TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Geoinformatik

Graph-basierte Analyse und Visualisierung von Metadaten im Kontext Urbaner Digitaler Zwillinge

Bachelor's Thesis im Studiengang
Geodäsie und Geoinformation

Felix Fuchsloch

München 2024

Bachelor's Thesis im Fach
Geoinformatik

eingereicht an der Technischen Universität München

Lehrstuhl für Geoinformatik
TUM School of Engineering and Design
Technische Universität München

Thema:

Graph-basierte Analyse und Visualisierung von Metadaten im Kontext Urbaner Digitaler Zwillinge

Thema Englisch:

Graph-based analysis and visualization of metadata in context of Urban Digital Twins

Verfasser:	Felix Fuchsloch
Matrikelnummer:	03737982
Referent:	Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe
Betreuer:	Dr.-Ing. Andreas Donaubauer
	Mag. Ing. Marija Knezevic

Begonnen am:	15. Dezember 2023	
Eingereicht am:	15. Mai 2024	
Abschließend beurteilt am:		Note:

Erklärung

Ich versichere hiermit, dass ich die von mir eingereichte Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Petershausen, 15. Mai 2024

Felix Fuchsloch

Abstract

Mit Urbanen Digitalen Zwillingen (UDZ) können Städte, Quartiere und Regionen digital repräsentiert werden. Sie sind Sammlungen von vielfältigen Ressourcen, die durch Analysen und Beobachtungen dieser, Rückschlüsse auf den aktuellen Zustand des Repräsentierten Gebiets ermöglichen. Im Rahmen der Realisierung eines UDZ mit der Smart District Data Infrastructure (SDDI), werden sämtliche Ressourcen anhand ihrer Metadaten in einem SDDI-Katalog registriert. [1] Da die Ressourcen untereinander in Beziehung stehen, kann der SDDI-Katalog auf einen Graphen abgebildet werden. Diese Arbeit untersucht, anhand einer Auswahl von Use Cases, welche Informationen aus der Graphstruktur der Metadaten abgeleitet werden können und welche Aspekte bei ihrer Visualisierung beachtet werden müssen. Darüber hinaus wurde eine eigene Schnittstelle erstellt, in der die Graphstruktur, im Rahmen der erarbeiteten Use Cases, visualisiert und analysiert werden kann. Es konnte festgestellt werden, dass eine graph-basierte Analyse und Visualisierung der Metadaten dazu beitragen kann, die Datenqualität eines SDDI-Katalogs zu erhalten und Informationen zur Struktur und Organisation des Katalogs zu erlangen.

Inhaltsverzeichnis

1 Einleitung

Mit der fortschreitenden Urbanisierung der Gesellschaft, stehen Städte und Kommunen immer mehr vor großen Herausforderungen, die effektive Entscheidungen und Planungen erfordern. Urbane Digitale Zwillinge (UDZ) bieten eine Möglichkeit, sie dabei zu unterstützten. Sie sind Sammlungen von vielfältigen Daten von verschieden Teilhabern, mit denen relevante Teile der realen Welt digital abgebildet werden können. Durch Analysen und Beobachtungen am UDZ, können dann Erkenntnisse über den aktuellen Zustand des repräsentierten Gebiets gewonnen werden. Das Smart District Data Infrastructure Konzept (SDDI) des Lehrstuhls für Geoinformatik an der TU München, bietet ein Rahmenwerk für die Umsetzung der Dateninfrastruktur eines UDZ mittels eines offenen und standardisierten Datenaustausches zwischen den diversen Teilhabern. Im Zentrum dieses Konzepts steht der SDDI-Katalog, in dem die digitalen Ressourcen eines UDZ, mittels ihrer Metadaten registriert werden. Da die digitalen Ressourcen untereinander verknüpft sind, kann dieser Katalog auch auf einen Graphen abgebildet werden. [1]

1.1 Zielsetzung der Arbeit

Das Ziel dieser Arbeit ist es, zu untersuchen, welche Informationen sinnvollerweise aus der Graphstruktur der Metadaten eines SDDI-Katalogs abgeleitet werden können. Hierfür sollen Use Cases identifiziert und analysiert werden, die von einer graph-basierten Analyse und Visualisierung Gebrauch machen und neue Erkenntnisse über die Ressourcen eines Urbanen Digitalen Zwilling ermöglichen. Dabei soll auch betrachtet werden, welche Anforderungen es an die Visualisierung des Graphen gibt. Die gefundenen Use Cases sollen dann mithilfe einer eigens erstellten Schnittstelle praktisch umgesetzt werden. Dadurch könnten neue Ansätze gefunden werden, mit denen die Nutzbarkeit und Verwaltung eines SDDI-Katalogs verbessert werden können.

1.2 Aufbau der Arbeit

Diese Arbeit ist aus 3 wesentlichen Abschnitten aufgebaut. Im ersten Abschnitt werden die theoretischen Grundlagen aus der Graphentheorie und dem SDDI-Konzept erläutert. Im darauf folgenden Abschnitt werden die in der Zielsetzung erwähnten Forschungsfragen beantwortet, indem die Anforderungen an die Visualisierung der graphischen Abbildung eines SDDI-Katalogs diskutiert werden und Informationen, die aus der Graphstruktur abgeleitet werden können, im Rahmen von Use Cases vorgestellt werden. Im Anschluss daran wird die eigene Implementation einer Schnittstelle zwischen den Metadaten in einem SDDI-Katalog und Analyse und Visualisierungswerkzeugen präsentiert, welche die theoretisch konzipierten Use-Cases praktisch umsetzen kann. Zuletzt werden die Erkenntnisse dieser Arbeit diskutiert, Grenzen der eigenen Implementierung erläutert und abschließend Vorschläge zur Weiterentwicklung des SDDI-Katalogs gegeben.

2 Theoretische Grundlagen

2.1 Graphentheorie und Knowlegegraphs

Allgemein betrachtet sind Netzwerke, eine Sammlung von Knoten (eng. nodes) welche durch Kanten (eng. edges) verknüpft sind. Hierbei ist der Kontext des Netzwerks dafür ausschlaggebend, was die Knoten und Kanten repräsentieren. In einem sozialen Netzwerk könnten Knoten einzelne Personen, und die Kanten Freundschaften zu anderen Personen im Netzwerk repräsentieren. Im Kontrast dazu können sie in einem Netzwerk wie dem Internet einzelne Computer und ihre Verbindungen zu anderen darstellen. [2] Mithilfe von Graphen können Netzwerke mathematisch modelliert werden[3], und durch verschiedene Untersuchungen des Graphs, mittels Algorithmen und Metriken, können verschiedene Aspekte der Netzwerkstruktur analysiert werden (vgl. Kapitel 3). [4]

Es wird zwischen gerichteten und ungerichteten Graphen unterschieden (vgl. Abbildung 2.1). Bei gerichteten Graphen ist eine Kante ein Paar von Knoten, wovon einer der Start und der Andere der Endknoten ist. Bei ungerichteten Graphen ist für Kanten keine Richtung definiert. Darüber hinaus können Kanten und Knoten auch mit Werten versehen werden, wodurch sie einen markierten Graphen bilden. Sind diese Werte Zahlen, welche zum Beispiel Fahrzeiten in einem Straßennetz repräsentieren, wird von einem gewichteten Graphen gesprochen. Im Folgenden werden nun die für die Arbeit relevanten Begriffe, Algorithmen und Metriken aus der Graphentheorie erläutert. [3]

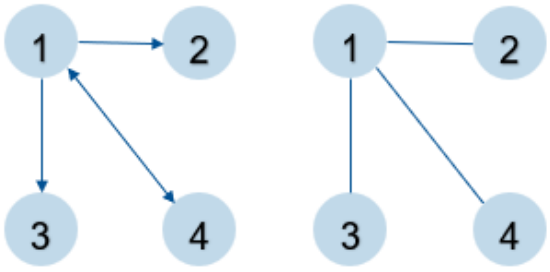


Abbildung 2.1: Beispiel für gerichtete Graphen (l.) und ungerichtete Graphen (r.)

Adjazenzmatrix

Knoten, welche durch eine Kante verbunden sind, werden als benachbart oder adjazent bezeichnet. Die sogenannte Adjazenzmatrix A ist eine Datenstruktur, mit der Graphen repräsentiert werden können. Sie hat eine Größe von $n \times n$, wobei n die Anzahl der Knoten ist. Für ungewichtete, ungerichtete Graphen wird sie durch den folgenden Ausdruck definiert: [2]

$$A_{ij} = \begin{cases} 1 & \text{falls Knoten } i \text{ zu Knoten } j \text{ adjazent ist} \\ 0 & \text{sonst} \end{cases} \tag{2.1}$$

Bei gewichteten Graphen wird von einer gewichteten Adjazenzmatrix gesprochen, bei der Kantengewichte direkt in der Adjazenzmatrix repräsentiert werden. Sie sind wie folgt definiert: [2]

$$A_{ij} = \begin{cases} k_{ij} & \text{falls Knoten } i \text{ zu Knoten } j \text{ adjazent und } k_{ij} \text{ das Gewicht ist} \\ 0 & \text{sonst} \end{cases} \tag{2.2}$$

Ist der Graph gerichtet, wird der Eintrag A_{ij} nur befüllt, falls eine Kante von Knoten i zu Knoten j existiert [2]. Beispielsweise lautet die Adjazenzmatrix des ungewichteten, gerichteten Netzwerk in Abbildung 2.2 folgendermaßen:

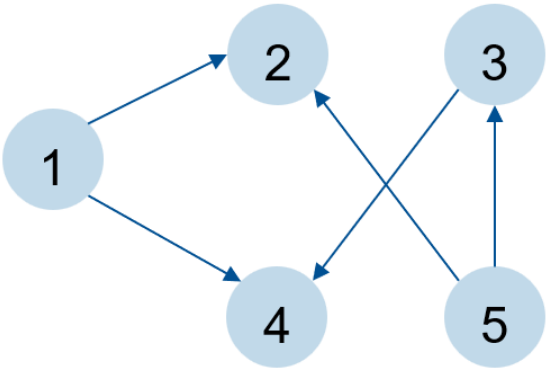


Abbildung 2.2: Beispielgraph

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.3}$$

Knotengrad

In ungerichteten Graphen beschreibt der Grad k eines Knotens die Anzahl der mit ihm verbundenen Kanten. Bei gerichteten Graphen wird außerdem zwischen dem Eingangsgrad und dem Ausgangsgrad unterschieden. Der Eingangsgrad k_i^{in} beschreibt die Anzahl der Kanten, die zu einem Knoten hinführen, und der Ausgangsgrad k_i^{out} die Anzahl der wegführenden Kanten. Der Grad eines Knotens, in einem ungerichteten Graphen, kann aus der Adjazenzmatrix A mittels der Formel 2.4 bestimmt werden, wobei k_i der Knotengrad des Knotens i, und n die Anzahl der Knoten im Netzwerk sind. [2]

$$k_i = \sum_{j=1}^{n} A_{ij} \quad (2.4) \qquad k_i^{in} = \sum_{j=1}^{n} A_{ij} \quad (2.5) \qquad k_j^{out} = \sum_{i=1}^{n} A_{ij} \quad (2.6)$$

Bei gerichteten Graphen kann für einen Knoten i, der Eingangsgrad mit Formel 2.5 und der Ausgangsgrad mit Formel 2.6 bestimmt werden [2]. Der Knoten 3 aus dem Beispielgraphen (vgl. Abbildung 2.2) hat zum Beispiel einen Eingangs- und Ausgangsgrad von jeweils 1. Mithilfe des Knotengrades kann eine Metrik über die Zentralität eines Knotens definiert werden, welche im anschließenden Abschnitt erklärt wird.

Zentralität

Die Zentralität eines Knotens ist ein häufig genutztes Maß, welches die Wichtigkeit oder den Einfluss eines Knotens im Gesamtgraph quantifiziert. Entsprechend der vielen Definitionen von Wichtigkeit in diesem Kontext gibt es viele verschiedene Arten von Zentralitätswerten. Der simpelste Zentralitätswert ist, anschließend zum vorherigen Abschnitt, der Grad eines Knotens. Die ***Gradzentralität*** misst somit lediglich die Anzahl der Nachbarn, die ein Knoten besitzt. Bei gerichteten Graphen kann, je nach Anwendungsfall, eine getrennte Betrachtung von Eingangs- und Ausgangsgrad oder die Betrachtung des Gesamtgrades, also der Summe von Eingangs- und Ausgangsgrad, nützlich sein. Trotz der Simplizität dieses Wertes, können dadurch bereits einleuchtende Beobachtungen zur Struktur eines Netzwerks getroffen werden. Im Fall eines sozialen Netzwerks kann zum Beispiel davon ausgegangen werden, dass Personen, welche viele Bekannte haben, entsprechend einflussreich sind. Es kann allerdings sein, dass der Einfluss der einzelnen Nachbarn nicht äquivalent ist. In vielen Fällen steigt der Einfluss eines Knotens, wenn dieser Verknüpfungen zu Knoten hat, welche

selbst sehr einflussreich sind. Die ***Eigenvektor-Zentralität*** und ihre Varianten bauen auf diesem Konzept auf, betrachten aber nicht nur die Anzahl der adjazenten Knoten, sondern auch die Zentralitätswerte dieser. Sie wird mittels des Eigenvektors **x** des größten Eigenwerts der Adjazenzmatrix bestimmt und kann für ungerichtete Graphen mit der Formel 2.7 und für gerichtete Graphen mit der Formel 2.8 berechnet werden. [2]

$$x_i = \kappa^{-1} \sum_{j=1}^{n} A_{ij} x_j \qquad (2.7) \qquad\qquad x_i = \kappa^{-1} \sum_{j} A_{ij} x_j \qquad (2.8)$$

Hierbei ist x_i die Eigenvektor-Zentralität des Knotens i, und die Konstante κ entspricht dem größten Eigenwert. Durch die Summe über j und dem Faktor A_{ij} wird sichergestellt, dass nur adjazente Knoten in den Term einfließen. Somit kann ein Knoten eine hohe Zentralität erlangen, indem er viele Nachbarn mit einer geringen Zentralität besitzt, oder wenige Nachbarn mit einer hohen Zentralität besitzt. Die Eigenvektor-Zentralität ist allerdings nur begrenzt für gerichtete Graphen geeignet. Aufgrund der Asymmetrie der Adjazenzmatrix bei gerichteten Graphen, wodurch eine Mehrdeutigkeit des Eigenvektors des größten Eigenwerts entsteht, muss entschieden werden, ob Nachbarn, die auf einen Knoten gerichtet sind oder Nachbarn, auf die der Knoten zeigt, in den Term einfließen. Unter der Annahme, dass Zentralität in gerichteten Graphen durch Nachbarn verliehen wird, die auf einen Knoten gerichtet sind, werden in diesem Fall nur solche Nachbarn beachtet, die auf den Knoten zeigen. Dadurch können Probleme entstehen, wenn ein Knoten beispielsweise nur ausgehende Kanten besitzt. Dieser Knoten hat dann eine Eigenwert-Zentralität von 0 und übertragen diesen Wert auch an Nachbarn, wenn diese nur eine eingehende Kante von diesem 'irrelevanten' Knoten besitzen. Dies kann dazu führen, dass auch Knoten, die einen hohen Eingangsgrad haben, einen Zentralitätswert von 0 erhalten. Die sogenannte ***Katz-Zentralität*** verhindert dies, indem jeder Knoten, unabhängig von der Zentralität der Nachbarn, eine konstante Menge Zentralität β gutgeschrieben bekommt: [2]

$$x_i = \alpha \sum_{j} A_{ij} x_j + \beta \qquad (2.9)$$

Mit der Konstante α kann in der Formel 2.9 die Balance zwischen dem Anteil der Eigenvektor-Zentralität und dem konstanten Anteil bestimmt werden. Wird für α ein Wert gewählt, welcher nahe bei κ^{-1} liegt, wird der Anteil des Eigenvektor-Terms maximal und der des

konstanten minimal. Somit können Zentralitäten bestimmt werden, welche numerisch sehr ähnlich zur regulären Eigenvektor-Zentralität sind, ohne dass Knoten eine Zentralität von 0 erhalten. Die *Katz-Zentralität* hat allerdings auch eine negative Eigenschaft, die von dem Zentralitätswert ***PageRank*** beachtet wird. Hat ein Knoten mit hoher Katz-Zentralität viele Verknüpfungen, die zu anderen Knoten zeigen, erhalten diese alle einen hohen Zentralitätswert. Allerdings kann argumentiert werden, dass sie nur einer von vielen Knoten sind, auf die gezeigt wird und die Zentralität durch das Teilen der Verknüpfung gemindert wird. Aufgrund dessen ist bei PageRank der Zentralitätsanteil der von benachbarten Knoten bezogen wird, proportional zu deren Zentralitätswert geteilt durch ihren Ausgangsgrad (vgl. Formel 2.10). Somit übertragen zentrale Knoten, die zu vielen Anderen zeigen, nur wenig Zentralität auf die jeweiligen Nachbarn. Allerdings muss beachtet werden, dass der Ausgangsgrad von Knoten ohne ausgehende Kanten künstlich auf 1 gesetzt werden muss, um eine Division durch null zu vermeiden.

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j^{out}} + \beta \tag{2.10}$$

Link Prediction

Bei Link Prediction handelt es sich um Methoden, die für die Korrektur von Fehlern in Graphen genutzt werden können. Sie werden für die Vorhersage von Kanten genutzt, die im Graphen fälschlicherweise fehlen, oder in selteneren Fall solche, die fälschlicherweise im Graph vorhanden sind. Es werden also Beziehungen gesucht, welche im realen Netzwerk, das ein Graph repräsentiert, existieren, aber im Graphen nicht vorhanden sind. Sie sind allerdings nicht in der Lage, jederzeit sämtliche fehlende Kanten vorauszusagen, sondern geben in der Regel eine Liste von Knoten-Paaren aus, die nach der Wahrscheinlichkeit, dass diese durch eine fehlende Kante verbunden sind, sortiert ist. Hierbei ist zu beachten, dass selbst die wahrscheinlichen Paare inkorrekt sein können. Die Methoden der Link Prediction basieren nicht auf bestimmten Modellen, sondern auf heuristischen Ansätzen. Zu diesen gehören unter anderem die Annahmen, dass Knoten-Paare, welche viele Nachbarn teilen oder ähnlich zueinander sind, eher in Beziehung stehen als solche mit wenigen gemeinsamen Nachbarn oder geringer Ähnlichkeit. Sie sind somit zwar nicht fähig, dem Nutzer mit Sicherheit die Lage einer fehlenden Kante zu zeigen, können aber Hinweise dazu geben, wo geschaut werden sollte. [2] Eine Methode für die Vorhersage von Verbindungen anhand der Anzahl von gemeinsamen Nachbarn ist der Adamic-Adar Index. Dieser beruht auf der Annahme, dass

gemeinsame Nachbarn eines Knotenpaars, welche selber eine große Nachbarschaft besitzen, bei der Vorhersage einer Verbindung, weniger signifikant sind als Nachbarn mit einer kleinen Nachbarschaft. Ein höherer Adamic-Adar Index deutet eine höhere Wahrscheinlichkeit einer Verbindung an. Für die Knoten x und y ist der Index durch die Summe der inversen, logarithmischen Gradzentralität der gemeinsamen Nachbarn definiert:

$$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log|\Gamma(z)|} \tag{2.11}$$

Hierbei beschreibt $\Gamma(x)$ die zu x adjazenten Knoten und $|\Gamma(z)|$ die Gradzentralität des gemeinsamen Nachbarn z. [5] [6]

Graph-Komponenten

Im Allgemeinen müssen Graphen nicht aus einer einzigen Menge von verbundenen Knoten bestehen. Sie können auch aus mehreren voneinander getrennten Komponenten aufgebaut sein. Komponenten beschreiben bei ungerichteten Graphen, Teilmengen der Knoten eines Graphen, in denen jedes Paar durch einen Pfad, verbunden werden kann, und kein weiterer Knoten hinzugefügt werden kann, ohne dieses Kriterium zu verletzen. Bei gerichteten Graphen wird hierbei zwischen schwach und stark verbundenen Komponenten unterschieden. Ein Paar von Knoten befindet sich dann in einer schwach verbundenen Komponente, wenn sie von mindestens einem Pfad verbunden werden können, wobei der Pfad auch entgegen der Richtung einer Kante verlaufen kann. Bei stark verbundenen Komponenten werden nur Pfade beachtet, die definitionsgemäß entlang der Richtung einer Kante verlaufen. Hierbei werden Knoten ohne Kanten auch als stark verbundene Komponente betrachtet. [2] Auf der Abbildung 2.3 ist zum Beispiel ein Graph abgebildet, welcher aus einer stark verbundenen Komponente (orange Knoten) und einer schwach verbundenen Komponente (blaue Knoten) besteht.

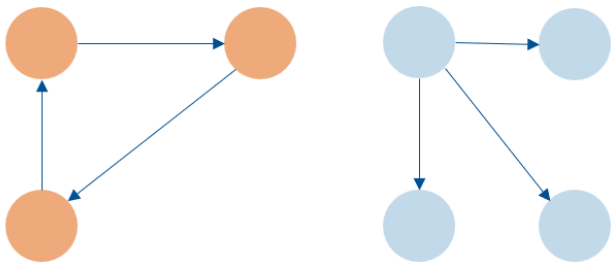


Abbildung 2.3: Beispielgraph mit 2 Komponenten

Detektion von Communities

Die Detektion von Communities, auch *Clustering* genannt, befasst sich mit der Unterteilung eines Graphen in Gruppen von Knoten, sodass innerhalb von Gruppen viele Kanten und zwischen den Gruppen wenige Kanten existieren. Dieses Verhältnis der Dichten von Kanten innerhalb zu der Dichte außerhalb wird auch Modularität genannt. Die Unterteilung des Graphen in Communities, kann dazu genutzt werden, um Strukturen und Anordnungen in einem Netzwerk aufzudecken, die den Maßstab weniger Knoten überschreiten. Community Detection kann auch genutzt werden, um sehr große Netzwerke in handhabbare Teile zu unterteilen, welche separat voneinander untersucht werden können, um die Analyse und Interpretation zu vereinfachen. [2] Eine Methode für die Detektion von Communities ist die sogenannte Louvain Methode, welche heuristische Algorithmen für die Optimierung der Modularität nutzt. Durch eine lokale Optimierung der Modularität findet sie zunächst kleine Communities. Diese werden dann zu einem Knoten zusammengefasst und der erste Schritt wird wiederholt, bis es kein Anstieg der Modularität gibt. [7] Ein Beispiel zur Aufdeckung von Strukturen wird an der Abbildung 2.4 deutlich. Auf dem Graphen sind die Kollaborationen zwischen Wissenschaftler eines Lehrstuhls zu erkennen. Die Knoten repräsentieren die einzelnen Wissenschaftler und die Kanten zwischen den Knoten deuten an, dass die verbundenen Wissenschaftler für ein Paper kooperiert haben. Auf der Abbildung sind eine Anzahl von dicht verknüpften Communities erkennbar, welche für Betrachter, die mit der Organisation von Lehrstühlen vertraut sind, als Forschungsgruppen interpretierbar sind. Betrachter können aus der Betrachtung der geclusterten Struktur des Graphen, auch wenn sie nicht mit der Organisation von Lehrstühlen vertraut sind, ableiten, dass innerhalb eines Lehrstuhls einzelne Gruppen existieren. [2]

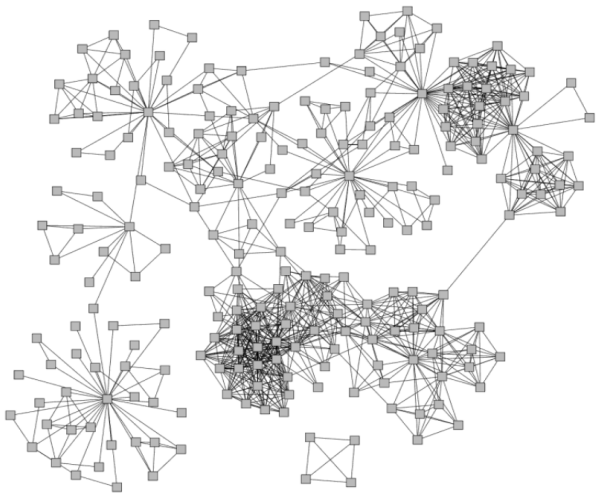


Abbildung 2.4: Netzwerk von Mitautorenschaften in einem Lehrstuhl, [2]

Breitensuche

Die Breitensuche, auf Englisch *Breadth First Search* (BFS), ist ein fundamentaler Algorithmus für das Durchlaufen von Graphen. BFS erkundet, aus gehend von einem Startknoten, zunächst alle adjazenten Knoten und markiert diese mit ihrem Abstand zum Startknoten, bevor sie zu Knoten der nächsten Ebene übergeht. Dies geschieht so lange, bis alle erreichbaren Knoten erkundet wurden. Sie wird unter anderen bei der Entdeckung von kürzesten Pfaden zwischen Knoten und von verbundenen Komponenten, benutzt. [8] Die Knoten der ersten Ebene, also jene, die adjazent zum Startknoten sind, haben hierbei einen Abstand von 1 zum Startknoten. Die Knoten der zweiten Ebene, also die Nachbarn der Nachbarn des Startknotens haben einen Abstand von 2, und so weiter. [2]

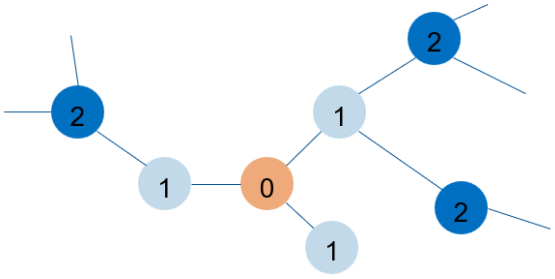


Abbildung 2.5: Breitensuche, nachgestellte Grafik, basierend auf [2]

Zyklen

In gerichteten Graphen beschreiben Zyklen, geschlossene Schleifen von Kanten. Also solche Pfade, welche den Kantenrichtungen folgen und deren Startknoten dem Endknoten entspricht (vgl. Abbildung 2.6). Enthält ein Graph mindestens einen Zyklus, wird dieser zyklisch genannt. Ein Graph ohne einen Zyklus wird als azyklisch bezeichnet. Unter der Annahme, dass ein azyklischer Graph mindestens einen Knoten besitzt, welcher nur eingehende Kanten besitzt, kann ein simpler Algorithmus zur Bestimmung der Azyklizität definiert werden. Die Annahme basiert darauf, dass in einem Graphen, der keinen Knoten mit Ausgangsgrad 0 enthält, ein unendlicher Pfad konstruiert werden kann, der zwangsläufig wieder durch den Startknoten verläuft. Dieser lautet folgenderweise: [9]

1. Suche nach Knoten mit einem Ausgangsgrad von 0.
2. Gibt es keine, ist der Graph zyklisch. Ansonsten wird der gefundene Knoten und zu ihm gerichtete Kanten aus dem Graph entfernt.

3. Enthält der Graph keine Knoten mehr, ist der Ausgangsgraph azyklisch. Ansonsten wiederhole ab Schritt 1.

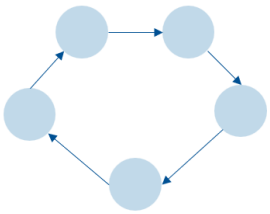


Abbildung 2.6: Beispiel für einen zyklischen Graphen

Knowledge-Graphen

Knowledge-Graphen, auch semantische Netzwerke genannt, repräsentieren Netzwerke von Objekten der realen Welt und illustrieren ihre Beziehungen zueinander. Sie bilden markierte Graphen (mit Werten versehene Knoten und Kanten) und können dementsprechend auch als Graph visualisiert werden. Sie bestehen typischerweise aus Datensätzen aus unterschiedlichen Quellen, welche sich oft in ihrer Datenstruktur unterscheiden. Aufgrund dessen werden Schemen für das Rahmenwerk des Knowledge-Graphen, Identitäten für die Klassifizierung der Knoten und Kontext genutzt, um Struktur in diese heterogenen Daten zu bringen. Beispiele für bekannte Knowledge Graphen sind Wikidata[1] für Daten der Wikipedia oder der Google Knowledge Graph [10], dessen Informationen zu Personen, Orte oder Dingen bei passenden Suchanfragen in einem Panel angezeigt werden. [11]

Ein besonderer Anwendungsfall von Knowledge-Graphen sind Cities Knowledge Graphen (CKG). Das Konzept der CKG, wurde von dem Future Cities Laboratory der ETH Zürich, speziell für die Verbesserung der Planung und des Designs von Städten entwickelt. Sie bilden eine digitale Plattform für den Austausch und die Zusammenführung von Daten über eine Stadt. Ziel der CKG ist es, Dynamiken einer Stadt aufzudecken, um die Ansprüche verschiedener Domänen aufeinander abstimmen zu können, bestehende Daten- und Wissenssysteme der stetig wachsenden Menge an Daten anzupassen und die Stadtplanungsprozesse verschiedener Domänen, oft mit verschiedenen Ansätzen der Datensammlung, zu integrieren. [12]

[1] https://www.wikidata.org/

2.2 Metadaten

Die Professoren Marcia Lei Zeng und Jian Qin definieren Metadaten in Ihrem Lehrbuch "*Metadata, 3rd Edition*", folgenderweise [13]:

> *Strukturierte, kodierte Daten, die Merkmale von informationstragenden Einheiten beschreiben, um die Identifizierung, Entdeckung, Bewertung, Verwaltung und Bewahrung der beschriebenen Informationen zu unterstützen.* [Übers. d. Verf.]

Metadaten sind simpel ausgedrückt Daten über Daten. [14] Sie werden für die Verwaltung und Organisation von informationstragenden Einheiten (d.h Dinge) genutzt. [13] Organisationen des Kulturerbes, wie Museen oder Archive nutzen sie, um ihre Bestände zu katalogisieren. Viele Anwendungen aus dem Alltag beruhen auf der Nutzung von Metadaten. Webseiten wie Wikipedia und YouTube nutzen die Metadaten ihrer Inhalte, um den Nutzern relevante Ergebnisse zu ihrer Suche bieten zu können. [15] Anhand von Metadaten können Ressourcen nach bestimmten Kriterien organisiert und gefunden werden, der Austausch und die Interoperabilität von Ressourcen sichergestellt werden, sowie eine digitale Identifikation und Beschreibung für eine Archivierung bereitgestellt werden. [13] Metadaten können in verschiedene Typen kategorisiert werden, welche für unterschiedliche Anwendungsfälle in Informationssystemen genutzt werden können. Zum Beispiel bieten beschreibende Metadaten Informationen, die dazu dienen, Ressourcen zu finden und zu verstehen. Sie können Attribute wie Titel oder Thema einer Ressource umfassen. Administrative Metadaten umfassen Informationen, welche für die Verwaltung notwendig sind oder sich auf die Erstellung einer Ressource beziehen. Zu ihnen zählen die unter andern, Informationen zu Datentyp, Lizenzrechte oder Erstellungsdatum. Zu den strukturellen Metadaten gehören solche Informationen, die Teile einer Ressource in Beziehung zu anderen Teilen setzt, wie die Sequenz von Seiten. [15]

Metadaten sind allerdings nur nützlich, wenn diese für die nutzenden Personen und Anwendungen verständlich sind. Um eine Interoperabilität zu gewährleisten, werden Standards genutzt, die Vorgaben zur Implementation der Metadaten definieren. [13] DCAT2 (Data Catalog Vocabulary - Version 2) des World Wide Web Consortium (W3C) ist solch ein Standard, welcher weltweit verwendet wird und die Interoperabilität zwischen Datenkatalogen im Internet ermöglicht. Mittels DCAT können Datensätze basierend auf einem standardisierten Modell und Vokabular in einem Katalog beschrieben werden, wodurch der Austausch von Metadaten zwischen unterschiedlichen Akteuren erleichtert wird. [16] Dies ermöglicht auch einen dezentralen Ansatz bei der Veröffentlichung von Datenkatalogen und eine vereinigte

Suche nach Datensätzen in verschiedenen Katalogen mit demselben Abfragemechanismus. [17] Das im Rahmen des SDDI-Konzepts entwickelte Modell für Daten im Metadatenkatalog von Urbanen Digitalen Zwillingen (vgl. Kapitel 2.4), kann auf diesen Standard abgebildet werden, wodurch der Katalog DCAT2-Daten bereitstellen und darauf zugreifen kann.[16]

2.3 Urbane Digitale Zwillinge und SDDI

Die Digitalisierung der Verwaltung ermöglicht viele neue Möglichkeiten zur verbesserten Planung und Vorbereitung auf zukünftige Herausforderungen. Das Konzept des Urbanen Digitalen Zwillings (UDZ), ist einer dieser Möglichkeiten. Ein UDZ ist die digitale Repräsentanz einer Stadt, eines Quartiers oder einer ganzen Region und bildet eine Schnittstelle, welches die vielfältigen Daten, welche unter anderem aus Modellen, Simulationen, Algorithmen oder Services bestehen, organisiert und nutzbar macht. Sie sind zweckgebunden und sammeln nur digitale Ressourcen, welche alle zweckgemäß erforderlichen Aspekte der realen Welt repräsentieren. Durch den kontinuierlichen Abgleich eines UDZ mit der realen Welt, ist es möglich aus Analysen und Beobachtungen des UDZ Rückschlüsse auf den aktuellen Zustand des repräsentierten Gebiets zu ziehen. Hierdurch können Gebietskörperschaften in ihrem Umgang mit Herausforderungen, zum Beispiel in den Domänen Energie, Mobilität oder Klimaschutz, bei der Entscheidungsfindung unterstützt werden. [1] Für die Realisierung von Urbanen Digitalen Zwillingen wurde am Lehrstuhl für Geoinformatik der Technischen Universität München das Smart District Data Infrastructure Konzept (SDDI)[2] entwickelt. SDDI ist ein Rahmenwerk für die Umsetzung der Dateninfrastruktur eines UDZ mittels eines offenen und standardisierten Datenaustausches zwischen den diversen Teilhabern. Hauptmerkmal dieses Rahmens ist unter anderen das **Virtuelle Distriktmodell**. In dem auf offenen, standardisierten Schnittstellen basierenden Virtuellen Distriktmodell werden alle wesentlichen Objekte eines Gebiets durch ein entsprechendes digitales Objekt, welches es in Thematik, Lage und Ausdehnung beschreibt, repräsentiert. Dies ermöglicht nicht nur die räumliche Visualisierung, sondern auch die Verknüpfung und Verortung von Informationen aus verschiedenen Quellen über ein Objekt des Virtuellen Distriktmodells. Das SDDI-Rahmenwerk ist modular aufgebaut und umfasst, mit dem Virtuellen Distriktmodell, Sieben in Austausch stehende Komponenten (vgl. Abbildung 2.7). [18]

In diesem Rahmen umfasst die Komponente der **Akteure**, alle Personen oder Organisationen, die etwas zum UDZ beitragen können oder Interesse daran haben. Die Beispiele dafür

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

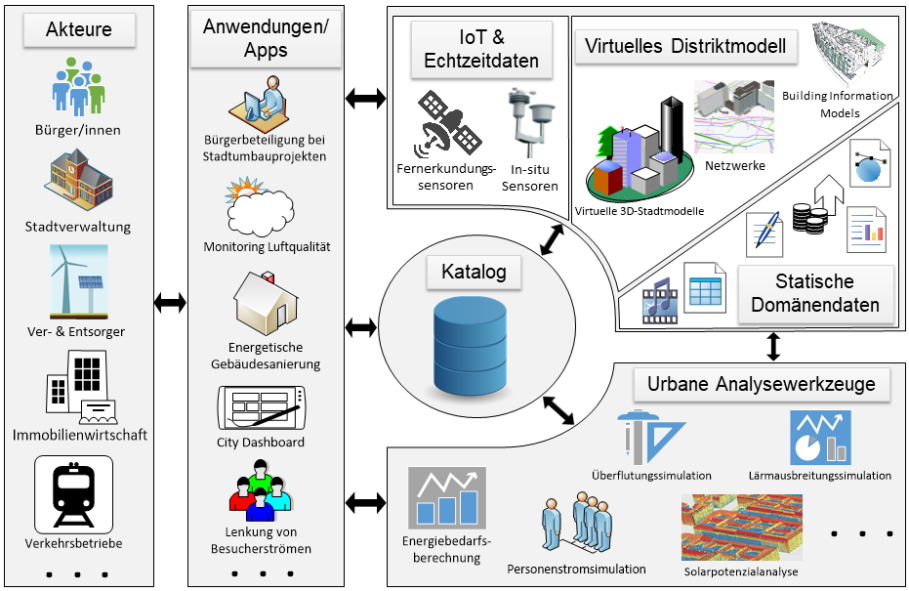


Abbildung 2.7: Schema des SDDI-Rahmenwerks, [1]

sind der Bürger und die Kommunen, aber auch andere Akteure wie öffentliche Verkehrsbetriebe, Energieversorger oder andere, die aktiv am Prozess beteiligt sind. Die Komponente **Internet of Things (IoT) und Echtzeitdaten** umfasst unter anderem Daten von Sensoren oder Software, welche den Zustand von physischen Objekten in Echtzeit beschreiben können und ihre Daten mit anderen Systemen oder Geräten austauschen. Ein solcher Sensor könnte zum Beispiel eine Verkehrskamera sein, welche Daten zur aktuellen Verkehrslage liefern kann. Im Gegensatz dazu sind **Statische Domänedaten** nicht dynamisch und werden lediglich anlassbezogen oder zyklisch aktualisiert. Sie umfassen fachgebietsspezifische Daten (z.B. Einwohnermeldedaten, Bevölkerungsstatistik), welche in der Regel als Datei oder über Webservices/APIs verfügbar sind. Bei **Urbanen Analysewerkzeugen** handelt es sich um Softwarekomponenten, welche Analysen und Simulationen durchführen können. Die **Anwendungen und Applikationen** erlauben den Akteuren einen Zugriff auf die verfügbaren Daten eines entsprechenden Anwendungsfalls (Use Case). Diese nutzen, je nach Use Case, eine Synthese von Daten aus dem Virtuellen Distriktmodell, Statischen Domänedaten, IoT-Sensoren und Urbanen Analysewerkzeugen. Der **SDDI-Katalog** steht als wichtigste Komponente im Mittelpunkt der bisher genannten und verbindet diese miteinander. [18] Im folgenden Kapitel wird dieser genauer betrachtet.

2.4 SDDI-Katalog

Im Rahmen eines Urbanen Digitalen Zwillings (UDZ) können digitale Ressourcen von unterschiedlichen Teilhabern in verschiedenen Datenformaten bereitgestellt werden. Um den anderen Teilhabern die Nutzung der Ressourcen für ihre Anwendungen zu ermöglichen, muss diesen eine Schnittstelle bereitgestellt werden, um Herkunft, Datenformat, sowie die Beziehungen der Ressourcen untereinander, recherchieren zu können. Der SDDI-Katalog wurde basierend auf dem Spatial Data Infrastructures Konzept (SDI) entwickelt. Diese SDDI-Kernkomponente dient als zentraler Speicherort für die standardisierten Metadaten der verteilten Ressourcen. Der Katalog ist auf die Verwaltung von UDZs spezialisiert und kann über Geodaten hinaus, verteilte heterogene Daten, sowie Verknüpfungen zwischen ihnen verwalten. [19] Der Katalog ist dafür vorgesehen, alle Nutzer und digitale Ressourcen mittels ihrer Metadaten zu registrieren. Hierfür werden die Referenzen zu Speicherort und Nutzungsbedingungen von Ressourcen, zusammen mit weiteren beschreibenden Informationen, innerhalb eines Katalogeintrags gespeichert, wobei es unter Größenbeschränkungen auch möglich ist, die Daten der digitalen Ressource in den Katalog hochzuladen. Katalogeinträge können hierbei beliebig viele Ressourcen umfassen. Der Katalog fungiert somit als zentraler Knotenpunkt, der einen umfassenden Überblick über alle digitalen Ressourcen bietet und diese mit ihren jeweiligen Anwendungsfällen verbindet. [16]

Zur Beschreibung der digitalen Ressourcen und ihrer Verknüpfungen muss das Metadaten-Schema des SDDI-Katalogs entsprechend aussagekräftig sein. Im Folgenden wird der Aufbau und die einzelnen Komponenten dieses Schemas (vgl. Abbildung 2.8) erläutert. Im Mittelpunkt des Schemas liegt als wichtigste Klasse *InformationResource* dessen Instanzen die Katalogeinträge repräsentieren. Um die Suche nach bestimmten Ressourcen im Katalog zu erleichtern, wird jeder Eintrag im Katalog in einer der Neun, als Hauptkategorien agierende, Unterklassen von *InformationResource* (*Datensatz und Dokumente, Online-Dienst, Online-Anwendung, Projekt, Software, Methode, Gerät / Ding, Geo-Objekt* und *Digitaler Zwilling*) eingeordnet und kann durch weitere Themen zusätzlich kategorisiert werden. In folgender Tabelle (vgl. Tabelle 2.1) wird aufgeschlüsselt, wie ein Katalogeintrag, in exakt einer dieser Unterklassen klassifiziert wird: [18] [19]

Tabelle 2.1: Übersicht über die Unterklassen von *InformationRecsource*

Unterklasse	**Klassifizierungskriterium**
Datensatz und Dokumente	Dokumente und Informationen zu beliebigen Informations-Ressourcen der Modellregion.
Online-Dienst	Daten, welche über eine API abrufbar sind. (keine Benutzeroberfläche)
Online-Anwendung	Daten, welche über eine API abrufbar sind. (mit Benutzeroberfläche)
Projekt	Geplante, begonnene oder abgeschlossene Aktivitäten, die zur Erreichung eines bestimmten Ziels oder einer bestimmten Aufgabe führen.
Software	Spezifische Werkzeuge oder Systeme
Methode	Zielorientierte Funktionen und Algorithmen, zur Lösung ausgewählter Aufgaben. (z.B. für eine Analyse der Beschattung an Gebäuden im Laufe des Tages.)
Gerät / Ding	Physische Geräte (z.B. Sensoren)
Geo-Objekt	Objekte mit eindeutig definierter, geographischen Komponente. (z.B. Gebäude, Ampel)
Digitaler Zwilling	Sammlung von Ressourcen, in Rahmen eines Themas oder einer geographischen Region.

Diese Unterklassen werden darüber hinaus über die abstrakten Klassen *RealWorldInstance*, *Prototype* und *PhysicalThing* semantisch gruppiert. Ebenso kann die räumliche oder zeitliche Ausdehnung einer Ressource durch die Klassen *Location* und *PeriodOfTime* repräsentiert werden, wodurch eine Suche nach Ressourcen auch über eine Eingrenzung dieser möglich ist. Die Instanzen der Klassen *User* und *Organization* repräsentieren die einzelnen Stakeholder des Katalogs, und ermöglichen eine verteilte Verwaltung des Katalogs. Die Organisationen in einem Katalog verwalten die Zugriffsrechte auf ihre eigenen Katalogeinträge. Darüber hinaus kann der Zugriff auf die Datensätze in einem Katalogeintrag, welche in diesem Datenmodell als Instanzen der *Distribution*-Klasse realisiert sind, einzeln reguliert werden. Mittels der Klasse *Relationship* werden die semantischen Verknüpfungen zwischen den *InformationResource* Instanzen definiert, wobei kataloginterne Verknüpfungen (*InternalLink*) und solche zu externen Ressourcen (*ExternalLink*) möglich sind. [19][16]

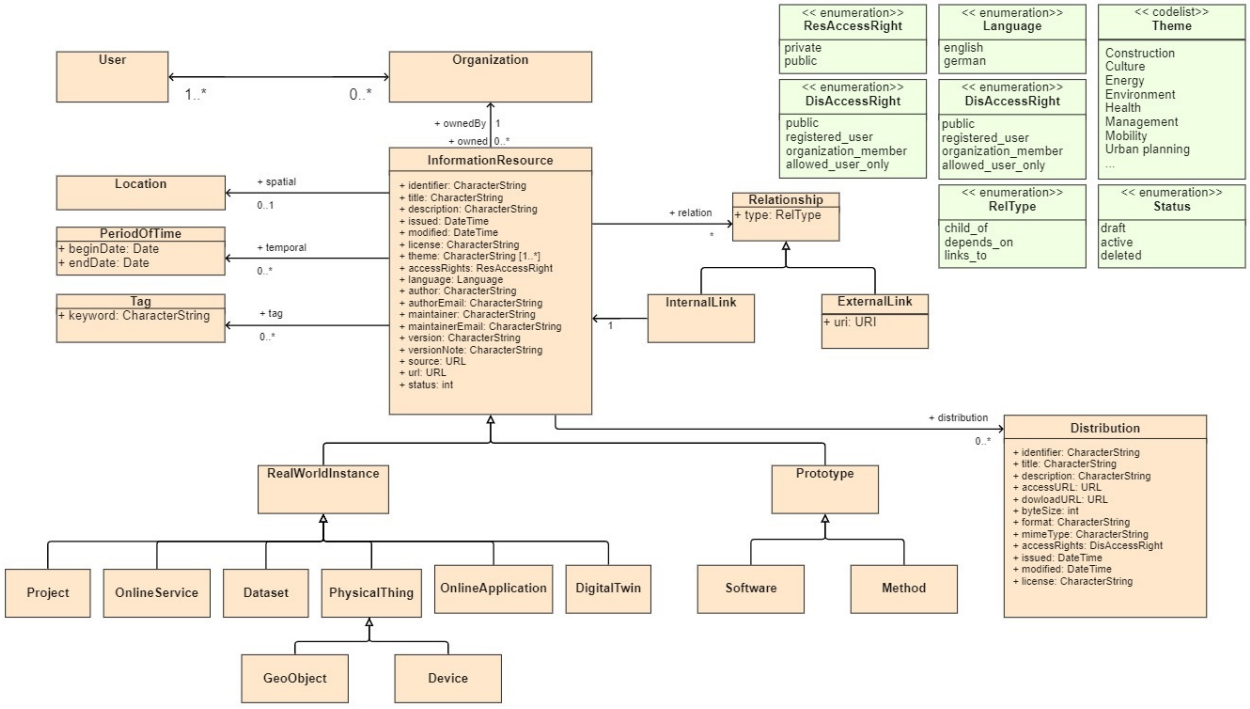


Abbildung 2.8: UML Klassen-Diagramm des Metadatenmodells, [19]

Im SDDI-Katalog werden Verknüpfungen zwischen Ressourcen, entsprechend der Art der Beziehung zwischen den Ressourcen, in eine von drei vordefinierten semantischen Beziehungstypen kategorisiert. *links_to* für Verbindungen zwischen eigenständigen Ressourcen, *depends_on* für Ressourcen, die nur im Kontext einer anderen Ressource interpretiert werden können, und *child_of* bei Ressourcen, die ein Teil von einer anderen darstellen. Dieses Metadaten-Schema ermöglicht somit die verteilte Verwaltung von Katalogeinträgen innerhalb und zwischen verschiedenen Organisationen, sowie die Verknüpfung von Einträgen,

innerhalb einer Organisation sowie mit Ressourcen einer anderen Organisation. [19] [16] Die einzelnen Einträge und ihre Verknüpfungen im SDDI-Katalog bilden die Grundlage für einen Knowledge-Graphen, wie im Kapitel 2.1 erwähnt. [16]

3 Analyse und Visualisierung

3.1 Bildung der Graphstruktur

Bevor in den folgenden Kapiteln die Anforderungen an die Visualisierung und Use Cases der graph-basierten Analyse eines SDDI-Metadatenkatalogs diskutiert werden, muss zunächst definiert werden, wie aus den Metadaten eine analysierbare und visualisierbare Abbildung der Graphstruktur des Katalogs gebildet werden kann. Da SDDI-Kataloge mithilfe der Katalogsoftware CKAN implementiert sind (vgl. Kapitel 4.1.1), können die Metadaten, der in ihm eingetragenen Ressourcen, mit bestimmten Anfragen an die API ausgelesen werden (vgl. Kapitel 4.2.1). Diese werden, einzeln für jeden Eintrag, in dem Datenaustauschformat JSON ausgegeben und beinhalten eine detaillierte Beschreibung der Eigenschaften der Ressource, wie bereits im Kapitel 2.4 beschrieben wurde. In der Abbildung des Katalogs werden die Metadaten einer Ressource durch Knoten repräsentiert. Die Beziehung einer Ressource zu anderen Ressourcen im Katalog ist in den Metadaten unter den Parametern *relationships_as_subject* und *relationships_as_object* beschrieben. Anhand dieser können die gerichteten Kanten zwischen den Ressourcen im Graphen gesetzt werden (vgl. Abbildung 3.1). Dies resultiert, sofern die Metadaten vollständig und korrekt sind, in einem gerichteten Graphen, welcher aus einer zusammenhängenden Komponente besteht. Weshalb der Graph zusammenhängend sein muss, wird in dem Kapitel 3.3.1 erläutert.

Abbildung 3.1: Ablauf der Abbildung eines SDDI-Katalogs

3.2 Anforderung an die Visualisierung

Ein Urbaner Digitaler Zwilling kann aus einer Vielzahl von Informationsressourcen bestehen. Für einen menschlichen Nutzer kann es, aufgrund der hohen Informationsdichte, sehr schnell kompliziert werden, aus einzelnen SDDI-Katalogeinträgen Rückschlüsse auf Strukturen und Beziehungen der Ressourcen im Netzwerk zu ziehen. Aus diesem Grund werden Methoden der Graph-Visualisierung benötigt, um die Nutzer bei dem Verständnis und der Navigation durch die Daten eines UDZs, aber auch von Graphen allgemein, zu unterstützen. Dieser Abschnitt wird sich mit den allgemeinen Anforderungen an die Visualisierung eines Graphens befassen und Ansätze für deren konkrete Umsetzung bei der Visualisierung der Graphstruktur von Metadaten eines SDDI-Katalogs bieten.

Layouts

Vor allem bei der Visualisierung von sehr großen Graphen müssen Einschränkungen bei der Repräsentation der Daten beachtet werden. Die Visualisierung sollte im Rahmen des beschränkten Bereiches eines Bildschirms und der kognitiven Leistung eines Menschen, ein intuitives und verständliches Layout bereitstellen können, welches das Verständnis über die Daten erhöhen kann. Für die Visualisierung von Graphen gibt es verschiedene Ansätze. Der triviale Ansatz hierbei sind die sogenannten *Node-Link Layouts*, welche Verknüpfungen durch Linien zwischen den Knoten darstellen. Ein weiterer Ansatz ist es, die Beziehungen implizit darzustellen, wie es bei raumfüllenden Techniken der Fall ist. [20] Diese werden in dieser Arbeit allerdings nicht weiter betrachtet. Im Rahmen von *Node-Link Layouts* sollte das Layout bei der Platzierung der Knoten und Kanten einige Kriterien beachten, um die Lesbarkeit des Graphen zu verbessern. Zu Ihnen gehören unter Anderen: [20] [21]

- Die Kantenlängen sollten möglichst kurz sein, um Verknüpfungen schneller erkennen zu können.
- Das Kreuzen von Kanten sollte minimiert werden, da Kreuzungen das Verfolgen von Verbindungen erschweren.
- Der Winkel zwischen Kanten sollte maximiert werden, damit diese an Knoten besser unterscheidbar sind.
- Die Krümmung von Kanten sollte minimiert werden, da ein Betrachter gerade Kanten besser folgen kann.

Dabei ist es zu erwähnen, dass es sehr schwierig sein kann, mehrere Kriterien zu kombinieren, wenn diese untereinander konkurrieren oder nur ineffizient implementiert werden können. Deshalb ist bei der Wahl des Layouts eine anwendungsbezogene Abwägung über die Priorisierung der Kriterien nötig.[20] [21] An den Visualisierungen desselben Graphens, auf den Abbildungen 3.2 und 3.3, wird deutlich, wie wichtig die Beachtung dieser Kriterien für die Lesbarkeit des Graphen ist.

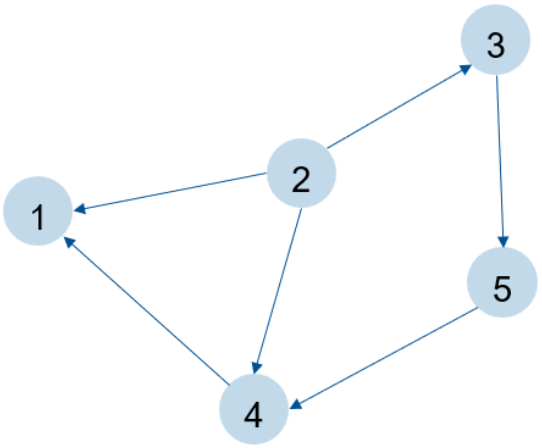


Abbildung 3.2: Graphvisualisierung unter Beachtung der Kriterien

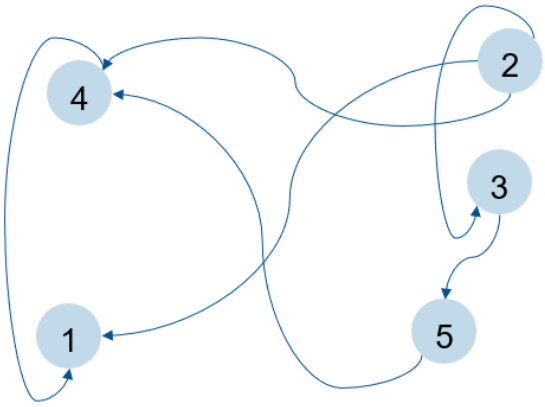


Abbildung 3.3: Graphvisualisierung ohne Beachtung der Kriterien

Unter Beachtung dieser Kriterien bieten sich für die Visualisierung eines SDDI-Katalogs sogenannte *force-directed* Layouts an. Diese betrachten Graphen als physikalische Systeme, in denen Knoten als geladene Teilchen betrachtet werden, die durch Federn verbunden sind. Dadurch besitzt jeder Knoten zum einen anziehende Kräfte zu adjazenten Knoten und abstoßende Kräfte zu allen Knoten des Graphen. *force-directed* Layouts (vgl. Abbildung 3.4) ordnen die Knoten so an, sodass die totale Energie im Graphen minimiert wird. [20] Dadurch liefern sie, im Hinblick auf die Kriterien der gleichmäßigen Verteilung von Knoten und der Minimierung der Kantenlänge, befriedigende Ergebnisse. Ein weiterer nennenswerter Aspekt von *force-directed* Layouts ist, dass diese genutzt werden können, um die Struktur eines geclusterterten Graphen (vgl. Kapitel 2.1) darzustellen, wie sie bereits auf Abbildung 2.4 zu erkennen war. [21] Für die Darstellung von Ressourcen eines Projektes in einer Hierachie bietet sich das sogenannte *breadthfirst* Layout an. Es ordnet die Ressourcen, basierent auf einer Breitensuche (vgl. Kapitel 2.1) ab einem benutzerdefinierten Startknoten, abhängig von ihrem Abstand zum Startknoten, in verschiedene Ebenen ein und stellt diese in einer Top-Down Ansicht dar (vgl. Abbildung 3.5). [22]

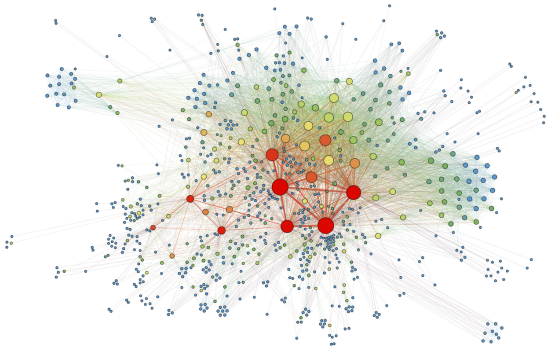


Abbildung 3.4: Visualisierung eines Graphen mit einem force-directed Layout, [23]

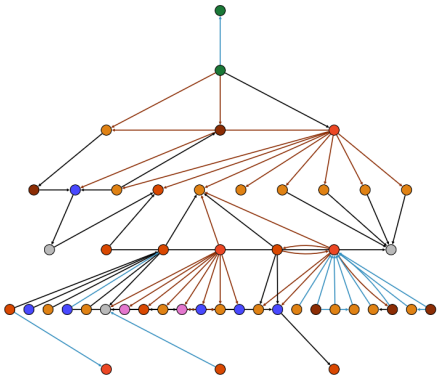


Abbildung 3.5: Visualisierung eines Graphen mit einem breadth-first Layout

Über die strukturellen Eigenschaften hinaus sollte ein Layout auch die Semantik des Graphen berücksichtigen. Zum Beispiel wäre es für den Betrachter eines Graphen, das ein ÖPNV-Netz repräsentiert, sehr verwirrend, wenn der Knoten einer Station an einer zufälligen Position dargestellt wird, und nicht an einer, die der Position der Station auf einer Stadtkarte entspricht.[21] Ebenso wäre es bei der Visualisierung des SDDI-Metadatenkatalog verwirrend, wenn voneinander unabhängige Ressourcen gruppiert dargestellt werden würden. Dies kann durch ein *force-directed* Layout verhindert werden.

Beschriftung und Stilisierung

Es ist auch zu erwähnen, dass eine zugrundeliegende Klassifizierung der Graph-Elemente in der Visualisierung dargestellt werden kann. Hierzu kann die Darstellung der Knoten und Kanten, abhängig von ihrer Klasse, zum Beispiel in ihrer Farbe, Form oder Größe variiert werden. Die umfangreichen Metadaten der Ressourcen in einem SDDI-Katalog ermöglichen es, diese ebenso umfangreich zu klassifizieren. Dabei ist der triviale Ansatz, Knoten, die Ressourcen der gleichen Hauptkategorie (vgl. Kapitel 2.4) und Verknüpfungen desselben Beziehungstyps in derselben Farbe darzustellen. Aber auch eine Färbung nach Eigenschaften wie der verwaltenden Organisation oder den Zugangsberechtigungen einer Ressource können genutzt werden, um dem Nutzer anhand der Visualisierung weiterführende Einblicke in die Struktur eines SDDI-Katalogs zu bieten.

Für die Lesbarkeit der Visualisierung, muss auch darauf geachtet werden, dass die Beschriftungen eines Graphens verständlich sind. In diesem Kontext sind folgende 3 Aspekte entscheidend [21]:

- Die Beschriftungen sollen in einer lesbaren Größe dargestellt werden.
- Jede Beschriftungen muss eindeutig einer Graphen-Komponente (Knoten/Kante) zugeordnet werden können.
- Die Beschriftungen sollten sich nicht mit anderen Beschriftungen oder Elementen überlappen.

Eine konkrete Umsetzung der Beschriftung eines SDDI-Katalogs wird im Kapitel 4.2.2 präsentiert.

Interaktion

Neben der Repräsentation der Daten ist auch die Interaktion eine Komponente der Visualisierung von Informationen. Sie ermöglicht einen Dialog zwischen dem Betrachter und der Visualisierung. Ohne Interaktion wird ihre Aussagekraft und die Fähigkeit Analysen an ihr durchzuführen für einen Nutzer, mit wachsendem Datenbestand, immer weiter eingeschränkt. Durch Interaktion können die Grenzen der Repräsentation überwunden werden und der Nutzer bei der Wahrnehmung der Visualisierung unterstützt werden.[24] Yi et al. (2007) unterscheiden hierbei sieben Kategorien von Interaktion bei der Visualisierung von Informationen. Im Folgenden werden diese Rahmen der Graph-Visualiserung, kurz erläutert: [24][20]

- *Selektion*: Ermöglicht es dem Nutzer, interessante Elemente visuell zu markieren.
- *Abstraktion/Elaboration*: Ändern der Detailstufe der Repräsentation.
- *Rekonfiguration*: Erlaubt es dem Nutzer, das Layout des Graphens zu ändern.
- *Kodierung*: Anpassung der visuellen Repräsentation der Elemente (z.B. Farbe, Größe oder Form).
- *Erkundung*: Änderung des Blickpunktes des Graphen. Hierzu gehören Zoom und Schwenken des Graphs.
- *Filterung*: Entfernen von irrelevanten Knoten, um relevante in einer angemesseneren Art darzustellen.
- *Verbindung*: Hervorhebung der Verbindungen des fokussierten Knotens und weiteren relevanten.

Möglichkeiten für die Umsetzung der Interaktion bei der Visualisierung eines SDDI-Katalogs werden im Kapitel 4.2.2 präsentiert.

Zusammenfassend lässt sich also sagen, dass bei der Visualisierung von Graphen einige Aspekte bezüglich Layouts, Beschriftungen und Semantik sowie der Interaktivität beachtet werden müssen, um eine intuitive und effektive Darstellung zu generieren, die den Nutzer bei dem Umgang mit den Daten unterstützt.

3.3 Use Cases der Qualitätssicherung

Es ist für alle Stakeholder des Urbaner Digitaler Zwilling von entscheidender Bedeutung, dass die Einträge des Katalogs korrekt und vollständig sind. Die im Folgenden betrachteten, Use-Cases der Qualitätssicherung, richten sich primär an die Bearbeiter eines SDDI-Katalogs. Durch die Visualisierung und Analyse der Graphstruktur des Katalogs, sollen sie bei der Sicherung der Datenqualität und bei der Integration neuer Daten unterstützt werden.

3.3.1 Erkennung isolierter Ressourcen

Das Ziel dieses Use Cases ist die Identifizierung von digitalen Ressourcen, die keine Verknüpfungen zu anderen Ressourcen besitzen oder Komponenten bilden, die vom restlichen zusammenhängenden Graphen isoliert sind. Ein UDZ ist, wie in Kapitel 2.3 beschrieben, eine Sammlung von digitalen Ressourcen, die alle zweckgemäß erforderlichen Aspekte der realen Welt repräsentieren. Die digitalen Ressourcen, die in einem SDDI-Katalog registriert sind, sind ein Teil des Urbanen Digitalen Zwillings und müssen deshalb im Graphen entweder direkt oder über andere Datensätze mit diesem verknüpft sein. Ressourcen, die dies nicht erfüllen, müssen im Sinne der Datenqualität des Katalogs entsprechend gehandhabt werden. Die graphbasiete Analyse des Katalogs ermöglicht es, einzelne isolierte Ressourcen mittels simpler Methoden zu identifizieren. Der gerichtete Graph muss lediglich auf Knoten mit Ein- und Ausgangsgrad (vgl. Kapitel 2.1) von 0 untersucht werden (vgl. Abbildung 3.6, rote Knoten). Für die Detektion von isolierten Komponenten, kann überprüft werden, ob der Graph, abgesehen von den einzelnen bereits entdeckten isolierten Ressourcen, aus einer einzigen Komponente besteht. Dies kann mittels einer Breitensuche (vgl. Kapitel 2.1) durchgeführt werden. Ist die Anzahl der von einem beliebigen Startknoten erreichbaren Knoten geringer als die Gesamtanzahl von Knoten im Graph, besteht der Graph aus mehreren Komponenten. Die Bearbeiter des Katalogs können diese isolierten Ressourcen dann, den

Umständen entsprechend bearbeiteten. Es können fehlende Verknüpfungen ergänzt werden oder die Ressource aus dem Katalog entfernt werden, falls diese fälschlicherweise registriert wurde.

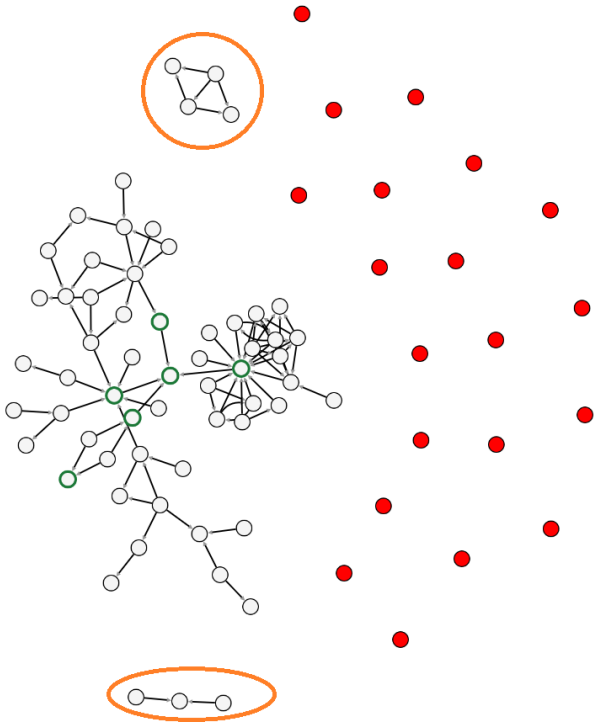


Abbildung 3.6: Beispielgraph mit isolierten Knoten und Komponenten, eigene Darstellung

3.3.2 Erkennung von Zyklen

Mithilfe einer graph-basierten Analyse des SDDI-Katalogs, sollen in Rahmen dieses Use Cases potenziell fehlerhaft angelegte Verknüpfungen zwischen Ressourcen erkennt werden. Zyklen im Graphen können auf Verbindungen der falschen Art oder Richtung hinweisen. Diese sollten im Sinne der Datenqualität unbedingt erkannt werden. Die Bearbeiter des Katalogs können diese dann auf ihre Korrektheit überprüfen und gegebenenfalls ausbessern. Die Existenz von Zyklen kann mittels der in Kapitel 2.1 erläuterten Methode detektiert werden.

Die Erkennung von Zyklen ist dann besonders wichtig, wenn alle Beziehungstypen innerhalb des Zyklus auf Abhängigkeiten deuten, also vom Typ *depends_on* sind. Dieser Spezialfall deutet an, dass alle Ressourcen, welche Teil des Zyklus sind, von sich selber abhängen. Dies ist unbedingt zu vermeiden und deutet an, dass der Katalog definitiv fehlerhafte Verknüpfungen enthält.

3.3.3 Hervorsage möglicher Verknüpfungen

Das Ziel dieses Anwendungsfalls ist es, Beziehungen zwischen Ressourcen eines Urbanen Digitalen Zwilling hervorzusagen, welche nicht im Katalog vermerkt sind (vgl. Abbildung 3.7). Aufgrund der begrenzten Möglichkeiten simpler Link-Prediction-Alogrithmen, wie bereits im Kapitel 2.1 erwähnt, ist dieser Anwendungsfall allerdings nicht dazu bestimmt, eigenständig fehlende Verknüpfungen zu ergänzen. Vielmehr können dem Bearbeiter des Katalogs, Vorschläge für mögliche Verknüpfungen zwischen Ressourcen gegeben werden. Ausgehend von der Annahme, dass Paare von Ressourcen, die beide mit ähnlichen Ressourcen in Verbindung stehen, ebenfalls in Verbindung stehen könnten, kann der im Kapitel 2.1 erläuterte Adamic-Adar-Index verwendet werden. Dieser folgt dem Ansatz, dass Knotenpaare in einem Graphen, mit einer ähnlichen Nachbarschaft, eher in Verbindung stehen als solche mit unterschiedlichen Nachbarschaften, gewichtet hierbei allerdings gemeinsame Nachbarn mit niedrigem Knotengrad mehr. Das Resultat ist ein Indexwert, welcher eine hohe Wahrscheinlichkeit einer Verbindung zwischen einem Knotenpaar, mit einem im Vergleich hohen Wert bemisst.

Im Sinne der Qualitätssicherung eines SDDI-Katalogs bietet es sich an, bereits bei dem Anlegen eines neuen Katalogeintrages, Ressourcen vorzuschlagen, die in Beziehung zu der angelegten Ressource stehen könnten. Dadurch könnten fehlende Verknüpfungen vermieden werden. Allerdings muss beachtet werden, dass die Vorgeschlagene Link-Prediction Methode erst verwendet werden kann, wenn bereits eine korrekte Beziehung angegeben wurde. Darüber hinaus ist es mit dieser Methode auch nicht möglich, die Richtung oder die Art der Beziehung hervor zusehen. Die Entscheidung, ob und in welcher Form eine vorgeschlagene Beziehung besteht, muss von einem Bearbeiter getroffen werden. Auch eine periodische Überprüfung eines Katalogs könnte empfehlenswert sein, um Beziehungen zu entdecken, die trotzdem übersehen wurden.

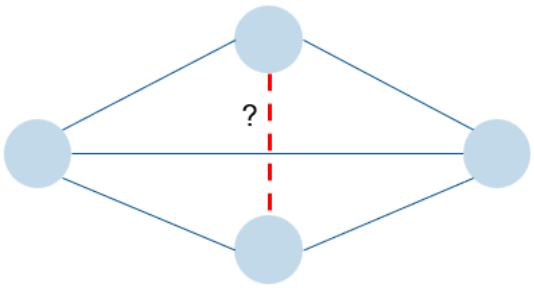


Abbildung 3.7: Allgemeine Fragestellung der Linkprediction

3.4 Use Cases der Informationsgewinnung

Die Use Cases der Informationsgewinnung, welche im Folgenden betrachtet werden, zielen darauf ab, aus der Analyse und Visualisierung der Graphstruktur eines SDDI-Katalogs neue Erkenntnisse über die in ihm enthaltenen Ressourcen zu erlangen. Sie richten sich nicht nur an die Bearbeiter des Katalogs, sondern könnten Informationen aufdecken, welche für alle Nutzer des Katalogs interessant sein könnten.

3.4.1 Erkennung wichtiger Ressourcen

Mithilfe einer graph-basierten Analyse können, aus der Struktur des Graphen, Metriken abgeleitet werden, mit denen die Wichtigkeit einer Ressource im Katalog bewertet werden können. Zunächst muss dafür definiert werden, was eine wichtige Ressource in einem SDDI-Katalog ausmacht. Mit wichtigen Ressourcen werden in diesen Kontext solche Ressourcen gemeint:

- von denen viele weitere Ressourcen abhängig sind. (*depends_on*)
- bei denen viele weitere Ressourcen einen Teil von dieser darstellen. (*child_of*)
- auf die von vielen weiteren Ressourcen verwiesen wird. (*links_to*)

Wichtige Ressourcen werden somit im Graph durch Knoten repräsentiert, die viele eingehende Kanten besitzen. Diese Definition kann darüber hinaus erweitert werden, indem argumentiert wird, dass Ressourcen, auf die von wichtigen gezeigt wird, selber wichtig sind. Aufgrund dessen bietet sich für die Erkennung von wichtigen Ressourcen eine Analyse des Graphs auf die Zentralitäswerte der Knoten an. Genauer betrachtet erscheint, unter Berücksichtigung erwähnten Kriterien, der Zentralitätswert *PageRank* als geeignete Metrik, um die Wichtigkeit einer Ressource zu bewerten (vgl. Kapitel 2.1). Diese beachtet neben der Anzahl der eingehenden Kanten auch die Wichtigkeit der Knoten, die auf einen Knoten zeigen, berücksichtigt dabei aber auch, ob der betrachtete Knoten nur eine von vielen ist, auf die gezeigt wird, wodurch die Bedeutung einer Referenz als gemindert angesehen werden kann.

Der so bestimmte Zentralitätswert eines Knotens könnte dann bei der Verwaltung der repräsentierten Ressource im Katalog berücksichtigt werden. Es wäre zum Beispiel denkbar, dass für die Bearbeitung von wichtigen Ressourcen, besondere Bearbeitungsrechte eingeführt werden, sodass solche nur im Vier-Augen-Prinzip verändert werden, oder sodass nur der Administrator einer Organisation Veränderungen durchführen kann. Ebenso wäre es denkbar,

dass den Verwaltern von Ressourcen, welche in Verbindung mit einer wichtigen Ressource stehen, eine Mitteilung darüber gegeben wird, wenn diese verändert wird. Die Verwalter könnten dann die Auswirkung der Veränderung auf ihre eigenen Ressourcen bewerten. Darüber hinaus ist es wichtig, dass solche wichtigen Ressourcen immer auf dem neusten Stand sind und gut erläutert werden. Es könnte auch in Betracht gezogen werden, die Ressourcen auf der Webseite des Katalogs nach ihrer Wichtigkeit zu sortieren.

3.4.2 Erkennung von Communities

Dieser Anwendungsfall für eine graph-basierte Analyse eines SDDI-Katalogs dient dazu, die zugrunde liegende Struktur und Organisation der Ressourcen innerhalb eines Katalogs aufzudecken. Diese Strukturen können dafür genutzt werden, um Einblicke in die verschiedenen Themen oder Anwendungsfälle eines Katalogs zu erlangen. Die Annahme hierbei besteht darin, dass Ressourcen, die eng miteinander verknüpft sind, thematisch ähnlich sind oder für denselben Anwendungsfall des Urbanen Digitalen Zwilling verwendet werden. Mithilfe von Community-Detection Algorithmen, wie dem Louvain-Algorithmus, können solche eng verknüpfte Gruppen von Ressourcen in der graphischen Abbildung eines SDDI-Katalogs aufgedeckt werden. Der Louvain-Algorithmus teilt die Knoten eines Graphen so in Gruppen ein, sodass innerhalb der Gruppen möglichst viele Kanten sind und möglichst wenige zwischen den Gruppen (vgl. Kapitel 2.1). Diese gefundenen Ressourcen-Gruppen können dann weiterführend analysiert werden, um die thematischen Schwerpunkte des UDZ aufzudecken. Da in den Metadaten der Ressourcen bereits Tags vorhanden sind, welche die Knoten beschreiben, bietet es sich an, die Tags der Gruppenmitglieder zu untersuchen. Aus ihnen kann ein übergeordnetes Thema extrahiert werden, das die Gesamtheit der detektierten Gruppe beschreiben kann. Dadurch könnte ein Nutzer, welcher mit einem Katalog nicht vertraut ist, bei der Navigation durch den Katalog unterstützt werden. Aktuell ist es der Katalogseite zwar möglich Ressourcen nach ihren Tags und Kategorien zu filtern, allerdings ist dabei nicht ersichtlich, ob und wie die gefundenen Ressourcen im Zusammenhang stehen. Durch die Aufteilung des Katalogs in Gruppen könnte der Nutzer ein für ihn relevantes Themengebiet wählen. Basierend auf dieser Auswahl können Ressourcen aus anderen Bereichen, also Knoten, die einer anderen Community zugeordnet wurden, ausgeblendet werden, wodurch eine gezieltere Erkundung der in Zusammenhang stehenden Ressourcen ermöglicht wird. Dadurch bietet sich die Möglichkeit, dem Nutzer interessante Ressourcen vorzustellen, ohne dass durch die Spezifität einzelner Tags eventuell relevante Ressourcen ausgelassen werden, oder dass irrelevante Ressourcen aus anderen Themenbereiche aussortiert werden müssen.

4 Implementierung der Use Cases

Im bisherigen Kapitel wurden einige Use Cases der graphbasierten Analyse und Visualisierung eines Metadatenkatalogs theoretisch untersucht. Das folgende Kapitel befasst sich nun mit der praktischen Umsetzung einer Auswahl dieser Use Cases. Im Mittelpunkt steht die eigene, in Python 3.11.9 implementierte Schnittstelle, welche die Graphstruktur der SDDI-Metadatenkataloge visualisieren kann und Analysewerkzeuge für die Umsetzung der Use Cases bereitstellt. Als Datengrundlage dienen die SDDI-Kataloge der Projekte SAVeNoW[1], TwinBy[2] und AgriHub[3]. Es werden zunächst die wichtigsten Abhängigkeiten von Software dritter erläutert, anschließend werden wichtige Komponenten der Schnittstelle präsentiert und zuletzt die Umsetzung der Use Cases in der Schnittstelle vorgestellt.

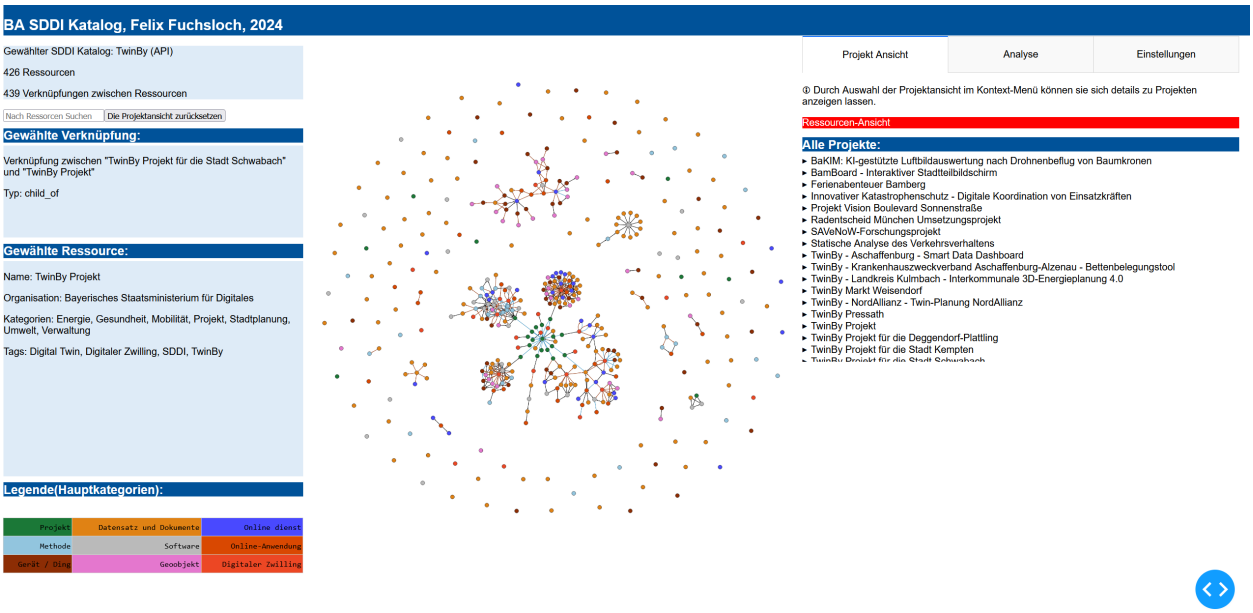


Abbildung 4.1: Übersicht der Schnittstelle

[1] https://savenow.de/de/
[2] https://www.bayern-innovativ.de/de/seite/twinby
[3] https://agrihub.hef.tum.de/, nur aus dem Netzwerk der TU München erreichbar

4.1 Abhängigkeiten

4.1.1 CKAN API

Der SDDI-Katalog wurde mit der Katalog-Software CKAN implementiert. [16] CKAN ist ein Open Source Datenmanagementsystem für Datenzentren und Datenportale, welches die Veröffentlichung, Nutzung und das Teilen von Daten vereinfachen soll. Mithilfe der CKAN-API (Application Programming Interface), können die Funktionalitäten des Systems durch externen Code abgerufen werden. Dadurch kann eine eigene Applikation nach Datensätzen eines Katalogs suchen und auf diese zugreifen. Mithilfe bestimmter Anfragen können hierdurch sämtliche Datensätze eines Katalogs im JSON-Format abgerufen werden. [25] In der eigenen Schnittstelle wird die API genutzt, um die Metadaten der digitalen Ressourcen eines SDDI-Katalogs zu erhalten.

4.1.2 NetworkX

Das Open Source Python-Paket NetworkX kann für die Erstellung und Manipulation von komplexen Netzwerken, sowie für die Untersuchung von Struktur, Dynamik und Funktionen dieser genutzt werden. Es bietet Datenstrukturen für ungerichtete, gerichtete und Multigraphen an und beinhaltet viele verbreitete Graph-Algorithmen und Metriken für die Netzwerkstruktur und Analysen. Die Datenstrukturen ermöglichen es, willkürliche Python-Objekte mit optionalen als Knoten zu speichern und Kanten mit beliebigen Daten in Form von Schlüssel-Wert Attributen zu markieren. Die Abbildung des SDDI-Katalogs wird in der Schnittstelle als NetworkX-Objekt der Klasse *DiGraph* gespeichert. Dieses nutzt Python *dict*-Objekte, um die Knoten und ihre Nachbarn zu speichern und lässt gerichtete Kanten zu. [9] Die in den Metadaten gespeicherten Attribute werden als Schlüssel-Wert Paar in den dicts der entsprechenden Knoten und Kanten gespeichert.

4.1.3 Dash Cytoscape

Dash ist ein Open Source Python Paket für die Erstellung von Web-basierten Anwendungen. Mithilfe dieses Pakets können Applikationen mit diversen interaktiven Elementen erstellt werden. Aufgrund der Unterstützung einer umfassenden Auswahl von Diagrammen ist es besonders für Programme der Datenanalyse und Visualisierung geeignet. [26] Dash Cytoscape ist eine Graph-Visualisierungskomponente, die es ermöglicht interaktive Netzwerke,

im Zusammenspiel mit anderen Dash-Elementen, mittels einer Auswahl von Layouts und einer benutzerdefinierten Farbgebung, zu visualisieren. [27] Mit Dash wurde die GUI der eigenen Schnittstelle und mit Dash Cytoscape die Visualisierung des Graphen erstellt. Die benutzerdefinierte Farbgebung mittels von *stylesheets*, ermöglicht es die Kanten und Knoten eines Graphen, basierend auf deren Attibuten, einzufärben. Sie werden in der Schnittstelle im Rahmen der Use Cases benutzt, um wichtige Informationen visuell darzustellen.

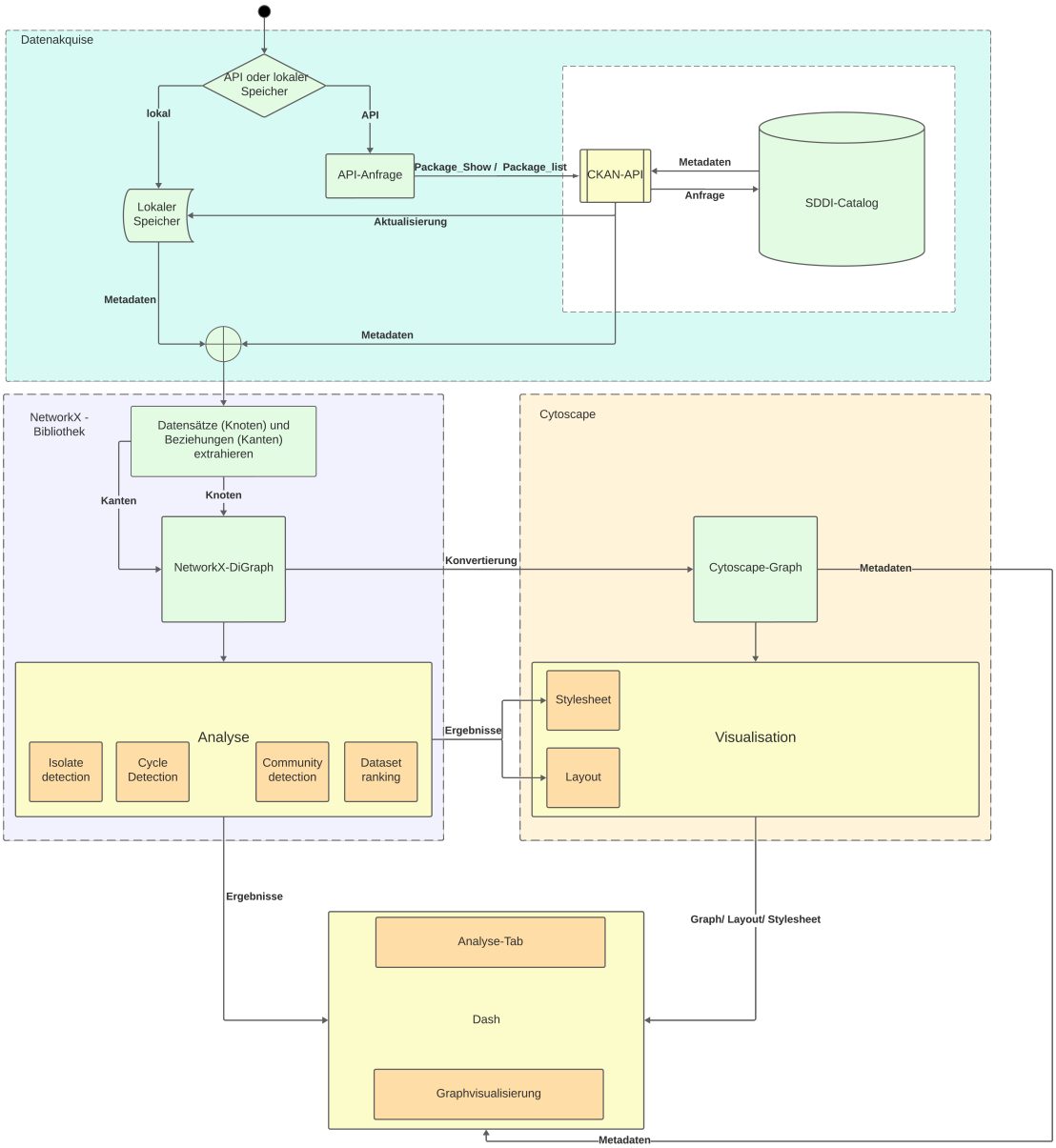


Abbildung 4.2: Genereller Aufbau der Schnittstelle

4.2 Schnittstellen-Komponenten

4.2.1 Datenakquise

Nach dem Start des Programms erhält der Nutzer in einem Pop-up-Fenster (vgl. Abbildung 4.3) die Wahl, welcher SDDI-Katalog durch das Programm analysiert und visualisiert werden soll. Hierbei kann entschieden werden, ob die aktuellen Metadaten durch eine Anfrage an die CKAN-API (vgl. Kapitel 4.1.1) heruntergeladen werden sollen, oder falls vorhanden, ein eigener lokaler Speicherstand genutzt werden soll (vgl. Abbildung 4.3). Im Rahmen einer API-Anfrage wird zunächst, mittels des Befehls *package_list*, eine Liste mit den Namen aller im Katalogeinträge angefordert. Anschließend werden anhand des *package_show*-Befehls die Metadaten der jeweiligen Einträge angefragt. Im Anschluss wird ein lokaler Speicherstand der Metadaten angelegt, oder ein bereits existierender aktualisiert (vgl. Abbildung 4.2). Für die Nutzung der API werden bei den Katalogen eigene API-Token benötigt, welche bei der Registrierung im Katalog generiert werden. Um auf den AgriHUB-Katalog des HEF zugreifen zu können, muss sich der Nutzer darüber hinaus im Netzwerk der TU München befinden.

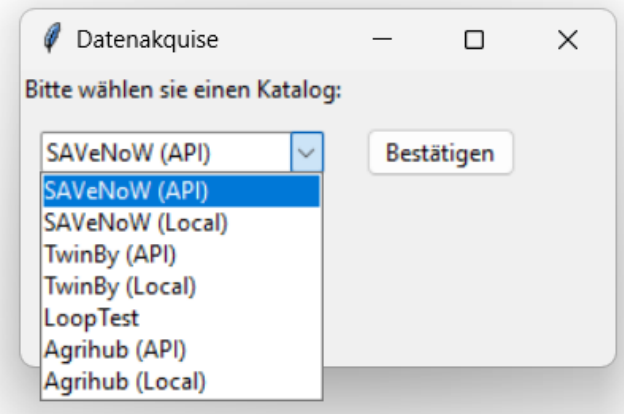


Abbildung 4.3: Pop-up-Fenster zur Datenwahl

4.2.2 Graph-Visualisierung

Die Visualisierung der Graphstruktur des SDDI-Metadatenkatalogs steht im Zentrum der Schnittstelle (vgl. Abbildung 4.1) und verbindet die genannten Ansichten und die Analysewerkzeuge der Use Cases. Sie soll, wie im Kapitel 3.2 beschrieben, den Nutzer der Schnittstelle bei der Interpretation der Struktur und der Beziehungen eines Urbanen Digitalen Zwillings

unterstützen, und ebenso bei dem Verständnis der Ergebnisse der Analysewerkzeuge der Use Cases helfen.

Zur Visualisierung des gesamten Graphen wird von der Schnittstelle standardmäßig das Cytoscape-Layout *euler* angewandt, da die Konfiguration dieses force-directed Layouts, dem eigenen Empfinden nach besonders leserliche und ästhetische Graph-Visualisierungen produziert. In den Einstellungen hat der Nutzer die Möglichkeit, das Graph-Layout der Visualisierung bei Bedarf zu ändern und kann dabei aus verschiedenen, von der Cytoscape-Dash Bibliothek vordefinierten, Layouts wählen. Auch die Stilisierung der Knoten kann durch den Nutzer angepasst werden. Die Farbgebung der standardmäßigen Klassifizierung nach der Hauptkategorie einer Ressource, kann durch eine Farbgebung nach Zugehörigkeit zu einer Organisation oder einer Community, geändert werden. Darüber hinaus bietet die Visualisierung eine Auswahl von Interaktionen an, welche dem Nutzer bei der Interpretation des Graphen unterstützen sollen. Zunächst ist zu erwähnen, dass der Blickpunkt auf den Graphen verändert werden kann. Es kann auf einzelne Bereiche des Graphen gezoomt werden und der Graph als Ganzes oder auch einzelne Knoten verschoben werden, wodurch der Nutzer die Ansicht auf für ihn interessante Bereiche fokussieren kann.

Die zentralen Methoden der Interaktion können allerdings durch Klicken und Schweben über den Graphelementen abgerufen werden. Durch einen Linksklick auf einen Knoten kann ein Knoten ausgewählt werden. Basierend auf dem ausgewählten Knoten, werden in der Ressourcen-Ansicht (vgl. Kapitel 4.2.3) Eigenschaften der repräsentierten Ressource angezeigt, und in der Projekt-Ansicht (vgl. Kapitel 4.2.4), Eigenschaften des Projektes, dem die repräsentierte Ressource angehört. Durch einen Rechtsklick in der Visualisierung kann ein Kontextmenü geöffnet werden, in dem zwischen den Ansichten gewechselt werden kann, und der Katalogeintrag der ausgewählten Ressource auf der Webseite des SDDI-Katalogs abgerufen werden kann. Wenn die Maus über einen Knoten gehalten wird, werden zwei Aktionen ausgelöst. Zum einen wird die Beschriftung der repräsentierten Ressource, also ihr Name, angezeigt. Die Beschriftungen sind zur Vermeidung von Überlappungen im Sinne der Lesbarkeit standardmäßig ausgeblendet. Bei einem Mouseover wird die Beschriftung eines Knotens allerdings eingeblendet (vgl. Abbildung 4.4). Diese hat zusätzlich einen farbigen Hintergrund, welcher die Hauptkategorie der Ressource anzeigt, auch wenn der Nutzer eine andere Farbgebung gewählt hat. Zum anderen werden die mit dem Knoten adjazenten Knoten hervorgehoben. Dies geschieht, indem die Opazität aller anderen Knoten und irrelevanten Kanten reduziert wird. Dadurch erhält der Nutzer eine bessere Übersicht über die Beziehungen einer einzelnen Ressource.

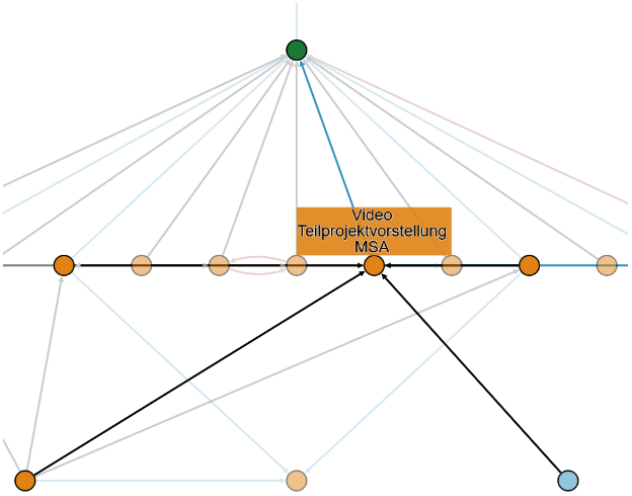


Abbildung 4.4: Verhalten der Visualisierung bei einem Mouseover

4.2.3 Ressourcen-Ansicht

Die auf der Abbildung 4.5 zu sehende Ressourcen-Ansicht der Schnittstelle, ermöglicht es umfassende Informationen zu Ressourcen im Katalog zu erlangen. Durch Anklicken eines Graphelements (Knoten/Kante) in der Visualisierung können Metadaten zu der repräsentierten Ressource oder Beziehung in einem Fenster angegeben werden. Standardmäßig werden bei Knoten der Name, die zugehörige Organisation, sowie Kategorien und Tags angezeigt und bei Auswahl einer Kante die Namen der in Beziehung stehenden Ressourcen und die Art der Beziehung. Falls spezifischere Informationen benötigt werden, kann die Anzeige in den Einstellungen angepasst werden, sodass die vollständigen Metadaten im JSON-Format angezeigt werden. Diese sind zwar nicht so leicht zu verstehen, bieten aber einen tieferen Einblick in die Eigenschaften einer Ressource.

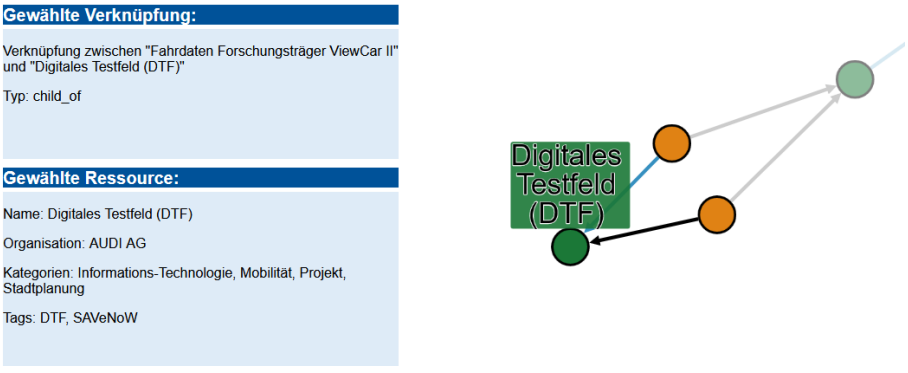
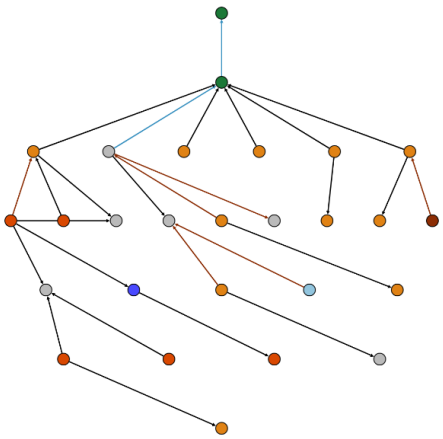


Abbildung 4.5: Ressourcenansicht der Schnittstelle, eigene Darstellung

4.2.4 Projekt-Ansicht

In der Projekt-Ansicht erhält der Nutzer weiterführende Informationen zu Projekten im gewählten Katalog. Nach der Auswahl einer beliebigen Ressource kann sich der Nutzer, durch Wahl des Projekt-Ansichtsmodus im Kontextmenü, eine Zusammenstellung von Informationen, über das mit ihr assoziierte Projekt ausgeben lassen (vgl. Abbildung 4.6). In dieser Ansicht werden zur besseren Übersicht Ressourcen, die nicht Teil des Projekts sind, aus der Visualisierung entfernt. Um die Hierarchie der Ressourcen im Projekt zu verdeutlichen, wird für die Graph-Visualisierung ein Breadth-First-Layout (vgl. Kapitel 3.2) angewandt, wobei hier als Startknoten das Projekt mit der höchsten Zentralität benutzt wird. Die Zusammenstellung der Informationen enthält zum einen, eine Übersicht darüber, welchen Hauptkategorien (vgl. Kapitel 2.4) die Projekt-Ressourcen angehören, und wie viele es jeweils sind. Zum anderen werden die Organisationen aufgelistet, zu denen die Ressourcen eines Projekts gehören, um Rückschlüsse auf Kooperationen ziehen zu können. Zuletzt gibt es auch einen Überblick über alle Projekte im Katalog, in der durch Klicken des nebenstehenden Dreiecks (vgl. Abbildung 4.6, unten rechts), die zugehörigen Ressourcen aufgelistet werden.



Projekt-Ansicht

Projektansicht: Simulation und Wirkungsanalyse von Maßnahmen (SWM)

Ressourcen im Project:

·datasets: 11

·online-services: 1

·methods: 1

·softwares: 6

·online-applications: 5

·devices: 1

Das Projekt enthält Ressourcen von:

·Lehrstuhl für Verkehrstechnik

·3D Mapping Solutions GmbH

·Lehrstuhl für Geoinformatik

·Bayerische Vermessungsverwaltung

·Technische Hochschule Ingolstadt (THI)

·Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)

·Softwareanbieter

·AUDI AG

Alle Projekte:

- Digitales Testfeld (DTF)
- Gesellschaftliche Teilhabe und begleitende Fragestellungen (GTBF)
- Modellstrukturierung und Architektur (MSA)
- SAVeNoW
- Simulation und Wirkungsanalyse von Maßnahmen (SWM)
- Virtuelles Testfeld (VTF)

Abbildung 4.6: Projekt-Ansicht der Schnittstelle, eigene Darstellung

4.3 Use Cases

4.3.1 Erkennung isolierter Ressourcen

Im Rahmen der Detektion von isolierten Ressourcen, wird eine Kombination von Analysewerkzeugen der NetworkX-Bibliothek genutzt. Mittels der `isolates()` Funktion können Knoten mit den Ein- und Ausgangsgraden 0 erkannt werden. Mit der Funktion `number_weakly_connected_components()` kann ein von den vollständig isolierten Knoten bereinigter Graph, auf die Anzahl von schwach verbundenen Komponenten (vgl. Kapitel 3.3.1 und 2.1) analysiert werden. Werden durch diese Funktion mehrere Komponenten detektiert, existiert mindestens eine isolierte Zusammenhangskomponente im Graphen. Während die einzelnen isolierten Knoten mit einer signalisierenden roten Farbgebung dargestellt werden, haben die restlichen Graphkomponenten eine schlichte Farbgebung, um die Aufmerksamkeit des Nutzers auf die isolierten Knoten zu richten (vgl. Abbildung 4.7). Aufgrund der abstoßenden Kräfte zwischen den Knoten bei dem *euler*-Layout stoßen sich getrennte Graphkomponenten voneinander ab, wodurch die Existenz mehrerer an der Visualisierung schnell zu erkennen ist. Darüber hinaus wurde neben der Visualisierung ein Kreisdiagramm platziert, das den Anteil der vollständig isolierten Knoten am Gesamtgraph darstellt. Des Weiteren wird der Nutzer bei einem Befund gewarnt, indem ihm die Namen der isolierten Ressourcen mitgeteilt und über die Anzahl der überschüssigen Komponenten aufgeklärt wird. An der Abbildung 4.7 kann erkannt werden, dass sie Schnittstelle die isolierten Knoten und Komponenten im Graphen des SAVeNoW-Katalogs erfolgreich erkannt hat.

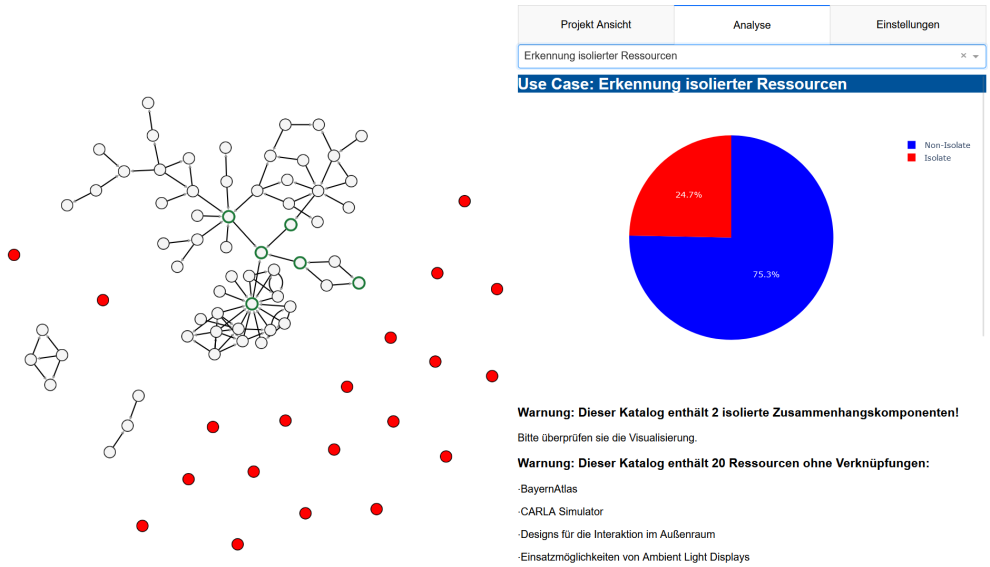


Abbildung 4.7: Use Case der Erkennung von isolierten Ressourcen im SAVeNoW-Katalog

4.3.2 Erkennung von Zyklen

Für die Erkennung von Zyklen im Graphen (vgl. Kapitel 3.3.2) wird in der Schnittstelle die NetworkX-Funktion `simple_cycles()` verwendet. Diese kann alle Zyklen erkennen, in denen ein Knoten exakt ein Mal vorkommt. Sie unterscheidet hierbei nur zwischen Zyklen, solange sie keine Permutationen voneinander sind, also unterschiedliche Abfolgen von Knoten und Kanten besitzen [9]. Das Resultat dieser Funktion sind Listen, in denen die Knoten der gefundenen Zyklen nach ihrer Reihenfolge aufgelistet sind. Mittels der Knoten kann anschließend im Graph die Art der Verknüpfung zwischen ihnen recherchiert werden. Die Selbstabhängigkeit einer Ressource kann dann festgestellt werden, wenn eine Schleife nur Verknüpfungen des Typs *depends_on* enthält. Diese wird verwendet, um die detektierten Schleifen mithilfe der Farbgebung vom restlichen Graphen visuell abzuheben. Die Kanten und Knoten die Teil einer Schleife sind, werden in der Visualisierung des Graphen rot eingefärbt. Ergänzend werden die Namen der Ressourcen Schleifenweise angegeben, wobei bei Schleifen, die auf eine Selbstabhängigkeit deuten, dies zusätzlich angemerkt wird. Auf Basis dieser Ergebnisse können dann die Bearbeiter des Katalogs die Selbstabhängigkeiten beseitigen und potenziell fehlerhafte Verknüpfungen überprüfen.

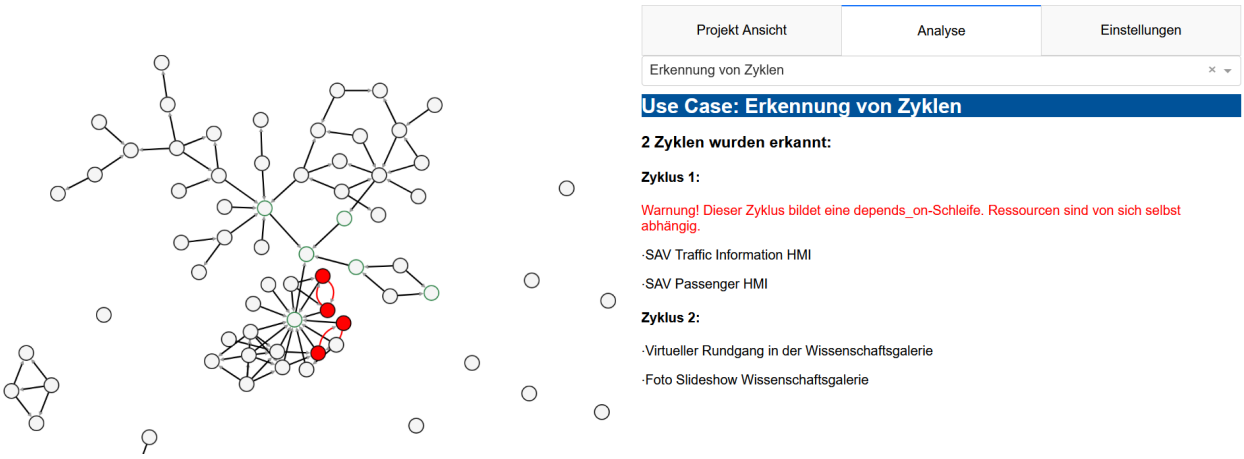


Abbildung 4.8: Use Case der Erkennung von Zyklen im SAVeNoW-Katalog

4.3.3 Hervorsage möglicher Verknüpfungen

Ziel der Umsetzung dieses Use Cases ist es, dem Nutzer Vorschläge zu eventuell fehlenden Verknüpfungen im Katalog zu geben. Hierfür berechnet die Schnittstelle mittels der NetworkX-Funktion `adamic_adar_index()` den Adamic-Adar Index (vgl. Kapitel 2.1) aller Knotenpaare im Graphen. Da diese Funktion allerdings nur für ungerichtete Graphen implementiert ist, wird für die Berechnung eine ungerichtete Abbildung des SDDI-Katalogs

verwendet. Aufgrund dessen kann die Schnittstelle zwar mögliche Beziehungen zwischen Ressourcen im Katalog andeuten, allerdings keine Richtung dieser Beziehungen angeben. Da die Funktion die Indizes aller Knotenpaare im Graphen berechnet, können viele niedrige Ergebnisse der Funktion ausgefiltert werden, da diese sehr wahrscheinlich eine non-existente Beziehung beschreiben. Die Schnittstelle zeigt dem Nutzer lediglich die 10 von dem Adamic-Adar Ansatz am wahrscheinlichsten angesehenen Verknüpfungen an. Diese werden in der Visualisierung durch künstlich ergänzte Kanten zwischen dem entsprechenden Knotenpaar signalisiert, wie auf Abbildung 4.9 zu erkennen ist. Die von der Schnittstelle hervorgesagten Beziehungen sind hierbei durchaus plausibel. Beispielsweise wird im SAVeNoW-Katalog eine Beziehung zwischen den Ressourcen *Erläuterung der Gruppen in der Katalogplattform* und *SDDI Glossary* hervorgesagt. Diese sind beides Ressourcen des Lehrstuhls für Geoinformatik und verweisen beide auf dieselbe Webseite, welche das Glossar des SDDI-Konzepts hostet. Da *Erläuterung der Gruppen in der Katalogplattform* auf einen Glossareintrag im Glossar verweist, welches durch *SDDI Glossary* repräsentiert wird, wäre eine Verknüpfung zwischen ihnen durchaus angebracht.

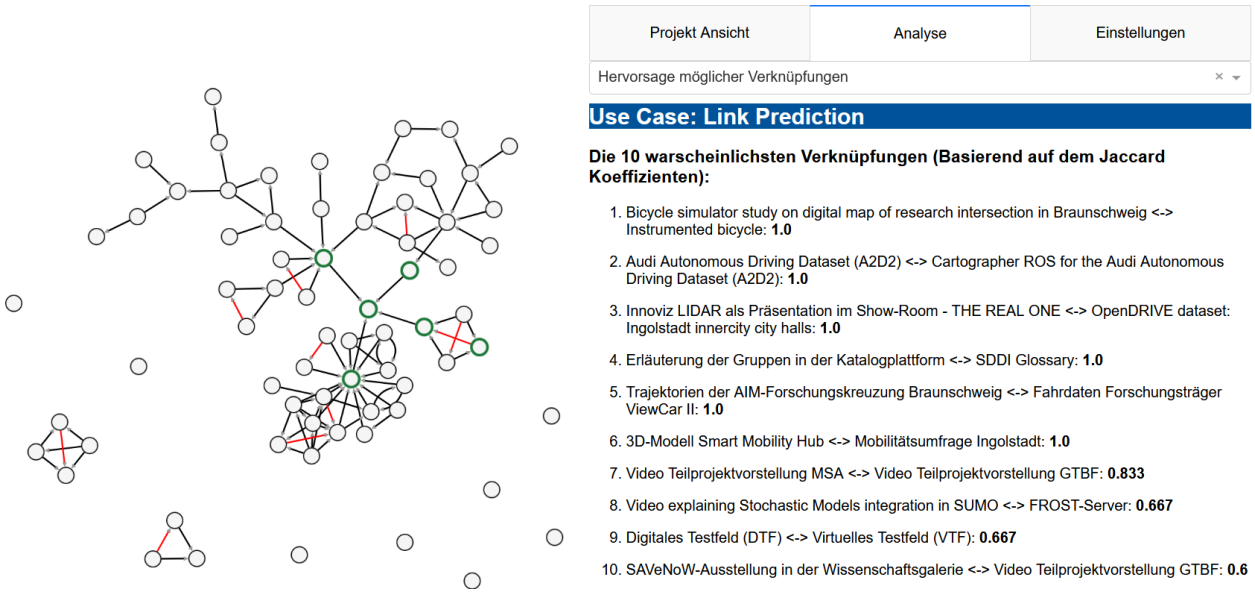


Abbildung 4.9: Use Case der Hervorsage von Beziehungen im SAVeNoW-Katalog

4.3.4 Erkennung wichtiger Ressourcen

In der Schnittstelle werden die PageRank-Zentralitäten (vgl. Kapitel 2.1) der Knoten, mittels der NetworkX-Funktion *pagerank()* berechnet. Für die Visualisierung wird in diesem Anwendungsfall eine besondere Farbgebung für die Knoten des Graphens genutzt. Die Färbung der Knoten ist im Rahmen dieses Use Cases, von dessen Zentralitätswert abhängig.

Die Knoten werden rot eingefärbt, wobei mit abnehmender Zentralität auch die Intensität der Farbe reduziert wird. Hierfür werden im Rahmen der Farbgebung die Zentralitätswerte, durch eine Normalisierung auf einen Bereich von 0 bis 1 skaliert. Somit erscheint der zentralste Knoten rot (maximale Intensität) und solche mit dem niedrigsten Wert im Graphen weiß (minimale Intensität). Durch diese Farbgebung können wichtige Ressourcen des Katalogs in der Visualisierung schnell erkannt werden. Darüber hinaus wird auch eine Rangliste der 10 zentralsten Knoten und ein Histogramm über die Verteilung der Zentralitätswerte bereitgestellt. Da es besonders für die Verwalter der Ressourcen einer Organisationen interessant sein kann, welche ihrer Ressourcen im Katalog besonders wichtig für den UDZ sind, gibt es auch organisationsspezifische Ranglisten. In diesen werden nur Ressourcen angezeigt, die von der spezifischen Organisation erstellt wurden.

An der Abbildung 4.10, welche diesen Use Case am Beispiel des SAVeNoW-Katalogs darstellt, ist gut zu erkennen, dass der Katalogeintrag des Projektes *SAVeNoW* die wichtigste Ressource des Katalogs ist. Dies entspricht auch der Erwartung, da dieser als "Überprojekt" fungiert und der Verknüpfungspunkt anderer Projekte im UDZ ist.

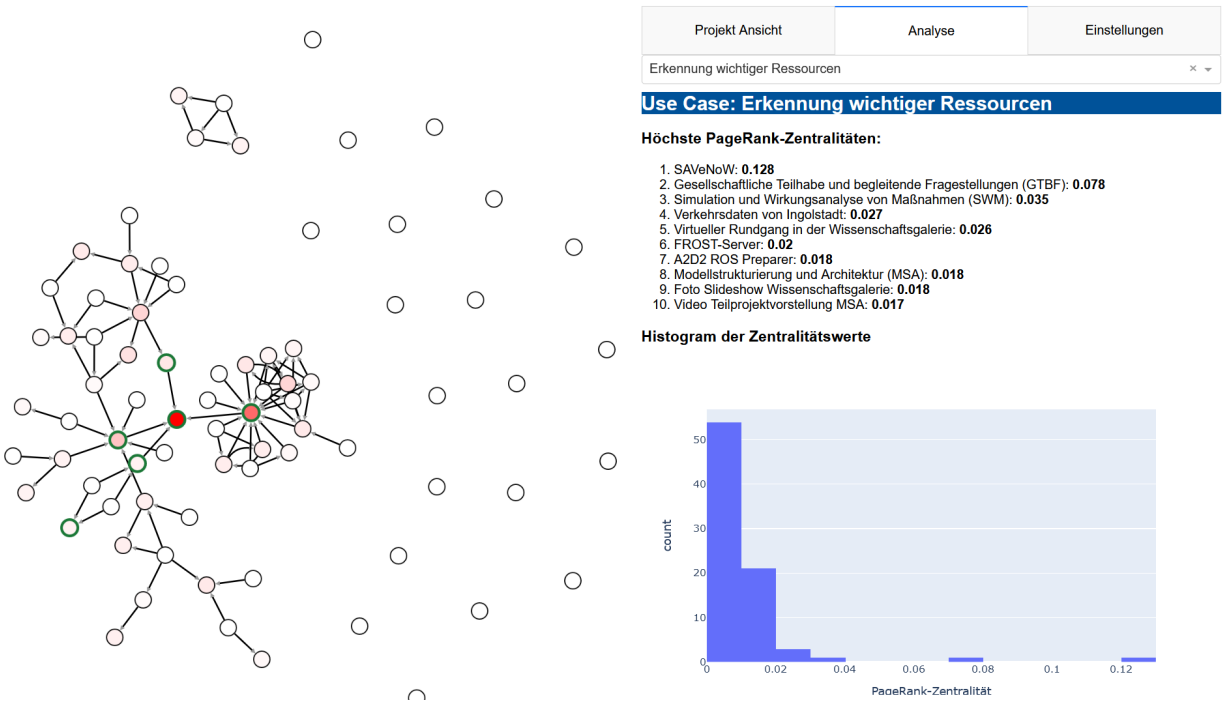


Abbildung 4.10: Use Case der Erkennung von wichtigen Ressourcen im SAVeNoW-Katalog

4.3.5 Erkennung von Communities

Die Erkennung von Communities im Katalog ermöglicht es, Rückschlüsse auf die Struktur und Organisation der darin enthaltenen Ressourcen zu ziehen. Die Schnittstelle nutzt hierfür die NetworkX-Funktion `louvain_communities()`. Diese bestimmt anhand des Louvain-Algorithmus (vgl. Kapitel 2.1), die beste Aufteilung des Graphen in Communities und gibt eine Liste von *sets* aus, in denen die Knoten der jeweiligen entdeckten Community gespeichert sind [9]. Für die Visualisierung der detektierten Communities wird diesen zunächst eine Farbe zugeordnet. Dadurch können alle Knoten, die Teil einer Community sind, in eben dieser Farbe dargestellt werden. Zusätzlich zu der Visualisierung werden weitere Informationen zu den Communities präsentiert. Es wird die Anzahl der entdeckten Communities angegeben und des Weiteren, die verschiedenen Tags der Ressourcen einer Community aufgeführt. Diese Tags sind nach ihrer Häufigkeit sortiert, wodurch dominierende Tags schnell erkannt werden können. Auf der Abbildung 4.11 ist anhand der Farbgebung und des *force-direced* Layout zu erkennen, dass der Katalog in mehrere eng verknüpfte Gruppen aufgeteilt werden kann. Aus den Tags dieser Gruppen lassen sich dann Rückschlüsse auf den Themenbereich ihrer Ressourcen ziehen. Aus den Tags der violetten Gruppe kann zum Beispiel interpretiert werden, dass ihre Ressourcen Teils eines Digitalen Zwillings sind, welcher einen Winterdienst repräsentiert. Ist ein Nutzer an dieses Themengebiet interessiert, könnten diesem die Gruppenmitglieder als möglicherweise interessante Ressourcen vorgeschlagen werden.

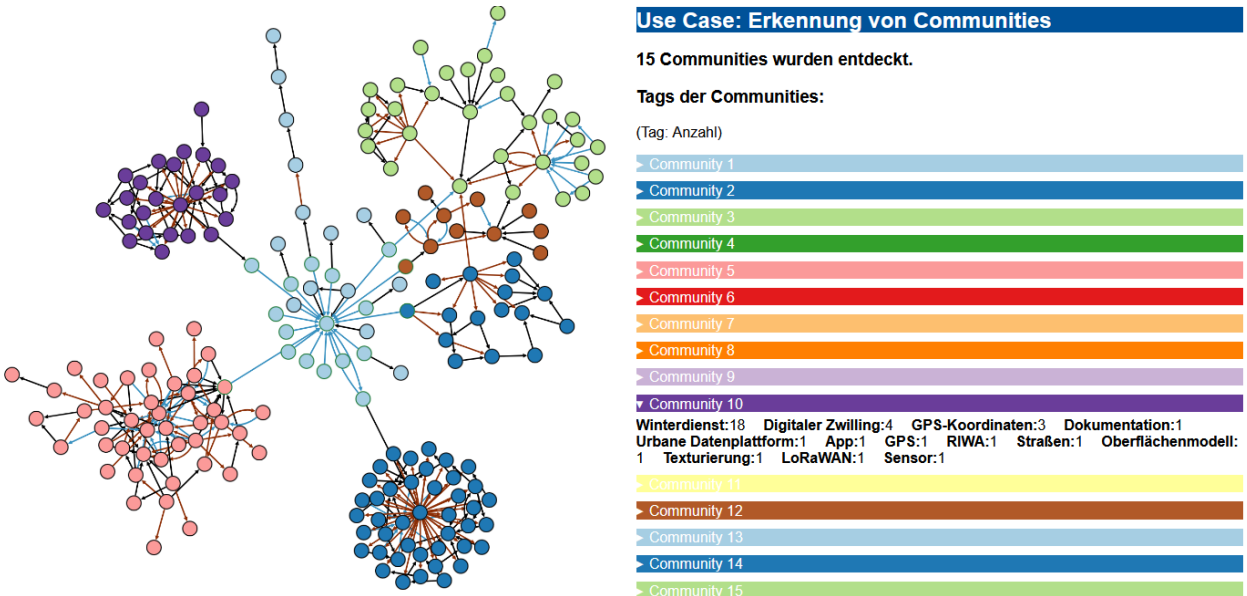


Abbildung 4.11: Use Case der Erkennung von Communities im TwinBy-Katalog

5 Fazit

5.1 Diskussion

Aus den Use Cases, die in den Kapiteln 3 und 4 betrachtet wurde, konnte festgestellt werden, dass durch eine graph-basierte Analyse und Visualisierung der Metadaten in einem SDDI-Katalog, Einblicke in die Struktur und Organisation eines SDDI-Katalogs gewonnen werden können. Die Erkenntnisse der Analysen können dazu verwendet werden, die Qualität der Metadaten von Ressourcen eines Urbanen Digitalen Zwilling zu sichern und interessante Informationen zu ihrer Organisation abzuleiten. Mit den Use Cases der Qualitätssicherung können isolierte Ressourcen und problematische Verknüpfungen identifiziert werden und durch Link-Prediction Algorithmen kann dazu beigetragen werden, dass die Verbindungen im Katalog vollständig sind. Mithilfe der Use Cases der Informationsgewinnung können in Rahmen der Detektion von wichtigen Ressourcen, Ressourcen gefunden werden, die zum einen besonders schutzwürdig sind und eine besondere Aufmerksamkeit bezüglich ihrer Aktualität und Korrektheit erfordern. Mit der Erkennung von Communities könnte ein Katalog in verschiedene Themenbereiche untergliedert werden, wodurch der Nutzer bei der Navigation durch die Ressourcen unterstützt werden kann. Darüber hinaus konnten in dieser Arbeit auch generelle Punkte erarbeitet werden, auf die bei der Visualisierung der Graphstruktur der Metadaten eines Urbanen Digitalen Zwilling geachtet werden müssen.

Die eigene Implementierung der Use Cases stößt allerdings teilweise auch auf Grenzen. Bei der Erkennung von wichtigen Ressourcen wurden die Nachbarschaftsbeziehungen verwendet, um die Wichtigkeit einer Ressource zu bewerten. Allerdings konnten hierbei die verschiedenen Beziehungstypen (*depends_on*, *child_of*, *links_to*) nicht unterschiedlich gewichtet werden, wodurch bei der Berechnung der Zentralität gleichwertig behandelt werden, obwohl einer Abhängigkeitsbeziehung theoretisch eine höhere Bedeutung zugesprochen werden könnte, als einem simplen Verweis. Im Kontext der Hervorsage von Verknüpfungen ist es zu erwähnen, dass die eigene Schnittstelle nicht in der Lage ist, die Art oder Richtung der hervorgesagten Verknüpfung zu bestimmen, da die benötigte Funktion aus der NetworkX-Bibliothek nur für ungerichtete Graphen implementiert ist.

Zusammenfassend lässt sich allerdings sagen, dass verschiedene nützliche Anwendungsfälle einer graph-basierten Analyse und Visualisierung der Metadaten eines SDDI-Katalogs gefunden werden konnten und anhand dieser die Teilhaber an einem Urbanen Digitalen Zwillings,

bei der Erkundung, Nutzung und Organisation der digitalen Ressourcen unterstützt werden können.

5.2 Vorschläge zur Weiterentwicklung/Ausblick

Für die Weiterentwicklung der Hervorsage von Verknüpfungen könnte es sinnvoll sein, Link Prediction Algorithmen zu testen, die neben der Netzwerkstruktur auch die Attribute der Metadaten betrachten, um die Wahrscheinlichkeit einer Beziehung zu bewerten. Hierfür gibt es bereits Machine Learning Ansätze, welche bessere Ergebnisse als simple Link Prediction Methoden liefern könnten. Ein nützlicher Use Case, welcher im Rahmen dieser Arbeit nicht näher betrachtet werden konnte, ist die Duplikatserkennung. Diese könnte vor allem bei der Kombination mehrerer Kataloge interessant sein. Da ein Objekt der realen Welt in verschiedenen Katalogen registriert sein könnte, muss bei der Zusammenführung darauf geachtet werden, dass der kombinierte Katalog am Ende nicht mehrere Einträge enthält, die sich auf dasselbe Objekt beziehen. Diese Duplikate müssen als solche erkannt werden, um diese in einen einzelnen Eintrag kombinieren zu können. Ein Ansatz für die Erkennung dieser könnte der Vergleich der Namen, der Tags oder der geographischen Lagen sein. Darüber hinaus könnte auch eine kartographische Visualisierung des Graphen in Betracht gezogen werden, welche nützlich sein kann, wenn die räumliche Ausdehnung (*Location*) der Ressourcen weit verbreitet ist.

Basierend auf den Erkenntnissen dieser Arbeit bietet es sich an, die genutzten Analyse und Visualisierungsmethoden direkt in die Webseiten der SDDI-Kataloge zu integrieren, um die Bearbeiter des Katalogs bei Organisation und Verwaltung des Katalogs zu unterstützen, sodass eine korrekte und vollständige Repräsentation der Ressourcen eines Urbanen Digitalen Zwilling gewährleistet werden kann. Auch die Nutzer des Katalogs könnten bei der Erkundung und Nutzung davon profitieren, wenn weiterführende Analysen und Visualisierungen direkt auf der Webseite des Katalogs verfügbar sind, und nicht über ein externes Werkzeug abgerufen werden müssen.

Literaturverzeichnis

[1] Lehrstuhl für Geoinformatik der TU München. Geobasierter digitaler zwilling bayern - leitfaden. https://collab.dvb.bayern/display/TUMdzb/, o.J. Zuletzt abgerufen 12. Mai 2024.

[2] M. Newman. *Networks*. OUP Oxford, 2018. ISBN 9780192527493. URL https://books.google.de/books?id=YdZjDwAAQBAJ.

[3] T.H. Kolbe. Geoinformatik: Vorlesung 9: Graphen und kürzeste wege, 2022. URL https://www.moodle.tum.de/pluginfile.php/3778895/mod_resource/content/0/Geoinformatik%201%20-%2009%20-%20Graphen%20%20ku%CC%88rzeste%20Wege.pdf. Zuletzt abgerufen 22. April 2024.

[4] P. Tittmann. *Graphentheorie*. Carl Hanser Verlag GmbH & Co. KG, München, 4., aktualisierte und erweiterte auflage edition, 2021. doi: 10.3139/9783446472471. URL https://www.hanser-elibrary.com/doi/abs/10.3139/9783446472471.

[5] Wikipedia contributors. Adamic–adar index, 2024. URL https://en.wikipedia.org/w/index.php?title=Adamic%E2%80%93Adar_index&oldid=1213445747. Zuletzt abgerufen am 9. Mai 2024.

[6] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. *Proceedings of the twelfth international conference on Information and knowledge management*, Nov 2003. doi: 10.1145/956863.956972.

[7] Wikipedia contributors. Louvain method, 2024. URL https://en.wikipedia.org/w/index.php?title=Louvain_method&oldid=1222455388. Zuletzt abgerufen am 13. Mai 2024.

[8] o.V. Breadth first search or bfs for a graph, Apr 2024. URL https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/. Zuletzt abgerufen am 13. Mai 2024.

[9] o.V. Networkx documentation, 2024. URL https://networkx.org/. Zuletzt abgerufen 28. April 2024.

[10] o.V. What is a knowledge graph?, o.J. URL https://www.ibm.com/topics/knowledge-graph. Zuletzt abgerufen am 13. Mai 2024.

[11] o.V. Funktionsweise des knowledge graph von google, 2024. URL https://support.google.com/knowledgepanel/answer/9787176?hl=de. Zuletzt abgerufen 25. April 2024.

[12] o.V. Cities knowledge graph, o.J. URL https://fcl.ethz.ch/research/research-projects/cities-knowledge-graph.html. Zuletzt abgerufen 25. April 2024.

[13] M.L. Zeng and J. Qin. *Metadata*. American Library Association, 2020. ISBN 9780838948637. URL https://books.google.de/books?id=Y5W1EAAAQBAJ.

[14] T.H. Kolbe. Geoinformatik: Vorlesung 15: Geodateninfrastrukturen, 2022. URL https://www.moodle.tum.de/pluginfile.php/3804341/mod_resource/content/0/Geoinformatik%202%20-%2015%20-%20Geodateninfrastrukturen.pdf. Zuletzt abgerufen 23. April 2024.

[15] R. Jenn. Understanding metadata. *Washington DC, United States: National Information Standards Organization*, 23:7–16, 2017. URL http://www.niso.org/publications/press/UnderstandingMetadata.pdf.

[16] M. Knezevic, A. Donaubauer, and T. H. Kolbe. Sddi - minimal ecosystem for the establishment of urban digital twins. *Publikationen der DGPF*, 32:151–168, 2024.

[17] R. Albertoni, D. Browning, S. Cox, A. G. Beltran, A. Perego, and P. Winstanley. Data catalog vocabulary (dcat) - version 2, Feb 2020. URL https://www.w3.org/TR/vocab-dcat-2/. Zuletzt abgerufen am 13. Mai 2024.

[18] B. Willenborg S. Bobinger A. Donaubauer, M. Knezevic and M. Lutz. Leitfaden – urbaner digitaler zwilling nach der methodik der sddi. 2023.

[19] M. Knezevic, A. Donaubauer, M. Moshrefzadeh, and T. H. Kolbe. 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, 2022. doi: 10.5194/isprs-annals-X-4-W3-2022-119-2022. URL https://isprs-annals.copernicus.org/articles/X-4-W3-2022/119/2022/.

[20] Raga'ad M. Tarawaneh, Patric Keller, and Achim Ebert. A General Introduction To Graph Visualization Techniques. In Christoph Garth, Ariane Middel, and Hans Hagen, editors, *Visualization of Large and Unstructured Data Sets: Applications in Geospatial*

Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011, volume 27 of *Open Access Series in Informatics (OASIcs)*, pages 151–164, Dagstuhl, Germany, 2012. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-939897-46-0. doi: 10.4230/OASIcs.VLUDS.2011.151. URL https://drops-dev.dagstuhl.de/entities/document/10.4230/OASIcs.VLUDS.2011.151.

[21] Michael Kaufmann and Dorothea Wagner. Drawing graphs. *Lecture Notes in Computer Science*, pages 17–22,249, 2001. doi: 10.1007/3-540-44969-8.

[22] Max Franz, Christian Lopes, Dylan Fong, Mike Kucera, and Gary Bader. Cytoscape.js documentation. URL https://js.cytoscape.org/#layouts. Zuletzt abgerufen 09. Mai 2024.

[23] Martin Grandjean. Introduction à la visualisation de données : l'analyse de réseau en histoire. *Geschichte und Informatik*, 18/19:109–128, 01 2015.

[24] Ji Soo Yi, Youn ah Kang, John Stasko, and J.A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007. doi: 10.1109/TVCG.2007.70515.

[25] o.V. Ckan, o.J. URL https://ckan.org/. Zuletzt abgerufen 28. April 2024.

[26] o.V. introducing dash, o.J. URL https://medium.com/plotly/introducing-dash-5ecf7191b503. Zuletzt abgerufen 28. April 2024.

[27] o.V. Dash cytoscape, 2024. URL https://dash.plotly.com/cytoscape. Zuletzt abgerufen 28. April 2024.

Abbildungsverzeichnis

Tabellenverzeichnis

Digitaler Anhang

- Digitale Version dieser Arbeit im PDF-Format
- Python Projektordner der eigenen Schnittstelle