNADE datasets encoded in GML into LPG instances suitable for loading into a graph database. Using GML as the source format preserves alignment with existing CityGML workflows, while the LPG representation retains the semantic classes, relationship constraints, and topological structure defined in the UNADE UML schema. Two case studies, a schematic example and a real-world water network from Frankston, Melbourne, are used to demonstrate the applicability of the model. In addition to structural and semantic correctness, the Frankston case study also includes an analytical path-finding example that retrieves the shortest sequence of connected pipeline segments between two InteriorFeatureLink elements. This demonstrates how the UNADE-LPG representation supports practical network analysis using the semantic and topological information embedded in the graph. Descriptive graph statistics, including node–edge counts, connected components, and node-degree distributions, further illustrate the internal consistency of the generated graph. The key contributions of this study are:

- A generalisable UNADE-LPG graph data model for the CityGML UNADE, aligning its UML-based schema with the LPG formalism.
- A conversion pipeline that transforms standardised CityGML GML files into LPG instances, enabling the population of graph databases with semantically enriched, topologically structured utility network data.
- Feasibility of the proposed UNADE-LPG graph data model is demonstrated through two use cases: (i) a schematic example scenario and (ii) a real-world water network, together with validation covering structural checks, topological continuity, classification behaviour, descriptive graph statistics, and an analytical path-finding query.

2. Literature Review

2.1. Relational Database 3D Urban Spatial Models

CityGML is a widely adopted standard for semantically and geometrically modelling 3D urban environments. Traditionally, CityGML implementations have relied on relational databases, particularly the 3DCityDB built on PostgreSQL/PostGIS, which supports GML-based imports and tabular schema management [16]. This relational foundation has been widely applied in managing 3D urban data. For instance, in Rotterdam, where utility network data were transformed from 2D shapefiles into 3D CityGML using FME and subsequently stored in an extended version of 3DCityDB to enable analysis and visualisation of both above- and below-ground utility infrastructure [4]. As the complexity of domain-specific extensions to CityGML has grown, efforts have shifted towards automating the generation of relational data model schemas. To support this, a graph-based transformation framework was proposed, leveraging typed and attributed graphs to derive optimised relational structures directly from CityGML ADE schemas. This approach allows spatially enabled relational databases to be dynamically extended while preserving semantic consistency [17]. Building on this relational foundation, recent developments have focused on improving accessibility and usability by integrating 3DCityDB version 5.0 with QGIS

through a dedicated plugin, enabling real-time visualisation and querying of CityGML data within a widely used desktop GIS environment [18].

However, despite their widespread adoption, relational databases exhibit significant limitations in managing highly connected infrastructure such as utility networks [19]. The rigidity of tabular schemas, the need for complex join operations, and the lack of native support for topological traversal hinder performance and scalability in scenarios requiring deep relationships and real-time querying [6,20]. These limitations are particularly critical in the context of smart cities, where seamless integration across utility, sensor, and infrastructure layers is essential [21].

2.2. Graph-Based 3D Urban Spatial Model

To overcome these challenges, graph-based approaches have emerged as a powerful alternative, offering greater flexibility, semantic richness, and traversal efficiency [22,23]. Two prominent graph-based approaches in the scientific domain are the RDF and the LPG model [24].

2.2.1. RDF-Based Approaches

The RDF graph model represents data as triples (subject, predicate, and object) providing a semantic structure that promotes interoperability and data exchange [25]. According to W3C standards, RDF facilitates integration with linked data systems, making it an effective choice for enhancing semantic relationships within CityGML datasets [26]. Several studies have explored RDF-based knowledge graphs to enrich CityGML by linking spatial and non-spatial attributes with external datasets by linking spatial and non-spatial attributes with external datasets, such as OpenStreetMap (OSM). This enrichment improves semantic connectivity while enabling integration across diverse geospatial domains [22]. RDF has also been used to transform CityGML's tree-like structure into a semantic graph. This transformation enables semantic querying and seamless integration with other datasets [27]. Other approaches have proposed integrating RDF with CityGML to create a dynamic geospatial knowledge graph for intelligent city modelling. By transforming the CityGML schema into an RDF-based semantic ontology, these methods enable the integration of multi-domain data for comprehensive urban analysis [28]. The primary emphasis of existing works has been on enriching semantic relationships, with limited attention to the operational and physical connectivity queries required for complexity in utility network datasets. Additionally, RDF's triple-based structure is appropriate for global data integration and machine interface focus. These limitations indicate that RDF, while effective for semantic enrichment, is less suited for dynamic, real-time, travers queries and high-connectivity systems such as those required by CityGML's UNADE [29].

2.2.2. LPG-Based Approaches

LPGs provide a more suitable solution for modelling utility networks. By representing data as nodes, edges, and properties, and by explicitly labelling nodes and edges, LPGs support the traversal of complex relationships, enabling real-time updates, hierarchical dependencies, and functional connectivity [29,30]. Unlike RDF's triple-based format, LPGs allow entities and relationships to carry multiple labelled attributes, which reduces modelling redundancy, simplifies query formulation, and improves execution speed in representing complex utility network structures. By representing CityGML elements as nodes and semantic relationships (e.g., hierarchies, XLinks) as edges, LPG-based approaches made it possible to compare different versions in detail and to apply step-by-step updates [15]. This model was later expanded to support semantic interpretation of changes from multiple perspectives, such as syntactic, structural, and thematic, improving its alignment with

urban digital twin applications [14]. In other efforts, topological connectivity was enhanced by combining IFC and CityGML data within a graph database, using well-defined transformation rules and bidirectional relationships for efficient querying [31]. Similarly, [32] proposed a method to parse UML-based OGC schemas into LPG elements by binding XSD definitions to JSON and converting them into graph nodes and edges, enabling batch insertion and Cypher-based analysis.

2.3. Identified Gaps and Motivation for UNADE-LPG

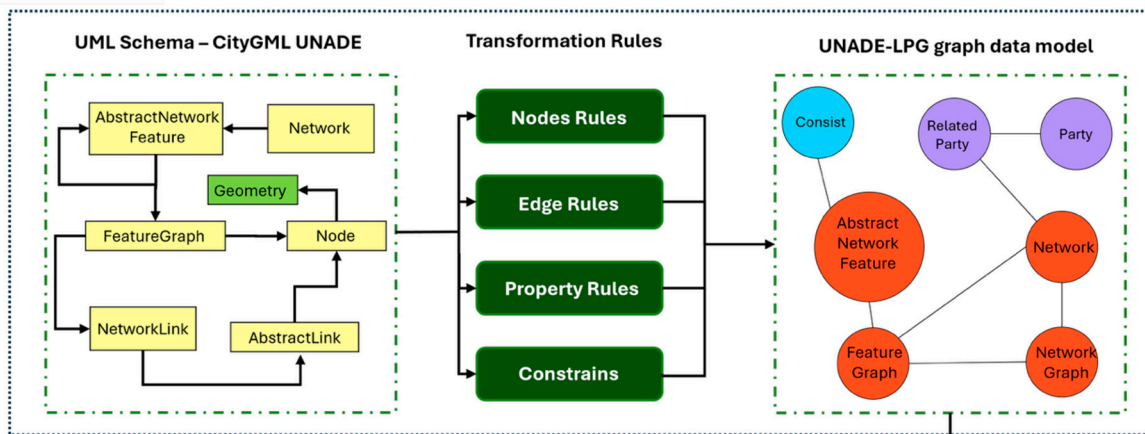
While LPG-based approaches offer a promising alternative to RDF for modelling CityGML UNADE within graph databases, existing efforts fall short in delivering a comprehensive and standards-compliant framework such as the proposed UNADE-LPG graph data model. First, semantic completeness is often not maintained. Many existing studies treat UML relationship types, such as association, aggregation, and composition, as regular or generic edges in the LPG. This approach removes the important differences between these relationship types, which are carefully defined in the UML model. As a result, the original meaning and structure intended by the CityGML schema can be ignored in the graph representation. Second, there is no standard or reusable mapping framework that fully supports all components of CityGML, particularly complex ADEs like UNADE. Without such a framework, elements from these specialised modules are often left unmapped or misrepresented, and most implementations rely heavily on instance-level conversions rather than a generalisable logical model. Third, graph database optimisation principles have often been ignored in previous transformations from UML to graph data model. Important design aspects such as schema normalisation, relationship cardinality control, node degree balancing, and efficient query structures, are rarely considered. This lack of attention can lead to inefficient graph structures that perform poorly and struggle to scale, especially when used with large, real-world smart city datasets.

These limitations highlight the need for a comprehensive, semantically aligned, and performance-aware solution, which motivates the development of the UNADE-LPG graph data model tailored specifically to the CityGML UNADE. The UNADE-LPG graph data model is designed to go beyond direct instance-level transformation by integrating the conceptual semantics of the UNADE UML schema with graph-database design principles that support scalable, query-efficient, and semantically consistent representations.

3. Methodology

The methodological framework adopted in this study follows a two-phase workflow to translate the CityGML UNADE schema into an operational graph-based representation and populate it within a graph database environment (Figure 1). Phase 1 develops the UNADE-LPG graph data model by mapping the UNADE UML concepts into the LPG approach through a set of node, edge, property, and constraint rules that preserve semantic fidelity and topological structure. Phase 2 implements a conversion pipeline that ingests standardised CityGML UNADE GML files and automatically instantiates the UNADE-LPG model using Python 3.13 and Cypher query language within Neo4j 5.26, enabling seamless deployment within Neo4j.

Phase 1: Transforming CityGML UNADE Concepts to UNADE-LPG Graph Data Model



Phase 2: Conversion Pipeline

Figure 1. Methodology workflow for translating CityGML UNADE schema into UNADE-LPG graph data model. In the UML section, yellow denotes classes and green denotes geometry elements. Colours in the node illustration represent the three node types, Directed Nodes [33], Entity Nodes (purple), and Relational Nodes (blue), which are formally defined in Section 3.1.

3.1. Phase 1—Transforming CityGML UNADE Concepts to UNADE-LPG Graph Data Model

This phase outlines the transformation rules to map the CityGML UNADE into the proposed UNADE-LPG graph data model. The aim is to establish a graph representation that reflects the semantic structure and topological patterns of the original UNADE schema. The transformation process consists of optimisation principles, including schema normalisation and relationship cardinality control. These rules reduce redundancy, minimise the creation of unnecessary nodes and edges, and streamline the representation of complex relationships, while maintaining full semantic and spatial fidelity to the UML-based source model.

3.1.1. Nodes Rules

In the LPG approach, nodes serve as the primary building blocks for representing entities, capturing both their semantic attributes and structural roles within the network. To preserve the conceptual clarity and functional richness of the CityGML UNADE schema, nodes are categorised into three distinct types: Directed Nodes, Entity Nodes, and Relational Nodes. This classification ensures that both domain-specific semantics and connectivity patterns are accurately mapped into the graph data model.

- Directed Nodes

Directed Nodes are derived from specific FeatureTypes in the CityGML UNADE schema. In CityGML, FeatureTypes define domain-specific entities such as pipes, valves, or network connections, along with their associated properties and relationships. Directed Nodes represent the FeatureTypes that inherently convey directional relationships within the utility network. For example, in the UNADE Core module, InterFeatureLink and

NetworkGraph are transformed into Directed Nodes as they structurally define connections between components with an identifiable source and target. In the resulting graph data model, each Directed Node has a label corresponding to its original FeatureType, thereby preserving the semantic role defined in the UML schema within the LPG representation.

- Entity Nodes

Entity Nodes are generated from properties in a FeatureType that refer to other DataTypes or FeatureTypes, particularly when the referenced elements possess their own attributes or internal structure. In the CityGML schema, DataTypes are reusable class-like constructs that group related attributes but are not standalone spatial features. They typically represent complex attribute structures within a feature. Entity Nodes in the graph data model do not represent directional flow; instead, they hold critical metadata and hierarchical relationships. For instance, the Network feature includes a property relatedParty: RelatedParty [0..*]. Rather than treating this as a primitive attribute, it is modelled as a separate RelatedParty node, with attributes such as role: RoleValue and share: Scale. Furthermore, the party: Party property within RelatedParty refers to another feature, so it is modelled as a separate Party node. This Party node may include attributes such as url: URL [0..1] and connects to a pointOfContact node, which stores contact information (e.g., email, phone, contactRole, etc.).

- Relational Nodes

In the proposed UNADE-LPG graph data model, Relational Nodes are introduced as an additional modelling construct to explicitly represent key relationships within the utility network that benefit from being treated as first-class entities in the graph. In the original UML schema of CityGML UNADE, such relationships are defined as associations between FeatureTypes. However, when these associations are semantically meaningful, occur frequently, or require metadata (e.g., timestamps, roles, or status), the proposed model represents them as nodes to support richer querying and semantic interpretation. A representative example is the subnetwork relationship, which logically connects a group of infrastructure components (e.g., pipes, valves, and pumps) within the same Network feature. In this case, a Subnetwork node is introduced to serve as an intermediary, linking all participating features. This approach not only maintains semantic clarity but also enables more sophisticated querying. For instance, users can efficiently retrieve all assets belonging to a specific subnetwork, filter based on subnetwork type or function, and perform topological traversals constrained within or across subnetworks. Table 1 presents all node types derived from the CityGML UNADE UML schema and their corresponding representations in the proposed UNADE-LPG graph data model. As shown, different types of nodes are generated from FeatureType, DataType, and Association or Composition relationships defined in the UML schema. This mapping reflects a transformation process that is guided both by the semantic structure embedded in the UML model and the practical requirements of querying utility networks within graph database systems.

Table 1. Mapping of CityGML UNADE UML elements to UNADE-LPG graph data model node type.

UML Element Type	UML Class Name	Mapped UNADE-LPG Graph Data Model
FeatureType	AbstractNetworkFeature	Directed Nodes
	Network	
	NetworkGraph	
	FeatureGraph	
	Node	
	InteriorFeatueLink	
	InterFeatureLink	
DataType	NetowrkLink	Entity Nodes
	RelatedParty	
	ContactType	
FeatureType	Party	Relational Nodes
Association	subOrdinateNetwork	
	superOrdinateNetwork	
Composition	subNetwork	
	consistsOf	

3.1.2. Edges Rules

In the proposed UNADE-LPG graph data model, edges represent the relationships between nodes, encoding both semantic meaning and structural logic. When translating the CityGML UNADE schema into an LPG model, each edge is explicitly labelled according to the property or association name defined in the original UML schema. This labelling ensures semantic traceability and consistency with the source model. In addition, each edge includes a type property that records its UML relationship type such as association, aggregation, or composition, to support semantic reasoning and structural interpretation during graph-based analysis.

The graph data model is implemented as a directed graph, consistent with the LPG principles. Edge direction is preserved from the UML schema by mapping the source class (e.g., the class that contains the reference) to the target class (e.g., the referenced or linked class). This directional mapping supports effective path traversal, functional flow modelling, and topological reasoning.

Importantly, the graph structure in the graph data model is not a one-to-one reflection of the original UML schema. While the UML model was primarily designed for XML-based data exchange, the graph data model introduces optimisations tailored for graph querying. As such, the graph data model includes additional edges that may not be explicitly defined in the UML schema but are inferred based on practical usage patterns and semantic logic.

One key optimisation in the proposed UNADE-LPG graph data model is the direct edge between Network and FeatureGraph, bypassing the intermediary AbstractNetworkFeature. Semantically, this connection represents an aggregation, reflecting that FeatureGraph forms a logical component of the Network but may also exist independently or participate in multiple network contexts.

As introduced earlier, Entity Nodes and Relational Nodes represent new structures not directly present in the original UML model. To incorporate them into the graph data model, additional edges are defined that link these nodes back to their source elements. Although these edges are not explicitly part of the CityGML schema, they are derived from its conceptual intent and follow appropriate UML relationship types. Whether the

connection represents ownership (composition), containment (aggregation), or a general association, these distinctions are maintained as edge properties in the LPG model.

- *RelatedParty* Relationship

The *RelatedParty* property is found in the *Network* and *AbstractNetworkFeature* classes. In the LPG model, this property is transformed into a distinct *RelatedParty* node, which contains its own set of attributes. Based on UML semantics, the relationship between *Network* and *RelatedParty* is best modelled as an aggregation. According to UML standards, aggregation denotes a whole–part relationship in which the part (here, *RelatedParty*) is associated with but not dependent on the lifecycle of the whole (*Network*). Aggregation is non-exclusive, meaning the same part may be associated with multiple wholes, and its existence does not depend on the container. This interpretation aligns with the nature of a *RelatedParty*, which may participate in multiple network features and is conceptually independent, it can exist outside the specific context of a given network.

Thus, in the graph data model, the edge between *Network* and *RelatedParty*, or between *AbstractNetworkFeature* and *RelatedParty*, is labelled using the property name *relatedParty*, consistent with the CityGML UNADE schema. The type of this relationship is explicitly classified as aggregation, ensuring that the graph data model reflects both the structural flexibility and semantic independence of the *RelatedParty* node.

- *Party* and *ContactType* Relationship

The relationship between *RelatedParty* and *Party* in the CityGML UNADE is interpreted as a UML association. According to UML standards, association represents a general link between two classes where neither class has ownership or lifecycle control over the other. In this case, *RelatedParty* refers to a *Party* entity (e.g., an organisation or individual), but it does not control its existence nor imply containment. The *Party* may exist independently and can be referenced by multiple *RelatedParty* instances across different networks. Therefore, in the LPG model, the edge from *RelatedParty* to *Party* is labelled using the property name *party*, and its relationship type is stored as *association*, preserving the loose coupling and reusability defined in the original schema.

In contrast, the relationship between *Party* and *ContactType* (represented by the *pointOfContact* attribute in the UML schema) is more appropriately modelled as composition. UML composition represents a strong ownership and lifecycle dependency, where the composite class controls the existence of the component. Here, the *ContactType* contains detailed contact information (e.g., name, role, phone, email) and is entirely dependent on the *Party*. It has no standalone identity or reuse across multiple parent objects. When the *Party* is removed, the associated *ContactType* should be deleted as well. Accordingly, in the graph data model, the edge from *Party* to *ContactType* is labelled *pointOfContact* and the relationship type is stored as *composition*, enforcing this lifecycle dependency and preserving the semantics of tightly coupled structural components.

Table 2 summarises how relationships defined in the CityGML UNADE UML schema are represented as edges in the proposed UNADE-LPG graph data model. Each row specifies the source and target nodes, the corresponding edge label, and, where applicable, the semantic classification of the relationship

Table 2. Mapping of CityGML UNADE UML relationships to UNADE-LPG graph data model.

UNADE-LPG Graph Data Model Node Source	UNADE-LPG Graph Data Model Node Target	UNADE-LPG Graph Data Model Edge Label	UNADE-LPG Graph Data Model Edge Attribute
NetworkGraph	FeatureGraph	featureGraph-NG	Aggregation
Network	FeatureGraph	networkFeature	
Network	SubNetwork	subnetwork	
Network	RelatedParty	relatedparty	
AbstractNetworkFeature	RelatedParty	relatedparty	
AbstractNetworkFeature	ConsistOf	consistof	
FeatureGraph	InteriorFeatueLink	interiorFeatueLink	Composition
FeatureGraph	NetworkLink	networkLink	
FeatureGraph	Node	node	
NetworkGraph	InterFeatureLink	interFeatureLink	
Party	ContactType	pointOfContact	
AbstractNetworkFeature	FeatureGraph	featureGraph_ANF	
InteriorFeatueLink	Node	startNode/endNode	Association
NetowrkLink	Node	startNode/endNode	
InterFeatureLink	Node	startNode/endNode	
Network	SuperOrdinateNetwork	superOrdinateNetwork	
Network	SubOrdinate Network	subOrdinateNetwork	
Network	NetworkGraph	networkGraph	
RelatedParty	Party	party	

3.1.3. Property (Attribute) Rules

In the proposed UNADE-LPG graph data model, attributes in the CityGML UNADE schema are represented as properties attached to both nodes and edges. For nodes, attributes such as function, measuredLength, or dateOfConstruction are stored as key–value properties, supporting efficient filtering, retrieval, and analytical queries. For edges, properties play an important role when a relationship includes additional metadata beyond connectivity. For example, edge properties may store the UML relationship type (e.g., association, aggregation, or composition). Representing these attributes as edge properties maintains semantic fidelity of the CityGML schema while enabling fine-grained graph queries in graph databases.

- AbstractLink

In the CityGML UNADE schema, the AbstractLink class serves as a superclass for InteriorFeatureLink, InterFeatureLink, and NetworkLink, providing shared attributes such as direction and the geometric primitive (GM_Curve). In the proposed UNADE-LPG graph data model, AbstractLink is not represented as a separate node. Instead, its shared attributes are incorporated directly into the inheriting nodes. For example, geometric primitive and direction are stored as a property of each link node.

In the CityGML UNADE schema, classes such as AbstractLink serve as abstract superclasses that define shared attributes and conceptual semantics for their concrete subclasses (InteriorFeatureLink, InterFeatureLink, and NetworkLink). Although these abstract classes are semantically meaningful within the UML hierarchy, they are not instantiable features in real datasets. Representing them as standalone nodes in the LPG model would introduce additional traversal steps and index overhead without contributing analytical value.

This approach aligns with best practices in LPG modelling, where abstract inheritance structures are typically flattened to avoid unnecessary intermediate nodes [34,35]. Accordingly, the inherited attributes of `AbstractLink` (for example, direction and geometric primitive) are embedded directly into the concrete link nodes during transformation. This flattening preserves the semantic intent of the UML inheritance hierarchy while improving structural clarity and query efficiency. As a result, each link type becomes a self-contained, semantically complete, and directly traversable entity, maintaining both topological meaning and directional flow without requiring indirect inheritance queries.

- Node

The `Node FeatureType` is represented in the graph data model as a `Directed Node`, enriched with both semantic and spatial properties. Key attributes include `type`, which specifies whether the node is interior or exterior, and `connectionSignature`, derived from the `SignatureType` enumeration (e.g., `230 V/50 Hz_SinglePhase`, `120 V/60 Hz_SinglePhase`, `100 Bar`, `10 Bar`). To capture spatial positioning, the node geometry is modelled using the `GM_Primitive` class, specifically `GeometricPrimitive::GM_Point`. This ensures that each node in the LPG is not only semantically defined but also spatially anchored, enabling accurate representation of topological connectivity within utility networks.

- Cardinality

In the graph data model representation, cardinality constraints defined in the CityGML UNADE schema are explicitly captured as edge properties to ensure that relationship multiplicities are preserved during graph-based querying and analysis. For each relationship, cardinality is recorded separately for incoming and outgoing edges relative to the connected node types:

1. `in-edge-(ConnectedNode)`: Specifies the maximum or exact number of incoming edges from a given node type (e.g., `in-edge-InteriorFeatureGraph` defines the allowable number of incoming connections from `InteriorFeatureGraph` to the current node).
2. `out-edge-(ConnectedNode)`: Specifies the maximum or exact number of outgoing edges from the current node to a given node type (e.g., `out-edge-FeatureGraph` indicates the allowable number of outgoing connections from the current node to `FeatureGraph`).

Encoding cardinality in this way provides two main advantages. First, it enables schema-level validation within the graph database, allowing automated checks for violations of UML-defined multiplicities during data import or modification. Second, it enhances query precision and optimisation, as traversal paths can be constrained to valid relationship patterns without the need for additional filtering logic.

For optimisation-driven relationships introduced in the graph data model (i.e., those not explicitly defined in the UML but added for query efficiency), corresponding cardinalities are also defined to preserve structural integrity. For example:

A `Network` or `AbstractNetworkFeature` may be connected to one or many `RelatedParty` nodes, but each `RelatedParty` is associated with at most one `Network` or `AbstractNetworkFeature`. Each `RelatedParty` must be linked to exactly one `Party` node, while a `Party` can be associated with multiple `RelatedParty` nodes. Each `Party` must contain exactly one `ContactType` node, and each `ContactType` belongs to exactly one `Party`. Each `FeatureGraph` is associated with exactly one `Network`, while a `Network` may contain multiple `FeatureGraph` nodes.

Cardinality values are encoded by translating the multiplicities defined in the UNADE UML schema into explicit property on the corresponding edges. Fixed multiplicities, such as the requirement that every `InteriorFeatureLink` must connect to exactly two `Node` elements (start and end) and one `FeatureGraph`, are transferred directly as minimum and maximum

values attached to those relationships during the construction of the graph structure. For relationships whose upper bounds depend on the structure of the input dataset such as the number of InteriorFeatureLink segments belonging to a FeatureGraph or the number of pipelines converging at a junction node the cardinality values are derived from the observed instance-level connectivity during import. In these cases, the minimum values reflect the UML constraints, while the maximum values capture the set of valid connections present in the data. Encoding multiplicities in this manner allows the instantiated graph to retain the semantic intent of the original UML model while supporting subsequent validation of structural correctness, ensuring that each relationship adheres to both the conceptual rules of UNADE and the actual topology of the utility network.

3.1.4. Constraints

In the CityGML UNADE UML schema, semantic and topological constraints are defined to ensure model validity. For example, the InteriorFeatureLink class specifies that “both nodes must belong to the same FeatureGraph,” thereby enforcing intra-feature connectivity and preventing invalid cross-graph links. All such constraints from the original UML schema are retained in the LPG representation to guarantee semantic equivalence between the two models.

However, the translation of CityGML UNADE into an LPG structure also requires additional graph specific constraints. These constraints safeguard the integrity, consistency, and practical applicability of the graph data model, particularly when additional edges or optimisations are introduced to improve query performance. Without these constraints, the graph could permit invalid or ambiguous configurations, weakening its reliability for network analysis and operational use. The additional constraints fall into the following categories:

1. Composition Rules

In UML, composition represents a whole–part dependency in which the child’s lifecycle is bound to the parent. In the LPG model, this principle is retained: if a parent node in a composition relationship is deleted, all dependent child nodes must also be removed. However, if a child node maintains other valid connections beyond the composition, it remains in the graph to preserve those relationships. This approach ensures that deletion operations reflect UML semantics without introducing unnecessary data loss.

2. Aggregation and Association Rules

In UML, aggregation and association represent weaker dependencies than composition, where the child node can exist independently of the parent. In the LPG model, these relationships do not trigger automatic deletion if the parent node is removed. Instead, the relationship type (aggregation or association), inherited from UML semantics, is stored as a property to preserve its meaning. This distinction is important in utility networks, where shared resources must be modelled correctly. For example, a RelatedParty node may be linked to multiple Network or AbstractNetworkFeature nodes, and it should not be deleted simply because one of those features is removed.

Figure 2 presents the proposed UNADE-LPG graph data model for the CityGML UNADE after applying the transformation rules. The model captures the main feature classes (Network, AbstractNetworkFeature, FeatureGraph, and Node) as Directed Nodes [33], ensuring that connectivity and flow direction are explicitly encoded. Supporting classes such as RelatedParty, Party, and ContactType are represented as Entity Nodes (purple), while associations such as SubNetwork and SuperOrdinateNetwork are expressed as Relational Nodes (blue) to improve semantic reasoning and query granularity. Edges are labelled according to their source UML relationships (e.g., startNode, endNode, consistsOf, related-

Party), with relationship semantics (association, aggregation, composition) preserved as edge properties.

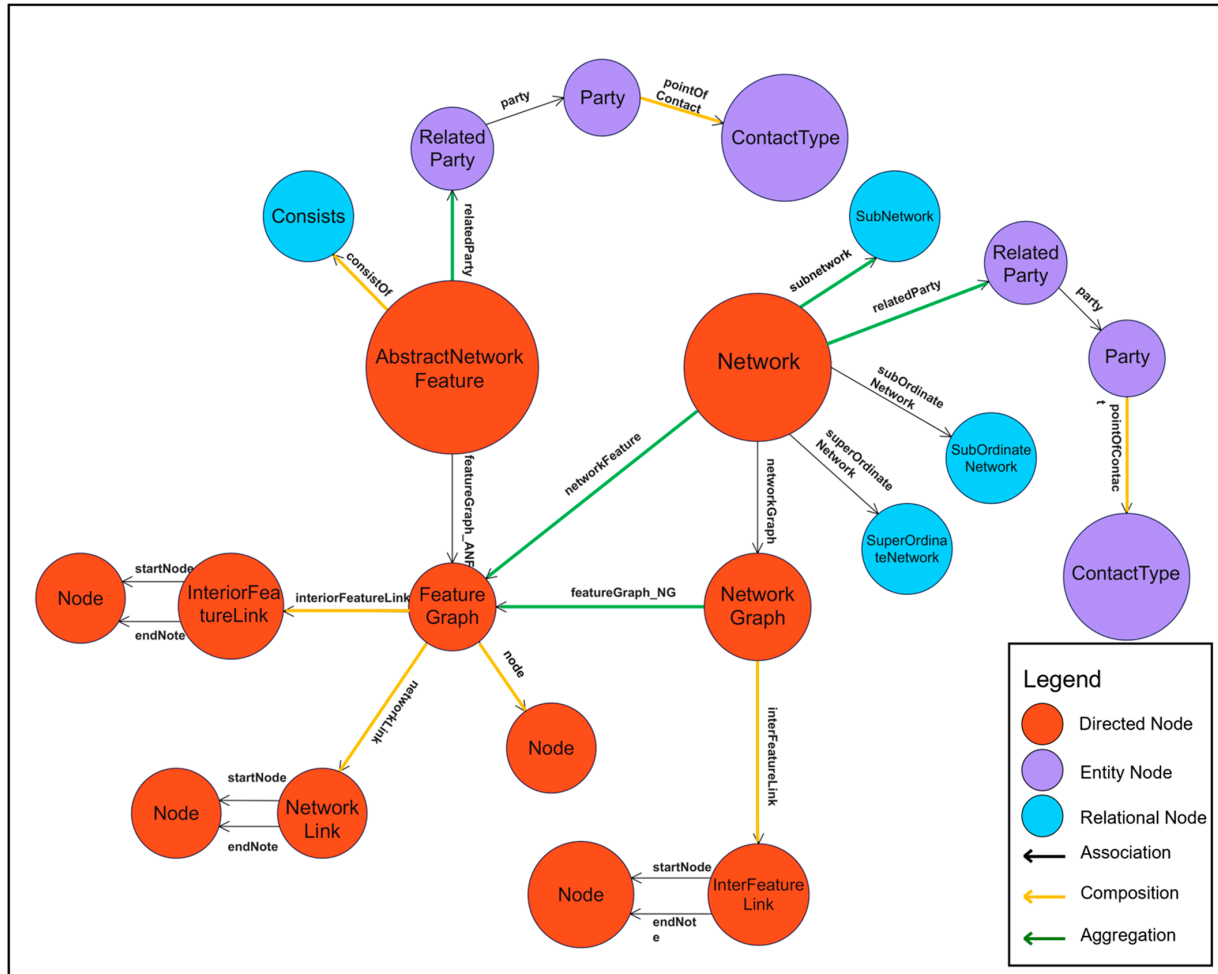


Figure 2. Proposed UNADE-LPG graph data model.

3.2. Phase 2—Conversion Pipeline

The second phase operationalises the UNADE-LPG graph data model by transforming standardised CityGML UNADE datasets encoded in GML into LPG instances suitable for population in a graph database environment. To maintain interoperability with existing OGC-compliant workflows, the pipeline accepts CityGML UNADE GML files as input, bridging the UML-based schema and its graph-based representation.

Figure 3 summarises the conversion pipeline workflow. The process begins with parsing the CityGML GML file and resolving the required namespaces. The pipeline then extracts the core UNADE feature classes—such as Network, FeatureGraph, Node, NetworkLink, InteriorFeatureLink, and InterFeatureLink, and converts their geometries into Well-Known Text (WKT) for spatial compatibility. Each extracted class is subsequently mapped to its corresponding LPG construct following the UNADE-LPG transformation rules: classes become labelled nodes, attributes become node or edge properties, and UML-defined associations become directed edges. The final stage prepares these instantiated elements for graph population, during which the required graph operations are generated to create nodes, assign properties, and establish relationships in accordance with the defined cardinality and constraint rules. This workflow ensures that the semantic structure, geometry, and connectivity of the original CityGML UNADE schema are systematically

propagated into the resulting graph representation, independent of the specific graph-database platform used for storage.

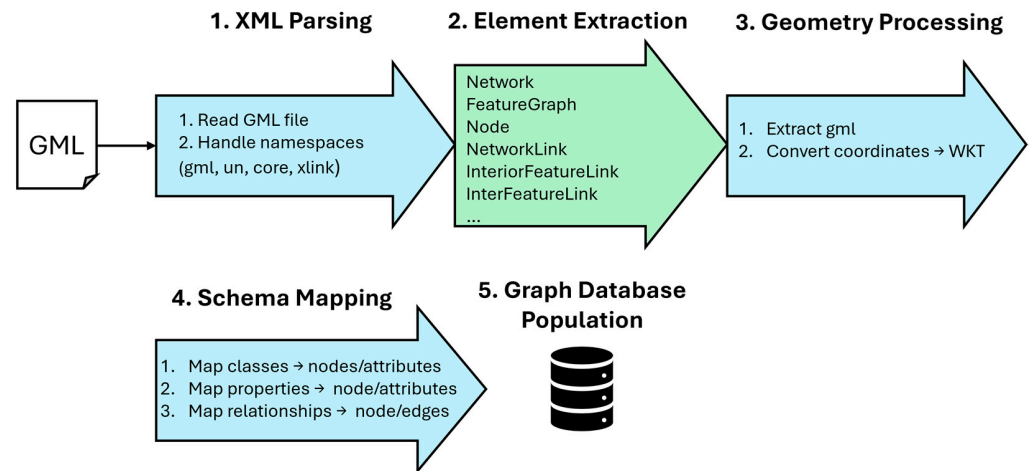


Figure 3. Conversion pipeline workflow.

To maintain interoperability with existing OGC-compliant workflows, the pipeline accepts CityGML UNADE GML files as input, serving as the bridge between the UML-based schema and its graph-based representation. After parsing the dataset, utility network elements, including Network, FeatureGraph, Node, and NetworkLink, are extracted, and their geometric primitives are converted into WKT to support spatial integration within the graph environment. Each extracted feature is mapped to its corresponding LPG construct in accordance with the UNADE-LPG rules, ensuring that semantic attributes and topological relationships (e.g., featureGraph, startNode, endNode) are preserved in the resulting structure.

Once the mapping is complete, the pipeline operationalises this structure by generating the necessary graph-population operations to instantiate nodes, assign properties, and create directed edges that represent connectivity. The workflow concludes with the construction of a complete, semantically aligned graph representation of the original CityGML UNADE dataset, enabling subsequent structural and topological analysis within the graph database environment.

4. Implementation and Results

This section presents the implementation of the proposed UNADE-LPG graph data model and the results obtained from applying it to two case studies. The section begins with an overview of the implementation workflow used to instantiate the model from standardised CityGML UNADE GML datasets. It then presents two case studies demonstrating how the model captures semantic structures, branching configurations, cross-pipeline connectivity, and geometric relationships in schematic and real-world utility networks. Finally, the section reports validation outcomes assessing the structural correctness, semantic consistency, topological integrity, and database-level behaviour of the instantiated graphs. Together, these components provide a comprehensive evaluation of the proposed model across both controlled and operational scenarios.

4.1. Implementation Overview

The implementation of the proposed UNADE-LPG graph data model operationalises the methodological framework by translating standardised CityGML UNADE datasets into an instantiated labelled property graph suitable for analysis within a graph database environment. This implementation follows a structured workflow that mirrors the trans-

formation rules defined in the Methodology. The pipeline begins with the ingestion of CityGML UNADE files encoded in GML, ensuring consistency with OGC-compliant data exchange practices. The GML content is parsed to extract the core utility components defined in the UNADE schema, including Network, FeatureGraph, InteriorFeatureLink, InterFeatureLink, and Node elements. Their semantic attributes and geometric primitives are then interpreted and converted into representations compatible with the UNADE-LPG structure, thereby preserving both semantic fidelity and spatial coherence during the transformation process.

Following data extraction, the semantic and topological components are mapped to nodes and relationships in alignment with the UNADE-LPG model. Each entity is instantiated with its relevant attributes, while edges are created to reflect the directional and structural relationships defined in the UML schema. Multiplicity rules, inheritance flattening, and relationship semantics (association, aggregation, composition) are incorporated during this step to maintain the conceptual integrity of the original schema. Once the graph structure is prepared, the graph population stage is carried out within Neo4j, where the nodes, edges, and properties are created, indexed, and validated. This process completes the system-level implementation of the UNADE-LPG graph data model and provides the foundation upon which the case studies and validation results presented in the following subsections are based.

4.2. Case Study 1: Schematic Example

The first case study (Figure 4) employs a schematic utility network model adapted from [36], representing two pipeline segments within the same network, one of which contains a branch connection. This example captures two recurrent connection patterns in utility networks:

- Inter-pipeline connectivity between two separate pipeline segments, each represented by a distinct FeatureGraph, connected via an InterFeatureLink.
- Branching within a pipeline, modelled using an InteriorFeatureLink with a shared interior node.

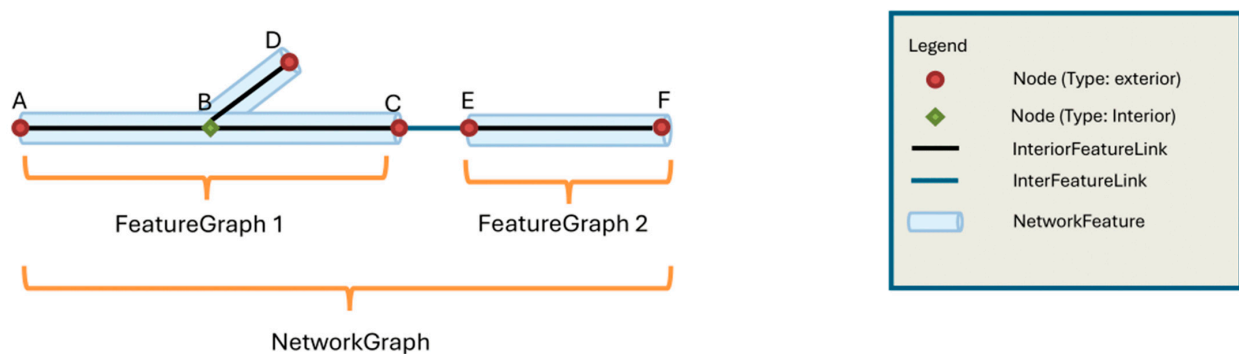


Figure 4. Case Study 1: Schematic example.

Figure 5 presents the UNADE-LPG graph representation of this schematic example. The purple nodes denote elements belonging to FeatureGraph 1. Here, Node B functions as a shared node linking three InteriorFeatureLink segments and is classified as an interior node. The connection between FeatureGraph 1 (purple) and FeatureGraph 2 (blue) occurs within the same NetworkGraph and is represented using an InterFeatureLink (orange).

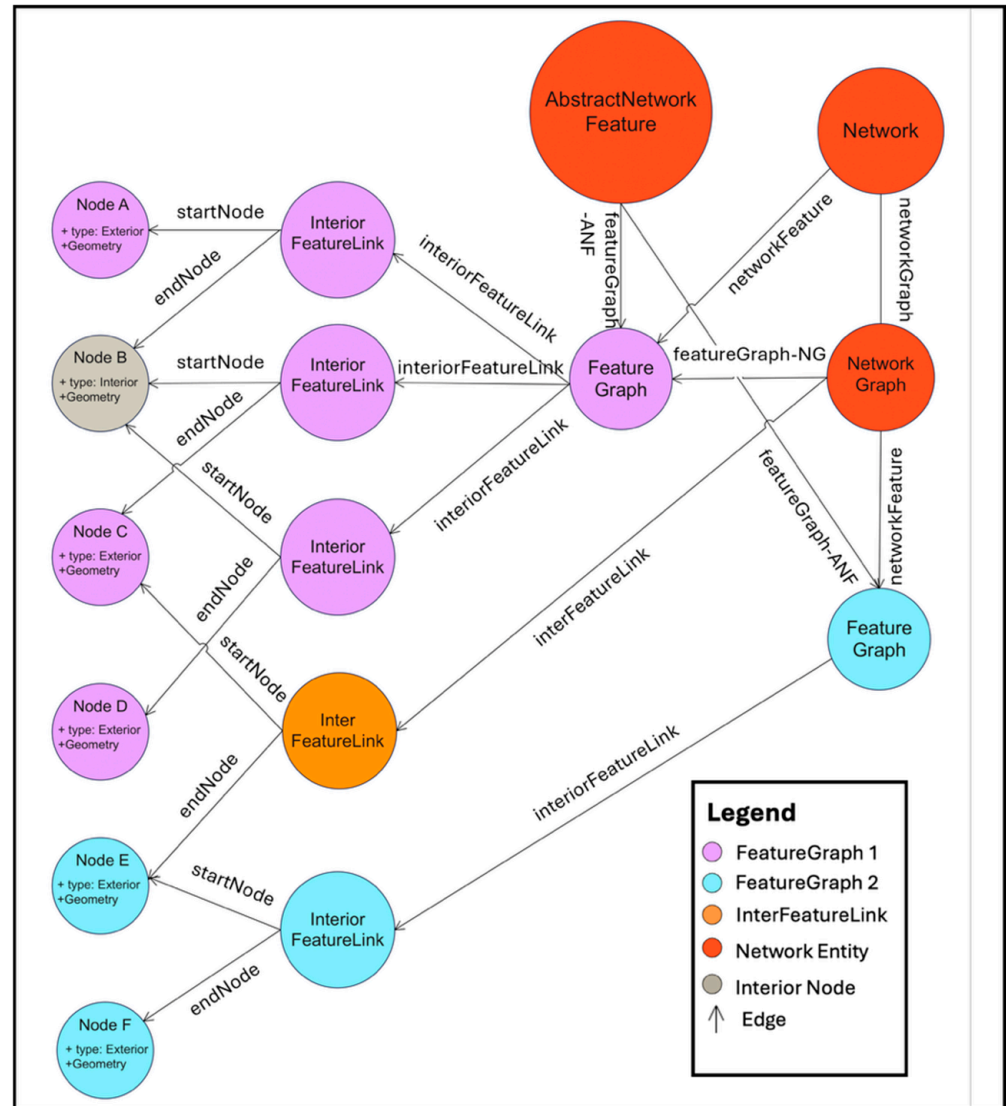


Figure 5. Proposed UNADE-LPG graph data model for schematic example.

FeatureGraph 2 is modelled more simply, comprising only two exterior nodes (E and F) connected by a single InteriorFeatureLink.

4.3. Case Study 2: CityGML UNADE GML File

The second use case (Figure 6) focuses on the water pipeline network in Frankston, covering the extended area defined by the coordinates (329,845.4635, 5,768,543.6843)–(330,440.6768, 5,768,922.7040). The dataset was provided by South East Water (SEW), a government-owned corporation responsible for supplying water, wastewater, and recycled water services to Melbourne’s south-eastern suburbs.

Figure 7 presents the corresponding CityGML GML file encoded according to the CityGML UNADE 3.0, while Figure 6 illustrates the same water pipeline network in a real-world 3D context. GML, an XML-based OGC standard, is widely adopted for encoding and exchanging geospatial information. Its popularity stems from its interoperability, semantic richness, and compatibility with CityGML, making it the default exchange format for 3D city models and utility networks.



Figure 6. Real-world water utility network in Frankston, Australia.

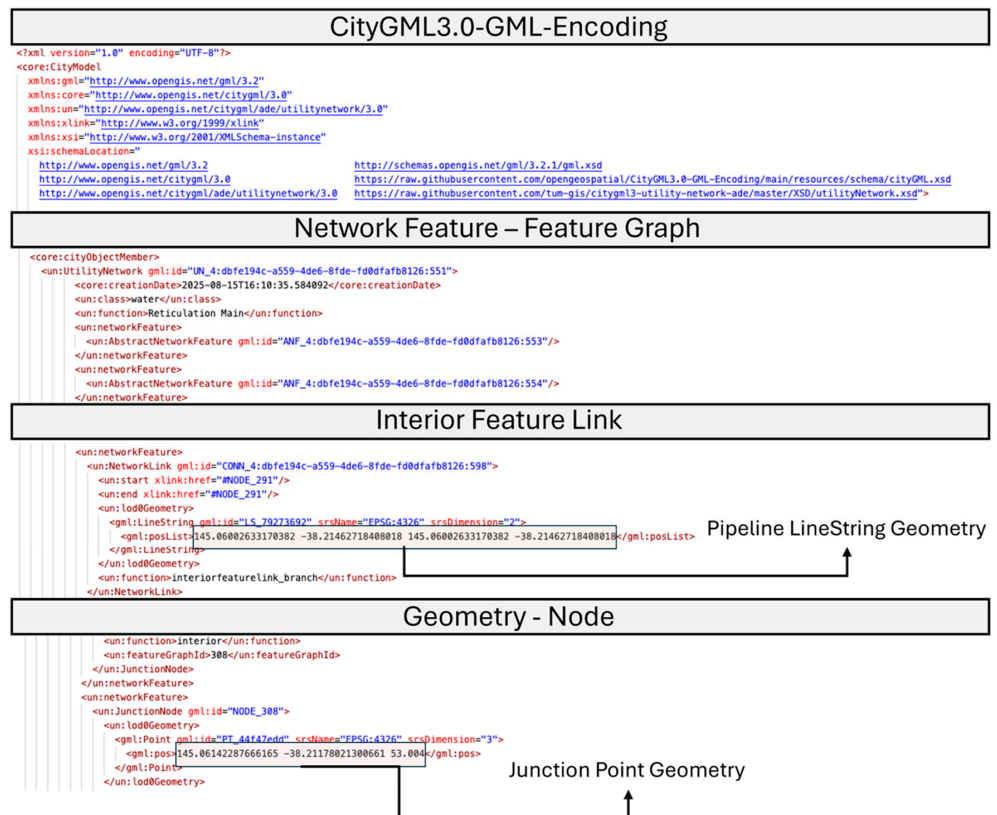


Figure 7. Water utility network CityGML GML file.

Figure 8 presents the developed conversion pipeline, referred to as the CityGML-to-Neo4j Importer, which provides a user-friendly interface for populating the proposed

UNADE-LPG graph data model within a graph database environment. The importer used in this study corresponds to the initial release (version 1.0) of the custom-developed tool. The interface supports connections to both local and remote Neo4j instances through user-specified URLs and login credentials. Once a CityGML UNADE dataset encoded in GML is selected, the connection can be tested to confirm successful communication with Neo4j prior to launching the import. During execution, the application parses the CityGML GML file, resolves XML namespaces, and automatically instantiates the corresponding LPG structure in Neo4j, including both the schema elements (e.g., FeatureGraph, InteriorFeatureLink, Node) and their associated data instances.

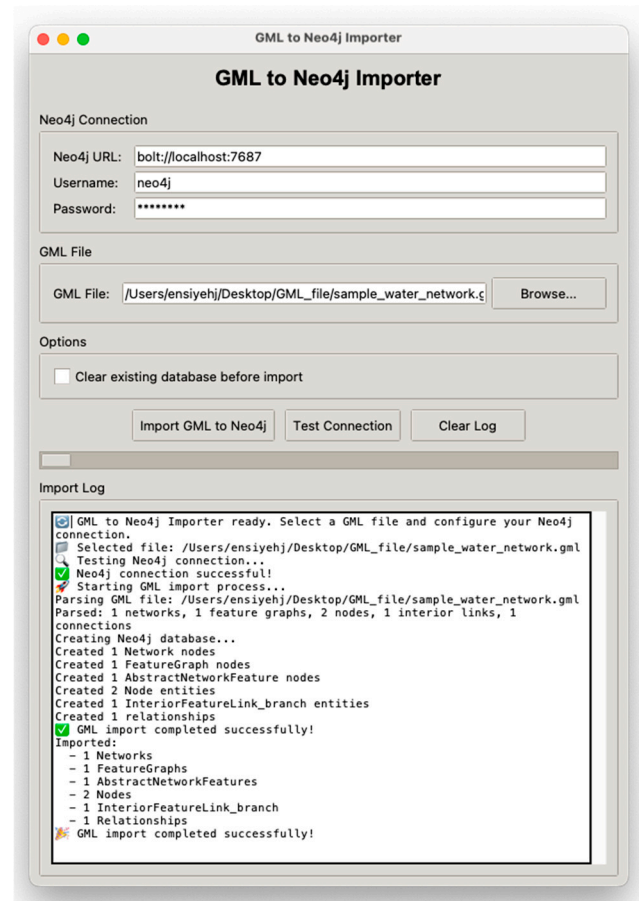


Figure 8. Populate the Neo4j using the CityGML GML file.

The import log displayed in the interface provides feedback on each stage of the process. It lists the CityGML elements that were successfully parsed (e.g., Network, FeatureGraph, InteriorFeatureLink, Node), confirms the creation of nodes and relationships in the graph database, and reports the total number of entities imported. Any errors or inconsistencies during import are also flagged in real time, allowing users to validate data quality. This workflow not only ensures transparency but also helps in debugging and confirming semantic alignment between the original CityGML schema and the populated LPG model. The tool has been made publicly available on GitHub 10 to encourage reproducibility and adoption in other utility network projects.

Figure 9 illustrates the schema of the water utility network dataset used in the case study. The schema is generated when the dataset is imported into Neo4j using the developed conversion pipeline. It represents the instantiation of the proposed UNADE-LPG graph data model within a graph database, following the LPG formalism. While the exact schema may vary depending on the characteristics of the input dataset, the overall structure

consistently reflects the transformation rules defined in the proposed UNADE-LPG graph data model. In this representation, the NetworkGraph is connected to both InterFeatureLink and FeatureGraph, while the FeatureGraph itself is associated with Network and AbstractNetworkFeature. The Network stores the functional role of the pipeline, for example, whether it serves as a fire service line or a reticulation main. The AbstractNetworkFeature records asset-related properties, including the ownership (e.g., SEW or private entities) and the operational state of the network, such as operational (OPR), abandoned (ABND), or migration (MIG). The FeatureGraph property `id_obj` represents a pipeline identifier that aggregates smaller pipeline segments (InteriorFeatureLinks). Branch connections are also modelled as InteriorFeatureLink_branch objects, which are functionally equivalent to InteriorFeatureLinks but differentiated in labelling to facilitate targeted querying of branch structures. Both types of InteriorFeatureLinks are linked to Node entities via `startNode` and `endNode` relationships, thereby preserving the geometry of each pipeline segment. Furthermore, the type property within Node indicates whether a point is classified as interior (branch connection) or exterior (pipeline-to-pipeline connection), ensuring accurate representation of topological relationships.

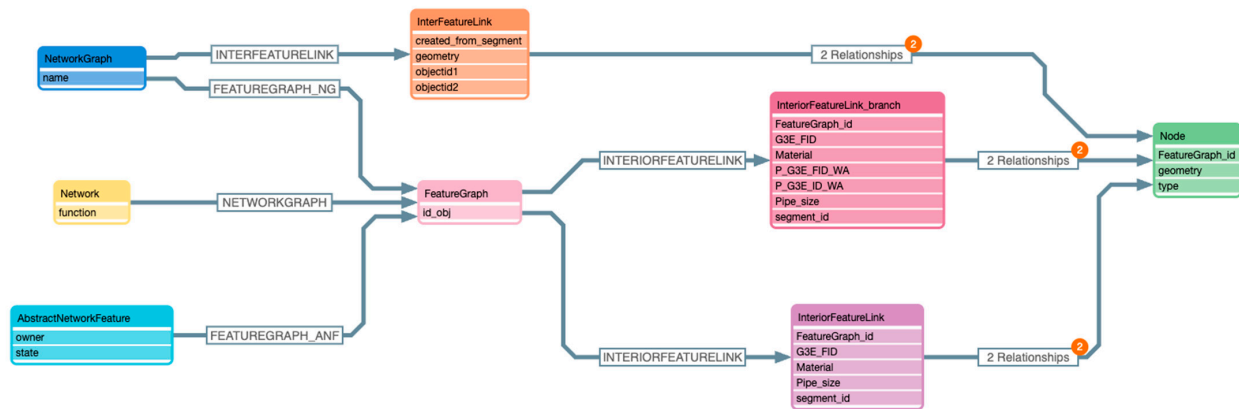


Figure 9. Water utility network UNADE-LPG graph data model schema. The number “2” denotes that two relationships connect each InterFeatureLink, InteriorFeatureLink, or InteriorFeatureLink_branch entity to the corresponding Node, representing the `startNode` and `endNode` associations defined in the UNADE-LPG schema.

Figure 10 presents an instantiation of the proposed UNADE-LPG graph data model for a real-world pipeline sample, illustrating three intersecting networks (yellow, red, and green) modelled in the LPG structure. This area was selected because of its structural complexity: it includes pipelines with multiple branching connections (e.g., the yellow and red pipelines) as well as simpler linear segments without branches (e.g., the green pipeline). The sample thus captures a diverse set of patterns representative of utility networks, providing a robust test case for evaluating the proposed UNADE-LPG graph data model.

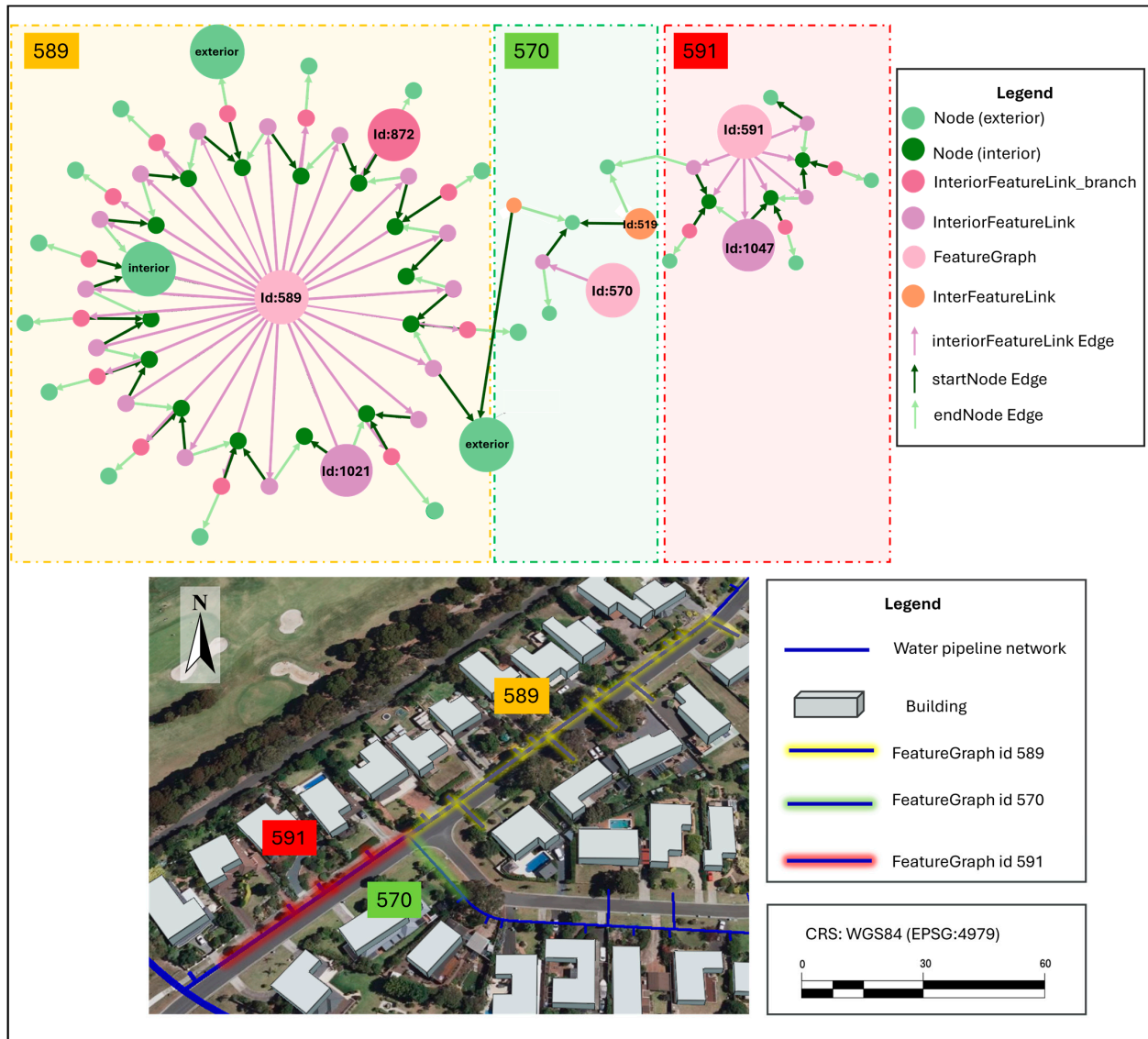


Figure 10. UNADE-LPG graph data model for representation of the water utility network for FeatureGraph 589, 570, and 591.

The coloured boundaries correspond to the three pipeline networks in the real-world sample: yellow, red, and green.

- **Orange nodes** denote InterFeatureLink elements, which explicitly capture intersections between the three pipelines. In this example, two InterFeatureLinks connect the yellow, red, and green networks.
- **Pink nodes** represent FeatureGraph elements, which group pipeline segments (InteriorFeatureLinks) and their associated branches under a common identifier. For instance, the yellow pipeline includes 13 branch connections, each modelled as InteriorFeatureLink_branch nodes attached to the same FeatureGraph.
- **InteriorFeatureLink nodes** correspond to individual pipeline segments between branches, preserving the real-world structure within the graph.
- **Green nodes** correspond to Node elements that capture 3D geometry for pipeline start and end points. Shared nodes represent interior junctions where segments meet, while exterior nodes mark free ends that either terminate or connect to other pipelines.

Numerical labels correspond to unique identifiers for FeatureGraph and Node elements, enabling precise retrieval and query execution within the graph database.

In addition to demonstrating the structural and semantic correctness of the instantiated graph, the Frankston case study also highlights how the UNADE-LPG representation enables analysis-oriented queries that operate directly on the topological and semantic information preserved in the graph. To illustrate this analytical capability, an end-to-end connectivity path search was implemented (Figure 11) using the real-world dataset. Given a pair of InteriorFeatureLink identifiers representing an origin and a destination pipeline segment, the query retrieves the shortest connecting path between them in terms of the number of intervening segments. Operationally, this corresponds to a common utility management task—such as tracing the chain of downstream or upstream pipes linking a damaged segment to a trunk main, facility, or isolation point.

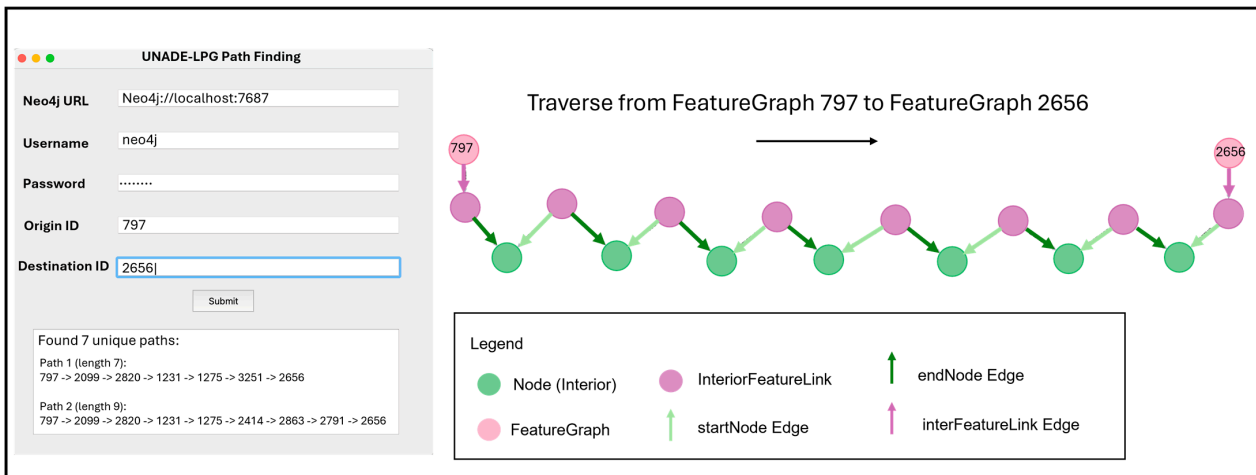


Figure 11. UNADE-LPG path finding.

The query process aligns with the hierarchical semantics embedded within the UNADE-LPG structure. It first locates the FeatureGraph containing the origin InteriorFeatureLink and traverses adjoining InteriorFeatureLinks via shared Node elements, using node type (interior or exterior) and stored link direction to restrict traversal to flow-consistent connections. Where two FeatureGraphs intersect, traversal continues through the corresponding InterFeatureLink nodes, thereby enabling movement across separate pipeline groups while maintaining the network hierarchy. The output is an ordered list of InteriorFeatureLink segments, together with their associated nodes, which can be exported or visualised for operational uses such as impact assessment, connectivity diagnostics, or maintenance planning. The implementation of this analytical query is available in the project’s public GitHub repository, supporting reproducibility and practical adoption.

4.4. Validation

This section reports the outcomes of the validation procedures applied to the instantiated UNADE-LPG graph data model. Three categories of validation were performed: structural conformance, topological consistency, and graph database integrity. These checks verify whether the resulting graph instance matches the structural patterns, degree requirements, and connectivity features specified in the UNADE-LPG transformation rules.

4.4.1. Structural Conformance

This validation ensures that the graph structure generated from the graph data model is correct and consistent, avoiding anomalies that could compromise network analysis. Two key structural validation rules are applied:

- Rule 1: Isolated Nodes Rule

Every node in the UNADE-LPG model must participate in at least one relationship. Isolated nodes indicate incorrect mapping, incomplete data import, or errors in construction.

Valid Case: A Node representing a junction is connected to two InteriorFeatureLinks, ensuring it contributes to the continuity of the pipeline.

Invalid Case: A Node appears in the graph with no edges (e.g., a junction geometry imported without connecting pipelines).

Result: No isolated nodes were detected in the dataset (Figure 12).

```
CALL { MATCH (n) RETURN count(n) AS total_nodes }
CALL { MATCH ()-[r]->() RETURN count(r) AS total_relationships }
CALL { MATCH (n) WHERE COUNT { (n)--() } = 0 RETURN count(n) AS isolated_nodes }
RETURN total_nodes, total_relationships, isolated_nodes, isolated_nodes = 0 AS no_isolated_nodes
```

total_nodes	total_relationships	isolated_nodes	no_isolated_nodes
1106	1674	0	true

Figure 12. Isolated node validation.

- Rule 2: Node Degree Rule

Each node type in the UNADE-LPG has a defined degree constraint based on its semantic role. Degree checks confirm that connections align with the intended structure.

Check 1: InteriorFeatureLink/InteriorFeatureLink_branch Degree

InteriorFeatureLink/InteriorFeatureLink_branch must have degree 3, consisting of one edge to its parent FeatureGraph and two edges to Node (via startNode and endNode relationships).

Valid Case: A branch segment links its parent FeatureGraph and two geometry nodes.

Invalid Case: A pipeline segment missing a connection with Node (degree < 3).

Check 2: Node Degree

Node (type: interior) must have degree 2 (two connected pipeline segments) or 3 if also connected to a branch. However, Node (type: exterior) must have at least degree 1 (endpoint) but can be higher when multiple pipelines meet.

Valid Case: A node connects exactly two InteriorFeatureLinks at an intersection.

Invalid Case: An exterior node with degree 0 (no pipeline attached).

Check 3: InterFeatureLink Degree

InterFeatureLink must have degree 3 (two Nodes (type: exterior) + one NetworkGraph).

Valid Case: An InterFeatureLink connects exterior nodes of two pipelines and the overarching NetworkGraph.

Invalid Case: An InterFeatureLink connected to only one exterior node.

Result: Table 3 presents a consolidated validation of node degrees across all entity types in the instance graph data model. The results show that all 387 InteriorFeatureLinks and all 113 InteriorFeatureLink_branch elements conform to the expected degree of three, reflecting correct connections to their parent FeatureGraph and associated start and end nodes. For the 348 interior nodes, 238 exhibit a degree of two and 110 a degree of three, both consistent with the defined connectivity rules for intersections and shared points.

Finally, all 16 InterFeatureLinks have a degree of three, corresponding to their connections with two exterior nodes and the overarching NetworkGraph.

Table 3. Node degree validation.

Validation Type	Rule	Total Node	Nodes with Degree 1	Nodes with Degree 2	Nodes with Degree 3
InteriorFeatureLink	Expected degree: 3 (connects to FeatureGraph, Node)	387	0	0	387
InteriorFeatureLink_branch	Expected degree: 3 (connects to FeatureGraph, Node)	113	0	0	113
Node (type:interior)	Expected degree: 2 (connects two InteriorFeatureLinks) or 3 (connects two InteriorFeatureLinks and one InteriorFeatureLink_branch)	348	0	238	110
InterFeatureLink	Expected degree: 3 (connects two Nodes and one NetworkGraph)	16	0	0	16

Any deviation from the expected degree values would indicate either incorrect mapping during the CityGML GML file conversion process or inconsistencies in the underlying data, both of which would compromise subsequent network analysis and querying.

4.4.2. Topological Consistency

- Rule 3: Topological Consistency Rule

The LPG model must preserve the physical and logical continuity of pipelines through exclusive membership, correct segment alignment, and valid branch geometry.

Check 1: Exclusive Membership

Each InteriorFeatureLink must belong to exactly one FeatureGraph.

Valid Case: A pipeline segment is linked only to its parent FeatureGraph.

Invalid Case: A pipeline segment connected to two different FeatureGraphs, creating ambiguity in network ownership.

Check 2: Segment Alignment

The end geometry of one InteriorFeatureLink must coincide with the start geometry of the next, ensuring continuous flow.

Valid Case: Two segments meet at a shared Node, with their end and start coordinates exactly matching.

Invalid Case: Two segments are connected in the graph, but their coordinates are misaligned, producing a geometric gap or overlap.

Check 3: Branch Geometry

Each InteriorFeatureLink_branch must share its junction geometry with the two InteriorFeatureLinks it connects.

Valid Case: A branch pipeline joins a main pipeline at a node shared by both.

Invalid Case: A branch segment connected in the graph without matching the spatial coordinates of the main pipeline, resulting in a “floating” connection.

Result: All tested cases conformed to these topological rules. Each InteriorFeatureLink was assigned exclusively to one FeatureGraph, all joined segments showed exact coordinate

alignment, and all branch geometries were correctly shared with their parent pipelines. Table 4 provides a consolidated summary of these checks, showing zero violations across exclusive membership, segment alignment, and branch-geometry conditions.

Table 4. Topological consistency validation.

Checks	Expected Condition	Violations	Notes
Exclusive membership	Each InteriorFeatureLink belongs to exactly one FeatureGraph	0	All 387 segments correctly assigned
Segment alignment	End coordinates of segment A coincide with start coordinates of segment B	0	No gaps or overlaps observed
Branch geometry	InteriorFeatureLink_branch shares junction node with its connected InteriorFeatureLink	0	All 113 branch links are correctly aligned

4.4.3. Graph Database Integrity

This validation ensures that the LPG implementation of the proposed UNADE-LPG graph data model is both structurally sound and semantically consistent. Unlike the previous checks, which focused on local structure and connectivity, this stage examines the overall robustness of the graph database. It validates that entities are uniquely represented, attributes are properly classified, and network connections reflect their real-world counterparts. Two key integrity rules are applied.

- Rule 4: Normalisation Rule

The graph must avoid redundancy and duplication of entities, ensuring structural normalisation and geometric consistency.

Check 1: Duplicate Geometry Nodes

No two nodes may share identical coordinates and type within a tolerance of 0.001 m. This threshold does not represent the measurement accuracy of the dataset; rather, it functions as a computational safeguard to absorb minor numerical variations introduced during GML parsing and coordinate processing. By treating coordinate differences below this value as equivalent, the validation procedure avoids false detection of duplicated or coincident nodes while remaining sufficiently strict to identify genuine geometric inconsistencies.

Valid Case: Two pipelines intersect at the same junction, represented by a single shared node.

Invalid Case: Two separate nodes with identical coordinates represent the same junction, causing redundancy.

Check 2: Coincident Nodes at a Junction

No distinct nodes may occupy the same spatial location ($\epsilon = 0.001$ m) while both are used in connectivity.

Valid Case: A junction is represented by one node reused in multiple pipeline connections.

Invalid Case: Two distinct nodes, both connected to different segments, occupy the same location and fragment connectivity.

Check 3: Reuse of Shared Points

Connected pipeline segments must share a common node instead of duplicating coincident endpoints.

Valid Case: Two InteriorFeatureLinks use the same node for their shared endpoint.

Invalid Case: Each segment introduces a new endpoint node at the same location, breaking continuity.

Check 4: Unique Start and End Nodes

Every pipeline segment must have exactly one unique start node and one unique end node; these cannot be identical.

Valid Case: A segment starts at Node A and ends at Node B, representing a real pipeline.

Invalid Case: A segment starts and ends at the same node, creating a zero-length segment.

Result: Table 5 summarises these checks. Across the dataset, no violations were detected, confirming that the graph is normalised, geometrically consistent, and free of redundancy.

Table 5. Graph database integrity validation.

Checks	Tolerance (ϵ)	Violations
Duplicate geometry nodes	0.001 m	0
Coincident but distinct nodes at a junction	0.001 m	0
Reuse of shared points	N/A	0
Unique start and end nodes	N/A	0

- Rule 5: Classification Rule

The LPG model must preserve attribute-based classifications so that graph elements align with real-world organisational and functional roles. In this rule the pipelines must be correctly grouped according to ownership (e.g., SEW, Private) and type (e.g., reticulation, fire service).

Valid Case: All SEW-owned pipelines are classified as reticulation mains, while private pipelines are classified as fire service.

Invalid Case: A reticulation main is classified as privately owned, contradicting the operational reality.

Result: Table 6 presents the classification validation. It shows that all 40 reticulation mains are SEW-owned, while 2 fire service pipelines are privately owned. No misclassifications were observed. These results confirm that the classification attributes in the graph database faithfully represent the real-world ownership and functional responsibilities of the utility network.

Table 6. UNADE-LPG classification query validation.

Query	Reticulation Main	Fire Service	Total
Owner: SEW	40	0	40
Owner: Private	0	2	2
Total	40	2	42

4.4.4. Descriptive Graph Statistics

To complement the rule-based validation, descriptive statistics were generated from the instantiated UNADE-LPG graph (Table 7). The full dataset comprises 1106 nodes and 1674 edges, forming two connected components corresponding to the SEW reticulation network and the privately owned fire-service network. The SEW network contains 1073 nodes and 1528 edges, while the private fire-service system consists of 13 nodes and 14 edges. Classification attributes were also preserved, with 206 nodes labelled as OPR and 881 as MIG. No isolated nodes were detected, consistent with the structural conformance

validation. The highest-degree nodes in the dataset were the NetworkGraph nodes (degree 58), reflecting the fact that each NetworkGraph aggregates all FeatureGraph elements belonging to the same functional network (e.g., reticulation main, fire service). Because every FeatureGraph is linked to exactly one NetworkGraph, the degree of a NetworkGraph directly corresponds to the number of pipelines it governs, leading to naturally high degree values. These statistics corroborate the correctness and internal consistency of the transformed graph structure.

Table 7. UNADE-LPG descriptive graph statistics.

Statistic	Value
Total nodes	1106
Total Edges	1674
Number of connected components	2 (SEW network + private fire-service network)
Nodes in reticulation main (SEW)	1073
Edges in reticulation main (SEW)	1528
Nodes in fire-service network (private)	13
Edges in fire-service network (private)	14
SEW-owned nodes	1073
Private-owned nodes	13
Nodes in operational (OPR) state	206
Nodes in migration (MIG) state	881
Highest-degree node	58
Lowest degree node	1

5. Discussion

The outcomes of this study show that the UNADE-LPG graph data model provides a semantically aligned and topologically consistent representation of the CityGML UNADE. The transformation rules developed in this work ensured that classes, attributes, and relationships defined in the UNADE UML schema were transferred into the labelled property graph structure without altering their conceptual meaning. Both case studies demonstrated that the graph representation preserved the expected connectivity relationships and semantic associations while enabling more direct traversal than would be possible in a relational schema.

One modelling decision that influenced the structure of the resulting graph was the removal of intermediary abstract classes between Network and FeatureGraph. This decision did not alter the semantics of the original UML model, as the abstract classes do not carry instance-level objects. Its effect was structural simplification: fewer intermediary nodes reduced the number of hops required during traversal and therefore shortened typical query paths. This type of flattening follows standard graph-design practices where non-instantiated UML abstractions are omitted to achieve clearer and more efficient graph structures. Similarly, normalising shared Nodes, representing common junctions between InteriorFeatureLinks, avoided duplication and reduced the total number of nodes stored in the database, which is beneficial for both storage efficiency and topological reasoning.

The schematic case study demonstrated that the UNADE-LPG model can represent branching and non-branching pipeline configurations using the same set of node and relationship rules. The use of InterFeatureLink nodes provided a controlled mechanism for modelling connections across pipeline groups, ensuring that cross-feature interactions were explicit and traceable. This aligns with the intent of the UNADE schema, where physical

intersections between different pipeline features must be semantically represented rather than inferred implicitly.

The real-world case study confirmed that the conversion pipeline can populate the proposed graph model directly from CityGML UNADE GML files. The resulting graph accurately reflected the connectivity patterns of the Frankston water network, including branching points, shared nodes, and multiple ownership types. The addition of the path-finding query illustrated that the proposed model supports basic network-analysis workflows, such as determining the shortest topological path between two pipe segments. This shows that the model is not only semantically aligned with UNADE but is also functional for operational queries commonly required in utility-network management.

The validation framework further confirmed the correctness of the transformed graphs. Structural checks verified that node degrees matched the expected multiplicities of the UNADE schema and that no isolated nodes were present. Topological checks confirmed that adjacent segments were spatially aligned, that branch geometry was correctly shared, and that each InteriorFeatureLink belonged to a single FeatureGraph. Classification queries showed that ownership and operational state were correctly preserved during transformation. The descriptive graph statistics provided additional assurance that the resulting structure was complete and internally consistent.

6. Conclusions

This study developed the UNADE-LPG data model as a graph-based representation of the core elements of the CityGML UNADE. The model formalises how UNADE classes, relationships, and constraints can be expressed within a Labelled Property Graph structure while retaining the semantic intent of the original UML specification. A conversion pipeline was implemented to generate these graph instances directly from CityGML UNADE GML files, offering a reproducible and standards-aligned workflow for graph-based utility-network representation.

The work is limited to the structural and semantic elements of the core UNADE schema. Components such as valves, pumps, flow states, and temporal behaviour are outside the current scope, as the focus is on verifying semantic preservation and structural correctness rather than modelling full operational behaviour. The study also does not include performance benchmarking on city-scale datasets; performance outcomes depend heavily on database configuration, indexing decisions, and hardware rather than the data-model design itself.

Future research will extend the UNADE-LPG framework to additional UNADE modules and broader utility domains and will apply the model to larger city-wide datasets as they become available. Further development will also include the integration of spatial and temporal analysis capabilities, more advanced traversal strategies, and extended support for graph-based analytical workflows. These directions aim to enhance the model's applicability within digital-twin environments and to support more comprehensive utility-network management tasks.

Author Contributions: Conceptualization, Ensiyeh Javaherian Pour, Behnam Atazadeh, Abbas Rajabifard, Soheil Sabri, and David Norris; methodology, Ensiyeh Javaherian Pour, Behnam Atazadeh, Abbas Rajabifard, and Soheil Sabri; software, David Norris; validation, Ensiyeh Javaherian Pour; formal analysis, Ensiyeh Javaherian Pour; investigation, Ensiyeh Javaherian Pour; resources, David Norris; data curation, Ensiyeh Javaherian Pour and David Norris; writing—original draft preparation, Ensiyeh Javaherian Pour; writing—review and editing, Ensiyeh Javaherian Pour, Behnam Atazadeh, Abbas Rajabifard, and Soheil Sabri; visualization, Ensiyeh Javaherian Pour; supervision, Behnam Atazadeh, Abbas Rajabifard, and Soheil Sabri; project administration, Behnam Atazadeh and Abbas

Rajabifard; funding acquisition, Behnam Atazadeh and Abbas Rajabifard. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Australian Research Council through two grants: DE220100094 (Behnam Atazadeh) and IH210100048 (Abbas Rajabifard).

Institutional Review Board Statement: Ethical review and approval were waived for this study because it did not involve human participants, human data, or animal experiments.

Informed Consent Statement: Not applicable. This study did not involve humans.

Data Availability Statement: The data will be made available upon request.

Software Availability: The source code supporting this study is available on GitHub at: [https://github.com/AJjavaheer/UNADE_GMLtoGDB] (accessed on 8 December 2025).

Acknowledgments: This research was supported by industry partners South East Water and Emerson. All individuals acknowledged have provided their consent.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

References

1. OpenGeospatialConsortium. OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard. 2021. Available online: <https://docs.ogc.org/is/20-010/20-010.html> (accessed on 8 December 2025).
2. Kutzner, T.; Kolbe, T.H. Extending semantic 3D city models by supply and disposal networks for analysing the urban supply situation. In Proceedings of the Lösungen für eine Welt im Wandel, Dreiländertagung der SGPF, DGPF und OVG, 36. Wissenschaftlich-Technische Jahrestagung der DGPF, Bern, Germany, 7–9 July 2016; pp. 382–394.
3. Javaherian Pour, E.; Atazadeh, B.; Rajabifard, A.; Sabri, S. Review and assessment of 3D spatial data models for managing underground utility networks. *Tunn. Undergr. Space Technol.* **2025**, *157*, 106219. [[CrossRef](#)]
4. Duijn, X.d.; Agugiaro, G.; Zlatanova, S. Modelling Below- and Above-Ground Utility Network Features with the CityGML Utility Network ADE: Experiences From Rotterdam. In Proceedings of the 3rd International Conference on Smart Data and Smart Cities, Delft, The Netherlands, 4–5 October 2018.
5. Vishnu, E.; Sameer, S. OGC CityGML 3D City Models Enriched with Utility Infrastructures for Developing Countries. *J. Indian Soc. Remote Sens.* **2021**, *49*, 813–826. [[CrossRef](#)]
6. Anuyah, S.; Bolade, V.; Agbaakin, O. Understanding graph databases: A comprehensive tutorial and survey. *arXiv* **2024**, arXiv:2411.09999. [[CrossRef](#)]
7. Bechberger, D.; Perryman, J. *Graph Databases in Action*; Manning Publications Co.: Shelter Island, NY, USA, 2020.
8. Ilonen, J. A Case Study on Transitioning from Relational Data models to Graph Data models in an Industrial Context. Master's Thesis, Åbo Akademi University, Turku, Finland, 2023.
9. Javaherian Pour, E.; Atazadeh, B.; Rajabifard, A.; Sabri, S. Developing a CityGML-based Graph Data Model for Utility Infrastructure in Smart Cities. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2025**, X-G-2025, 405–412. [[CrossRef](#)]
10. Ji, Q. Geospatial Inference and Management of Utility Infrastructure Networks. Ph.D. Thesis, Newcastle University, Newcastle upon Tyne, UK, 2020.
11. Dunn, S.; Holmes, M. Development of a hierarchical approach to analyse interdependent infrastructure system failures. *Reliab. Eng. Syst. Saf.* **2019**, *191*, 106530. [[CrossRef](#)]
12. Li, T.; Rui, Y.; Zhu, H.; Lu, L.; Li, X. Comprehensive digital twin for infrastructure: A novel ontology and graph-based modelling paradigm. *Adv. Eng. Inform.* **2024**, *62*, 102747. [[CrossRef](#)]
13. Peng, F.-L.; Qiao, Y.-K.; Yang, C. Building a knowledge graph for operational hazard management of utility tunnels. *Expert Syst. Appl.* **2023**, *223*, 119901. [[CrossRef](#)]
14. Nguyen, S.H.; Kolbe, T.H. A multi-perspective approach to interpreting spatio-semantic changes of large 3D city models in citygml using a graph database. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **2020**, VI-4/W1-2020, 143–150. [[CrossRef](#)]
15. Nguyen, S.H.; Yao, Z.; Kolbe, T.H. Spatio-semantic comparison of large 3D city models in CityGML using a graph database. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 99–106. [[CrossRef](#)]
16. Kasprzyk, J.-P.; Nys, G.-A.; Billen, R. Towards a multi-database CityGML environment adapted to big geodata issues of urban digital twins. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *48*, 101–106. [[CrossRef](#)]
17. Yao, Z.; Kolbe, T.H. Dynamically extending spatial databases to support CityGML application domain extensions using graph transformations. *Kult. Erbe Erfassen Und Bewahr.-Von Der Dok. Zum Virtuellen Rundgang* **2017**, *37*, 316–331.

18. Tsai, B.-S. Extending the 3D City Database 5.0 to Support CityGML Application in QGIS. Master's Thesis, Delft University of Technology, Delft, The Netherlands, 2024.
19. Sadavare, A.; Kulkarni, R. A review of application of graph theory for network. *Int. J. Comput. Sci. Inf. Technol.* **2012**, *3*, 5296–5300.
20. Bonifati, A.; Fletcher, G.; Voigt, H.; Yakovets, N. *Querying Graphs*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
21. Kousis, A. Managing Smart City Linked Data with Graph Databases: An Integrative Literature Review. In *Graph Databases*; CRC Press: Boca Raton, FL, USA, 2023; pp. 118–141.
22. Ding, L.; Xiao, G.; Pano, A.; Fumagalli, M.; Chen, D.; Feng, Y.; Calvanese, D.; Fan, H.; Meng, L. Integrating 3D city data through knowledge graphs. *Geo-Spat. Inf. Sci.* **2025**, *28*, 780–799. [[CrossRef](#)]
23. Sahu, S.; Mhedhbi, A.; Salihoglu, S.; Lin, J.; Özsu, M.T. The ubiquity of large graphs and surprising challenges of graph processing: Extended survey. *VLDB J.* **2020**, *29*, 595–618. [[CrossRef](#)]
24. Angles, R.; Thakkar, H.; Tomaszuk, D. RDF and Property Graphs Interoperability: Status and Issues. *AMW* **2019**, *2369*, 1–11.
25. Arenas, M.; Gutierrez, C.; Pérez, J. Foundations of RDF databases. In *Reasoning Web International Summer School*; Tessaris, S., Franconi, E., Eiter, T., Gutierrez, C., Handschuh, S., Rousset, M.-C., Schmidt, R.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 158–204.
26. Vinasco-Alvarez, D.; Samuel, J.; Servigne, S.; Gesquière, G. Towards an Automated Transformation of an nD Urban Data Model to a Computational Ontology Network: From UML to OWL, From CityGML 3.0 to “CityOWL”. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *X-4/W4-2024*, 231–238. [[CrossRef](#)]
27. Lam, P.D.; Gu, B.H.; Lam, H.K.; Ok, S.Y.; Lee, S.H. Digital Twin Smart City: Integrating IFC and CityGML with Semantic Graph for Advanced 3D City Model Visualization. *Sensors* **2024**, *24*, 3761. [[CrossRef](#)]
28. Chadzynski, A.; Li, S.; Grišute, A.; Chua, J.; Hofmeister, M.; Yan, J. Semantic 3D city interfaces—Intelligent interactions on dynamic geospatial knowledge graphs. *Data-Centric Eng.* **2023**, *4*, e20. [[CrossRef](#)]
29. Donkers, A.; Yang, D.; Baken, N. Linked Data for Smart Homes: Comparing RDF and Labeled Property Graphs. In Proceedings of the LDAC2020—8th Linked Data in Architecture and Construction Workshop, Dublin, Ireland, 17–19 June 2020.
30. Gelling, E.; Fletcher, G.; Schmidt, M. Bridging graph data models: RDF, RDF-star, and property graphs as directed acyclic graphs. *arXiv* **2023**, arXiv:2304.13097. [[CrossRef](#)]
31. Hor, A.H.; Sohn, G.; Claudio, P.; Jadidi, M.; Afnan, A. A semantic graph database for BIM-GIS integrated information model for an intelligent urban mobility web application. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *4*, 89–96. [[CrossRef](#)]
32. Agoub, A.; Kunde, F.; Kada, M. Potential of graph databases in representing and enriching standardized Geodata. *Tagungsband Der* **2016**, *36*, 208–216.
33. Chen, Z.; Pouliot, J.; Hubert, F. A Hierarchy of Levels of Detail for 3D Utility Network Models. In Proceedings of the International 3D GeoInfo Conference, Munich, Germany, 12–14 September 2023; pp. 543–561.
34. Robinson, I.; Webber, J.; Eifrem, E. *Graph Databases: New Opportunities for Connected Data*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
35. Angles, R.; Gutierrez, C. Survey of graph database models. *ACM Comput. Surv. (CSUR)* **2008**, *40*, 1–39. [[CrossRef](#)]
36. Becker, T.; Nagel, C.; Kolbe, T.H. Integrated 3D modeling of multi-utility networks and their interdependencies for critical infrastructure analysis. In *Advances in 3D Geo-Information Sciences*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–20.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.