Graph-based Analysis and Visualization of Metadata in the Context of Urban Digital Twins

Marija Knezevic*[1], Felix Fuchsloch [1], Andreas Donaubauer [1], Thomas H. Kolbe [1]

[1] Chair of Geoinformatics, Technical University of Munich, Arcisstr. 21, Munich, Germany
(marija.knezevic, ge59mer, andreas.donaubauer, thomas.kolbe)@tum.de

Keywords: Urban Digital Twins, Knowledge Graphs, Smart District Data Infrastructure, Metadata Catalogs, Graph Analysis for Metadata, Data Quality Assessment.

Abstract

Urban Digital Twins (UDT) depend on integrating various heterogeneous data sources to represent complex urban environments. Metadata management is an essential component of such systems. This paper introduces a framework for analyzing and visualizing semantic relationships between digital resources in the UDT metadata catalogs, focusing on the Smart District Data Infrastructure (SDDI). Conversion of metadata relationships into directed graphs enables intuitive exploration of the dependencies between resources. Utilizing technologies such as CKAN, NetworkX, Cytoscape, and Dash allows users to perform advanced analytic tasks such as detecting important resources, community detection, detection of isolated resources, cycle detection, and link prediction. It will show how graph theory can help in improving the quality evaluation of metadata, enhance transparency and usability, and contribute to the scalable and effective implementation of UDTs.

1. Introduction

Urban Digital Twins (UDTs) are crucial in managing detailed digital representations of the city. Integrating real-world virtual models provides a solid foundation for simulations and analyses. However, their implementation faces significant challenges, as UDTs rely on integrating a wide range of heterogeneous data sources to describe dynamic urban environments (Lnenicka et al., 2024; Somanath et al., 2024). The effective use of UDTs depends on the management and analysis of a large amount of available information distributed over many actors. Without standardized approaches, cataloging, managing, and analyzing metadata becomes complex and inefficient, reducing its effectiveness and usability (Lopez et al., 2012). It is important to understand the relations between the resources used to create UDTs, as these links form the basis for understanding the interactions and functionalities of urban systems (Gil et al., 2006). Graphical visualization of relations provides a powerful solution for navigating and analyzing complex metadata structures, which allows stakeholders to gain valuable insights and share knowledge.

The Smart District Data Infrastructure (SDDI) framework[1] provides a structured approach to address those challenges. SDDI integrates and links metadata about digital resources within a standardized framework, enabling effective visualization and analysis of datasets for UDT development. At the core of SDDI is the metadata catalog, which serves as a structured repository for organizing resources and mapping their interconnections through graph-based representations (Knezevic et al., 2024, 2022; Kolbe et al., 2020).

Transforming metadata catalogs into structured graphs enables the detection of inconsistencies and weaknesses that are difficult to identify using conventional methods. Although graph-based methods and knowledge graphs are effective for visualizing relationships, the literature largely focuses on structural aspects and support information discovery rather than semantic evaluation (Frasincar et al., 2006; Mutton & Golbeck, 2003). Similarly, Linked Data approaches improve interoperability and discoverability in metadata catalogs (Bauer & Kaltenböck, 2012; Ullah et al., 2018; Khan et al., 2022), but they concentrate on publishing and connecting data, not on evaluating the semantic and structural quality of metadata catalogs. Catalogs such as *MetaVer*[2], provide descriptive parent-child relationships between entries, but do not distinguish between other types of relationships within the catalog, which limits deeper semantic and structural evaluation. The *European Data Portal*[3] catalog focuses on linking to external vocabularies but does not evaluate the structural or semantic quality of metadata. For UDTs, however, analyzing both the structure and semantics of catalog entries is essential, as reliable metadata supports interoperability, discoverability, and long-term sustainability of digital urban ecosystems. This paper discusses the role of graph-based analysis and visualization within the SDDI framework with a particular focus on the semantics of urban metadata catalogs. By modeling metadata entities representing urban information resources as typed nodes and their relationships as directed, typed edges, a typed graph is generated, which not only captures structural information but also expresses semantics. Based on this typed graph, graph theory enables systematic evaluation of metadata quality through analyses such as identifying hidden structures, detecting logical inconsistencies, finding critical resources, and predicting missing connections. The rest of the paper is laid out in the following way: Section 2 reviews related work, Section 3 outlines metadata challenges, Section 4 presents use cases, Section 5 describes the system implementation, and Section 6 concludes.

2. Related Work

Metadata ensures interoperability and data integration across urban data systems (Conde et al., 2024; Gray, 2023). Despite promising achievements in UDTs, many existing approaches have difficulties addressing scalability and complexity challenges in large urban systems. Furthermore, the integration of distributed resources can lead to inconsistencies in the quality of metadata, which can become a barrier to analysis and effective decision-making (Ost et al., 2023; Somanath et al., 2024).

[1] https://www.asg.ed.tum.de/en/gis/projects/smart-district-data-infrastructure/

[2] https://metaver.de/

[3] https://data.europa.eu/en

Metadata standards, such as the Data Catalog Vocabulary[4] (DCAT), have been developed to enable seamless data exchange and integration within metadata catalogs. These standards provide guidelines on how to represent metadata in a way that facilitates discoverability, accessibility, interoperability and enhances the usability of such catalogs (Lisowska, 2016; Löbe et al., 2022; Lopez et al., 2012). Traditional metadata management systems often struggle with interoperability issues, lack of advanced interactivity, and the inability to represent complex relationships dynamically (Knezevic et al., 2022).

Graphs are important in metadata visualization and analysis, especially when dealing with diverse, large-scale, interconnected urban datasets (Sensarma, 2023). They can address challenges by offering intuitive network representations that capture relationships between metadata entities more naturally and interactively. Graph-based metadata analysis can improve the quality of metadata and facilitate decision-making by enabling stakeholders to visualize dependencies between urban data resources (Desimoni & Po, 2020; Po et al., 2020).

3. Metadata Challenges and Structural Modeling within the SDDI Framework for UDTs

Metadata plays a key role in ensuring consistent data integration, discoverability, and interoperability in UDTs. The way relationships between metadata catalog entries are defined has a direct impact on how effectively information can be retrieved, understood, and reused. Each metadata entry is represented as a node and can correspond to an entire dataset or service rather than an individual data instance. For example, one node may correspond to a complete digital city model, while another may represent a sensor data service aggregating measurements from all sensors in a particular district. Modelling semantically enriched metadata as typed graphs enables structured representation and supports knowledge discovery.

3.1 Graph-Responsive Metadata Challenges in UDTs

UDT catalogs often handle all resources equally and have limited mechanisms for evaluating the importance of the information resources based on their relations. Typical problems that arise in UDT metadata catalogs include the following:

1. Isolated Resources: Some resources appear without relationships to others, resulting in isolated nodes that limit the discoverability and reusability of information. Such information resources often indicate gaps in semantic linking or misclassifications, restrict interoperability between systems, and prevent the development of automated workflows. Identifying and reintegrating these isolated nodes through graph analysis can significantly improve metadata completeness and help to ensure that valuable resources contribute meaningfully to the UDT ecosystem.

2. Inadequate prioritization of highly important resources: In UDTs, important resources often include 3D city models or aerial images, which are usually provided by government authorities, such as land registry offices, as they form the basis for most other models and solutions. The UDT catalogs have limited mechanisms to promote them. Without relational analysis, stakeholders may overlook resources that are essential for simulations, planning, or visualization. The quality of the resources has a direct impact on the effectiveness of the entire system and are often referenced by various services, and applications, such as urban simulations, planning tools or visualisation platforms. Graph-based visualization and analysis make it possible to reveal the significance of such resources, which is crucial for prioritizing their quality assurance and maintenance.

3. Incomplete Links: Even when metadata is available, the relationships between entities may be weakly defined or entirely missing. This prevents automated workflows and limits the interoperability between systems and applications. Some relationships may be incomplete or missing due to human error or incomplete data entry. Link prediction helps to validate metadata structure by suggesting plausible missing edges, improving the catalog's completeness and usability. This can improve metadata discoverability and help users find relevant resources more efficiently.

4. Limited Detection of Domain-Specific Groupings: UDT catalogs often lack mechanisms to logically organize resources according to their thematic or domain-specific applications. Graph-based community detection algorithms can reveal clusters of closely related resources based on their semantic relationships, tags, or shared dependencies, and can help stakeholders to identify related datasets, to detect synergies between projects, and to avoid duplication of existing work and support discovering new use cases.

5. Circular Dependencies: Catalog entries may sometimes form dependency cycles, where a chain of relationships leads back to the original resource. These circular dependencies violate logical consistency and can cause significant problems. While such dependency loops may be valid in specific contexts, they must be explicitly defined and carefully validated to avoid errors.

3.2 Graphs in Smart District Data Infrastructure (SDDI)

The Smart District Data Infrastructure (SDDI) provides a standardized infrastructure for managing resources in UDT (Donaubauer et al., 2023). The SDDI metadata catalog is specialized for UDTs and manages distributed heterogeneous data and their links. The data model of the SDDI catalog is an extension of the DCAT schema. This catalog supports classifying information resources into nine sub classes: *Project, Online Service, Dataset, GeoObject, Device, Online Application, DigitalTwin, Software,* and *Method* (Knezevic et al., 2022). Three different types of relationships can be defined between the information resources in the catalog: *links to*, *depends on*, and *part of*. *Links to* relation describes a general reference between two resources (e.g., *Project* is linked to an *Online Service* which indicates that the service is relevant to the project context). A resource that requires another to function or to be interpretable (e.g., the relationship between the energy certificate and the building for which it is calculated) is linked with the *depends on* relation. *Part of* is used when one resource is a part of or belongs to a parent resource within a hierarchical structure where the child node depends on the parent node (e.g., a sub-project as a child node is part of a larger project realized as a parent node). Those relationships can be interpreted as a typed graph in which nodes represent the resources along with their type and further semantic information, such as metadata tags, and the edges represent the defined relationships between them, and the type of relationship (Knezevic et al., 2024). One example of such a graph is shown in Figure 1. The nodes' colours represent the main category assigned to this resource. Relations between nodes are represented as directed edges. Three different colours represent different relation types (Knezevic et al., 2022).

The semantic characteristics and mathematical properties of defined relationship types play an important role in specifying the meaning and interpretation of graph structures. Transitivity helps

[4] https://www.w3.org/TR/vocab-dcat-3/

in understanding indirect dependencies or hierarchical inheritance. A relationship is said to be transitive if, whenever A is related to B and B is related to C, then A is also implicitly related to C. Antisymmetry helps enforce clear directional relationships and prevents cycles in hierarchies. A relation is said to be antisymmetric if A is related to B, then B must not be related to A unless they refer to the same entity. Irreflexivity prevents logical errors like self-dependency. These three properties are illustrated in Figure 2, which shows a transitive relation (left), an antisymmetric relation (middle), and an irreflexive relation (right).

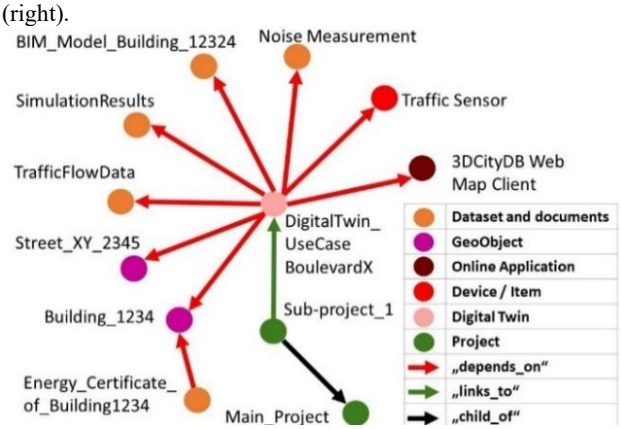

Figure 1. Graph-Based Illustration of Catalog Resources and Their Dependencies (adapted from Knezevic et al., 2022). Each node represents a catalog entry, which may represent individual world objects, but also datasets providing data on entire object sets.

The *Depends on* relation is transitive because if A depends on B and B depends on C, then A depends on C. For example, suppose *Dashboard A* displays real-time traffic analysis and relies on *Predicted Dataset B*, which is generated using *Sensor Data C* from traffic detectors. In that case, if *Sensor Data C* is missing or corrupted, *Dashboard A* will be indirectly affected. It is antisymmetric because if A depends on B, then B should not depend on A. For example, *3D Building Model* depends on *LiDAR Point Cloud Dataset*, and the *3D Building Model* is derived from *LiDAR Point Cloud Dataset*, not the other way around. This prevents circular dependencies, which are a serious problem for system logic, automation, and consistency. This relation is irreflexive since resources should not depend on themselves.

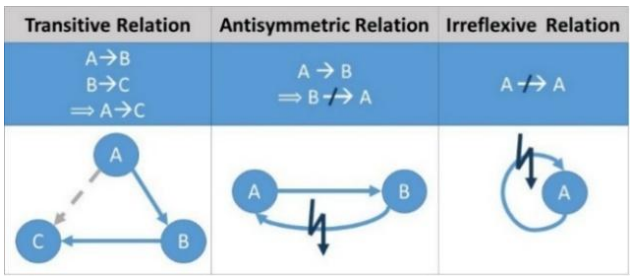

Figure 2. Transitive (*left*), Antisymmetric (*middle*), and Irreflexive (*right*) Relations in Directed Graphs.

The *Part of* relation is a hierarchical relation. It is transitive because if A is a part of B, and B is a part of C, then A is a part of C. For example, if *Noise Sensor Task A* is child of *Smart Mobility Project B* and *Smart Mobility Project B* is child of *City Digital Twin Initiative C,* then *Noise Sensor Task A* is also part of *City Digital Twin Initiative C* even if not directly related. This relation is antisymmetric since if A is a part of B, B cannot be a part of A. That would destroy the hierarchy. For example, if *Urban Energy Monitoring Project A* is child of *Electric Vehicle Charging Study B,* then *Electric Vehicle Charging Study B* cannot be part of *Urban Energy Monitoring Project A* since this creates an illogical loop and breaks the concept of a hierarchy. *Part of* relation is irreflexive because a resource can't be a part of itself.

The *Links to* relation is not transitive because A links to B and B links to C doesn't necessarily imply A links to C. For example, if *Mobility Dashboard App A* links to *Real-Time Traffic API B*, which links to *Sensor Network Maintenance Schedule C*, *Mobility Dashboard App A* does not implicitly link to *Sensor Network Maintenance Schedule C* since *Mobility Dashboard App A* does not directly use that schedule. This relation is not antisymmetric since A might link to B, and B might link back to A, which can be valid in bidirectional workflows. For example, if *Interactive Map Viewer A* links to *Geospatial Tile Server B* it could be that *Geospatial Tile Server B* links to *Interactive Map Viewer A* because it's configured to log or prioritize client requests from viewer.

Table 1 summarizes the transitivity, antisymmetry, and irreflexivity of the *depends on*, *part of*, and *links to* relationship types in the SDDI metadata graph. These logical properties help ensure consistency and avoid structural conflicts such as circular dependencies or self-referential links.

Relationship type	**Transitive** [Yes/No]	**Antisymmetric** [Yes/No]	**Irreflexive** [Yes/No]
Depends on	Yes	Yes	Yes
Part of	Yes	Yes	Yes
Links to	No	No	Yes

Table 1. Logical Properties of Metadata Relationships in Directed Graphs.

4. Analysis of Use Cases and Findings

In the context of UDTs, graph theory offers a powerful way to structure, represent, and explore the semantic relationships between various digital resources (Bondy & Murty, 1976). Graph-based approaches are used to identify patterns and dependencies to provide knowledge that can be applied to urban management strategies (Nguyen., 2024).

This section will address the challenges introduced in Section 3.1. The following examples show how the implemented system enhances quality assessment, relationship analysis, and decision-making in real-world urban data scenarios. The evaluation was based on the SDDI catalog created as part of the SAVeNoW[5] research project. The main goal of the project was to analyse the development and operation of a digital twin for urban mobility, with the city of Ingolstadt in Germany being used as a case study. The SDDI catalog was used to organize and link the diverse digital resources involved in the real and simulation environment, serving as a foundational tool for data integration, consistency, and metadata analysis between thirteen project partners. The SAVeNoW catalog[6] has 81 publicly available datasets. Manual inspection confirmed that the applied methodology correctly identified issues that represent actual problems in the catalog.

4.1 Detection of Important Resources

In the graph, important resources are represented by nodes with many incoming edges (see challenge 2, section 3.1). These resources are considered important because they are widely

[5]https://savenow.de/de/

[6] https://catalog.savenow.de/

reused or referenced across various components of the UDT system. An analysis of the nodes' centrality is helpful in identifying important resources and is detected with the centrality measure *PageRank*. *PageRank* is a graph-based ranking algorithm that measures the importance of nodes in a directed network by evaluating the number of incoming links (Xing & Ghorbani, 2004). This algorithm is well-suited for directed graphs, as it not only counts incoming edges, like the *In-degree centrality* algorithm, but also ranks them according to the importance of their sources. While the *Eigenvector centrality* algorithm is conceptually similar to *PageRank*, it performs poorly in directed and weakly connected networks. *PageRank* ensures stable and non-zero scores across all nodes, even in disconnected graphs, and it was therefore chosen as a more robust and scalable centrality measure for identifying important resources in UDT catalogs. The centrality value of the nodes is calculated using the *pagerank()* function from NetworkX. Figure 3 shows the detection of important resources in the UDT metadata catalog. The scale from red to white represents the centrality value, where intense red indicates high centrality and white indicates low centrality. This visualization is providing a quick visual indication of key information resources within the catalog. In addition, a ranking of the five most central nodes and the distribution of the centrality values are provided (shown in Figure 3 on the right). This enables the detection of resources in the catalog that should be prioritized regarding their maintenance and validation.

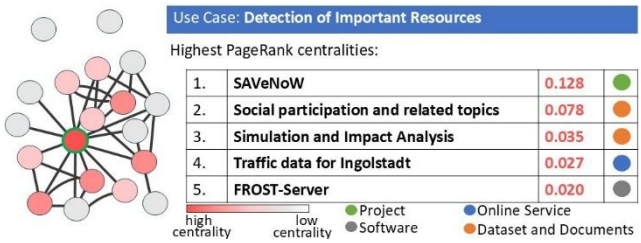

Figure 3. Example Graph from the *Detection of Important Resources* Use Case (Fuchsloch, 2024). Each node represents a resource in the catalog, and the edges represent any of the three types of relationships.

4.2 Community Detection

Community detection helps to identify clusters of closely related resources (see challenge 4, section 3.1). For community detection, the function *louvain_communities()* is used. The Louvain method is chosen because it is an efficient, hierarchical, and modular-based algorithm designed to detect communities by optimizing the modularity of a network. Modularity is a function that measures the density of links inside communities compared to links between communities. The algorithm operates in two main phases: initially, each node is assigned to its own community, and nodes are repeatedly compared with neighbouring communities. In the second phase, nodes belonging to the same community are aggregated into super-nodes, and the process repeats until no further improvement in modularity is possible (De Meo et al., 2011). Community detection can also divide large networks into smaller, more easily manageable parts that can be analyzed and interpreted separately (Newman, 2018). It helps stakeholders to explore new use cases, identify missing data, and promote collaborations. In this approach, it is assumed that resources that are closely related, thematically similar, have defined the same tags, or are used for the same application of the UDT. Alternative algorithms such as *Girvan–Newman*, which is computationally expensive and impractical for large graphs, or *Label Propagation,* which is fast but produces unstable and difficult-to-reproduce results, are less suitable for UDT metadata catalogs.

Detected communities are shown in Figure 4, where the nodes are grouped by community, using color to show clusters of closely connected resources. Each node represents one resource, and each edge represents a semantic relationship. The community represents a set of resources that are more closely connected to each other than to the rest of the graph. Associated metadata tags help interpreting thematic groupings of resources. The tags attached to these groups can be used to conclude the subject area of their resources. Figure 4 shows tags from the purple group, which can be interpreted, for example, as their resources part of an UDT that represents a winter road maintenance service because *Winter maintenance* is one of the most frequently used tags in that cluster. It is also possible to see that in total 15 communities were identified, with five of them highlighted in the legend using distinct colors. Each color matches the nodes grouped in the network graph, enabling visual identification of individual clusters. Below the community labels, a summary of the most frequent metadata tags found within each community is provided.

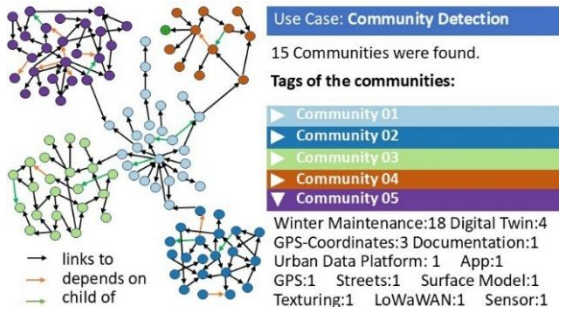

Figure 4. Example graph from the *Community Detection* Use Case (Fuchsloch, 2024).

4.3 Detection of Isolated Resources

The completeness and connectivity of metadata are crucial to enable meaningful analysis and simulations in UDTs. Large-scale metadata catalogs often contain isolated resources (see challenge 1, section 3.1). The *Detection of Isolated Resources* use case aims to identify resources isolated from the rest of the connected graph without links to other information resources. It helps to indicate missing relationships, incomplete catalog entries, or poor data integration within the catalog. The identification of disconnected nodes is an important step in quality assurance.

In a directed graph, isolated nodes can be identified by analyzing nodes with both an in-degree and out-degree of 0, which can be detected using the *isolates()* function. Alternative methods, such as *k-core decomposition* or *connected component* analysis, could also reveal isolated or weakly connected resources. However, since the detection of completely disconnected nodes is a well-defined and computationally simple task, the use of degree-based identification (*isolates()*) is the most efficient and interpretable choice for large-scale metadata catalogs. The function *number_weakly_connected_components()* allows for the analysis of a graph once completely isolated nodes have been removed to determine the number of weakly connected components. The individual isolated nodes are displayed in a red colour (Figure 5). In addition, a pie chart has also been placed next to the visualization, showing the proportion of completely isolated nodes in the overall graph.

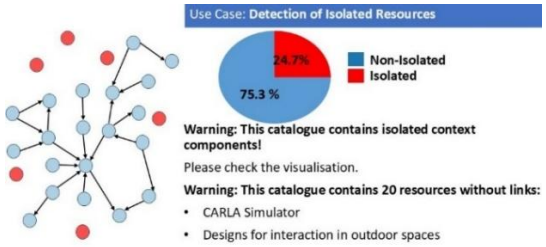

Figure 5. Example graph from the *Detection of Isolated Resources* Use Case (Fuchsloch, 2024).

4.4 Cycle Detection

In directed graphs, cycles are defined as paths that follow the edge directions and where the start node corresponds to the end node (see challenge 5, section 3.1). As defined in Section 3.2, it is possible to define three types of semantics relations between resources. In terms of cycle detection, three distinct categories must be distinguished (*depends on cycle*, *part of cycle*, *links to cycle*) as each represents a fundamentally different semantic relationship. To ensure consistent and interpretable metadata structures in the UDT catalog, the semantic characteristics of relationships must be clearly defined since each type of relationship in the SDDI graph implies different constraints.

A *depends on cycle* results when two or more resources are linked so that each depends on the others, creating a closed dependency circle. Mutual dependencies may be acceptable in some cases, but they should be explicitly validated, as they can lead to ambiguous interpretations and complicate processing. A cycle in a transitive *depends on* relationship means that every resource in the cycle is dependent on every other resource even if not all pairwise *depends on* links are explicitly present. This behaviour is similar to a graph clique, where every node of the clique is connected to every other node. Recognising such cycles is crucial, as they can identify tightly coupled resources or problems in architecture modelling in the context of the Urban Digital Twin. For example, if *Dataset A* depends on *Dataset B*, and *Dataset B* depends on *Dataset A*, then neither can exist without the other, creating a closed loop of mutual dependency. Cycles in *depends on* relationships are therefore problematic, as they may indicate logical conflicts or tightly coupled systems that should instead be merged or modularized. Rare, mutual dependencies may exist in practice (e.g., the traffic signal A provides real-time traffic data to the optimization software B, which depends on the traffic simulation method C to evaluate and select optimal traffic strategies, whose outputs are then used to update the behaviour of the traffic signal A). A *depends on cycle* occurs in optimization and simulation workflows when the system is adaptive, real-time, or interactive with the physical world. These cycles reflect the feedback loops essential to closing the gap between sensing and controlling. However, such tightly coupled systems should be modelled and interpreted carefully. Because *depends on* is transitive, a cycle implies that each resource in the loop is implicitly dependent on every other one, forming a tightly interlinked group. A *part of cycle* occurs if a hierarchical relationship is identified as a subcomponent of itself (e.g., *Project A* is part of *Sub-Project B*, *Sub-Project B* is part of *Project A*). Aggregation hierarchies are non-cyclical by nature and a cycle represents a structural error. If a *links to cycle* is recognized, it is typically not incorrect. Such cycles should be examined in order to eliminate possible human mistakes and to justify why the cycle was defined. Sometimes, resources are naturally interconnected - for example, an *Online Service A* links to a *Dashboard B*, and *Dashboard B* links back to *Online Service A* since the service provides the functionality for the dashboard and the dashboard helps managing the service. However, *links to* cycles may reflect redundant, ambiguous, or poorly modeled relationships that can reduce the interpretability and efficiency of the metadata structure – for example *Dataset A* links to *Dataset B*, and *Dataset B* links back to *Dataset A* without a clear reason. This is especially problematic if both datasets contain similar information or if one is a derived version of the other, but the catalog lacks proper indication of versioning or data lineage. In such cases, the mutual links may suggest equivalence or dependence that doesn't exist, leading to confusion and possible misinterpretation by users or automated systems. *Links to* is intentionally more flexible than the other two relation types and lacks strict logical constraints. While this flexible approach allows a wide range of associations, it also makes such links open to redundancy and ambiguity, especially when forming cycles.

The function *simple_cycles*() is used to identify cycles within the directed graph created from the metadata catalog. The result of this function is a list of the nodes of the found cycles. The type of relationship can be used to determine the dependency of a resource if a loop that only contains links of the type *depends on*. By identifying such loops, redundant or incorrect dependencies can be revised, improving metadata consistency. This method was selected because SDDI metadata graphs are directed and have different semantic relations that must be distinguished. Alternative approaches, such as *Depth-First-Search* (DFS) based detection or topological sorting, can identify whether cycles exist but do not enumerate or classify them. Detected loops are visually highlighted in the graph visualization to make them easy to identify. The edges and nodes that are part of a loop are shown in red (Figure 6). This helps users to quickly locate and inspect problematic relationships in the metadata structure.

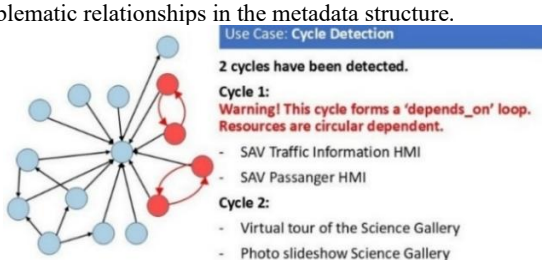

Figure 6. Example graph from the *Cycle Detection* Use Case (Fuchsloch, 2024).

4.5 Link Prediction

The aim of the *Link Prediction* use case is to identify missing, potential, or future relationships between information resources (see challenge 3, section 3.1). Based on the assumption that pairs of resources linked to similar resources could also be linked, an Adamic-Adar index can be used. This index measures the likelihood of a connection between two nodes by evaluating the number of shared neighbors, giving higher weight to neighbors that are less connected overall, thus emphasizing more informative or rare connections (Adamic & Adar, 2003; Liben-Nowell & Kleinberg, 2003). Compared to simpler metrics such as *Jaccard Coefficient* or *Preferential Attachment*, *Adamic–Adar* is consistent with the goal of identifying missing semantic links in UDT metadata. More advanced metrics, such as the *Katz Centrality* index or *SimRank*, are less practical for integration into an interactive tool due to their higher computational costs and complexity. Using the *adamic_adar_index()* function, it is possible to compute the link prediction for all node pairs in the graph. It should be noted that the proposed method for predicting relationships can only be applied if a correct relationship has already been defined. Based on this index, the user interface displays the top ten suggested links, enabling users to explore

new relationships and illustrating the potential for enriching the catalog with relevant connections. This method can help avoid missing links, but it does not specify a direction or type of relationship. A stakeholder must decide to edit this resource in the catalog.

Figure 7 illustrates the results of a link prediction use case where predicted potential links are highlighted in red. The right side lists the top five most probable new connections, where the strength of each predicted link is quantified by its similarity score.

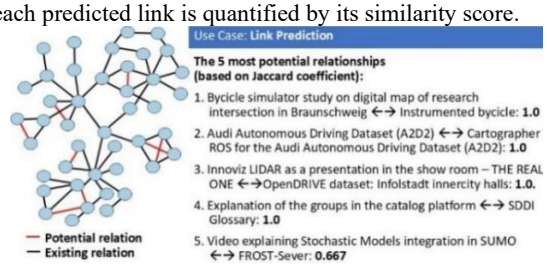

Figure 7. Example graph from the *Link Prediction* Use Case (Fuchsloch, 2024).

5. Graph-Based Framework and System Implementation

Graphs are defined as mathematical structures that represent networks of entities as nodes and their connections as edges, illustrating relationships between different elements in a structured way. Graph analysis algorithms can be used to evaluate various aspects of network structure (Bondy & Murty, 1976). Graph theory has become increasingly relevant in urban data systems, particularly metadata visualization and analysis. For example, graph-based approaches can identify important resources, track dependencies between resources, detect redundant resources or identify relations between components in a smart city environment.

5.1 Graph-Based Metadata Visualization: System Development and Integration

This section describes the system development and integration of a graph-based metadata visualization framework, focusing on data retrieval, graph creation, and visualization techniques. The proposed system provides a structured approach to explore metadata relationships through graph-based representations (Fuchsloch, 2024). Figure 8 illustrates the process of collecting, analyzing, and visualizing metadata from the SDDI Catalog using CKAN-API, NetworkX, Cytoscape, and Dash.

5.2 Data Retrieval

The SDDI Catalog is based on the open-source data management software CKAN[7]. CKAN provides a robust framework for organizing metadata and managing data catalogs worldwide. An important feature of CKAN is its RESTful API, which allows external code to access catalog functionalities and metadata about information resources. All information resources registered in the catalog can be retrieved through specific queries, with the responses returned in JSON format.

The Data Retrieval process is shown in the top-left section of Figure 8 and is highlighted in red. The CKAN API provides structured access to metadata entries using specific endpoints, such as *package_show* and *group_list*, which return detailed descriptions of registered information resources. To accurately model metadata dependencies, the SDDI catalog specifies relationships between resources using two key parameters: *relationships as subject* and *relationships as object*. If a resource appears in *relationships as subject*, this indicates that the resource is the starting point of the relationship and represents an outgoing edge. If a resource appears in *relationships as object*, this indicates that the resource is the endpoint of the relationship and represents an ingoing edge. This structure ensures that directed graphs correctly represent dependencies and references within the metadata catalog and that directed edges between the resources can be set in the graph. The retrieved metadata is fed into the NetworkX Library, where it is structured as a directed graph (Di-Graph) for further analysis and visualization.

5.3 Graph Construction and Analysis

The open-source Python package NetworkX[8] is used to create and analyze complex networks (green section of Figure 8). NetworkX provides a range of functionalities for constructing graphs, including adding nodes and edges, calculating metrics such as centrality, and executing graph algorithms for tasks like community detection and cycle identification. The input in this package is the metadata retrieved from the SDDI Catalog via the CKAN API. It maps information resources as nodes and relationships are mapped as directed edges. The mapping of the resources is stored in the interface as a NetworkX object of the DiGraph[9] (Directed Graph). The attributes defined in the metadata are stored as key values of the corresponding nodes and edges. The constructed graph is then analyzed using various graph-based techniques, including *Community Detection, Dataset Ranking, Cycle Detection, Link Prediction, Detection of Important Resources,* and *Detection of Isolated Resources* (Section 4).

5.4 Graph Visualization, Interactive Interface and User Interaction

The Cytoscape and Dash components are used for graph visualization and interactive exploration of metadata relationships. To enable interactive visualization, the NetworkX graph is converted into a Cytoscape[10] graph for visualization (blue section of Figure 8). It provides an interactive graph representation, enabling users to dynamically explore metadata relationships. For the visualization stylesheets and layouts are used. Stylesheets customize the appearance of nodes and edges based on metadata attributes (e.g., resource type). Layouts organize graphs to improve readability, using force-directed layouts for intuitive structural representation (Chadzynski et al., 2021).

The layout should follow certain criteria and ensure the readability of the graph (Fuchsloch, 2024; Kaufmann & Wagner, 2001; Tarawaneh et al., 2013):
- the edge lengths should be as short as possible to be able to recognize links more quickly,
- crossing edges should be minimized, as crossing makes it more challenging to follow connections,
- the angle between the edges should be as large as possible to improve readability and distinguish arrow directions,
- the curved edges should be avoided, as a user can easily follow straight edges,
- the labels should be displayed in a readable size, font, and colour,
- each label must be clearly assigned to a graph component (vertex/edge),

[7] https://ckan.org/
[8] https://networkx.org/
[9] https://networkx.org/documentation/stable/reference/classes/digraph.html
[10] https://dash.plotly.com/cytoscape

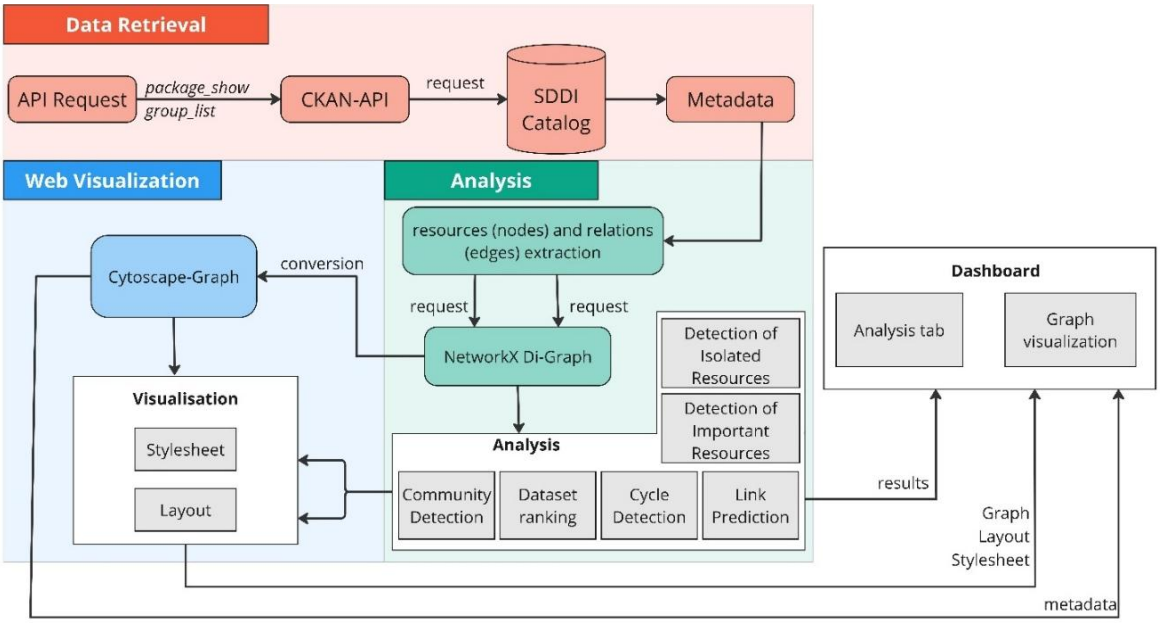

Figure 8. Workflow for Graph-Based Analysis and Visualization of Metadata in the SDDI Catalog (adapted from Fuchsloch, 2024).

- the labels should not overlap with other labels or elements.

As such, *force-directed* layouts meet these requirements and are used for graph visualization. These layouts make complex graph structures more readable and can display clusters, hierarchies, and dependencies between metadata resources.

The final visualization and analysis results are integrated into a Dash-based web interface (white section of Figure 8). Dash[11] is an open-source Python package for creating web-based applications. It provides a graphical user interface (GUI), enabling interaction with metadata visualization. This package enables application creation with various interactive elements. Due to the support of a wide range of diagrams, it is particularly suitable for data analysis and visualization. The graph visualization enables the user to visually mark elements and to change the level of detail in the representation. This includes zooming and panning of the graph and highlighting the connections of the nodes. The analysis tab provides detailed insights, while the graph visualization enables dynamic exploration of metadata relationships using graph layouts and stylesheets. This integration ensures that users can visualize the graph structure intuitively, facilitating better engagement and understanding of the data. In this workflow, Cytoscape handles the rendering and interactivity of the graph, including node positioning, styling, zooming, and selection. Dash provides the overall web application framework, organizing the layout of the interface, dealing with user inputs, and executing backend logic via Python. Using Dash callbacks, interactions with the Cytoscape graph (e.g., selecting a node or edge) can trigger real-time updates within the application, such as displaying detailed metadata, performing analyses, or modifying visual components. This integration enables the development of rich, interactive, graph-based tools.

6. Conclusion and Future Work

This study presents a comprehensive graph-based approach for analyzing and visualizing metadata within Urban Digital Twins (UDTs), specifically within the Smart District Data Infrastructure (SDDI) framework. The study contributes a novel perspective on graph-based analysis of metadata in UDTs by explicitly addressing the semantics of metadata catalogs. The proposed graph-based approaches improve the data quality and usability of the UDT metadata catalog and enhance the transparency and usability of UDT catalogs demonstrated through practical use cases. Graph-based approaches provide a structured and scalable metadata visualization and analysis solution in UDTs. The integration of graph-based metadata representation allows stakeholders to explore urban data networks more efficiently, supporting data-driven decision-making in smart city environments.

Other graph metrics and algorithms could also be used, although this could lead to the loss of important semantic information that need to be kept. The current implementation is tested on small to medium-sized catalogs and relies on metadata completeness and accuracy. As metadata catalogs grow in the number of registered resources, some relationships between resources may be overlooked or not explicitly documented, which leads to disconnected resources. Automatic link prediction will help to identify potential or missing connections between information resources based on existing structural patterns within the graph. The work extends graph-based analysis to the level of metadata semantics in the context of UDT metadata catalogs, which has been scarcely researched, and thereby complements research on knowledge graphs and linked data visualization. Future research will focus on optimizing performance while ensuring that visualizations remain easy to understand, even with large amounts of data. Future work in this field will focus on the integration of advanced machine learning techniques for automatic link prediction. Further development will aim to identify new use cases, such as duplicate recognition. This can be especially useful when harvesting metadata from external catalogs, as the same information may already be registered in the target catalog. One approach to detecting these could be to compare the names, tags, or geographical locations. Given the practical relevance of graph-based analyses for managing UDT, both graph-analyses and graph-based visualization should be natively integrated into the end-user interface of the SDDI catalog. This integration would allow users to conduct analysis and access visualizations directly within the catalog interface, avoiding the need for a separate application.

[11] https://github.com/plotly/dash

Acknowledgements

This work was supported by the Bavarian State Ministry for Digital Affairs in the TwinBy Project, and by the Federal Ministry of the Interior and Community in the Connected Urban Twins Project, in cooperation with the City of Munich, and by the Federal Ministry for Digital and Transport in SAVeNoW Project, and the EU INTERREG program in the Twin4Clim Project.

References

Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the Web. Social Networks, 25(3), 211–230.

Bauer, F., & Kaltenböck, M. (2011). Linked open data: The essentials. Edition mono/monochrom, Vienna, 710(21).

Chadzynski, A., Krdzavac, N., Farazi, F., Lim, M. Q., Li, S., Grisiute, A., Herthogs, P., von Richthofen, A., Cairns, S., & Kraft, M. (2021). Semantic 3D City Database – an enabler for a dynamic geospatial knowledge graph. Energy and AI, 6, 100106.

Bondy, J. A., & Murty, U. S. R. (2008). Graph theory with applications (Vol. 290). London: Macmillan.

Conde, J., Pozo, A., Munoz-Arcentales, A., Choque, J., & Alonso, Á. (2024). Fostering the integration of European Open Data into Data Spaces through High-Quality Metadata. CoRR, abs/2402.06693. https://doi.org/10.48550/ARXIV.2402.06693

De Meo, P., Ferrara, E., Fiumara, G., & Provetti, A. (2011). Generalized Louvain method for community detection in large networks. 2011 11th International Conference on Intelligent Systems Design and Applications, 88–93.

Desimoni, F., & Po, L. (2020). Empirical evaluation of Linked Data visualization tools. Future Generation Computer Systems, 112, 258–282. https://doi.org/10.1016/j.future.2020.05.038

Donaubauer, A., Knezevic, M., Willenborg, B., Bobinger, S., Morich, L., & GmbH, B. I. (2023). Leitfaden – Urbaner Digitaler Zwilling nach der Methodik der SDDI (Version 1.2). https://mediatum.ub.tum.de/node?id=1725270&change_language=en

Frasincar, F., Telea, A., & Houben, G.-J. (2006). Adapting Graph Visualization Techniques for the Visualization of RDF Data. In V. Geroimenko & C. Chen (Hrsg.), Visualizing the Semantic Web: XML-Based Internet and Information Visualization (S. 154–171).

Fuchsloch, F. (2024). Graph-basierte Analyse und Visualisierung von Metadaten im Kontext Urbaner Digitaler Zwillinge, Bachelor's thesis, Technical University of Munich. https://mediatum.ub.tum.de/node?id=1747862

Gil, Y., Ratnakar, V., & Deelman, E. (2006). Metadata Catalogs with Semantic Representations. In L. Moreau & I. Foster (Hrsg.), Provenance and Annotation of Data (Bd. 4145, S. 90–100). Springer Berlin Heidelberg.

Gray, J. W. Y. (2023). What do data portals do? Tracing the politics of online devices for making data public. 5(e10).

Kaufmann, M., & Wagner, D. (Hrsg.). (2001). Drawing Graphs: Methods and Models (Bd. 2025). Springer Berlin Heidelberg.

Khan, H., DeMarco, C., Fernsebner Eslao, C., Folsom, S., Kovari, J., Warner, S., ... & Usong, A. (2022). Using linked data sources to enhance catalog discovery. KULA, 6(3), 1-26.

Knezevic, M., Donaubauer, A., & Kolbe, T. H. (2024). SDDI - Minimal Ecosystem for the Establishment of Urban Digital Twins. Publikationen Der DGPF, 32, 151–168.

Knezevic, M., Donaubauer, A., Moshrefzadeh, M., & Kolbe, T. H. (2022). Managing Urban Digital Twins with an Extended Catalog Service. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, X-4/W3-2022, 119–126.

Kolbe, T. H., Moshrefzadeh, M., Chaturvedi, K., & Donaubauer, A. (2020). The Data Integration Challenge in Smart City Projects. Chair of Geoinformatics, Technical University of Munich. https://mediatum.ub.tum.de/doc/1554725/1554725.pdf

Liben-Nowell, D., & Kleinberg, J. (2003). The link prediction problem for social networks. Proceedings of the Twelfth International Conference on Information and Knowledge Management, 556–559.

Lisowska, B. (2016). How can Data Catalog Vocabulary (DCAT) be used to address the needs of databases? (No. Discussion Paper No. 5, Joined-up Data Standards Project). https://www.w3.org/2016/11/sdsvoc/SDSVoc16_paper_31

Lnenicka, M., Nikiforova, A., Clarinval, A., Luterek, M., Rudmark, D., Neumaier, S., Kević, K., & Bolívar, M. P. R. (2024). Sustainable open data ecosystems in smart cities: A platform theory-based analysis of 19 European cities. Cities, 148, 104851.

Löbe, M., Ulrich, H., Beger, C., Bender, T., Bauer, C., Sax, U., Ingenerf, J., & Winter, A. (2022). Improving Findability of Digital Assets in Research Data Repositories Using the W3C DCAT Vocabulary. MedInfo, Volume 290 of Studies in Health Technology and Informatics, IOS Press, 61–65.

Lopez, V., Kotoulas, S., Sbodio, M. L., Stephenson, M., Gkoulalas-Divanis, A., & Aonghusa, P. M. (2012). QuerioCity: A Linked Data Platform for Urban Information Management. The Semantic Web – ISWC 2012, Springer Berlin, 148 –163.

Mutton, P., & Golbeck, J. (2003, July). Visualization of semantic metadata and ontologies. In Proceedings on Seventh International Conference on Information Visualization, 2003. IV 2003. (pp. 300-305). IEEE.

Newman, M. E. J. (2018). Networks (Second edition). Oxford University Press. https://books.google.de/books/about/Networks.html?id=YdZjDwAAQBAJ&redir_esc=y

Nguyen, S. H. (2024). Automatic Detection and Interpretation of Changes in Massive Semantic 3D City Models. Dissertation, Technical University of Munich. https://mediatum.ub.tum.de/doc/1765978/1765978.pdf

Ost, P., Shakeel, Y., & Tögel, P. (2023). Data Collections Explorer: An Easy-to-Use Tool for Sharing and Discovering Research Data. Proceedings of the Conference on Research Data Infrastructure, 1.

Po, L., Bikakis, N., Desimoni, F., & Papastefanatos, G. (2020). Linked Data Visualization: Techniques, Tools, and Big Data. Springer International Publishing.

Sensarma, D. (2023). Applications of Graphs in Smart Cities. In S. Pramanik & K. M. Sagayam (Hrsg.), Advances in Data Mining and Database Management (S. 40–54). IGI Global.

Somanath, S., Naserentin, V., Eleftheriou, O., Sjölie, D., Wästberg, B. S., & Logg, A. (2024). Towards Urban Digital Twins: A Workflow for Procedural Visualization Using Geospatial Data. Remote Sensing, 16(11).

Tarawaneh, R. M., Keller, P., & Ebert, A. (2013). A General Introduction To Graph Visualization Techniques. Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering-Proceedings of IRTG 1131 Workshop 2011, 151-164.

Ullah, I., Khusro, S., Ullah, A., & Naeem, M. (2018). An overview of the current state of linked and open data in cataloging. Information Technology and Libraries, 37(4), 47-80.

Xing, W., & Ghorbani, A. (2004). Weighted PageRank algorithm. Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004., 305–314.