Automatic conversion of real-world data from a German road infrastructure management system into a labelled property graph

Ina Heise[1]
[1]Chair of Computational Modeling, Technical University of Munich, 80333 Munich, Germany
E-mail(s): ina.heise@tum.de

Abstract: Maintenance of road infrastructure in many countries, including Germany, is carried out by various specialised institutions. These institutions use separate systems that often rely on relational databases. As a result, it becomes nearly impossible to conduct a comprehensive analysis of road infrastructure data. To address this challenge, graph-based representations are commonly used to enable more flexible and comprehensive data linking. This paper introduces an approach to represent geospatial data as a labelled property graph (LPG). The standardization of geospatial data provision through Web Feature Services is leveraged to develop a universal concept for creating the graph representation of geospatial data. The resulting LPG enables a more flexible evaluation of road management data stored in established GIS-based systems, facilitating integration into comprehensive systems. Real-world data from the road management system in Bavaria, a state in Germany, is transformed into an LPG to validate the concept, and an example of a comprehensive query is provided.

Keywords: road infrastructure, labelled property graphs, GIS, GML, digital twin

Erschienen in Tagungsband 35. Forum Bauinformatik 2024, Hamburg, Deutschland, DOI: 10.15480/882.13523

1 Introduction

Road infrastructure is crucial for society, making its management and maintenance exceptionally important. The sheer size and complexity of the system pose significant challenges. According to [1], digital twins, which accurately represent physical systems in their current state, are considered an effective tool for enhancing the operation of road infrastructure. These digital twins serve as comprehensive databases, facilitating in-depth data analysis and providing a basis for decision support.

A comprehensive infrastructure management system based on a digital twin differs significantly from the current practice of managing road infrastructure. Various aspects of road infrastructure are managed in separate, unconnected systems, mostly based on relational databases. The dispersed data management in place requires significant manual effort to conduct comprehensive data analysis, if possible at all. Therefore, the consideration of road infrastructure as a system consisting of different

interacting subsystems is not feasible.
Graph-based approaches are commonly used to facilitate flexible and comprehensive evaluations of (highly) connected data. As a result, graph-based representations of legacy data are frequently employed to incorporate legacy data into digital twins. Existing approaches commonly use ontologies describing the legacy data structures for this purpose. Subsequently, representations based on RDF graphs can be established. However, these approaches depend on the availability of the underlying data structure, which may not always be publicly accessible, such as in the case of Bavarian road data. Additionally, RDF graphs may not always provide the most compact information representation
Hence, this paper presents an alternative method for transforming existing infrastructure data into graph-based representations. The suggested concept focuses on a general approach for converting geospatial data provided through a Web Feature Service (WFS) into a graph. The selected graph model is the labelled property graph (LPG) model, which offers a more concise representation. Moreover, the underlying schemaless approach provides advantages when converting data with unknown data structures.
The result is an approach allowing the evaluation of data contained in GIS systems much more flexibly and integrating it with other data sources in a digital representation of a digital twin. The method was tested using the GIS system utilised in Bavaria for road infrastructure management. This system provides geospatial data on Bavarian infrastructure through a WFS. The geospatial data retrieved is automatically converted into an LPG. By inspecting the metamodel of the LPG in neo4j, the graph structure obtained could be derived, allowing the formulation of comprehensive queries as road conditions in conjunction with road cross-section data for all Bavarian roads.

2 Related Research

The research on digital twins in road infrastructure carried out by Taherkhani, Ashtari, and Aziminezhad [2] underscores their potential and the challenges posed by different existing systems. The major challenges identified include poor data quality and insufficient consideration of the overall infrastructure. Heise [3] discusses the existence of data silos in German infrastructure management and compares the available data in distributed systems with potential use cases, highlighting the necessity for comprehensive data evaluation. This emphasises the potential of a cross-system approach while revealing that existing relational database systems lack the necessary flexibility to conduct data evaluations for these use cases effectively. Consequently, the suggested solution is to employ graph-based representations of the legacy data instead.
Existing research already provides approaches for the creation of graph-based representations. Ismail, Strug, and Ślusarczyk [4] discuss a solution for generating graph representations of IFC models. However, the approach presented focuses more on buildings rather than infrastructure structures. Another approach, presented in [5], deals directly with the data of the system used in Germany for infrastructure asset management. The introduced ASB-ING ontology covers the data structure underlying the infrastructure asset management system, and an RDF graph is generated as a graph representation by instantiating this ontology with data from the legacy system. Beetz, Amann, and Borrmann [6] apply the same approach for road data corresponding to the OKSTRA-schema by introducing and instantiating okstraOWL. However, these concepts were developed based on the

knowledge of the underlying structure of the instance data. Furthermore, the focus was on establishing one-directional data conversion due to the existing interfaces of the systems under consideration. In the case of existing GIS systems for road data, there are already systems providing interfaces with implemented web services that theoretically facilitate connections to overarching digital twin systems. A graph-based representation would, therefore, serve as an extension, enabling semantic analyses and contextual linking of GIS data with other systems data. Hence, this paper outlines the development of a concept for creating a graph representation that serves as an extension by representing geodata, which can be queried via these web services. Unlike the previously presented approaches that used RDF graphs, this concept utilises LPGs, which offer benefits in terms of a more compact structure. Furthermore, LPGs are better suited for representing data whose data structure is not entirely known in advance. In this case, a schemaless approach, when combined with the use of self-describing database systems like neo4j, is particularly advantageous because the graph representation could be used additionally for their analysis of the data structure.

3 Background

Geospatial data from GIS systems is typically provided through Web Services, which adhere to standards set by the Open Geospatial Consortium (OGC). The choice of web service specifications needed depends on the geospatial data that needs to be accessed. When integrating geospatial data into a digital twin and connecting it to other infrastructure data sources, the primary emphasis is on the semantics. Therefore, the WFS, standardised by the OGC, is crucial. Although WFS provides extensive functionality beyond file-based geospatial data provision, its main purpose is to make vector data accessible. This functionality is utilised to create the graph-based representation. To retrieve data, an HTML request in accordance with the web service definition needs to be sent to the respective server endpoint. The returned data is typically formatted in XML and follows the Geography Markup Language (GML) Encoding Standard, which is also defined by the OGC.
According to the Spatial Data on the Web Best Practices by W3C, geospatial data is typically described using *Features* defined in DIN EN ISO 19101. According to DIN EN ISO 19101, Features are "abstractions of real-world phenomena" and mainly describe geographical units. The use of *Features* is rooted in the General Feature Model (GFM), standardised by DIN EN ISO 19109. The GFM defines *FeatureTypes* and *FeatureInstances*. Similar to class definitions in object-oriented programming, *FeatureTypes* outline generalisations, while *FeatureInstances* describe specific objects. *FeatureTypes* can have properties. These properties are described using *PropertyTypes*. While *PropertyTypes* define the characteristics of *FeatureTypes*, they are modelled separately. This separation results in a *FeatureType* not being defined by its properties. The subclasses of *PropertyType* include different types of properties: *AttributeType* for static properties, *Operation* for dynamic properties represented by function calls, and *FeatureAssociationRole* for characterising types of associations. The connection between *PropertyTypes* and *FeatureTypes* is defined by *ValueAssignments* and their subclasses, which also encompass *Operation*. Additionally, there may be associations or inheritance relationships between *FeatureTypes*. These associations are further delineated by *FeatureAssociationRole*, while the inheritance relationship is represented using *InheritanceRelation*. In summary, the geospatial data accessible via a WFS is described by instances of *FeatureTypes*. *FeatureTypes* may possess various

properties and relationships with each other, but they are not defined by these.
In contrast, there is the labelled property graph model, as defined, for instance, in [7]. Information is depicted as nodes linked by edges, which can be directional. Both nodes and edges are assigned labels and may possess other attributes. The significant advantage of using graphs for storing data is their effectiveness in representing interconnected information. Furthermore, by allowing attributes to be directly assigned to nodes and edges, LPGs offer a much more concise representation of information compared to RDF graphs.

4 Concept

4.1 Proposed graph structure for representing FeatureInstances

When geospatial data is represented as LPG, the main objective is to improve the capability to analyse relationships within data and link it to other data sources. Therefore, the emphasis is on capturing the semantics inherent in the geospatial data, and it is assumed that the graph representation supplements the WFS without entirely replacing it. This means that not all information available from the WFS needs to be explicitly represented in the graph.
In the LPG, all *FeatureInstances* are represented as nodes with their associated *FeatureTypes* as labels. When it comes to representing the properties of individual *FeatureInstances*, a distinction is made between *AttributeType*, *Operation*, and *FeatureAssociationRole* properties. Static properties (excluding attributes for describing geometry) are stored as properties at the *FeatureInstance* nodes. The attribute name serves as the property name, and the value is assigned accordingly. Moreover, the ID of a respective *FeatureInstance* is stored in a property, named following the schema `label name_ID`. For attributes that describe geometry or have dynamically changing values, we propose that the original data source (the WFS) be referred back to. One way to implement this reference is by storing the HTML request for the corresponding resource as a string in the respective property value. In the case of attributes with changing values, this approach helps prevent inconsistencies between the graph representation and the geodata. Regarding geometric attributes it ensures that the graph is not burdened with geometric information that may not be necessary for the semantic linking.
Relationships between different *FeatureInstances* are depicted through edges, utilising labels to describe the relationship type. Additionally, the option to assign properties to edges in LPGs allows for the potential modelling of other relationship properties. Figure 1 illustrates the proposed graph structure, depicting the representation of two *FeatureInstances* in relation to each other.

4.2 Creation of the graph representation

With a system depicted in Figure 2, the LPG can be created. The primary component is a Python script that initially retrieves geospatial data by sending an HTML request to the WFS server, followed by the interpretation of the returned GML data and generation of the graph by sending queries to a graph database server instance. Since neo4j was chosen as the graph storage system, CYPHER queries have to be used. In order to generalise the graph creation process, an object-oriented data structure is defined, as depicted in Figure 3, comprising three classes: *ConversionObject*, *ConversionLink*, and *ConversionProperty*. Instances of the *ConversionObject* class must have an *ObjectID* storing the unique *FeatureInstance*-ID and an *ObjectType* holding the associated *FeatureType* to enable clear

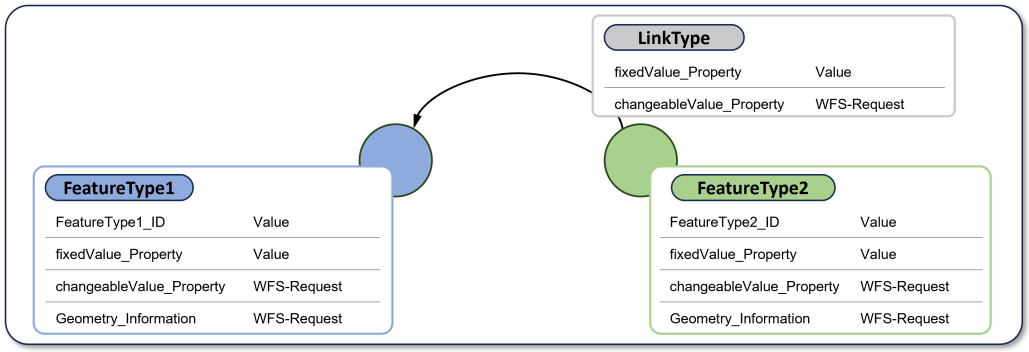

Figure 1: Graph structure of the resulting LPG, displayed using two example instances

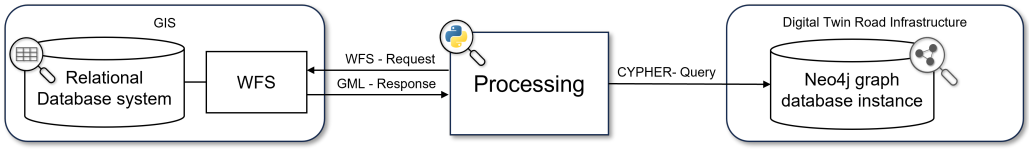

Figure 2: Prototype system for the conversion of geodata into LPG using the WFS of a GIS system

identification of the described *FeatureInstance* in the original data. Additionally, *ConversionObject* instances can possess a list of *ConversionLink* and *ConversionProperty* instances. *ConversionLink* instances store the relationship to another *FeatureInstance* through *LinkType*, which specifies the type of link, and *LinkedObjectType* and *LinkedObjectID* attributes, describing the target object of the link by specifying its *ObjectType* and *ObjectID*. *ConversionProperty* instances detail the attributes of the respective *FeatureInstance* with *PropertyType*, *PropertyName* and *PropertyValue*.

To generate the graph based on the list of instances of *ConversionObject*, it is important to ensure that all nodes representing specific *FeatureInstances* are uniquely identifiable. Therefore, property existence and property uniqueness constraints are set up for nodes with the respective label before the node itself is created. For generating the nodes, the *MERGE* clause is used to check whether a node with the corresponding label and ID already exists. If so, the remaining attributes stored in *ObjectProperties* are appended to it; otherwise, a new node with associated properties is generated. When creating the edges to represent the links, it's important to consider that edges can only be established between two existing nodes. Thus, if the 'target node' of the link doesn't exist when the edge is being created, a 'placeholder node' is required. This 'placeholder node' serves as a temporary representation of the target node, allowing the edge to be established. After the 'placeholder node' is generated using the information from the *ConversionLink* instance regarding the target node (label and ID), the edge can be established. The remaining information is then added to the 'target node' later during the conversion of the associated *FeatureInstance*, as it can be uniquely identified by its label and ID.

This approach allows for the automatic construction of the graph representation by iterating over the list of all *ConversionObject* instances. However, how the *ConversionObject* instances are to be generated may vary slightly depending on the underlying application schema in the WFS implementation.

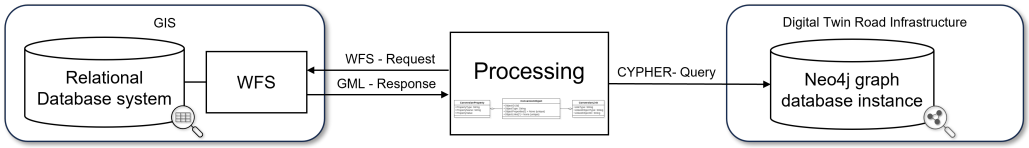

Figure 3: Object oriented datastructure used for parsing

5 Application Bavarian road data

The approach presented is implemented for BAYSIS, the road management system used in Bavaria. BAYSIS relies on ttSIB, a GIS system developed by novasib. The data stored in ttSIB's relational database system can be accessed through a WFS. However, neither the data structure in the database system nor the application schema used to create the complex features for WFS provision are publicly available. Consequently, the general structure of the returned XML files was deduced by querying and inspecting several example files. Subsequently, a Python script was created to automatically instantiate the conversation classes mentioned above.

First, the XML sub-elements that describe the individual *FeatureInstances* have to be identified since the hierarchy level of these instances within the XML tree is unknown. The script iterates through sub-elements and uses the existence of a unique ID to identify the relevant XML elements. The result is the information on the hierarchy level of the *FeatureInstances*, allowing the generation of a list of XML elements representing single *FeatureInstances*, which can be iterated over in the next step.

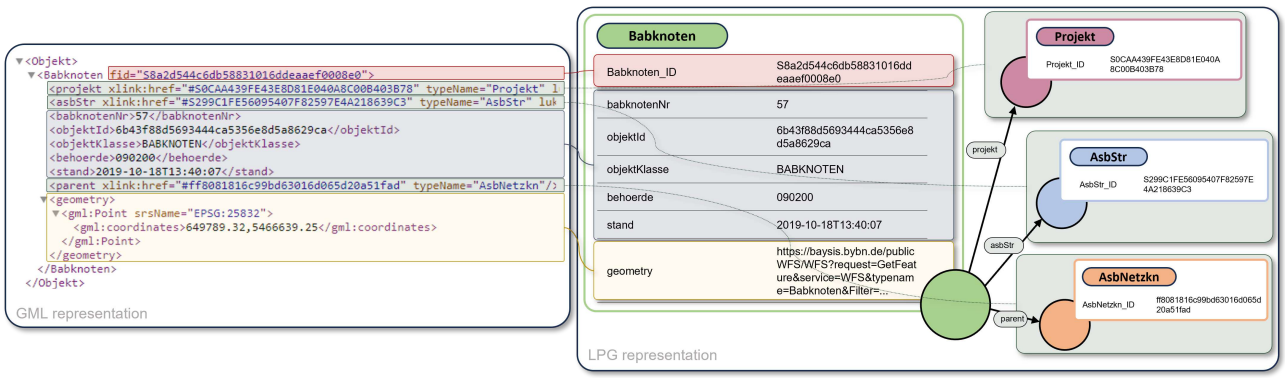

Figure 4: Representation of an example *FeatureInstance* in the GML file and in the LPG

Every XML element belonging to a *FeatureInstance* is searched for attributes and links to other *FeatureInstances*. Properties and links are stored as either subelements of the XML element or as attributes of the XML element, also visible in an excerpt of an example instance shown in Figure 4. The structure of the XML elements and subelements is analysed further to determine whether the information contained within is to be saved as a property, link, or geometry information. All direct attributes of the *FeatureInstance* XML element are considered properties. Direct subelements of the *FeatureInstance*-element are categorised based on their own subelements. If a subelement has no further subelements and no attributes of its own, it is considered a property, and the tag is used as the property name, while the content of the subelement is used as the property value. The remaining sub-elements are further divided into sub-elements holding information on links to other instances

of other *FeatureTypes* or geometric information, again based on their structure. If a subelement has more than one attribute but no other subelements, it is assumed to be describing a link. Then the tag name is interpreted as the name of the link. The attribute *href* with the http://www.w3.org/1999/xlink namespace is used to extract the referenced instance, since per the XLINK-schema definition, this attribute contains the IRI or URI of the referenced object. However, upon examining sample data, it was observed that the link objects in the GML data provided by the ttSIB WFS only contain the ID of the referred *FeatureInstance*, which must be interpreted in conjunction with another attribute containing the name of the *FeatureType*.
An additional property instance is created for attributes recognised as describing geometry, where the tag name is used as the property name, and a generated string containing a WFS query for the source resource is used as the attribute value. The list of *ConversionObject* instances created in this way is used to create the LPG according to the procedure described in section 4.2. Upon examining the metamodel of the generated LPG, it was feasible to discern the data structure that underlies the data and to construct comprehensive queries based on it. One possible query is showcased in Figure 5 on the right, where all road conditions are queried in conjunction with the road cross-section elements at the corresponding positions.

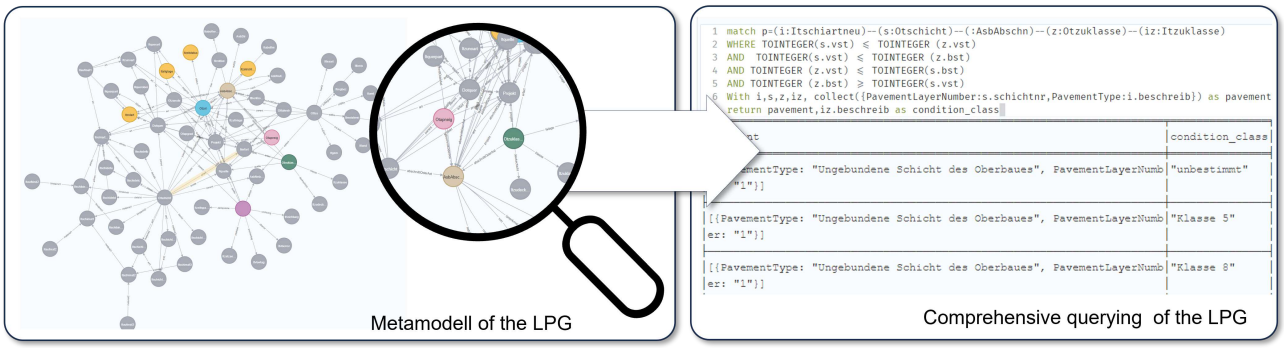

Figure 5: Metamodel of the LPG of Bavarian road data and a possible comprehensive query

6 Results and Discussion

This paper proposes a methodology to generate an LPG representation of geospatial data provided through a WFS using the example of the Bavarian road management system. Each individual *FeatureInstance*, along with its properties, is depicted as a node with the associated *FeatureType* as a label and its corresponding properties. Links between *FeatureInstances* are represented as edges between the respective nodes. Rather than explicitly encompassing geometry attributes and variable attribute values, they are integrated through a reference to the WFS.
Upon analysing the metamodel of the resulting LPG, it is possible to examine the data structure of the represented geospatial data. Using this information, a query was formulated to correlate road conditions to data about the road cross-section throughout the entire dataset. This serves as an example of the comprehensive data analysis that the LPG facilitates within the road data.
Future studies will seek to explore the potential linkage between road data represented by graphs and other infrastructure data, specifically structural data. This will involve integrating historical data

from existing infrastructure asset management systems and consideration of IFC models within the framework of BIM-GIS integration.
The approach presented has limitations, as it has not conclusively addressed the question of ensuring consistency between the original data and its graph representation. The possibility of changing attribute values in the representation of non-static properties is taken into account by using a reference to the original data. Although it's relatively easy to perform a consistency check for individual *FeatureInstances* using the information on *FeatureType* and ID stored at the node, how newly added or deleted *FeatureInstances* can be quickly and effectively detected in the original data in order to update the graph remains unclear.

Acknowledgements

The presented research has been funded by the Bavarian Ministry for Housing, Construction, and Transport in the frame of the project "Digital twin for operating road infrastructure".

References

[1] E. VanDerHorn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation", *Decision Support Systems*, vol. 145, Jun. 2021. DOI: 10.1016/j.dss.2021.113524.

[2] R. Taherkhani, M. A. Ashtari, and M. Aziminezhad, "Digital twin-enabled infrastructures: A bibliometric analysis-based review", *Journal of Infrastructure Systems*, vol. 30, 1 Mar. 2024. DOI: 10.1061/JITSE4.ISENG-2323/SUPPL_FILE/SUPPLEMENTAL.

[3] I. Heise, "Requirements analysis for a digital road infrastructure twin", *Proceedings of the 34. Forum Bauinformatik*, 2023. DOI: 10.13154/294-10105.

[4] A. Ismail, B. Strug, and G. Ślusarczyk, "Building knowledge extraction from bim/ifc data for analysis in graph databases", pp. 652–664, 2018. DOI: 10.1007/978-3-319-91262-2_57.

[5] A. Göbels, "Conversion of infrastructure inspection data into linked data models", *32. Forum Bauinformatik*, 2021. DOI: 10.18154/RWTH-2021-07268.

[6] J. Beetz, J. Amann, and A. Borrmann, "Analysis of application possibilities of linked information (Linked Data) and ontologies and related technologies (Semantic Web) in the road sector", Bundesanstalt für Straßenwesen (BASt), Abschlussbericht, 2018. [Online]. Available: https://mediatum.ub.tum.de/doc/1451874/xq8ya8ljdktao3vzqcf3wus0p.2018_Borrmann_EGICE.pdf (visited on 07/01/2024).

[7] R. Angles, "The property graph database model", *AMW*, 2018. [Online]. Available: https://ceur-ws.org/Vol-2100/paper26.pdf (visited on 07/01/2024).