TECHNISCHE UNIVERSITÄT WIEN

Understanding Relevance in Maps through the use of Knowledge Graphs

MASTERARBEIT

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Cartography

eingereicht von

José Pablo Ceballos Cantú
Matrikelnummer 12042494

ausgeführt am Institut für Geodäsie und Geoinformation der Fakultät für Mathematik und Geoinformation

der Technischen Universität Wien
Betreuung: Dr. Markus Jobst

Wien, 9. September 2022

José Pablo Ceballos Cantú	Markus Jobst

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-12601 ▪ www.tuwien.at

TECHNISCHE UNIVERSITÄT WIEN

Understanding Relevance in Maps through the use of Knowledge Graphs

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Cartography

by

José Pablo Ceballos Cantú
Registration Number 12042494

Completed at the Department of Geodesy and Geoinformation of the Faculty for Mathematics and Geoinformation at the Technical University Vienna

at the TU Wien
Advisor: Dr. Markus Jobst

Vienna, 9th September, 2022

José Pablo Ceballos Cantú | Markus Jobst

Cartography M.Sc.

Master thesis

Understanding Relevance in Maps through the use of Knowledge Graphs

José Pablo Ceballos Cantú

Technical University of Munich
TECHNISCHE UNIVERSITÄT WIEN Vienna University of Technology
TECHNISCHE UNIVERSITÄT DRESDEN
ITC UNIVERSITY OF TWENTE.

2022

Statement of Authorship

Herewith I declare that I am the sole author of the submitted Master's thesis entitled:

"Understanding Relevance in Maps through the use of Knowledge Graphs"

I have fully referenced the ideas and work of others, whether published or unpublished. Literal or analogous citations are clearly marked as such.

Vienna 09.09.2022, José Pablo Ceballos Cantú

Erklärung zur Verfassung der Arbeit

José Pablo Ceballos Cantú

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 9. September 2022

José Pablo Ceballos Cantú

Acknowledgements

I would like to thank my advisors Markus Jobst and Georg Gartner for their guidance and advice. I would ever be grateful to Juliane Cron who supported me throughout my cartographic journey, without her help my participation in this program would have not been possible. I am thankful to Markus Heiler and Zoltan Daroczi who allowed me to grow professionally. A great thank you to all my friends and family, who provided me every day with the support I needed to overcome the different challenges that I faced these last 2 years. Thank you Nele, Claudia, Genaro, Juan Carlos, Ana Paula and Vivien.

Abstract

The relation that the ontological and spatial dimensions have is not always evident. To better understand this relation, a tool called *SeMaptics* that connects the two domains were developed. Ontological mapping allows for discrete ontologies to be projected into the spatial field. Such ontologies are regularly seen in a continuous or overlapping layered format in the spatial dimension. However, integrating both spaces results in a novel method, which adds additional perspectives when designing a map.

The objective is to link and map both dimensions enabling user interactions, and allowing for proper semantic probing. This exploration contains some criteria of geographical relevance. This criteria allows for better identification of high-interest areas within any given dataset.*SeMaptics* implements graphDB Neo4j to accommodate a graph architecture and visualization JavaScript library D3js to render knowledge graphs on the screen. Visual representations of knowledge graphs are limited by performance as the data pool grows. In this work, an isolated administrative unit in Austria is analyzed to have a balance between semantics and data quantity. Criteria that comprise geographical relevance are over-viewed in this style of mapping to understand if it is the use relevance when exploring spatial data from a semantic viewpoint.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The ontological space presents several properties that cartography can take advantage of such as discretizing, connecting, and encapsulating semantics. The spatial domain is one of continuous fields and variables which can or can not be discrete[JT22]. A present challenge for map makers is to communicate effectively map semantics. Understanding semantics in such a dimension proves at times challenging for both humans and machines. Better integration of the semantic field into the spatial domain can enhance our ability to identify hidden semantics and behaviors of our mapped surroundings. As a result, we can better understand the data within a context to better design maps, enhance distribution services or increase performance on digitally based maps. There is a wide field contained within maps that implement knowledge graphs that is yet to be explored. Showcasing a simple tool that allows us to see the connection between the spatial and the ontological dimension should help us better understand the semantic relevance of data.

1.2 Thesis Outline

- **Chapter 2 Research Identification**: A presentation of the research objectives, research questions, hypothesis, and explanation of how this thesis contributes to cartography.

- **Chapter 3 Relevance**:Relevance and its different elements and types. The types covered in this chapter are i) physiological relevance as presented by Wilson (1986) and Harter (1992)[Har92][SW86]. ii) Logical relevance per Cooper (1973)[Coo73].iii) Situational relevance from Wilson's (1973) perspective. iv)Geographical relevance presented by Reichenbacher and Sabbata(2011 & 2012)[dSR12][RDS11]. The criteria that comprise geographical relevance and the evolution of such criteria are covered.

  Relevance, maps, and communication are interconnected and thus a brief overview of the role that neurocognitive relationships and the elements of the cartographic transmission model are discussed [Ken18] [SZRM07]. Section 3.2 provides insight in how *SeMaptics* as visual tool that communicates semantics influences relevance. Additionally, relevance is tightly coupled with the information extraction field. Inferring semantics has an underlying information extraction process following Boyce's (1982) and others' relevance is presented under the scope of information extraction [Boy82].

- **Chapter 4 Knowledge Graphs & Knowledge Networks**: Knowledge graphs are fundamental to the semantic web[MHC+22]. In this chapter, the basic elements of a knowledge graph are presented through the triplet model[MHC+22] and the subject, predicate, and object model as per Nickel (2016)[NMTG16].

  Knowledge Networks(KN) stem from decentralized linked data [Biz09]. Efforts to delimit concepts with KN in the form of Resource Description Framework Schema(RDF) as presented by the WWW3. This standardization allows for both humans and machines to understand linked semantics. KN in cartography is confined to 3 dimensions thematic, topographic, and connected [JG22]. How we integrate these fields in cartography will define categorically related spatial data infrastructures (SDIs), to formalize exchange and inter-usability[HKMH20]. Such standardizations include the use of geospatial vocabularies through the standard of GeoSPARQL.

  Knowledge graphs in data visualisation have 3 main variables *interaction*, *users* and *task*[MGP+04]. These three variables subdivide data visualization into 4 levels. How the Semantic web and other knowledge networks affect the geovisualisation field reflects on evolving standards such as the ones proposed by the World Wide Web Consortium(WC3). Graph visualisations are built following 4 main steps *data retrieval, building, calculations, layout* and *rendering*)[GRMSOG18]. The different types of knowledge graph layouts have advantages and disadvantages depending on different scenarios where navigation and interaction affect user preference and use.

Mainly visual knowledge graphs in cartography have been used to represent *flow maps, sketch maps* and *network maps*. The way a network is presented is defined by its topology, spatial accuracy, and connectivity[RA97]. Narrative cartography has found support in knowledge graphs to construct maps and define standards[MHC+22].

- **Chapter 5 Popular Technologies that surround Knowledge Graph and Knowledge Network systems**: All technologies related to knowledge graphs such as graph databases. These databases are schema flexible and follow the SPO model. The query language they use defines them. The three most popular languages for graph databases are SPARQL, Cypher, and Gremlin[Ang17]. These languages conform to a declarative and graphical structure. They also support existing database operations such as UNION, WHERE, FILTER, JOIN, and others. The popular graph databases on the market at the moment are AllegroGraph, ArangoDB, InfiniteGraph, and OrientDB. All databases support different query languages and have use cases where they perform better, whether it is scaling, learning curve, readability, deployment, and distribution. These performance points have defined the level of adoption that each one of these databases has. In the cartographic field, these databases are popular in the use of mobile cartography, digital river networks, and augmented reality navigation among others.

  Additionally, technologies relating to knowledge graphs subdivide into web browser-based and none web-browser based. The web browser offers cross-platform compatibility, while native technologies might offer better performance and allow for native optimization.

- **Chapter 6 Finding Relevance Through the combination of Maps, Knowledge Networks, and Web-based Knowledge Graphs**: Methodology used in this project alongside justification and hypothesis. A presentation of *SeMaptics*, the tool developed for this thesis, and the capabilities it showcases as part of the results.

  SeMaptics was made using the four main steps on graph visualization *Data retrieval, building, calculations, layout* and *rendering* proposed by Romero (2018)[GRMSOG18]. The data retrieval stage is comprised of collecting data from the Austrian Federal Office of Metrology and Surveying (BEV) and converting tabular data into a graphical structure on a Neo4j instance. The building processes follows a single-page application SPA structure. The SPA approach allows for flexible and scalable tools that require less time to develop, deploy and maintain. The main visual components are upgraded versions of Typescript of D3js and OpenLayers.

  The result is a tool that projects data to the spatial dimension from the ontological dimension with a one-degree connection. Interactions are fluid in both visualizations, with operations such as selection and drag being included. Simple readable legends are rendered on top of each node to support human readability of the network.

Different types of graphical visualizations are supported in *SeMaptics* such as Force-direct and disconnected-force graphs.

- **Chapter 7 Conclusion & Discussion**: Discussion on the results, challenges, and limitations of *SeMaptics*. Future work and final thoughts on this topic.

  It was found that graph visualizations add a level of understanding of the data and that could stimulate creative mapping. Important characteristics such as i)Ontology definition ii) Data harmonization, iii)Query language selection, iv)graph visualization selection, and v)Interaction design play a role in how a user interacts and perceives the presented semantics. Cluster identification on graph visualizations allows for quick pattern identification. Data navigation is made much easier since some of the data have no apparent spatial relation. The ontological space allows for all data to be displayed on a single layer, something that is not possible in the multilayered spatial dimension. Improvements can be done to explore the impact that they have on relevance. These include categorization, multiple degree selection, zooming, panning, and a dynamic state that changes according to interactions.

  In conclusion, the connection of both dimensions reveals hidden patterns and allows for efficient semantic reads. Several criteria points from geographical relevance are met such as *depth, specificity, availability, accuracy, tangibility, accessibility, dynamism, curiosity, spatial proximity, visibility, cluster* and *co-location*. This research leaves room for many areas to be explored. These areas include experimenting with the impact of implementing: different graph typologies and connectivity, cardinality implementation, cross-domain data integration, additional graph types, and understanding more semantic reads and aesthetics.

CHAPTER 2

Research Identification

2.1 Objectives

- Understand how the mapping of the ontological dimension projects relevance into. the spatial dimension.

- Present a proof of concept, which can be build upon to explore relevance that originates from the ontological dimension and projects itself to the spatial dimension.

2.2 Research Questions

- Does a web-based map tool, which contains a linked ontological and spatial dimension, enables geographical relevance criteria to be identified within in the ontological and spatial dimension?

- Which criteria from geographical relevance can be asserted from a web-tool which shows linked ontological(knowledge graphed space) and spatial dimension(map)?

2.3 Hypothesis

The connection of the ontological and spatial dimensions, while displaying both spaces and adding based event interactive filtering features, will allow us to explore new ways to understand relevance related to spatial features.

2.4 Contribution

This thesis explores the use and combination of maps, knowledge networks, and knowledge graphs taking data visualization and database implementation. Through the linkage and combination of these, the objective is to see the unseen and bring relevance forward. Popular-themed cartography is focused mainly on layered visual classification, where ontological relations are often overlooked. The reasons for this may vary. It could be that cartographers are not exposed to the ontological relations that are embedded in the data represented by the maps. The lack of maps that map ontologies or that the tools used in map-making lack the means to create such maps. Relevance is an appropriate topic that satisfies the need for maps that originate from an ontological dimension. The main goal is to help understand the relevance in the ontological dimension and project such relevance into the spatial dimension.

CHAPTER 3

Relevance

3.1 Brief Introduction into relevance

Relevance is a notion that people use without defining it. One definition given by Saracevic (1996) of relevance is The basic inner and outer human cognitive notion in constant use when we have a present matter. Its pragmatic nature and core are evident to us, yet we struggle to define it. Relevance is an action to access, filter, infer rank, accept, reject and classify information. Even though its pragmatic application and nature are evident to us its strict definition, fundamental particles, and classification type require a lot of research [Sar96b]. Since the sixties, there have been efforts in academics to understand the underlying nature of relevance. However, Sabbata states that relevance is still a not well-defined concept and that relevance subdivides itself into many types[SMR15].

Relevance as presented by Wilson (1973) is not a fundamental concept. But, as a composition of different fundamental relevances [Wil73]. A type of relevance mentioned by Wilson is psychological relevance, which focuses on how people use the information and how their views change after such information has been received. On the other hand, Wilson mentions logical relevance, a type of relevance where the level of consequence the received information has on an individual's personal view is subject to criticism. He states that logical relevance is often overlooked and ill-identified. An example he provides is the realization of valuable information that was available all along. But for any given reason was failed to be identified as relevant. Wilson defines logical relevance as, *"(...) the concept of a relation between an item of information and a particular individual's personal view of the world and his situation in it, and it is a concept in which relevance depends on logical bearing on some matter on which he has preferences."* [Wil73]. This definition bears importance because it states the relationship between the items and the individual's personal views at the center, showing that even relevances which are considered fundamental are subject to the individual's perception of reality. This concept resurfaces later on in the cartographic communication model proposed by Kolacny where the perception of reality by both the user and the cartographer affects the information transmission process[Li20]. Researchers such as Sabbata (2015) mention that relevance in terms of information extraction is the connection between the information available to the individual and the needs of the individual[SMR15]. Such ideas tie better to a static concept of situational relevance introduced by Wilson(1973).

In the following sections, the main aspects of relevance that are considered relevant in cartography will be covered. However, relevance is a vast well in which much research happens. I would not exempt myself from being tied to situational relevance, consequently omitting relevant research.

3.2 Communication & Relevance in Cartography

Since ancient times maps have been used as an interface of communication before written language was commonly used [UMS+09]. Maps are communication gateways in which spatial information concerning spatial ontologies transmits from the map maker to the map user. Maps are communication systems [Li20]. Cartography is the art and science that provides meaning to spatial relationships transforming space [Ken18].

The meaning a space receives through a map is always subject to the map maker and the user. The two parts inherently have interests that they project on the map. Such projection is the cartographic information transmission model [Ken18]. This model has three core elements: the cartographer, the map user, and the reality[SSJK02]. Both the cartographer and the user have different perceptions of reality. The cartographic community extended such a concept with new models accounting for the loss of information. Since maps are communication systems, terms were coined for these models such as encoding, decoding, and transmission [RP95]. However, these terms are not novel. They have been mentioned in the code model since the time of Aristotle[Sar96b]. These components are contained in what Sperber and Wilson defined as the improved inference model. A model based on a high payoff low processing effort system, where relevant information available receives more attention from the individual [WS95]. Sperber and Wilson propose two principles in relevance. The first oscillates around the concept of high payoffs for low processing efforts through cognitive processes. The second states that relevance is always clearly communicated[Sar96b]. Novel to this theory is the relationship between communication and information retrieval for high payoffs.

Swienty (2007) presented the neurocognitive model that a geovisualisation instantiates. This model splits into two paths. The *M-Path* & *P-Path* each of these is responsible of different tasks. The *M-Path* takes care of the spatial inference and the *P-Path* takes care of defining what the user is seeing. Swienty presents a table that shows graphical variables and neurocognitive pathways belonging to the neurocognitive model. Some factors such as contour size, shape, resolution, crispiness transparency, saturation-achromatic contrast, spatial depth disparity, and semantic motion signals belong to both pathways. However, size, contrast, velocity, and motion direction are related only to the *M-Path*[SZRM07]. Swienty argued that there is a need to create a geovisualisation that is relevance driven.

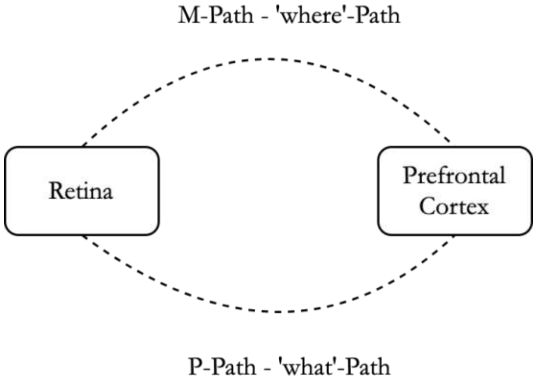


Figure 3.1: Neurocognitive relationships [SZRM07]

3.3 Information Retrieval and Relevance

Information extraction and relevance are tightly coupled. A drift exists between the definition of relevance within the information retrieval field and other fields that deal with relevance. For this reason, a general view of how information retrieval defines and uses relevance has been included. Other definitions of relevance, which are covered in the following sections, also include the information retrieval point of view. Information retrieval and relevance relate through topicality[Boy82]. The idea is that relevant requested information from an individual relates to topicality. This topicality acts as a binder between relevant and non-relevant information. However, Boyce thought that this was not necessarily true. He proposed a two-stage model where topicality is taken into account in the first step. But afterward, a matrix of topical probabilities is made to rank information with the highest conditional probability given a representation[Boy82].

Information retrieval, which started as a static task later evolved into a more interactive process. However, information retrieval does not reflect interaction [Sar96b]. There exist three proposed models known to me for information retrieval where relevance is treated ever so differently. Firstly, the *cognitive model* proposed by Ingwersen (1996). Here interactions are viewed as a set of cognitive processes under the justification that the user interacts with texts, which are defined as cognitive structures. Relevance in this theory is at the center of cognitive interactions from the users[Ing96]. Secondly, the *episode model* by Belkin (1995) where interactions are a separate set of episodes and relevance may be defining which type of interaction the user chooses to make[Bel95]. Thirdly, a model proposed by Saracevic (1996) called the *stratified model of IR interaction*. This model follows a 3 step foundation of *Acquisition-Cognition- Application*[Sar96b]. This approach separates human interaction into a cognition level, where relevance participates through intent pattern definition. From the computer side, an input or process level exists. At this point, the computer tries to identify its relevance. An interface exists which connects both participants[Sar96a]. Saracevic mentions that since humans made such systems and they made relevant assumptions, there are more cognitive structures at play[Sar96b]. With this insight, Saracevic explains that there is a plurality of relevances in the information retrieval field. This plurality is also mentioned in Wilson's (1973 & 1995) work. The novelty of Saracevic's work is that he presents what he calls a *strata*. These strata of interactions encapsulate different relevances which are independent of each other. Hence he concludes that we can not choose one type of relevance and ignore the rest [Sar96b]. This gives birth to a definition of relevance seen under the scope of information science. Saracevic (1996) defines such relevance as the indicator that shows how effective interaction between two parties, a user and an information system, is [Sar96b].

A proposed relevance criteria list was given and formalized by Barry & Schamber (1998). Their published list criteria included: depths, scope, presentation quality, geographic proximity followed by accuracy and validity (Object & Subject), content Novelty, tangibility, affectiveness, verifiability, and consensus among other criteria. This list was based on all past research at the time. Schamber's pointed out that geographical proximity was identified as relevant when categories such as weather or currency were in play[IS98]. Both Sabbata & Reichenbacher expressed that this was a list that was founded on a general perspective[dSR12]. Consequently, there was the possibility to use the same criteria within geographical relevance. This criteria list will be explained in further detail later on.

3.4 Types of relevance

In this section, the most prominent relevances mentioned in literature will be covered. However, this field is vast and has a long history of research. It is most likely that some types of relevance might be missed. It is encouraged that complementary read and research is done besides what is presented in this chapter.

Physiological relevance is one of the earliest relevances to be researched and defined. Initially presented by Sperber & Wilson (1986) and later on named by Harter (1992). Physiological relevance is the relation between a premise and its context. Where two elements from this framework must be satisfied [SW86].

- *"1) An assumption is relevant in a context to the extent that its contextual effects are large. [Har92] "*
- *"2) An assumption is relevant in a context to the extent that the effort required to process it is small. "*[Har92]

Understanding these two premises of physiological relevance. Harter states that we can consider it as a comparative concept. He explains that since Sperber & Wilson's proposal, there has been a gap in the concept definition. There is the idea that the individual will try to extract the most amount of value from any given piece of information, thus maximizing its contextual value. Even though we have discussed that these authors regard relevance as a comparative concept, psychological relevance can be classificatory when considering its value at a defined time within a defined context[Har92]. Saracevic (1996) felt that a more appropriate name for physiological relevance would be cognitive relevance. However, the naming has prevailed. A shortcoming also identified by Saracevic is the challenge of putting our thoughts into discrete verbal concepts. Hence, using language as a way to identify relevance is not advisable. Also, it restricts itself to project information and changes according to received answers, making any a-priory relevance obsolete [Sar96b].

Logical relevance as defined by Cooper (1973) is always dependent on the consequence of any proposed statement [Coo73]. The possible answers to any given statement are defined as component statements. Logical relevance focuses on the relationship between the initial statement and component statements. From this definition, Wilson explains that a piece of information holds its weighted relevance proportionally to the probability of confirmation of the conclusion, based on the relation that the information has between the proposed statement and the conclusion. If information holds no impact on the probability of confirmation of such a conclusion can then be considered irrelevant. Then situational relevance is at the forefront of relevance. Relevance is subject to the individual's perceived reality and not to the perceived reality of him by others or reality itself, if there is such a thing. Consequently, statements are formulated in a categorical order. The order is pertinent to the desired answers of the individual, which is a situational description.[Wil73]

Situational relevance is a core relevance Wilson(1973) places relevance as a consequence of the individual's situation and preferences. The dynamic concept within this relevance is that information can become relevant when added to the individual's knowledge stock. This relevance depends on the individual's value and preference stock. However, Wilson does not explicitly state what the individual's knowledge stock encloses. Situational relevance, as the name suggests, encircles information based on situational concerns. Some main drivers of situational relevance are preference, interests, time, degrees of relevance, and completeness. Situational categorizes information based on these factors, which act as situational to any given individual and are evaluated through a logical relevance lens. Wilson argues this type of relevance is a strong notion for system design that supplies information through an analysis that determines if a piece of information is situationally relevant to a system user or not. Therefore, a challenge would be to accurately project the user's views and concerns into the system[Wil73].

3.5 Relevance in Cartography

Cartography has a second side to relevance. Most relevant papers link relevance with information extraction or decision-making. However, in cartography, there are other relevant issues that relevance solves. Filtering is an important process/service in modern cartography. There has been a lot of interest in filtering objects on mobile maps due to the limited cognitive capacities of humans. Hardware limitations on over-rendering are aided by filtering as well. There has been an identified need to render and communicate efficiently. This presents a challenge since mobile map makers wish to communicate effectively and maintain map performance. On top of this, there is a need to be able to identify relevant information from an overwhelming well of data that ubiquitous modern mobile GIS provides. Geographic relevance has been the answer to some of these challenges [RDS11]. In geographic relevance, all entities in a geographic space have a quality attribute. The quality is the relation between the representation of such an entity and the use context. Such objects have a relevance that is contained in differing levels of intricacy. There are four levels to this relevance. The base level is relevance contained within discrete objects. The second level contains attributes related to such objects and/or their relationship at a: spatial, temporal, and spatio-temporal degree. The third level is the relevance found within groups such as clusters. Lastly, the fourth level is the relevance held as a function of such objects and places. [RDS11]

Geographical relevance has been related to situational relevance by Sabbata, under the argument that the relevance contained within geospatial features, projects the relevance of a real-world entity or event[dSR12]. However, Sabbata states that the current understatement of geographical relevance is to evaluate the relevance of any given object within an information system. As explained before, relevance is a dynamic state that depends on the user's interests and temporality. This relevance impacts how the map user interacts with the map. Discussion around how to properly relate geographical relevance and relevance that concerns information retrieval exits[dSR12]. There was an understanding that relevance was tightly coupled with topicality in information retrieval. Therefore, Euclidean distance could be used as a form of spatial topicality. However, there has been debate over how to go about this linkage[dSR12]. Additionally, topicality in information retrieval may require secondary stages to deliver relevant information[Boy82].

Considering the list proposed by Barry & Schamber(1998) and mentioned in the Information relevance section. The criteria relation to geographical relevance was made by Reichenbacher & Sabbata (2012) under the argument that this list could be also used as criteria for geographical relevance. The criteria stated were *depth, scope, specificity, availability of information, sources of information, effectiveness, accuracy, validity, clarity, currency, tangibility, reliability, quality of sources, accessibility, and verification*[dSR12]. Reichenbacher & Sabbata explain why each of the elements of the list can be used by geographical relevance and why are they valuable. Some of the criteria added were: *currency* where it shows how much the information regarding any entity is up to date.*Accessibility* relates to the fact that information might have some effort or cost to obtain. *Affectiveness* is the level of emotional response to any given information. Another criteria that Reichenbacher & Sabbata added through Schamber's work was *dynamism*, which refers to how alive or active the information is. An example would be zooming and graphical interaction[dSR12]. More elements in the list are contained. The criteria list was later extended with time. *Novelty* refers to information newness was also included as criteria. In the early 2000s, there was more focus on web relevance where the criteria list was extended by Reichenbacher & Sabbata. This new criteria where *familiarity,variety and curiosity*. Finally, both researchers explain that personalized relevance is in play, which they define as *appropriateness*[dSR12].

Relevance in GIS has also been the subject of research. Two core criteria are usually scoped. Those being *spatial* and *temporal proximity*[dSR12]. *Spatial proximity* is defined as the distance between entity and user. The same applies to *temporal proximity*. But concerning the entity's temporality and the temporality of the user. Other studies on the mobile realm argued that spatial proximity was a dynamic factor since individuals could trade space for time[MM07]. This concept plays into accessibility so Reichenbacher & Sabbata brought up the *spatio-temporal proximity* as a criteria for relevance. The mobile studies from Mountain (2007) made also possible for Reichenbacher & Sabbata to include *visibility* and *irrationality* as criteria of relevance in GIS.

The geographical environment is influential to the judgment of relevance and definition context. In such a context some fundamental geographical criteria that relate to any set of entities exist. This relation can manifest itself in the form of hierarchy or spatial hierarchy or through a cluster-type relation. Both, clustered and hierarchies, are judged under the relevance scope. For cluster size and shape shows relevance. For hierarchical entities, geographical units represent relevance. Other relevant factors that define relevance in the geographical environment are *co-location* & *association* which can be constrained by temporal factors[dSR12]. In their research Reichenbacher & Sabbata(2012) found that as long as *spatio-temporal proximity* and *topicality* where satisfied *co-location,cluster and hierarchy* where important criteria for users to define relevance in geographical entities. Contrary to what Mountain (2007) found in his mobile study where availability was a core criterion to define availability. Perhaps here is a good example where situational relevance plays a role in the results of these studies.

There is a proposed discrete way to measure graphical relevance proposed by Reichenbacher & Sabbata (2016). Using the criteria list previously defined in the field, some of which have been presented in this section, they account for *semantic distance, mobility, and geographical environment*. Each of which spans over some criteria. *Mobility* takes care of spatio-temporal proximity and directionality. *Geographical environment* takes care of cluster-related relevance and co-location. The idea behind this calculated geographical relevance was to compare it to the relevance given by regular users to see if the score matched the user's given relevance. They concluded that their estimation had a positive correlation with human relevance. They found, however, that their proposed model performed better in scenarios where spatio-temporal proximity approximated spatial distance. Even though they had a positive correlation on their model they concluded that there was still space to improve, to assess and estimate human relevance better. They also acknowledge that geographical relevance is expressed through multilayered relationships where geographical relevance is strongly attached to topicality and spatio-temporal factors concerning the user.[RSPF16]

Relevance in digital cartography can be affected by hardware and software variables, such as display size and displayed data amount, influencing the cognition and geographical relevance of the user[SZRM07]. In mobile cartography, Oliveri & Reichenbacher (2021) studied the impact graphical variables have on relevance. This elements where *hue , value and transparency*. The approach was to have subjects draw their attention to the screen. The participants' eyes were tracked in a free roam scenario. There was no immediate impact on what users perceived as relevant. However, in a second scenario where users were asked to complete a task, they found that transparency yields good results to draw attention[OR21]. Suggested future work by them was to add geographical encoding and test more graphical variables. In geovisualisation relevance has been treated as a two-part system of objective and subjective relevance[SZRM07]. The objective relevance is the direct result of any algorithmic determination and the subjective relevance is the influencing factor. Subjective relevance is the interface in which a system generated geovisualisation and the user interact[SZRM07].

CHAPTER 4

Knowledge Graphs & Knowledge Networks

4.1 Understanding Knowledge Graphs & Knowledge Networks

4.1.1 Brief Introduction to KG and KN

Ontologies are one of the fundamental units in a semantic web. Ontology is a blueprint to represent knowledge. Ontologies are the description of concepts and relations using deterministic structures. The construction of a determined semantic space is known as a domain.[MHC+22]

A standard definition of a Knowledge Graphs KG states that statements in the context of the Semantic Web can be expressed in the form of a triple (h,r,t), the head entity (i.e., subject), the relation (i.e., predicate), and the tail entity(i.e., object)[MHC+22]. Processing data using KGs is known as an ontological approach. A challenge that the ontological approach faces is the ambiguities contained within natural language. Formalism is required to map semantic spaces to strive for well-defined heterogeneous domains[HMBB17].

Knowledge Networks(KNs) stem from decentralized *linked data* [Biz09]. Defined as the use of typed links between data from different sources[Biz09]. The distance between these sources is irrelevant. The sources may be projected in the spatial dimension or it might be data held in the ontological dimension. Data is identified on the web via Union Resource Identifiers (URI´s)[Biz09]. URIs operate within Web of Data – Resource Description Framework Schema(RDF) technology, which is also based on a triplet structure as the one explained for KGs. This aggregation of linked data is a knowledge network [MBK]. KN aid in the creation of semantic interconnected spaces, to achieve such spaces metadata must be included in the different spatial data infrastructures (SDIs)[HMBB17].

4.1.2 Basic Elements on Knowledge Graphs & Knowledge Networks

Knowledge has the property to be surrogated by a representation[DSS93], which helps us reason knowledge. In any given knowledge graph we can use a Node, Relationship structure. This falls in line with the subject, predicate, Object (SPO) model. A relationship is born out of the link of two ontological objects, which are represented as nodes. When various SOP triplets combine they form a graph system. The directionality of the edge is what differentiates a subject from an object. The construction of a KN leads to accurate results. In these systems, we have the advantage of matching unique ontologies on a space to create a KG. KGs are known to have deterministic properties such as transitivity. Transitivity shows a logical path that knowledge follows[NMTG16]. An example of transitivity would be a node of type "park" that points to the year 2022 as its foundation, respectively 2022 points to the century 2000-2100. Hence, we can easily infer that such a park was found in the twenty-first century. The existence of statistical patterns in a KG is known as *homophil (autocorrelation)*. Entities tend to be related to similar entities.[NMTG16]

Figure 4.1: SPO model

According to Nickel (2016), There are different ways how to build a knowledge graph. First, through a curated set of SOPs handpicked by experts. Second, using crowdsource or group volunteer collaboration (Wikipedia). Third, automated approaches, extract triplets from structured data. Fourth, automated methods from unstructured data such as language. Nickel explains that even though a curated approach brings on the most accurate results, using this technique does not scale well. All data from a KG is stored in a Knowledge Base(KB). The main advantage of KGs is that semantic information stored exists in computer-readable structures, which enhances the intelligent capacities of computer systems. However, the quality and the trust worth of a knowledge graph are only as good as the source it comes from.[NMTG16]

```
Relationship {
        identity: Link ID,
        start: Start Node ID,
        end: End Node ID,
        type: definition of relation
    }
Node {
       identity: Node ID,
       labels: String[],
       properties: {}
    }
```

For KNs some basic metrics help evaluate them. Some of these are *the average degree of nodes, the average local cluster, the global cluster coefficient, and the number of connected components*[MBK]. There are more metrics in which a knowledge network can be measured. Additionally, RDF properties can be analyzed within data stacks depending on the sources. Mangaladevi (2017) found diverse KN structures through the measure of the basic metrics in combination where each property of a data structure (RDF) including spatial topologies resulted in different network structures and properties. The upward trend in Semantic Web ontologies is because tasks such as storing, publishing, retrieving, reusing, and integrating are supported. Standardized vocabularies such as Ontology Web Language (OWL) enhance RDF triplets[GRMSOG18].

4.2 Knowledge Graphs & Networks in Cartography

Three dimensions have been presented in KN cartography: thematic, topographic, and connection. The thematic dimension has the problem of informational structures, which demand hierarchy and are diverse. Combining such structures has always been a challenge in cartography. A concrete example is abstract administrative units represented in cadastral data and their relations. Semantic abstraction using KN benefits the implementation of this dimension on maps. The topographic dimension which is the host for the real-world referencing point acts as the centerpiece that benefits from networks for data processing and distribution. The connection dimension is the support of a common data space where the enablement of relation allows for the implementation of multi-layer systems, IT architect design, understanding of API mapping, and identification of redundancy. Defining a standardized ontology to topographic elements to provide information security has stagnated for some years now[JG22]. This study brings forward relevant areas of need in modern cartography. Where ontological standardization implementation has been set aside, due to the fact of questioning the relevance of doing so. More concrete examples of applied semantic web ontologies in cartography are needed to be made to demonstrate the relevance of semantic definition and standardization.

Our world moves into ubiquitous Location Base Services(LBS) systems. Knowledge bases become a viable service that cartography can rely on. Narrative cartography is a field of cartography that benefits from KGs & KBs. From a visualization perspective using KG can be projected into the spatial dimension. Crowdsource KG such as Wikipedia has been mapped to understand the process and also as an additional step to evaluate the accuracy of the data. KGs have not been properly exploited to generate geovisualizations. The geospatial domain is no different from other domains. There is a long-standing need to understand it. Consequently, maps have been subject to knowledge formalization to understand the underlying semantic structures within maps. A cartographic module fitted for narrative cartography has been proposed by Mai (2022) to better define the geospatial domain. This domain contains ontologies of different natures such as object, event, Spatio-temporal extent, Spatial-Extent, Temporal-Extent, map content Item, and map content-type. The relation of such ontologies is defined in predicates of type boolean (*'is of type', 'has property'*). However, also predicates that explain the nature of such ontologies can be used (*according to whom*). The KG cartographic module serves as an initial guideline to map the semantic space in the geospatial domain. We can even add styling features as ontologies to also define the styling domain as a subset of the geospatial domain. Ontologies such as feature-type-styled, symbol, and legend-item. stroke, fill and others can be graphed in the style domain. All these ontologies are connected through boolean predicates of type consists-of, has-symbolizer, has-stroke, and so on. The proposed cartographic module by Mai proves to be a powerful tool to define cartographic spaces on the semantic web domain.[MHC+22]

The challenge of mapping the geospatial domain divides itself into two main areas of application. The first is geospatial data integration. A well-defined problem is anointed to the spatial data infrastructures (SDIs). SDIs formalize data integrations. Since a standard allows for better data exchange and inter-usability[HKMH20]. Geoprocessing is another field that would benefit from semantic web integration[HMBB17]. Here ontologies are defined as process points and fed to geoprocessing tools as operators[HKMH20]. Since knowledge graphs have transitivity properties workflow automatizing is possible. This exercise has been explored by Huang (2020) on bicycle roads. Huang found that it is possible to overcome data heterogeneity between different sources to integrate and process semantic data using KGs. He makes use of semantic constraints on data integration in his work to achieve cross-domain integration. The second area of application that results from mapping the geospatial web domain is geovisualisation. There is a clear boundary to what corresponds to data acquisition. However, what data to show and how to show it is a different science[HKMH20]. Establishing KGs that define the display rules would be a step closer to automatizing the map-making process.

Maps can be used as a KB. This allows for maps not only to be human-readable but also machine-readable. Since there have been efforts by the WWW Consortium (W3C) to standardize graph vocabularies, many KGs have been generated under the recommended Resource Description Framework (RDF) classes. Nodes under the general convention are International Resources Identifiers (IRIs). Ontological design for geospatial features follows a *Class* design, where such classes are instantiated and extended. This permits clear deterministic concept creation. Classes can e part of a class, consequently having *Sub-classes*. Classes can contain values and properties which define interactions and projections of the physical world. Many of the vocabularies available are non-geospatial. However, geospatial vocabularies exist such as geo:lat and geo:long. These vocabularies define spatially in an object and are under the Basic Geo vocabularies. Map semantics are represented under the RDF vocabularies. However, there is still not a fixed convention for all ontologies, and depending on the field of application-specific vocabularies may be proprietary or of hybrid nature. Since RDF vocabularies are machine-readable, some ontologies within a map can be defined without a map being visualized. This allows for other maps to be generated on demand.[VU18]

4.3 Knowledge Networks in Data distribution Infrastructures

Vocabularies that represent geospatial data through the standard GeoSPARQL, depreciated since 2021 and refactored to OCG-GeoSPARQL, such vocabularies allow the embedding of a spatial predicament in queries. It makes use of simple fundamental concepts such as feature and geometry. Other more robust vocabularies have been implemented using GeoSPARQL as a foundation, INSPIRE being an example. Metadata provides context to any dataset which is associated thanks to graph representations of geospatial datasets. Cartographic scale definition has also been modeled under graph systems to develop a vocabulary that allows for better standard geometry design Huang2020. This work is relevant to understanding the standard ontological design vocabulary behind KG cartography and the underlying stepping stones that have been laid before in KG cartography.

Evaluation of data quality across the Semantic Web is another relevant topic. The value of a web can be seen as the number of links that exist among all participating ontologies since this increases the usefulness of the data as a whole. Where linked data not only refers to lower-level ontologies but also at a meta-level. Where the linked aggregate of such heterogeneous data sources brings forth a network of knowledge. The quality of the semantic web data can be seen in a 5-star system where One-star has the lowest grade and is considered demand-driven (PDF). The two-star rating puts data into context and is available as machine-readable structured data in proprietary formats (XLSX, Esri Shapefile, MapInfo, Geodatabase). Three-star rating supports conservation around data, but in a non-proprietary format (CSV, TXT, GeoJSON, TopoJSON, KML, KMZ). Four-star rating builds capacity, skills, and networks using open web standards from

W3C (IRIs for identifiers, RDF for data model, SPARQL for querying). Five start Linked to other Linked Open Data on the web providing the content. Collaboration on data common sources. As of 2020 most of the European detests were classified on a 3-star system OlhaOstrovna2020. More projects using Semantic Web technologies should promote a shift towards an upper star system, which benefits data security [JG22] and cartographic semantics.

Traditional SDIs have developed data storage in tabular structures. However, machines are efficient at understanding these structures. Hence a field that specializes in extracting semantics from tables exists[HKMH20]. Syed (2010) proposed a simple way for systems to interpret data, where table headings are categorical ontologies and each corresponding row to said categorical ontology is an instance of such category with a specific value[SFMJ10]. This process allows for data to be understandable to machines and humans alike and has been built upon other research[Zha14].

Network Analysis on GIS also brings valuable elements to the table. Where a network is not only seen as an equal weight network topology. But also as a dynamic system, where there is a flow preference depending on the network characteristics. An example of such a network is a traffic network. These graphs are defined as value graphs or alternative graphs. GIS relies upon data models as a core piece. Such models are abstract representations of the real world. A popular standard is the relational model, which separates the data attributes and the spatial attributes of any node edge representation. Even though we can manage these representations on Relation Data Base Management Systems RDBMSs, this does not allow for relations to have their own attributes[FM06]. Network Analysis is a vast field. However, the implementation of simple but basic network operations enhances our understanding of the semantical context.

These researches do set a foundation stone for standard vocabulary and ontological definition. However, there has been limited work done on the implementation of web integrated systems that make use of semantic web technologies since the principal cartographic studies still deal with KG basic definition concepts. Incursion in cartographic KG tool creation and technological stack proposal represents a challenge since it has not been thoroughly explored. However, the stage has already been set by fellow cartographers in the community to start implementing such technologies.

4.4 Knowledge Graphs in Data Visualisation

Geovisualisation has 3 main variables interaction(I)[high(h)-medium(m)-low(l)], users(U)[public(p)-specialist(s))] and task(T)[knowledge(k)-info(i)]. These 3 variables indicate define the function of the geovisualisation, which is ranges in 4 levels exploration(I-**h**,U-**s**,T-**k**), analysis(I-**h/m**,U-**s/p**,T-**k/i**), synthesis(I-**m/h**,U-**p/s**,T-**i/k**) and present(I-**l**,U-**p**,T-**i**)[MGP+04]. Graphical representations help us understand our surroundings and the nature of complex systems better. Since actual systems are complex, dynamic, and ubiquitous and such systems present SPO structures. Graph network type representations have been sought[DR22]. Popular network representations are protein pathways in biology, supply chain systems, and temporal events in narrative cartography just to name a few.

Efforts have also been poured into presenting geovisualisation using Semantic Web technologies, where a common framework is set by the World Wide Web Consortium (WC3) that allows data to be shared and used by different applications across the Semantic Web. Geovisualisation has broad coverage that contains multiple knowledge traits. These contain visual perspective, cartographic scale, data portrayal, and geometry. All these elements comprise geovisualisation knowledge [HKMH20]. Understanding the fundamental aspects of geovisualisation concerning the Semantic Web is key to develop a standardized framework that follows pre-established concepts by the cartographic community. Similar graph models have been implemented in narrative KG cartography to standardize Narrative Cartography Ontology [MHC+22].

Graphs are classified by the directionality of information they present or the lack of it. KG that contains directionality on the edge that relates two nodes is called a direct graph, while a graph that does not show such directionality is an indirect graph [Mat22]. Directionality normally is represented with an arrow [0–>0<==>0].

Graph visualization follows an outlined process for its estimation and rendering. The process consists of 4 main steps(*Data retrieval, building, calculations, layout* and *rendering*). *Building* extracts data from a source. In the case of a semantic graph, it extracts semantic context. In this step, RDF vocabularies help to provide interoperability between the semantic structures. The *building* process takes care of building the graph and has data in the most machine-readable way. Graph *calculations* estimate all the graph properties interpreted in the rendering process. Specific graph attributes such as relation weights and node relevance are calculated at this point. The graph *Graph Layout* calculates spatial values or coordinates within the viewport depending on the selected layout. The layout makes it possible for humans to understand the graph system. Finally, the *rendering* displays all estimated values from the previous steps. This process can vary specifically on the data retrieval since OWL vocabularies keep expanding, which facilitates extraction from raw data. [GRMSOG18]

A limiting factor for graph visualizations is the graph size. Problems such as viewing port saturation and performance arise with size. Saturation of the viewing port results in overall information becoming hard to understand since nodes and edges pile up together.

Hence, recommended way to display a graph is small. Graph visualization is responsible to solve the problem of giving a certain number of nodes and relations, estimating the positions of these, and drawing all elements with respect. Some terminology exists in graph drawing depending on the resulting drawn graph. Such as *planarity, aesthetic-rules, size, predictability and time complexity*[Her00]. Planarity refers to the possibility of drawing a graph without having edges crossing paths with each other. *Aesthetic* rules may set constraints to the drawing process such as edges must be straight lines and have the same length between nodes. *Planarity* should be minimized. Nodes in a drawn graph should be isomorphic to each other. From these aesthetic rules avoiding edges, and crossing is the most important[Pur98]. *Size* relates to how well a graph visualization will scale, accounting for readability and rendering performance. *Predictability* is a term that refers to the algorithms drawing the graph, where different drawing algorithms drawing the same data should lead to similar visual outcomes. Finally, *time-complexity* is how immediate can updates on a graph take place, which allows for better interactions and facilitates implementation when scaling up.[Her00]

4.5 Types of Knowledge Graphs Visualisations

Graph visualizations are classified according to their layout. Each layout has its challenges concerning the overall drawing variables (planarity, aesthetic rules, size, predictability, and time complexity).[Her00]

The tree layout is the traditional way of representing a graph. This layout subdivides into many types such as *simple tree-layout,ballon-view, hierarchical-tree, radial-view,tree-map*. Tree layouts are seen as the simplest graph visualizations to generate and the ones with the lowest complexity to generate, where complexity is proportional to the number of nodes. The Ballon-view (Figure 4.3) is generated by having sub-trees projected to attached father nodes. For the hierarchical tree (Figure 4.2) the most important factor to preserve is planarity. Treemaps are also a form of hierarchical representation, where the viewport is subdivided, however, relations can not always be identified. Relationships of type *isContained* is easily spotted. The radial-view (Figure (Figure 4.4)) places all nodes on a concentric circle concerning their hierarchical level. [Her00]

The *Sugiyama Layout* is used for directed acyclic graphs (DAGs), where the edge is directed to another and there is no closed loop. In this type of layout, the *layering* of the nodes is of most relevance. The placement of the nodes follows the layering order. The same information can be represented in different manners according to preset constraints.

The *Spring Layout* or *Force-Directed* is a graph generated based on Hooks law which describes attracting and repelling forces between bodies. This method is attributed to Eades and was later refined[FRW91] [Her00][HSoM91]. The result is generally a balanced graph system with good planarity. The disadvantage of this method computes time since all nodes have to be revisited for each calculation. Complexity is of the order $O(N^3)$ considering N as the number of nodes[Her00]. Another property of a spring layout system is that it is unpredictable, the same data could generate different graphs. A *Spanning Tree* is a popular graph representation that scales well with large graph systems that can be disconnected. The *Hyperbolic layout* is a graph contained in a distorted space, which often allows for navigation within the fish-eyed view-port (Figure4.6).

Navigation and *Interaction* are relevant areas that impact the level of the geovisualisation (exploration, analysis, synthesis and present). *Navigation* is composed of *zoom and pan*. Context preservation techniques can be implemented to preserve contextual information when a section of the graph is zoomed such as *Fish-eye distortion* or *Incremental exploration*[Her00]. Context preservation techniques are used as a counterbalance graph size. *Clustering* is another popular method used to abstract graphical information. The implementation is done based on semantics. Clustering can be done on the source of the information. *structural-based* clustering is done strictly with data contained within the graph. *content-based* uses semantic data that is bonded to each of the graph elements. Typical tasks for clustering are searching and task filtering. Algorithms that cluster graph systems usually are designed to find a balance in the filtered information, while maintaining the overall graphical structure at the proposed abstract level. Such algorithms usually focus on the network properties such as type of graph, degree, rank, edge weight, and such[HDM98][Her00]. A *clustered layout* is the resulting layout of a graph that has undergone a clustering process. The term *super-node* introduced by citeHerman I (2000) in this visualisation. Super-nodes are clustered nodes, when implementing super nodes the resulting graph is called a *compounded graph*. citeHerman) proposes some visualization techniques for compound graphs (*ghosting, hiding, and grouping*). *Ghosting* takes focus away from non-super-nodes or less clustered super-nodes. *Hiding* hides nodes based on a predefined constraint (selection, node-type among others ...). Compounded graphs can also be subject to navigation interaction through a folding process [HDM98].

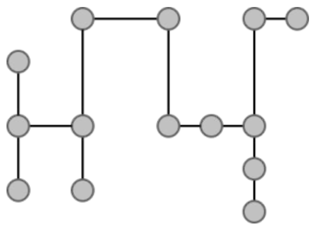


Figure 4.2: H-tree layout

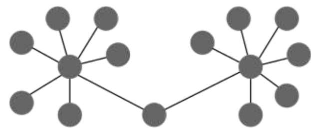


Figure 4.3: Ballon View

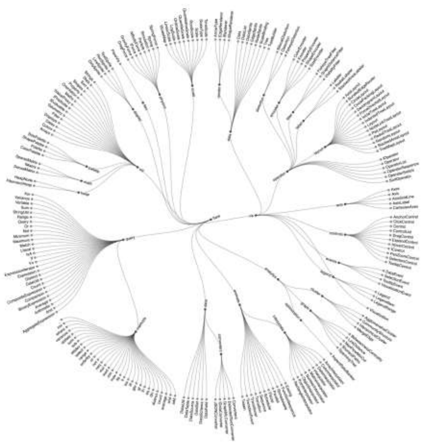


Figure 4.4: Radial-View [Bos12]

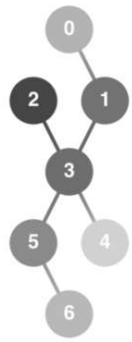


Figure 4.5: Sugiyama-Layout [Bos12]

Figure 4.6: Hyperbolic Tree[Glo21]

4.6 Visual Knowledge Graphs & Cartography

The earliest knowledge visualizations integrated into maps are *flow maps, connection maps* and *sketch maps*. *Flow maps* are not always considered formal network maps. Normally a flow map consists of a network-like view, where flows are represented by relative line widths. A pitfall from this map is that often lines overlap. It is difficult for humans to estimate the magnitude of the values expressed on line widths. *Connection maps* were introduced in the 1930s in London as subway maps.*Network sketch* maps as the name suggests are hand-drawn maps with a route and an end destination. Used typically for advertisement purposes. These maps have low detail and are considered mental representations of the maker from any given area. Also, network sketch maps show alternative flexible routes and navigation instructions. In this type of map, the focus is placed on landmarks that facilitate wayfinding on the road or in remote scenarios. Due to the simplicity of these types of maps often their network properties are overlooked. However, they all contain the characteristics that define a network map visualization. According to Ruggles (1997), a network is an organized collection of objects with cartographic nature, where degree, organization, flow-interaction, and contextual relationship influence their representation of such network. Network and geographic transformation can be combined to obtain the network types seen in figure 4.7. [RA97]

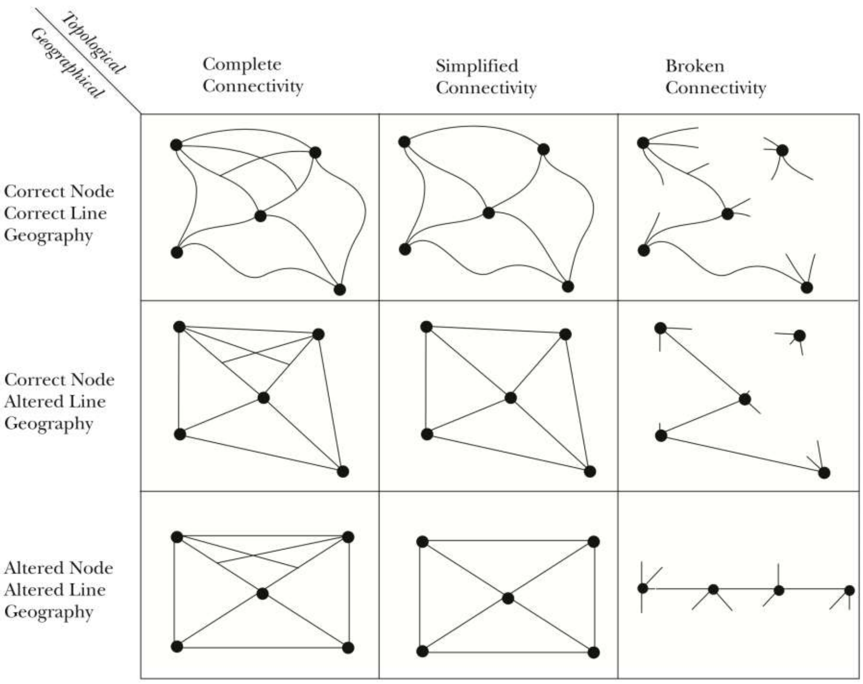


Figure 4.7: Network basemap transformations[RA97]

There has been some relevant work in the field of KGs and cartography. The use of KGs to present history combines technologies of GIS clients (QGIS, ArcGIS Pro) and Knowledge Network(KN) sources like Wikipedia. A drawback of this research was the limitation of the used GIS client and its toolbox extensions such as "Linked Data Relationship Finder" of proprietary nature. Challenges were identified such as data incompleteness, semantic incompatibilities, and geovisualisation. To these last drawbacks, a proposed ontological design pattern in the form of a "Cartographic Module" was presented, to define how narrative maps that use KGs should be visualized [MHC+22]. High volumes of data do not translate to better visualizations [KPS19]. Hence, we need to rely on effective ways to convey insights contained in data. The true benefit of visualizations is not contained in a single visualization form. But in the combination of various visualization types (line, bar, pie, stacked, polar, histogram, dot-matrix, etc ...)[KPS19].

Cartography of the mind is an area where KGs are used. Our creative process has a strong relation to KGs. The association that concepts have in our mind is related to the creative self[Med62]. More creative people are capable of traversing deeper into their mental structures. These people have higher semantical awareness. Ontologies are connected weakly and strongly. The way mental semantics are traversed is related to connectivity, ontological distance, and communities. Creative individuals normally reach weaker connected nodes. Each path explored path is a *stream thought*, where each concept visited has a semantic distance associated with it[KF19].*Stream thoughts* can be represented on a graph visualization to better understand the thought stream that happens in the human mind[KAF14].

CHAPTER 5

Popular Technologies that surround Knowledge Graph and Knowledge Network systems

5.1 Graph Databases

5.1.1 Brief introduction Graph Databases

Current map information structures in GIS revolve around traditional relational database management systems (RDBMSs) [CDW+19]. In such databases, information is stored in table-like structures which implement structure query language (SQL) to retrieve data and execute operations on such RDBMSs. This traditional method allows for practical bulk operations that enable effective statistical analysis. Apart from these relational databases exist graph database management systems (GDBMSs) which operate on a node-edge structure, where ontological instances are depicted as nodes and the relationship between these ontologies is defined by an edge. These ontological interactions are represented in a knowledge graph(KG)[MHC+22]. Analogous to RDBMSs SQL's GDBMSs use SPARQL to query and implement changes. SPARQL has the same expressive power as Relational Algebra [AG08].GDBMSs have emerged as a complement to RDBMSs. However, the limits of GDBMSs are still unknown[CDW+19]. The scarce implementation around GDBMSs results in a need to explore the potential application and use cases that could benefit map creators and users alike. The storage architecture works by having vertices that contain a direct reference to a list of direct vertex[FB18].

Graph databases support ACID transactions the same as RDBMS. They are better at gathering information than RDBMS. The data storage capacity is in the order of Petabytes. They are flexible when expanding due to their graph structure. They handle well irregular data. Tasks that relate to data mining perform well. Data type declaration is not required. Handle well on querying data that has multi-dimensional. Hence, having good performance in Big data scenarios[FB18]. Graph databases struggle with representing data or operations that involve data steamed from RDBMS. However, graph databases are flexible, feature good performance, and are quite agile. Today businesses evolve at high speeds and also the collection of data is at higher volumes than before, for these reasons the use of graph databases has become more popular.[FB18]

Both RDBMS and GDBMS are cyclical structures. An important differing factor between these two technologies is that relational structures use tuple array structures, while graph structures employ a triplet structure (RDFs). Tuples inherently predescribe data type. It is possible to execute RDF queries on relational structures. Using RDF queries provides unambiguous results since RDF has deterministic vocabularies.[GPE21]

As the use of the Semantic web becomes more popular performance challenges arise. Graph systems struggle when displaying big graph structures and in the early stages of retrieving and processing the data. Completion time of tasks such as queries, page rank, build, and layout scale concerning the number of edges that a network has. Layout tasks especially increase exponentially with the number of edges, while queries, page rank, and build scale linearly with some exceptional cases. However, depending on the dedicated technologies used the point of exponential growth can be displaced in any direction. This has been proven with both real-life data and synthetic generated graph systems.[GRMSOG18]

The core objectives of all GDBMS are *graph pattern matching* and *graph navigation*. Graph pattern matching works depending on the overall graph pattern from the GDBMS. The known patterns are of nature *basic* and *complex*. Basic graph patterns are query structures that directly match the database contents. Complex graph patterns are augmented query structures by use of unions, projection, optional, and difference. These patterns can be of *homomorphic* nature or *isomorphic*. A *homomorphic* pattern matches into a graph with no restrictions while a *isomorphic* pattern follow restrictions. Isomorphic patterns can allow repetition or not of edges or nodes. [Ang17]

Aside from basic graph patterns, a graph database is defined by its navigation capacity. Navigation graph patterns are basic graph patterns with the main difference being that the edge is being labeled. These are also known as regular path queries or (RPQs). RPQs are integrated as basic navigation graph patterns and can be extended as well as complex navigation graph patterns[Ang17].

5.1.2 Graph Database Query Languages

The most popular GDBMs operate under three different structure query languages (SPARQL, Cypher, and Gremlin). As mentioned before these query languages all operate under pattern matching and pattern navigation. SPARQL is derived by the W3C to extract RDF graphs. Cypher is an implement-ion of the Neo4j team, which uses patterns as the fundamental components to build queries. Cypher aims to be a more visual language with a visual node encoding. The main command used in Cypher to query is MATCH. Inside the MATCH command, the basic graph pattern is specified. Here are some simple queries that Cypher uses to do graph pattern matching. Cypher uses WHERE for conditional matching, WHERE NOT is also possible. To create a Node or a relation the command CREATE is used. [Ang17]

```
Node Creation

CREATE((Fruit){name:apple})
CREATE((Taste){flavour:sweet})

Simple query to get Fruits which are of name apple

-MATCH (a:Fruit) WHERE a.name='Apple' RETURN a;

This query returns all the relation of type 'tastes_Like'
connected to Fruit nodes of name 'Apple'

-MATCH (a:Fruit)-[:tastes_Like]-(tastes)
WHERE a.name='Apple' RETURN a,tastes

Relation Creation -> To create a relation first we need to
match the nodes and then create the relation between them.

-MATCH (a:Fruit) WHERE a.name='Apple'
-MATCH (b:Taste) WHERE b.flavour='sweet'
-CREATE(a)-[:tastes_Like]->(b)

In this example note that apple tastes like sweet and not the other
around(sweet tastes like apple). Relations have a directionality.
```

SPARQL is a declarative language, which means the user specifies the task. This W3C language queries RDF triplets of SPO nature. SPARQL supports basic patterns and complex graph patterns. Each statement is interpreted as a sentence. In the following example, it would read as follows. Select an apple that tastes like sweet and a taste that has a sweet flavor. This allows for deterministic sentences. That are structured under the predefined standard vocabularies set by the W3C. SPARQL supports other operations (UNION, MINUS, FILTER, OPTIONAL,...), which allows for more specific sentences. [Ang17]

```
SELECT ?a ?b
WHERE {
  ?a :tastes_like ?b . ?a :type :Fruit.
  ?b :flavour "sweet" . ?b :type :Taste.
  }
```

Gremlin is a language adopted by Apache Tinker-Pop3 and has a very distinct nature in comparison with SPARQL and CYPHER. It follows a G.V() command that returns the requested list of nodes, followed by .in() and .out() commands to include respective relations. It is considered to be a limited declarative language. [Ang17]

A popular alternative among developers is the query language GraphQL. A query language released in 2016 by Facebook and since has been popular in the programming community[HP18]. GraphQL is an implementation to partially replace REST-API requests on SQL and NoSQL commands. Since its release, it has seen adoption by some graph database developers such as Neo4j[TVSV18]. However, the GraphQL implementation is strictly an overlay on the existing graph querying language. This means that it is possible to emit GraphQL commands which are parsed to SPARQL, Cypher, or Gremlin. Each parsed command term executes the corresponding command in its specification. This means the user querying gains homogeneity at the expense of sacrificing some performance on parsing tasks. There is a challenge to implement semantic declarations in the case of SPARQL[HP18][TVSV18]. Tasks such as defining a graph structure are necessary to create the bridge between GraphQL and SPARQL[TVSV18].

5.1.3 Popular Graph Database Systems

The most prominent databases that have reached a considerable level of adoption are Neo4j, AllegroGraph, ArangoDB, InfiniteGraph, and OrientDB. They all support different query language architectures (SPARQL, Cypher, Gremlin ...) and have differences in the way they operate. All of these databases boast advantages and pitfalls. The main areas according to Fernandes (2018)that define a graph database are flexible schema, query language, sharding, backups, multi-model, multi-architecture, scalability, and cloud-ready.

AllegroGraphDB is based on RDF technology. It is an open-source project, which has been adopted by private and public entities alike. AllegroGraph is known to have high storage, due to its efficient memory handling. It can scale up to a billion nodes while maintaining performance. The AllegroGraphDB community has developed interfaces to connect with popular programming languages such as JAVA and python. It supports *adhoc* queries, which are dynamic or on-the-fly queries. Queries are done under SPARQL language[FB18]. It is possible to make queries that contain Geospatial, temporal and relational logic. It can be cloud hosted.

ArangoDB was started in 2012. However, the level of adoption is still small. Arango's query language (AQL) supports multiple data models. The AQL command remains unchanged even if the data model changes. This feature is powerful since it allows for simplicity and flexibility when developing products. It has strong data consistency, thanks to back-end management that follows the ACID rules[FB18]. Arango's client is embedded in the browser. ArangoDB's architecture implements REST protocol. Because of this, it has JavaScript integrations.[FB18] It can be run under a cluster architecture[Akh17].

InfiniteGraph is built on C++ and has JAVA implementations. This database is popular in commercial projects due to its scalability, distribution and processing features, and cloud storage capacities. InfiniteGraph allows for distributed process partitions and graph partitions. InfiniteGraph has the advantage of allowing multiple reads and writes. API is mostly based on JAVA and C++. It allows for data models to be exported as JSON. InfiniteGraph is not a free database.[FB18]

Neo4j is an open-source database built on JAVA released in 2007. Neo4j implements native graph storage. It follows a persistent transaction architecture. Neo4j implements Cypher as it is a query language. HTTP API is used for its client, which runs on the browser. Neo4j is considered the most popular graph database. Used in both private and public projects. It has been presented as a tool in use cases such as tax evasion, and access management. Neo4j has graph algorithms. Neo4j is flexible, scalable, and follows the graph model. Multiple drivers for JAVA, JavaScript, Spring, and Scala have been developed for Neo4j. It supports query data exports in JSON and CSV. Neo4j is a high-performance database since it is built on native Neo4j[FB18]. Neo4j supports sharding through its Neo4j Fabric implementation. Neo4j supports spatial queries due to its spatial plugin extension [TS19].

OrientDB is an open-source multi-model database released in 2011. OrientDB is implemented on JAVA. Query commands can be done on Gremlin, Java, or SQL. It is designed under free adjacency architecture, meaning that vertices have a stored list of all connected vertexes. It is free under the Apache license. OrientDB can be sharded. One of the biggest pitfalls of OrientDB is that it does not support import tasks. OrientDB has a web-built client[FB18].

Finally, Apache's AGE is a PostgreSQL extension that implements Cypher on any PostgreSQL (11,12,13, and 14) instance. This is done through the implementation of a parser that acts as a middle firmware to execute queries at an SQL level. The process takes place on a cashed layer with the following steps: *parsing, transformation, planner* and *executor*. At the *parsing* stage the caper query is parsed into Cypher specifications. During *transformation* the Cypher query is converted into a query tree. The *planner or optimizer* produces plan nodes to execute the corresponding node operations. Finally the *executor* executes the plan nodes. This extension is still under development and still, the performance impact or level of adoption that this parser will have is unknown.[hASF22]

5.1.4 General Adoption of Graph Databases in Cartography

As graph databases see higher adoption in other fields such as cartography, and paired with the fact that geospatial data is collected in higher volumes. Additionally, cartographers frequently deal with graph-like structures such as river networks, and road networks among others. Consequently, research is done using graph databases to explore the possibilities that implementing graph databases open.

Brazil holds 12 percent of the world's freshwater reserves. However, the distribution of it is not balanced across Brazil. Most of the water is stored in the Amazonas. The network faces many challenges such as pollution waste from illegal mining, industrial waste, sewage, and general waste. For this reason, is important to have efficient information infrastructures that provide on-demand information in regards to the network. Daltio (2016) migrated the river network from an RDBM to a GDBM to understand the benefits of such technology. He defines drainage points as nodes and drainage stretches as the edges. The river network he works with is a binary tree graph that is connected and acyclic. Relevant operations would be to retrieve all the water stretches from a drainage point and calculate upstream catchment areas. These operations on a regular RDBMS require multiple JOIN queries. In his research, he encountered three main challenges data structure types, operator implementation, and consistency in database states. Daltio concludes that due to data persistence issues it is recommended to use both database types to obtain the benefits of consistency from an RDBM while having the graph benefits that a GBDM offers. This type of architecture he describes as a hybrid. His implementation was done on a Neo4j instance.[DM16]

Trajectory analysis benefits from the implementation of GDBMs. Currently, navigation algorithms filter a lot of semantics because data comes in bulk with often an unnecessary overhead. These filtering techniques stem. These filtered elements are often relevant for other parts of the navigation system. For this reason, Tamilmani (2019) proposed a new methodology that involved trajectory estimation with the use of a GDMS. In his works, she uses Neo4j and PostgreSQL instances. Tamilmani uses a hybrid approach as well. The main methodology approach is to enrich the simplified line with semantics and inject this enriched line into Neo4j. Once in Neo4c Cypher can be used to further process and analyze the trajectories. An architecture approach he follows is to connect a JAVA server and have the JAVA server communicate with Neo4j and PostgresSQL then the data is transferred via HTTP to a custom front-end web client. He concludes that this approach simplifies trajectory analysis when talking about large data sets. [TS19]

Augmented reality navigation systems are also good candidates for graph database implementations according to Amirian (2015). Since LBS applications are becoming more ubiquitous, and as seen on Tamilmani's work there is a lot of drive for graph database implementation. Amirian proposes a Native mobile navigation stack that enforces the connection of both a spatial database and a graph database, with a front-end web application rendering tool. The proposed tool provides efficient navigation using augmented reality properties such as wayfinding in real-time in combination with the cellphones camera. The navigation tasks are handled by the graph database, while the distribution of other assets such as images and graphical elements are provided by RDBMS. OSM data would be embedded in the graph database and communication between these elements is handled via a REST API. He concludes that graph databases allow for easy connection of multi-dimensional data. [ABG+15]

5.2 Web browser-based

Modern cartography is trending more and more in the web browser. We see more often web data tools coming up. Big GIS clients such as ArcGIS or GRASS GIS can be now used from a web browser. Projects such as OpenLayers, Mapbox, and Leaflet have gained popularity over the last few years. There is a trend toward websites with visualizations that prioritize interactivity. The way a website is designed regarding how it functions and how it interacts with itself is normally called web Architecture[DB19]. There are many types of architectures such as client-server, multi-tier, service-oriented architecture (SOA), and component-based among others. The client-server model is based on the concept of a front-end or client requesting data from a logical server or back-end. Depending on the response the client might react in a specific way. A multi-layer as the name suggested is simply a tiered layer that divides the tasks. Each tier might be in charge of security, task management, data acquisition, or distribution. SOA is increasingly popular and it divides all the elements of a web client into small components and allows for efficient maintenance and scalability. It allows for flexible projects that can be modified easily. The component architecture is similar to SOA. However, the core concept of this architecture is re-usability where components can be instantiated multiple times and rendered according to predefined specifications. This last architecture normally is used to design singled page applications (SPAs).[DB19]

The way architecture is structured relates to the framework used to build it. Currently, popular component or SOA architectures to build SPA's are Angular and ReactJS. Angular is a framework developed by Google. It has been used to build popular web pages such as Gmail, Forbes, and PayPal. ReactJS developed by Facebook is responsible for websites such as Netflix and Dropbox(Hello-sign) built using this framework. These frameworks work under a language developed by Microsoft called Typescript. Typescript addresses JavaScript's lack of syntax[BAT14]. Typescript is a strictly typed language. This means syntax must be defined always. This allows for a clear code read and faster project development. Since it is an extension of JavaScript it can run on the browser and inherits all prototype methods that JavaScript has for each type of object (Array, object, integer, string ...). Typescript follows object-oriented programming rules. Typescript in comparison to other strictly typed languages (JAVA), allows type errors at compilation that fails at run-time[BAT14]. To build SPAs that contain maps and knowledge graphs it is required to have certain building blocks. Libraries are the compounded elements to build such products. A library is generally written in a language (javascript, typescript, python, JAVA, etc ...) and it may or not support other languages depending on the developer. They are subject to license usage which is defined by the author of said library.

A popular library for rendering maps on a website is Open Layers. This is an open-source library that follows the Open Geospatial Consortium (OGC) standards[DJS14]. Open layers are extensive and flexible. OpenLayers is written in JavaScript. However, a Typescript upgrade exits, which allows it to be used in Angular or React. OpenLayers allows for interaction with objects contained within the map, which adds a higher level of interactivity.

D3js is another popular JavaScript library used in data-related projects. Its popularity is due to visual flexibility and integrated scalable vector graphic(SVG) standards defined by the W3C. D3js supports many types of visualizations and has become a popular library to showcase data aesthetically since SVG preservers detail independent from scale (*resolution-independent*)[BC14]. D3js supports a long range of graphical display types that range from traditional charts (bar, line, bubble, etc ...) to a more experimental type such as hyperbolic trees [Glo21].D3js allows data binding Document Object Model (DOM) elements, which allows for interactivity. D3js's methods are designed under a declarative architecture which allows for more intuitive DOM manipulations. D3js has a well-documented API and a vast community, which allows for the project to be well maintained.

graphVizdb is an example of a graph project that is distributed online. graphVizdb architectures consist of the front-end, a graphVizdb core module, and a database. The prototype includes several services such as visualization, information, navigation, bird-view, string search, statistics, filter, and edition. graphVizdb rendering time is dependent on the window size (pixels) where bigger screens take longer to render. graphVizdb is built on JAVA, MySQL, mxGraph, JavaScript, and Metis[BLK+16]. graphVizdb runs a graph tool using only an RDBM as its database.

The main strength of web-based graph technologies is compatibility and distribution. It is common to have none standardized technological environments, where users have different software and hardware environments. Having a cross-platform project, which can run on any web browser allows for mass adoption. Distributing any project through the internet also gives it exploration and increases the user base, where scalability at a low cost is feasible.

5.2.1 Non Web browser-based

Even though cartography trends toward a more web-browser environment. There are technologies that run independently from the browser and still prove to be powerful tools. These tools can be integrated into web-based tools as back-end services[BC14] [FB18].

Python is a derived programming language from C++, C, ABC, and Unix among others[SFV19]. Python is the 3rd most commonly used language among programmers [ove21]. Python has an easy-to-learn approach since it resembles English reading. Python has a supportive community which has allowed for good maintenance. Even though python was not intended to be used as a web-based technology it has been implemented as a framework (Django and Flask) to build websites[SFV19]. Python has specialized in data analytics libraries such as OpenCV and Tensorflow, which are popular to handle big data. Python is simple in the way that more gets done with fewer lines of code compared to other languages[SFV19].

TorchKGE is an open-source python library/module that specializes in knowledge graphs [Bos20a]. Its main objective is to provide fast KG operations and predefined structured interfaces that allow for fast KG design and testing[Bos20b]. TorchKGE is integrated with *NumPy* [Bos20b], a popular python library that enables numerical computing with Python[Num]. TorchKGE has been recognized for its high efficiency and in comparison with other KG specialized modules such as OpenKE, AmpliGraph, Pykg2vec[Bos20b].

NetworkX is another popular network python library. NetworkX focuses on the exploration and analysis of networks and their algorithms[HSSC08]. The library ships with network data structures that conform to graphs and networks. Popular algorithms that NetworkX has included are shortest paths, betweenness centrality, clustering, and degree distribution just to name a few[HSSC08]. NetworkX is designed to work with SciPy, Numpy, and Matplotlib tools. NetworkX allows for simple graph instantiation and construction. Figure 5.1 shows Napoleon's Russian campaign rendered with NetworkX.

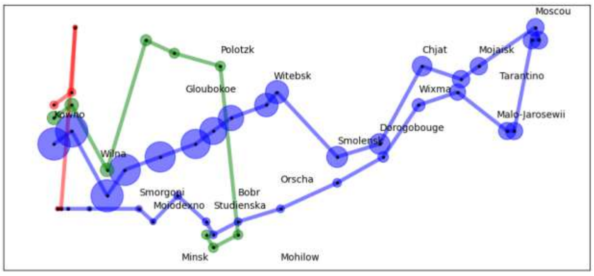


Figure 5.1: Napoleon's Russian Campaign with NetworkX [HSS08]

Non-web-based tools have the benefit of running sometimes in lower-level languages, which are often faster depending on the hardware. They feature powerful processing data libraries. A popular architecture is to combine non-web-based tools with web-based tools in the back-end of any web product (graphVizdb).

CHAPTER 6

Finding Relevance Through the combination of Maps, Knowledge Networks and Web-based Knowledge Graphs

6.1 Justification and Hypothesis Review

Present technology allows us to implement ontological mapping and project this mapping into the spatial domain. However, much research is needed still in this domain. Graph systems allow us to make maps from a different perspective. There is still not a well-defined standard on how to design maps fundamental to ontological constructs. There is also the challenge of integration. How can cartographers convert existing table-like data into knowledge graphs? How can map makers standardize vocabularies when making graph-based maps? We see projects such as the one proposed by Varanka (2018) where the map acts as both a human and machine-readable document[VU18]. We are entering a stage where we could start making maps where semantics are also understood by machines.

Additionally, current map information structures in GIS revolve around traditional relational database management systems (RDBMSs) [CDW+19]. In such databases, information is stored in table-like structures which implement structure query language (SQL) to retrieve data and implement operations on such RDBMSs. This traditional method allows for practical bulk operations that enable effective statistical analysis. Apart from these relational databases exist graph database management systems (GDBMSs) which operate on a node-edge structure, where ontological instances are represented as nodes and the relationship between these ontologies is defined by an edge. These ontological interactions can be represented in a knowledge graph(KG)[MHC+22]. Analogous to RDBMSs SQL´s GDBMSs use SPARQL to query and implement changes, SPARQL has been proven to have the same expressive power as Relational Algebra [AG08].GDBMSs have emerged as a complement to RDBMSs. However, the limits of GDBMSs are still unknown[CDW+19]. The scarce implementation around GDBMSs results in a need to explore the potential application and use cases that could benefit map creators and users alike.

Map makers struggle to convey and/or understand the relevance of a map. Knowledge graphs pose as an alternative or a complementary tool to understand and identify relevance considering semantics. This work serves as a proof of concept that the combination of knowledge graphs, graph visualizations, and maps lets the design of maps consider ontological definitions, which help in the understanding of geographical relevance.

6.2 Methodology

6.2.1 Project Objectives Review

This thesis work has two main objectives. First, understand how the mapping of the ontological dimension projects relevance to the spatial dimension. Some other researchers have explored the idea of mapping the semantic space as a conceptual idea[JG22]. However, the lingering question is in regards to the actual value that mapping such spaces bring considering human and time resources. This leads to the second main objective of this study which is to present a proof of concept that can be built upon to explore relevance that originates from the ontological space. Logically in cartography, there has been a focus on understanding geographical relevance. A relevance where all entities in geographical space have a quality attribute. This quality is the relation between the representation of a such entity and its context. Geographical relevance then goes into four levels discrete objects, relationships, groups, and functionality[RDS11]. Mapping the ontological space and projecting such space on a map helps us address the first level (discrete objects figure 6.1). When speaking about mapping the ontological space, two main challenges arise. The first one is the commonality of an understatement. This relates to what the W3C does in its RDF vocabulary. The second challenge is the level of semantic richness an ontological space can have. Both are vast challenges that are areas of research by themselves. This work presents a very simplified version of mapped ontological space. Simplifying the mapping process allows for an easier understanding of the overall subject.

Since modern cartography has been trending towards browser-based tools, building tools that run in such environments makes sense. The idea is to present a tool that runs on free-to-use software that stays away from proprietary technologies. This facilitates reproducibility and serves as a pathway that others can follow without the worries of financial constraints. For semantic web technologies to be adopted and used there is a need for commonality. This commonality disappears when we use proprietary closed-black-box software.

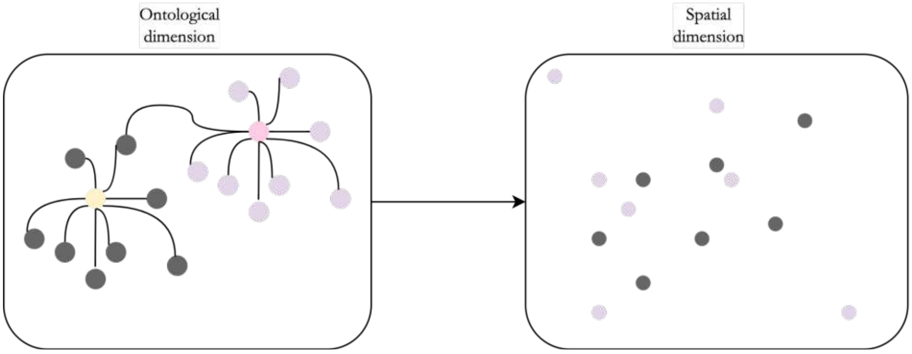


Figure 6.1: Ontological and Spatial Dimension mapped

6.2.2 Conceptual design

For simplicity, I will refer to this tool as *SeMaptics*. *SeMaptics* should run in the browser to have cross-platform compatibility. To build *SeMaptics* first, Some design layouts were maid. An SPA approach will be followed to allow for a layout design that is flexible and can be modified . However, the initial design indicates the components that need to be made to have this scalable and flexible tool(Figure 6.2). In this case, a graphing library is needed and a mapping library to achieve this design. Also, the ontological spaces should be interactive to allow the user to explore and interact with the data.

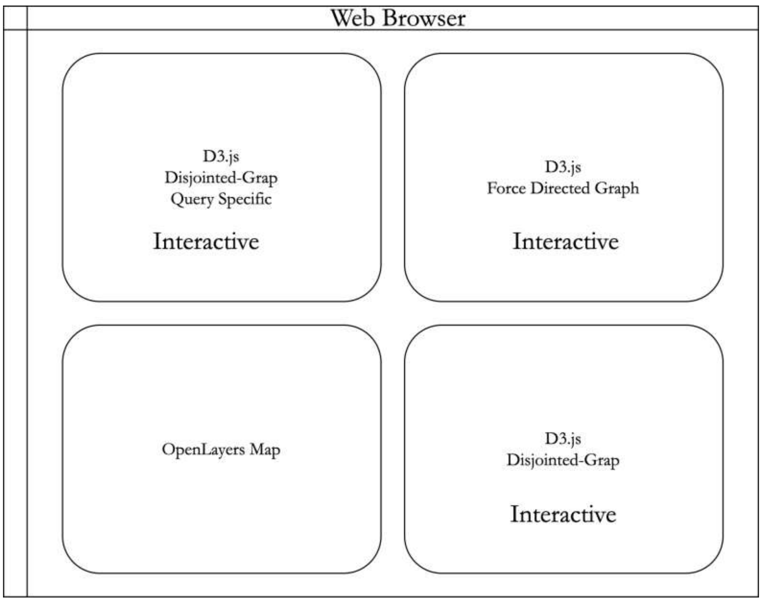


Figure 6.2: *SeMaptics* Conceptual Design

6.2.3 Environment setup

SeMaptics design follows the 4 main steps of graph visualization *Data retrieval, building, calculations, layout* and *rendering*[GRMSOG18].

For Data retrieval two sources were defined a static GEOJSON source which acts as the result of any SQL command done via HTTP. The data that needs to be queried needs to be simple and uncontroversial. If *SeMaptics* can bring relevance to a simple scenario it is likely to have a greater impact on data that is highly influenced by semantics. Following this logic, data from the Austrian Federal Office of Metrology and Surveying (BEV) was chosen. Specifically, a digital landscape model that contains points of interest of municipal, operative, cultural, recreation, and sports nature in Austria. Some data elements in the data sets could be lines in the case of ski tracks and power distribution lines[BEV]. Visual knowledge graphs do not perform well on scenarios with a vast amount of data[Her00]. Consequently, the data was brought down in size by a) selecting a category from the ones mentioned above and b) focusing only on an administrative unit of Austria. Recreational buildings were chosen as points of interest. Since recreational features are more uncontroversial and relevance might be more difficult to identify. For the administrative unit, 'Tulln an er Donau' (Figure 4) with the international id of NM 33-12-19 was chosen. Administrative units in this dataset have a similar extent. However, Tulln an er Donau unit is located on the northwest outskirts of Vienna, and having a unit contains both city and outer city feature points might show in the semantic space.

The data is distributed on a table feature with the following columns *featureID, featureCode, featureName, Name, type, addressCode,subCode, StreetName, DateOfConstruction ,DateOfService ,ObjectId, GlobalID, latitude* and *longitude*. To convert these columns from a table domain into an ontological domain a similar logic to the one implemented by Syed (2010) was used, where each row is an instance of several ontologies and column titles are ontologies themselves. Relations of type *isOf*[SFMJ10] where defined among the nodes. The approach used can be seen in Figure 6.3 using this logic a simple semantic network was generated. This is an approach similar as the one done by Nickel (2016) in his knowledge graph building methodology[NMTG16].

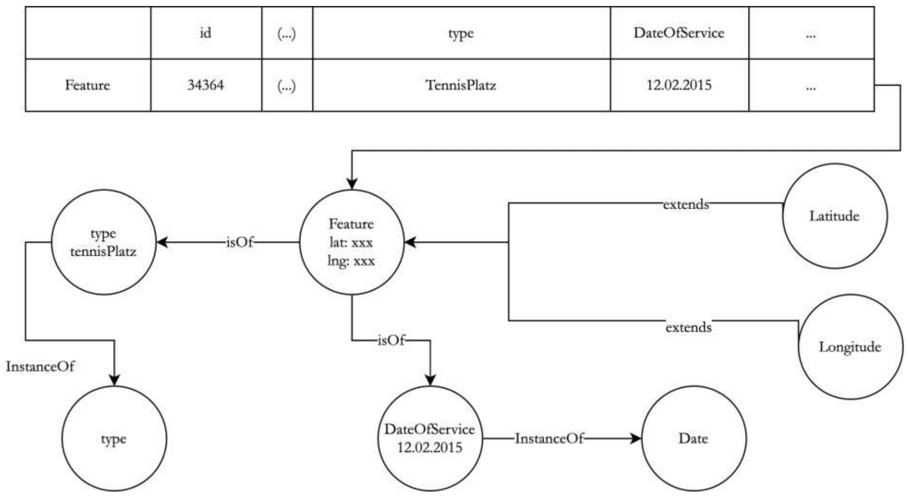


Figure 6.3: Parsing Tabular data to a semantic knowledge graph

With this logic, a decision was made on database technologies to create the proposed semantic network. Neo4j was chosen as the GDBM database since it is an open-source project and has widespread adoption in many cartographic and non-cartographic projects[FB18][ABG+15]. It also comes with a web client, which means that this methodology can be replicated regardless of the operating system. To create such a network a custom parser was written, which outputted a raw .txt from a .geojson file that contained batch instructions to create the network. The batch instructions were later fed into Neo4j's web client. The same logic to make the relations was used. The command that initiates a single node is the CREATE command(Figure 6.4).

```
'CREATE(node:Feature{
    name:"2403 Sportplatz",
    type:"Fußballplatz",
    dateOfService:12.02.2018,
    (...)
})'
```

Figure 6.4: Example on Creating a Node

Additionally, nodes can be created on demand from *SeMaptics* itself since it supports HTTP requests. To make relations in Neo4j it is necessary first to match node pairs (Subject & Object) and then instantiate the relationship between them (Figure6.5). These relations are of cardinal nature.

```
'MATCH(node:Ontologie) WHERE id(node)=34364
 MATCH(type:ART) WHERE id(type)=8471
 CREATE((node)-[:IS_OF]->(type))'
```

Figure 6.5: Example on Creating a Relationship

To set up Neo4j container technologies were used. This falls in with the philosophy of cross-platform compatibility. Docker containers allow for regulated replicable environments, which can be scaled and distributed efficiently and are compatible independent from the operating system they are run on. GraphStack's image (gdb:3.5.26) was used since it is an extension of Neo4j.

A design decision was made to have a non-temporal dependant ontological dimension. This means that time instance are considered as ontologies (Figure 6.3). This is a powerful property of graph systems, where temporal and thematic classification can be done in the same layer. This type of design approach is also used in narrative cartography[MHC+22]. However, there is the possibility of having temporal nodes that relate to an instance of a graph that contains a temporal state of all nodes.

After defining *Data Retrieval* the *build* process[GRMSOG18] was implemented. In this phase, the architecture and the building blocks (libraries) were determined. For web design, a SPA approach was followed, which implements Angular as my general framework. Since Angular projects are Typescript type consistency is preserved[DB19].

An extension that communicates via HTTP with Neo4j was needed. Traditionally this is handled by a back-end service. However, to keep simplicity on database requests a Typescript library called *angular-neo4j* was used, which allows for web-socket communication with any Neo4j database running on a network. Using angular-neo4j avoids the need to have a dedicated back-end. However, in a production environment, it is recommended to have a dedicated back-end for proper security and design. All Neo4j requests are handled by a dedicated Neo4j service. This service handles node and relation creation, deletion, and general queries.

For the graphical interface, *SeMaptics* uses D3js. Since it offers widespread graph visualizations and is in constant development. D3js ships with multiple methods to interact with the DOM and is a tool that not only displays but also enables interaction with the data. This design decision makes *SeMaptics* to be more of an exploration tool with high interactivity[MGP+04]. D3.js offers a wide range of graph visualizations, which are categorized into two groups hierarchical and network. Since the data is not of hierarchical nature network visualizations were more eligible for the project. Among the network visualisation options D3js offers two were chosen *Force Directed* and *Disjoint-force-directed*. *Spring-layout* or *Force-Directed* as mentioned earlier result on balanced graph systems that respect planarity, which is a desirable trait in visual graph representations[FRW91][Her00][HSoM91]. The main difference between *Disjoint-force-directed* and *Force Directed* is that force directed gives all nodes a repelling force and the graph is held together by the relationships. This means that if there is a node that is not related to the data bulk it will simply repel itself out from the viewport. However, nodes that are highly connected or relevant to the network tend to be pulled to the center of the viewport. Thus making relevant nodes easier to identify. The *Disjoint-force-directed* graph on the other hand has similar behavior. The main difference is that a cohesive force at the center of the viewport, which keeps all nodes within the frame, is similar to a gravity field. All forces on these visualizations can be adjusted. D3js can be highly customized. It allows for all sorts of modifications on nodes and edges such as fill-hue, border-hue, shape, thickness, transparency, size, and forces between elements.

D3js in *SeMaptics*follows an nine-step build process. i)It cleans the view-port from any pre-rendered elements. ii) It process the graph data. This includes instantiating d3js node, link, link target, node groups, node titles, styling objects, and force objects. iii) Build node and link arrays that include d3js prototype methods. iv) Define a color scale in which the nodes and links can be rendered according to any given value or classification. v) Sorts and the different node groups. vi) Assign each node and/or link a color from the defined scale. vii) Construct forces on nodes and Links. viii) Construct the force simulation. In these steps, a force simulation is generated depending on the assigned forces from step vii. ix) Set graphical elements. In this step, all SVG elements are constructed according to different calculated variables such as stroke, stroke-width, stroke-opacity, stroke-line cap, text, fill, primitive selection (line, circle, ...), and primitive attributes (the radius for circle). Also, prototype and/or custom methods are attached to each SVG element. These prototype methods are responsible for interaction tasks such as drag, click, and drop. Once this process is finished an interactive graph system is rendered on the viewport. The graph system that *SeMaptics* displays according to Ruggles (1997) classification system (Figure 4.7), is of type *altered node - altered line - complete connectivity*[RA97].

The implementation of OpenLayers(Figure 6.6) follows a simple pattern that goes from fundamental to compounded elements. Some elements can be defined by the application state such as zoom, map center, projection, and base layer. *SeMaptics* defines the map center, zoom level, and the feature data as variables set by the application set. The base layer is an Open Street Map which OpenLayers ships with. The selected hard-coded projection is EPSG:4326. However, data can be projected as well on EPSG:31255, which is the original projection of the data source (Figure 3). The center of the map is a coordinate point that is around the center of Tulln an er Donau.

The main idea behind *SeMaptics* is to have a tool where the user can see both the spatial dimension and the ontological dimension (Figure 6.1). Through this dual representation, a user is able to identify ontological relevance and project this relevance into the spatial dimension. To achieve this goal there needs to be communication between the distinct elements through a managed central data state. Additionally, many helper functions and methods are required, which prepare and harmonize the data for both D3js and OpenLayers. All of this tasks are compartmentalized within a utility manager(Figure 6.7). The utility manager takes care of: i)Managing the application state. ii) Communicating

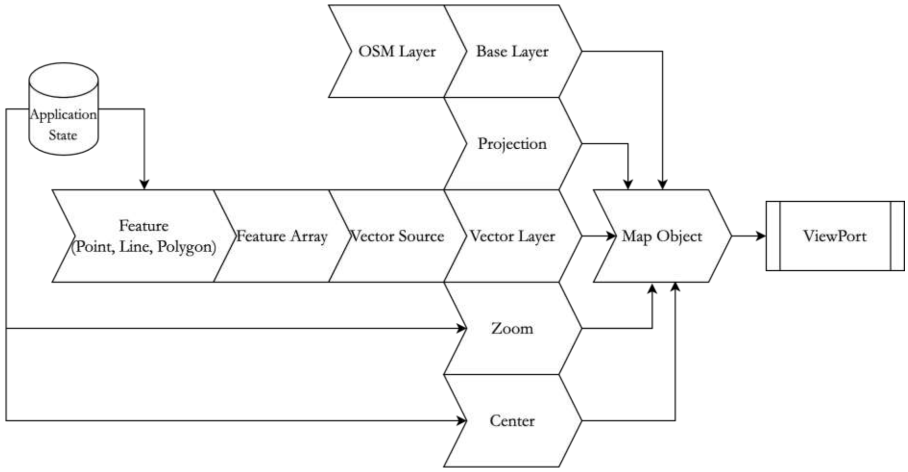


Figure 6.6: Open Layers Build Process in *SeMaptics*

with Neo4j. iii) Sending the updated state to all the corresponding components. iv) Managing interactions. This level of management is done through a design pattern called a facade design pattern. This means that components have independent services. But they communicate with other components through a facade service [PGT10]. This design approach hides complexity in the application and simplifies interactions.

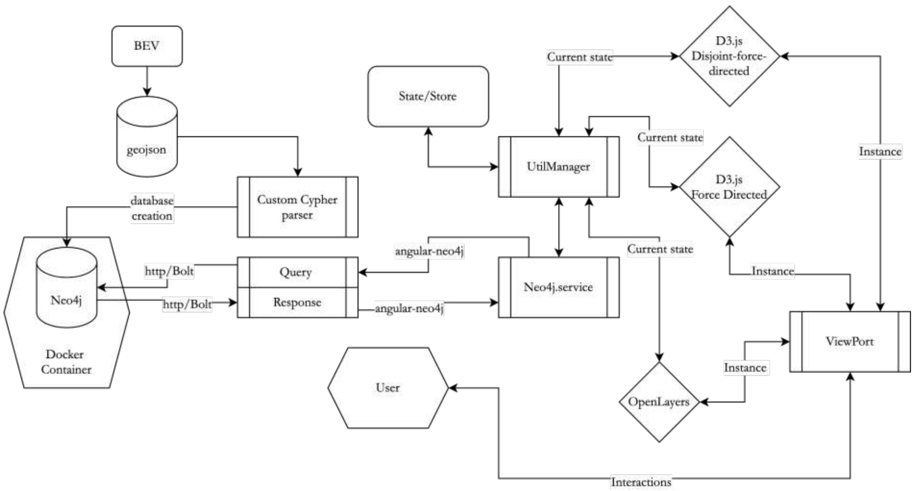


Figure 6.7: *SeMaptics* architecture

The Neo4j response comes in the form of an HTTP response and some parsing is required to have the data harmonized for use within the application. Nodes have three main attributes i) id, which serves as the node identifier. ii) Label, which generally contains the node's title or label. iii) Property object, which stores all other related data that pertains to the node. This last property object is used to have matching elements when selecting data. Relationships have the following attributes. i) Start, which is the source node of the relationship. ii) End, which is the target node of the relationship. iii) Type, the type of relationship that the relationship is ('IS_OF','ACTS_IN','WORKS_WITH')

Defining interactions on the graph is an important feature of *SeMaptics*. The user should be able to navigate the ontological space and see the spatial features tied to any given ontology(Figure 6.1). Here is where D3js DOM interaction and manipulation play a meaningful role. Since each node generated on the map has an id, it is possible to match any node to any feature in the map and query the node's information from Neo4j. Once a node is clicked the application store is notified and the application state updates. This prompts all linked components to update, through what is known as a subscription(Figure 6.9). This follows the observable design pattern, where instantiated classes are notified of specific variable changes[WLoSEoPL09].

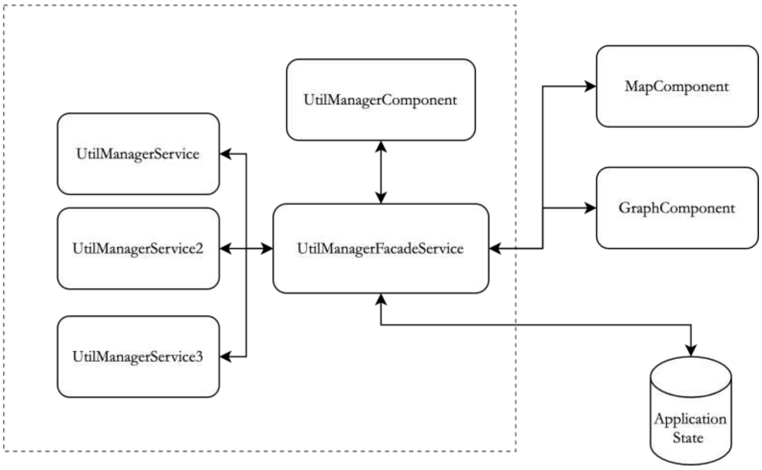


Figure 6.8: *SeMaptics* Utility Manager

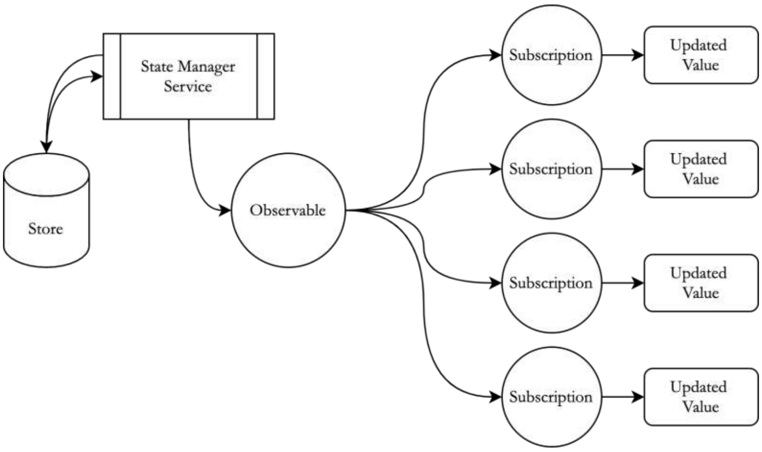


Figure 6.9: Observable pattern on Angular

Using the observable pattern OpenLayers component responds to the selected nodes of spatial nature or the related spatial nodes of any chosen non-spatial ontology such as 'Type' or 'Date'. When data is clicked on *SemMaptics* open layers adds an additional layer of the selected either spatial features or the connected spatial features. Thus the user can easily interpret the graph system and visually identify relevant nodes on any given structure.

6.3 Results

The resulting graph system from the BEV dataset is a 563 node graph that contains 528 *feature* nodes, 10 *categorical* nodes and 25 *date of service* nodes and 4 *street name nodes*. Feature nodes all contain an instance of their respective tabular data and are of geometry type 'point' (Figure 6.3). Categorical nodes refer to the type of installation(ball sport, Football, indoor pool, beach, tennis court, indoor and outdoor pool, outdoor stage). The date of service nodes contains an epoch number of the date of service of the feature. More attributes can be added to the network to enrich its informational value. However, to keep simplicity no other instances where added.

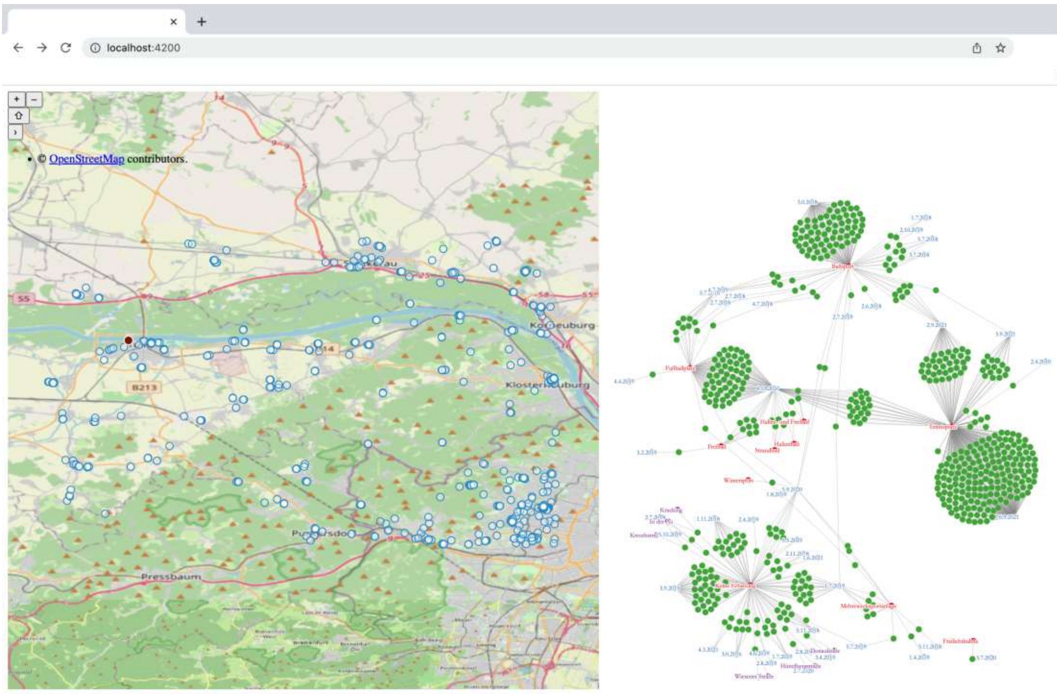


Figure 6.10: *SeMaptics* with a Disconnected-force-graph

SeMaptics general layout can be easily configured. The minimum requirement is that a map and graph component are rendered on screen. The graph has two main visual components i) color, which is assigned to categorization. ii) Labeling, which assists with information graph navigation. The resulting graph is an overlay of two graph systems. One that renders the nodes and another that renders the labels. The design approach takes a toll on performance since the rendering of 2 graph systems in one takes place. However, rendering times and interactions are still snappy and unnoticeable. *SeMaptics* supports dragging functions that allow for the user to modify the topology of the graphical structure without editing the data(Figure 6.12). Drag and drop functionality is available also on Isolated Graphs and on force-directed graphs (Figure6.17 & Figure 6.15)

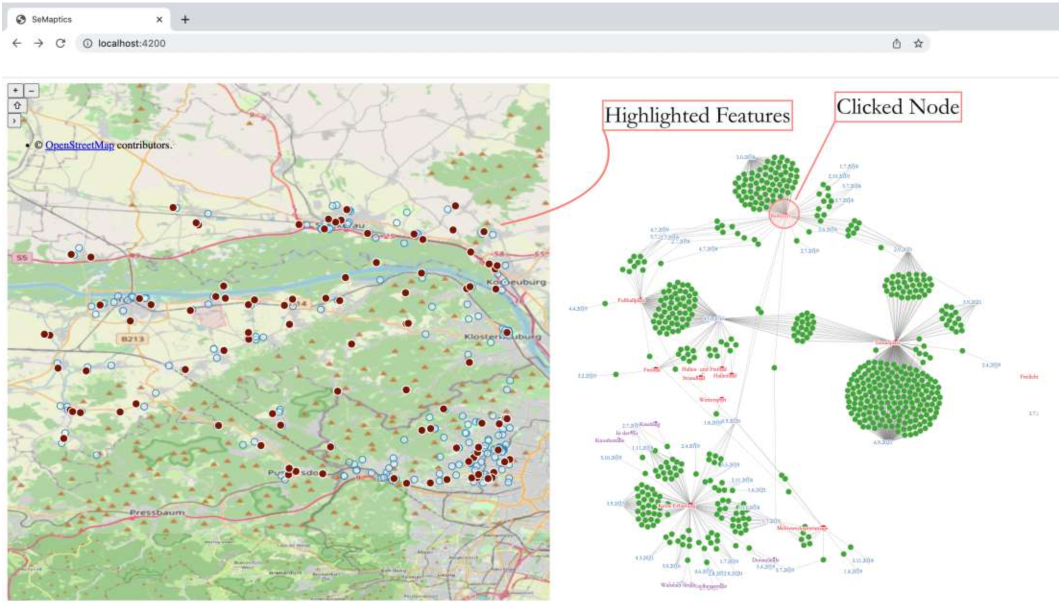


Figure 6.11: SeMaptics Interaction description

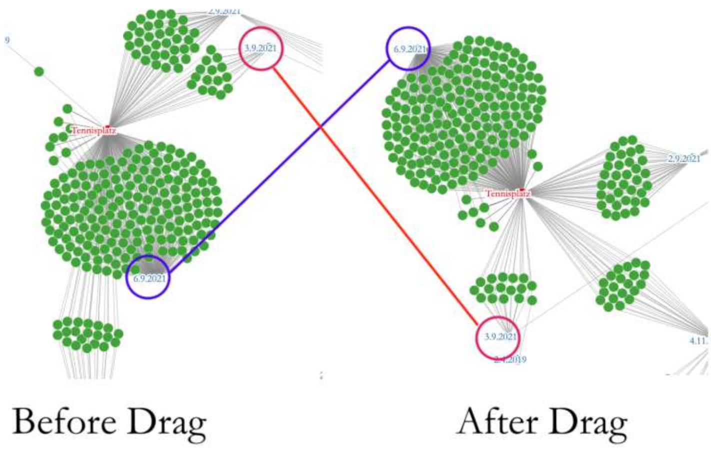


Figure 6.12: SeMaptics Drag

SeMaptics interactions are simple. The selection of a spatial node (green - Figure 6.13) will highlight a point rendered in the map (Figure 6.10) and the selection of a categorical or temporal node will highlight all the spatial nodes related to it. This allows the user to traverse the data and see the projection of ontological entities and their relations in the spatial dimension. This helps the user to define topicality in a map and make categorizations that benefit information extraction[Boy82]. The exploration and interaction with the graph systems allow for cognitive interactions that are chosen by the user as a key element of relevance[Ing96] [Bel95].

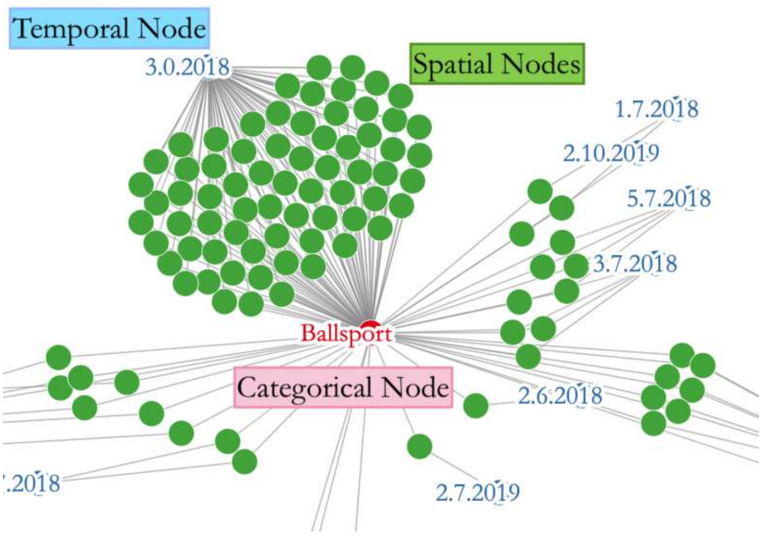


Figure 6.13: SeMaptics Node types

SeMaptics can also focus on single nodes and queries can be established and displayed on map and graph Figure(6.17).This isolates information from the graph under a force-directed graph visualization. This isolation is key to small display graph system visualizations and prevents graph visualizations of great size. An alternative view is using force directed graph where nodes with higher relevance tend to be drawn to the center Such as 'Tennis court –> Tennisplatz' in Figure 6.15. 'Tennis court' has great deal of nodes relating to it (Figure 6.11, Figure 6.12 & Figure 6.17), this translates into the force directed graph. However, some elements such as the Date node '2-7-2018' seen in Figure 6.16 does not relate to many nodes. However, is pulled to the center because is between relevant spatial nodes that are connected to important categorical nodes ('Football' and 'Ball sport'). The user can also interact with this graph and highlight either related nodes or spatial nodes. The force-directed graph view in *SeMaptics* is of broken connectivity with altered nodes, lines, and geography[RA97].

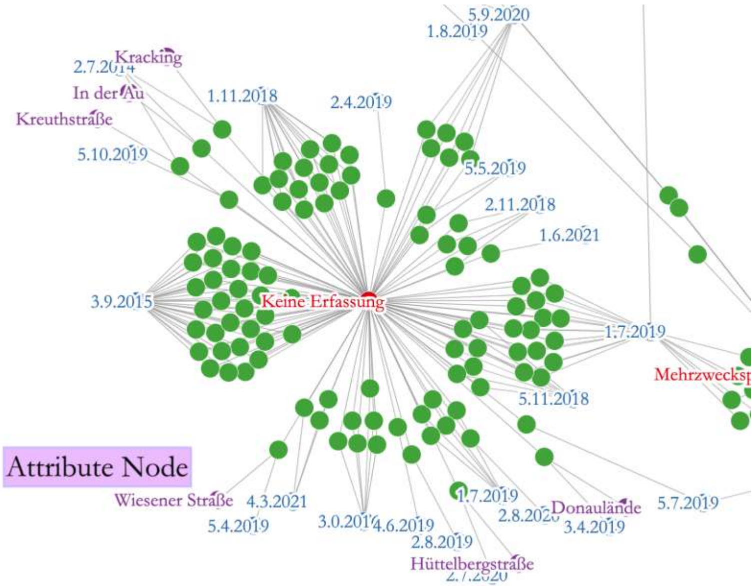


Figure 6.14: SeMaptics Attribute Node

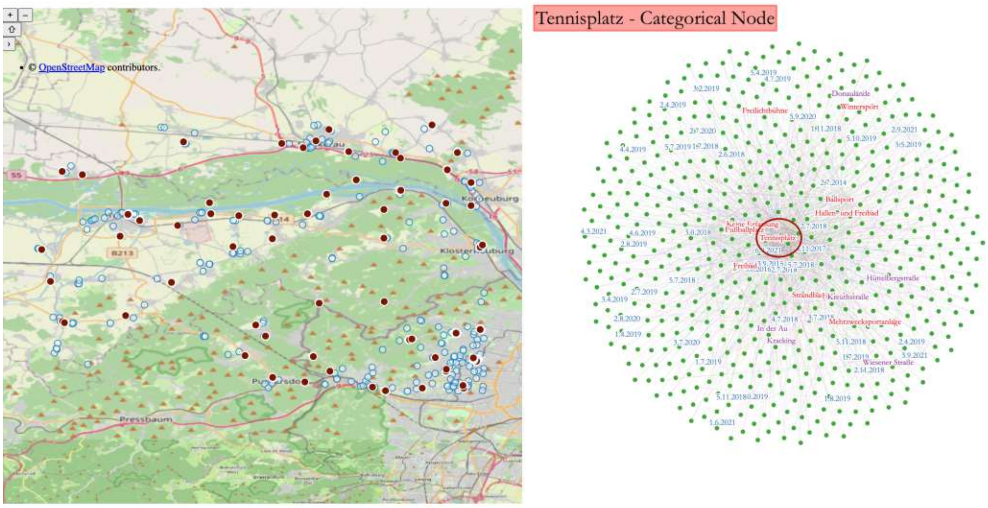


Figure 6.15: SeMaptics Categorical Node Selection force directed graph

Attribute nodes saturate the view port and do not provide a lot of graphical information since normally there is only one instance of each per spatial node. In this data scenario, only a few nodes contained a street name, hence we only see a few attribute nodes. However, we can already see in Figure 6.14 and Figure 6.11 that this nodes only exist on nodes that are related to the category 'Keine Erfassung - not defined'.

Data that is related to categorical nodes for this particular dataset is not spatially clustered (Figure 6.15). If the map had been designed in a traditional way, these relationships might not have been identified as relevant. This is a case where the spatial distribution is not related to the ontological relation. However, with this design approach, we can recognize that this category is relevant to the data set and project such relevance into the spatial dimension.

Any layout arrangement can be done on *SeMaptics* since it supports a SPA web architecture(Figure 1 & Figure 2).Many visual components can be rendered at the same time and they all can be connected to the central state, which is persistent among all visualizations. These visualizations stimulate both the *M-Path* through the use of the map and the graph force animation and its visual elements and the *P-path* through the graph systems[SZRM07].

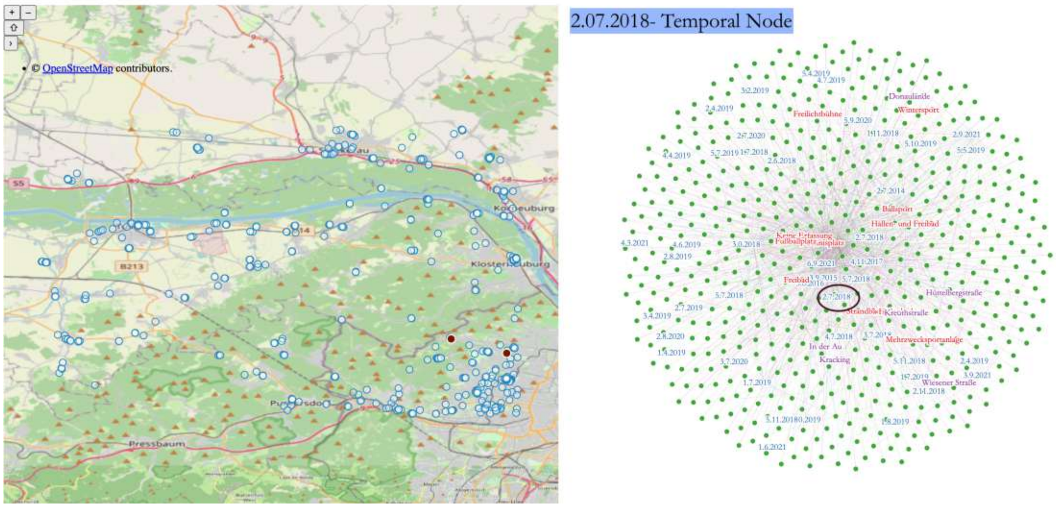
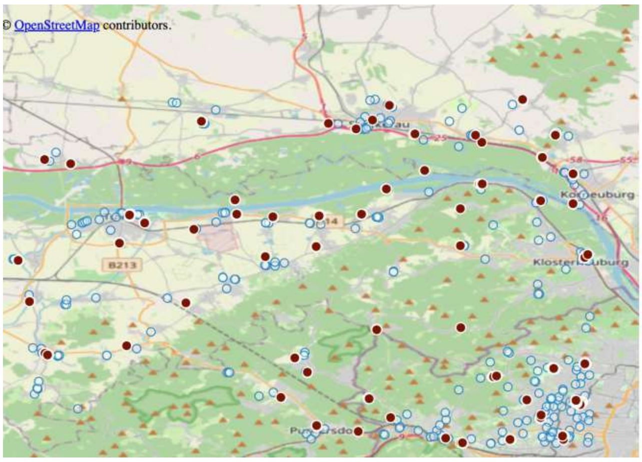


Figure 6.16: SeMaptics '2-7-2018' Node Selection



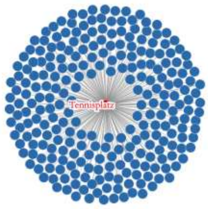


Figure 6.17: SeMaptics Graph Isolation

CHAPTER 7

Conclusion & Discussion

7.0.1 Discussion

Through the build and implementation of *SeMaptics* some points are important. i)Ontology definition ii) Data harmonization, iii)Query language selection, iv)graph visualization selection, v)Interaction design. *Ontology definition* is a step that needs to be considered at the beginning of the project. A characteristic that ontological-based maps have is that before starting the map maker needs to decide on how to define the ontological space. There are already some standards defined. However, it is easy to come up with personal ontological vocabularies that stem from the personal and cultural background of the map maker. Assuming the map maker is defining the space before making a map. A scenario could arise where the map maker starts the cartographic process from a predefined map space such as RDF vocabularies. ii) This leads to the second characteristic *Data harmonization*. This is not unique to graph-based maps all web maps that consume services have to have homologation processes, what stands out in a graph map is passing from a tabular logic to a graph logic. Graph logic presents a challenge not because it is a complex structure but because most pre-existing data comes in a tabular format Even if data comes in a graph structure there is the need for homologous vocabularies. This homologation process is an important step when making graph-based maps in order to facilitate a connected space. iii) Unlike RDBMS GDBMS do not have a standardized *Query language selection*, which is somewhat predefined by the chosen GDBMS that is selected. However, many databases have shared query languages and this boosts adoption and extension. As discussed earlier the query language selection ties to performance[Ang17], which carries on to the user experience. iv) *Graph visualization selection* defines the success of any tool that wants to reveal relevant data. It is important to choose a visualization tool that is flexible and that can scale well with the data. v) *Interaction design* is the synthesis of all previous steps since it allows for exploration and visual stimulation. Interaction favors an exploratory tool, which ranks higher in geovisualisations[MGP+04].

Data retrieval is always a topic in any cartographic project. The data source heavily influences the look and development of the project. For *SeMaptics* the influence was at the schema level. The schema design process is the result of the tabular parser (Figure 6.3). The fact that *SeMaptics* is based on a single point of source simplified this problem. However, projects which contain more than one data source might have to implement more strict approaches, where the schema is predefined before loading the data. Schema definition brings into play interesting elements such as temporal nodes. Seeing data from an ontological perspective removes some constraints of time linearity. Multiple dimensions can be projected into the same space and relate to features. This brings some discussion on the topic of relationship and classification. Since relationships have properties, we could also see relationships of spatio, temporal, spatio-temporal, or taxonomical nature. In *SeMaptics* only has spatio, temporal and taxonomic relations. Even though all nodes were classified under 'IsOf'. The nature of such relationships is different. Relation direction is also an important factor. One which does not end up being rendered on the graph view. This is a shortcoming of *SeMaptics* and is something that should be improved.

Neo4j had qualitatively speaking a good performance. However, batch processing was slow in comparison to relational databases. Specifically, batch processing of relationship creations. This is because creating a relationship requires two previous queries, which are the nodes, and then creating the indexing of the relation that binds such nodes takes time. There is room for improvement in terms of performance for the relationship creation processes. Additionally, making requests from the client to Neo4j was not very efficient for data creation. It is better to create data directly on the client of Neo4j. Because of the high waiting time on relationship creation, connections to the database get easily saturated, thus Neo4j drops the connection or needs to restart. This might be an issue if the tool is designed to create data based on any external service. From a query perspective, Neo4j performed efficiently and reliably. Connecting from a docker container was not an issue as long as there is a proper protocol configuration via Bolt or HTTP.

Displaying both dimensions on the screen, it is easy to identify ontological clustering points, which immediately catch the eye and invite the user to explore (Figure 7.1). These are relevant information hubs from the data set. However, when we click them we see that spatially no obvious pattern(figure 7.2). This is a scenario where this type of ontological exploration brings relevant behaviors forward and makes them more obvious. When seeing multiple variables (date of implementation and category) we also easily identify that two nodes hold a lot of relevance in the data set that was selected. i) Tennis court and ii) 6.9.2021. These two variables we might be able to visualize in a traditional system using a bi-variate choropleth map. The advantage of a graph system is that it does not require any specific type of training or long pondering to understand the relevance of the data. Additionally, bi-variate maps require more advanced GIS skills, while simple graph navigation feels intuitive and simple to understand. Another advantage that graph navigation has is that we can project and study multiple dimensions at the same time. In figure 7.2 we see that the tennis court is not only related to *6.9.2021* but also to 4 other relevant dates (02.09.2021, 4.11.2017, 3.09.2021, and 2.04.2019), where 4.11.2017 is also a node that relates to other clusters.

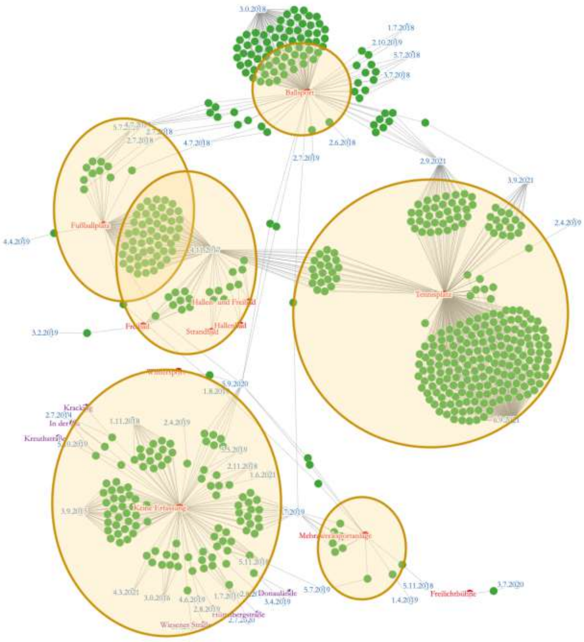


Figure 7.1: Ontological groups

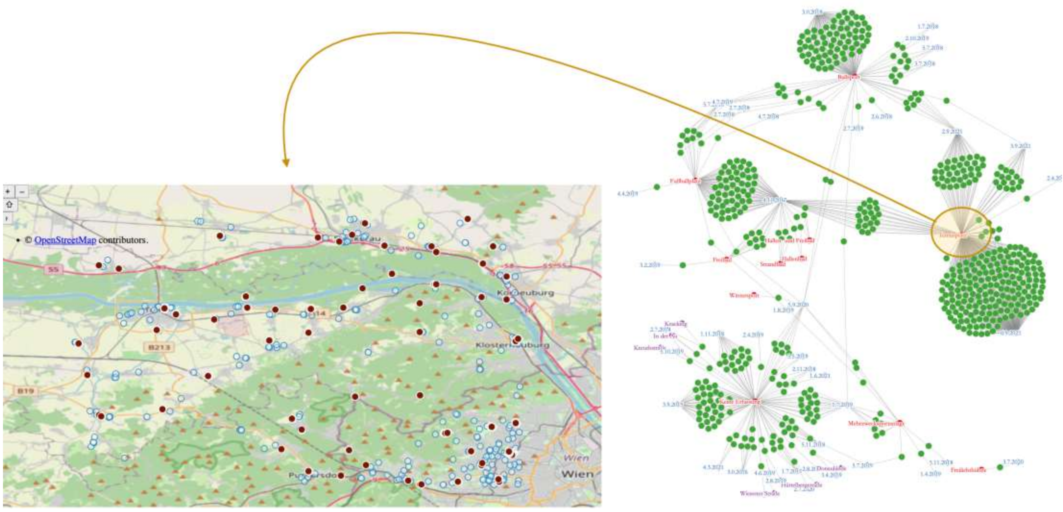


Figure 7.2: Ontological groups selected

Even though the graph is intuitive to navigate and it is easy to tell apart nodes from each other, there is a problem with node classification. A proposed solution is to have a type of classifier nodes that serve as element grouper to make grouping easy and intuitive as well (figure 7.3). This classifier might be able to suggest which categories we could make based on the available data. In a way making maps from graph data pushes the map-maker to design maps around the data that is relevant to the dataset.

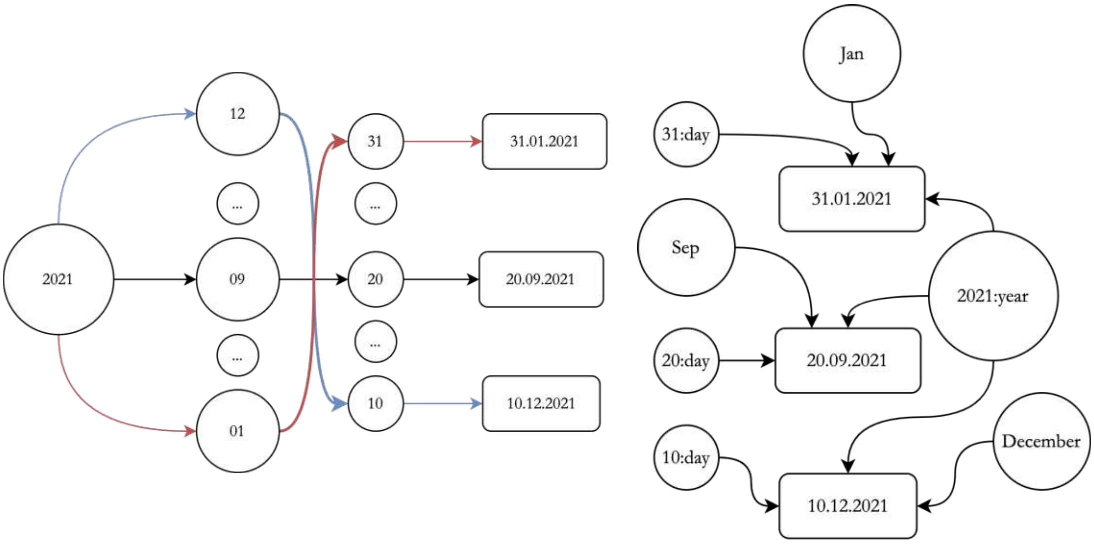


Figure 7.3: Ontological categories

SeMaptics interactions are elemental at the moment. Having a highlighting(filtering) function constitutes a core action of relevance [Sar96b]. Which is also a common task in cartography [RDS11]. Navigation and interaction, which impact the visualization level, also perform filtering tasks[Her00][HDM98] to balance the graphical load. All this tasks are executed in *SeMaptics* weather by the highlight selection or the isolated graph (figure 6.17). In regards to this point, *SeMaptics* delivers in bringing relevance forward to us. Other functions could be included to enhance more this capabilities such as zoom and pan. Context preservation is done well in *SeMaptics* the disconnected force graph (figure 6.10). However, context preservation would benefit also from the implementation of zoom and pan. On the other hand, the force-directed graph (figure 6.15) eliminates context by pulling elements away from the view. This is harmful for context preservation and overall relevance identification. However, this visualization is effective for isolated graph components, which could have an enhancing role of relevance. Selection reach is another area where relevance could be brought up currently selection degree is of the first order (figure 7.4). This means that if we select a category we only get the immediate nodes related to this node. Extending the filtering degree would most likely have a meaningful impact on data exploration, which benefits relevance identification.

The ability to navigate the ontological space and geo visualize it falls into both the *cognitive model* [Sar96b] and the *episode model* [Bel95]. Since it is both an interactive process and a process where interactions can be seen as episodes or snapshots. This blends well with the relevance of the information retrieval nature. If *SeMaptics* would be part of a larger service chain we could also consider the *Acquisition-Cognition-Application* model[Sar96a]. However at the moment *SeMaptics* has only *Acquisition and Cognition*. At present, the user is at the center of relevance identification. However, if we were to compute certain parameters such as the number of connections, page rank, betweenness, and other network statistics, we could put machines at the center of relevance. Since one of the main advantages of a graph map system is that it is machine-readable. With correct RDF vocabulary interactions, it is possible to develop map systems that identify relevance firsthand. This concept enables machines to define maps by themselves and have an interface of understanding between machine and human understatement in regards to what a map is.

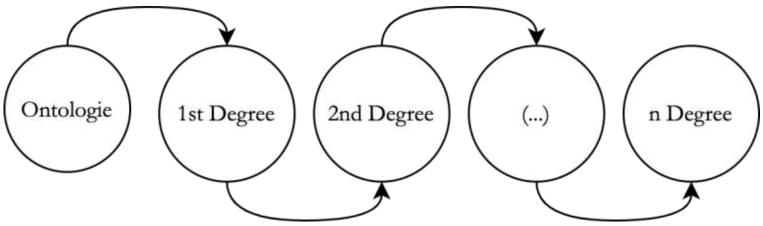


Figure 7.4: Ontological groups selected

Using a rich visualization library such as d3js helps when designing any project that intends to show relevance. However, implementing native JavaScript libraries in higher languages such as Typescript presents a technical challenge. *SeMaptics* has a custom built Typescript upgrade. There is little to no documentation on many of the network visualizations that D3 offers as examples, this is a challenge for any web map maker. However, D3js is an open source project which comes with the advantage of code transparency. Understanding what any library is doing when a method or function is called helps any web map designer create better tools. Even though documentation is limited in D3js, flexibility is high. However, other tools could be equally as suitable for graph mapping projects. The following aspects could define the usability of any graph library. i) Environment requirements, ii) Interaction support in the viewport, iii)graphical visualization catalog, iv)build efficiency, v)graphical primitives.

The environment requirements for D3js are any environment that supports JavaScript. However, as mentioned before a custom upgrade was necessary to overlay D3js on Typescript. The interaction support for d3js was good since any selected DOM element on the view-port can be mapped out, which allows for a convenient state change. This interaction support is crucial to address relevance revelation. The graphical visualization catalog for d3js is extensive and creative since it stems from a vast community of contributors. However, one must not assume that all examples are easily reproducible since sometimes the example implements custom functions, resulting in confusing code. Additionally, all examples are written in JavaScript, which has the drawback of not being a strict type of language, which adds a layer of ambiguity to an example. A pitfall when implementing D3js. Build efficiency for *SeMaptics* was snappy and responsive. When manually zooming with browser controllers the graph rendering slows down considerably. Considering that *SeMaptics* was rendering two full graph networks with labels(4 graph networks- 2000 nodes) and an isolated network with around 100 nodes considering labels. This first iteration proved that for isolated regions with 2000 plus nodes graph mapping is possible with current technology (software and hardware).

Psychological relevance changes as the user receive new pieces of information[Wil73]. *SeMaptics* has a role in this relevance, since every time the user interacts and explores spatial entities on the graph new questions arise that relate to dimensional (What, when, where, and who). Of course, there is always the current situation of the user and his/her personal preferences. *SeMaptics* allows for easy data navigation and exploration of ontologies. Since physiological relevance places items and personal views at the center, we can think of ontologies as information items that interact with the user. This itemization of relevance is a powerful concept that can be further explored.

Geographical relevance evaluates the object within the information system [dSR12] and *SeMaptics* allows for a visual evaluation of node relevance on any loaded information space. Also, this includes a dynamic state which changes with any given interaction. *SeMaptics* could also be extended to support on the fly data uploads from a traditional RDBMS, which builds upon dynamic states. *SeMaptics* directly address *scope* and *dynamism* when speaking of geographical relevance criteria points.

7.0.2 Conclusion

The visual and state connection of both the ontological and spatial dimension with event-based interaction allows us to explore hidden patterns and allows for efficient semantic reads where relevance is present. Where a semantic read can be interpreted as the overall look of interconnected ontologies towards a specific data point. A semantic read is currently not available in the traditional map-making process.

Mapping ontological spaces require to have a discrete concept definition. Existing vocabularies can serve as a guide to defining local ontologies. However, there are still no fixed methods for defining ontologies that stem from traditional tabular data. Once an ontological space has been constructed it is necessary to build proper relations that allow for SOP queries to retrieve information from a knowledge graph. Additionally, to map ontological interactions and relations to the spatial dimension, we need interface ontologies that are overlaid in both dimensions (points, lines, polygons). These overlapping ontologies are what allow the projection of one space into another.

The geographical relevance criteria[dSR12] [IS98] [RSPF16] found in ontological mapping is the following: i)*depth* with the display of all data and how it relates to other ontologies, with the added visual navigation. ii) *specificity*, with well-defined ontologies that allow for the mapping of a discrete dimension that projects to a continuous dimension(space)[JT22]. iii) *availability*, through state persistence and the presentation of the whole data set visually. iv)*accuracy*, since the ontological space is discrete thus accurate descriptions and data points can be derived. This of course is always dependent on the quality of the data source, which is also a criterion of geographical relevance. v)*tangibility* is another criteria that bonds well to a discrete mapped space. vi)*accessibility*, allowing interactions open data to be accessed and provide transparency. However, some criteria might only be reached at full when talking of data that is linked directly from the semantic web. vii) *dynamism* since data in any graph visualization is flexible to change and recording change can be done within the same ontological dimension. viii) *curiosity* is fed through semantic mapping where creativity is stimulated[KAF14] ix) *spatial proximity* is seen directly on the map. x)*Visibility* is present in both graph and map visualization and interaction services such as highlight and filtering. *Cluster and Co-location* can also be derived from graph patterns. The other criteria (*affectiveness, effectiveness, currency, temporal proximity*, and *verification*) are criteria that could also be proved achievable with the use of graph mapping. However, with the current reach of this study, it is not possible to make such statements yet.

SeMaptics most likely helps the user design a map in a creative way, since network data is shown and easily read. Creativity is defined by the ability that a person has to connect two distant concepts together[KAF14]. Using a network approach could lead to more creative map creation. Since today we are dependent on tabular data exploration or exploration through predefined tools within current GIS clients (QGIS, ArcGISPro, GRASS, etc ...). This creativity not only is sourced from humans but computers themselves.

Current web-based technologies are sufficient to combine ontological and spatial mapping. SPA design pattern fits well in graph mapping, due to its flexibility in scale and scope. Graph mapping is a young and upcoming field and many branches of such can be studied. Having a better understanding of how can we integrate ontological mapping into current map-making technical is important. The barrier of entry is still high. An extensive technical background is required to build such ontology-based projects and to understand previous work made in the field. Map makers need to find a way to lower this barrier to increase the level of adoption of all graph-related technologies.

It is possible to make web-mapping tools using only GDBMS. However, much technology is already developed for RDBMS that benefits from GDBMS. The desired pattern would be a hybrid one where both technologies are involved and the core benefits from both technologies can be used.

The graph visualization process proposed by Romero (2018) [GRMSOG18] serves as a good guideline to design ontological mapping as well. All 4 steps (Data retrieval, building, calculations, layout, and rendering) translate well to the graph building process.

In this dataset, some ontological patterns do not translate to spatial patterns. There are some instances where relevance is hidden in semantics and relevance can be identified through the use of graph tools. *SeMaptics* works as a proof of concept on how can we merge both spatial and ontological patterns.

Something that is not addressed by *SeMaptics* is the aesthetic design of both the map and the graph visualizations. These elements influence relevance perception. It would be important to see the impact that these factors have on relevance perception with a graph structure. However, even with the basic visual components, we can already access an additional layer of information that is traditionally not present in maps.

Relevance is an elusive concept that changes scope depending on the who, where, what, how, and when the data is accessed. Graph mapping opens a door of possibility having multiple dimensions (time, space, methods, technology, and user) interacting in the same space. However, there is a challenge in commonality and mutual understatement among all ontologies. Bridging this communication gap makes ontological mapping a power full approach to having context embedded not only in our perceived context. But also in the computer's context.

7.0.3 Future Work

It would be worth testing the other graph systems proposed by Ruggles (1997)(figure 4.7) and see how they impact relevance in comparison to the *complete connected-altered node, line, and geography* system implemented in *SeMaptics* [RA97]. How the graph is presented to the user translates into data perception and thus into data definition.

The presented exercise focused on the most simple spatial form (Point). However, ontologies are also embedded in higher forms such as lines and polygons. A future step would be to use graph mapping on these forms and study how topological interactions affect the mapping of the ontological space.

The type of relations used in *SeMaptics* where of type boolean. However, relationships can have other natures and carry meaning and weight to the link. Web mapping that takes into account the quality and nature of these relations is also likely to dig more into hidden relevance in the ontological dimension. Additionally, *SeMaptics* lacks cardinality in the display of its relationships, which limits the semantic value of such relationships. Extending cardinality display and understanding the impact that it has is a future step.

Cross-domain data integration such as proposed by Huang (2020), would be the next step to using graph mapping to find relevance[HKMH20]. This means using data from the semantic web to do semantic mapping and project such mapping into the spatial domain. This means a more formal implementation of RDF vocabularies and defining mapping modules for these domains such as done by Mai (2022) in narrative cartography[MHC+22].

The type of graph selection has an impact as well on relevance perception. More research should be done to understand how this relevance perception is affected according to graph visualization (force directed, h-tree, ballon-view, Sugiyama, hyperbolic tree, radial-view, etc ...). The graphical elements chosen for any graph visualization also impact relevance. Such elements include rendered primitives, shapes, sizes, hue, value, and width, among others. All visual elements influence relevance perception and there is still much unknown on how geographical relevance is affected by these elements.

Graph mapping could make an impact on creative map definition. Since data that is shown in a graph is connected it may lead to the correlation of distant nodes, which is a core definition of creativity[KAF14]. It is worth exploring how can we map spaces if we start with data that is projected by a knowledge network rather than approaching data with a preconceived notion. Since in some cases the map theme is defined by the availability of the data. However, the approach might change if the data is presented on graph visualization.

The way a user reads a graph, whether human or machine, defines relevance. Looking under the scope of psychological relevance [Wil73]. Doing further research on what calls the attention of the user in a graph visualization that projects into the spatial dimension add value to how we interact with relevance visually. Graph navigation patterns may be influenced by axioms of relevance, which may reveal the user's nature. Additionally, a higher degree search should also prove important in finding underlying data semantics (figure 7.4). In this graph, other types of nodes were not added to avoid visual cluttering. Adding more information enriches the data and increases semantics but reduces legibility from a human perspective. It is important to better attain this balance, where we can have rich semantics while preserving legibility for humans.

Using Harter's second definition of relevance in section 3.4 as a north star in contextual relevance. The data loaded into *SeMaptics* is as relevant as its connection to the environment. This means that for graph mapping to be effective data must be thoroughly enriched. The example presented in this work is limited to the data selected. However, we must think that this data is highly connected to other ontologies and there must be further development on the idea of having web semantic connections in this kind of map.

There are still many research areas that can be explored in graph mapping. Current technologies allow us to explore the ontological space and current vocabularies open the opportunity for deterministic space mapping. Aesthetics, functionality, interactions, and data sources all affect relevance. When we project the spatial into the ontological dimension we find new trends in data that reveal unexplored semantics. These underlying semantics are crucial to further understanding the nature of data spaces.

List of Figures

Bibliography

[ABG$^+$15] Pouria Amirian, Anahid Basiri, Guillaume Gales, Adam Winstanley, and John McDonald. The next generation of navigational services using openstreetmap data: The integration of augmented reality and graph databases. *Lecture Notes in Geoinformation and Cartography*, 0:211–228, 2015.

[AG08] Renzo Angles and Claudio Gutierrez. Lncs 5318 - the expressive power of sparql, 2008.

[Akh17] Akhil. Directed and Undirected Graphs. http://abiasforaction.net/cap-theorem/, 2017. [Online; accessed 31-July-2022].

[Ang17] Foundations of modern query languages for graph databases. *ACM Computing Surveys*, 50, 9 2017.

[BAT14] Gavin Bierman, Martín Abadi, and Mads Torgersen. Understanding typescript. pages 257–281, 2014.

[BC14] Fan Bao and Jia Chen. Visual framework for big data in d3.js. IEEE, 5 2014.

[Bel95] Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems, 1995.

[BEV] BEV. Bev. https://data.bev.gv.at/geonetwork/srv/spa/catalog.search#/metadata/c8ffe3f7-1cd4-4449-9bd3-511d9f2a9221. [Online; accessed 15-April-2022].

[Biz09] Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5:1–22, 7 2009.

[BLK$^+$16] Nikos Bikakis, John Liagouris, Maria Krommyda, George Papastefanatos, and Timos Sellis. graphvizdb: A scalable platform for interactivelarge graph visualization. 2016.

[Bos12] Mike Bostock. D3.js - data-driven documents, 2012.

[Bos20a] Armand Boschin. Torchkge: Knowledge graph embedding in python and pytorch. In *International Workshop on Knowledge Graph: Mining Knowledge Graph for Deep Insights*, Aug 2020.

[Bos20b] Armand Boschin. Torchkge: Knowledge graph embedding in python and pytorch. 9 2020.

[Boy82] Bert Boyce. Beyond topicality: A two stage view of relevance and the retrieval process. *Informorion Procesring Monogemenf*, 18:105–109, 1982.

[CDW+19] Yijian Cheng, Pengjie Ding, Tongtong Wang, Wei Lu, and Xiaoyong Du. Which category is better: Benchmarking relational and graph database management systems. *Data Science and Engineering*, 4:309–322, 12 2019.

[Coo73] William S Cooper. On selecting a measure of retrieval effectiveness. *Journal of the American Society for Information Science*, pages 87–100, 3 1973.

[DB19] Daniel Danielecki and Thomas Beekman. Security first approach in development of single-page application based on angular, 2019.

[DJS14] Blagoj Delipetrev, Andreja Jonoski, and Dimitri P. Solomatine. Development of a web application for water resources based on open source software. *Computers and Geosciences*, 62:35–42, 1 2014.

[DM16] Jaudete Daltio and Claudia M Bauzer Medeiros. Hydrograph: Exploring geographic data in graph databases hydrograph: Explorando dados geográfi cos em bancos de dados de grafos. *Brazilian Journal of Cartography*, 68:1181–1189, 2016.

[DR22] Sumit Dutta and Swarup Roy. Complex network visualisation using javascript: A review. volume 431, pages 45–53. Springer Science and Business Media Deutschland GmbH, 2022.

[dSR12] Stefano de Sabbata and Tumasch Reichenbacher. Criteria of geographic relevance: An experimental study. *International Journal of Geographical Information Science*, 26:1495–1520, 8 2012.

[DSS93] Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation representation? 1993.

[FB18] Diogo Fernandes. and Jorge Bernardino. Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. In *Proceedings of the 7th International Conference on Data Science, Technology and Applications - DATA,*, pages 373–380. INSTICC, SciTePress, 2018.

[FM06] M. Fischer and Manfred. *Spatial Analysis and GeoComputation*. Springer, 2006.

[FRW91] Thomas M J Fruchterman, Edward M Reingold, and John Wiley. Graph drawing by force-directed placement. *Software Practice And Experience*, 21:1129–1164, 1991.

[Glo21] Glouwa. D3-hypertree, 2021.

[GPE21] W3C2022 Gordon Prud'hommeaux Eric. Matching graph patterns against stem graphs, 2021.

[GRMSOG18] Juan Gómez-Romero, Miguel Molina-Solana, Axel Oehmichen, and Yike Guo. Visualizing large knowledge graphs: A performance analysis. *Future Generation Computer Systems*, 89:224–238, 12 2018.

[Har92] Stephen P Harter. Psychological relevance and information science. *Journal of the American Society for Information Science*, 43:602–615, 1992.

[hASF22] he Apache Software Foundation. Apache AGE. https://age.apache.org, 2022. [Online; accessed 15-June-2022].

[HDM98] Ivan Herman, Maylis Delest, and Guy Melancon. Tree visualisation and navigation clues for information visualisation. 17:153–165, 1998.

[Her00] Graph visualization and navigation in information visualization: A survey. *Graph Visualization and Navigation in Information Visualization: A Survey*, 6:24–43, 2000.

[HKMH20] Weiming Huang, Khashayar Kazemzadeh, Ali Mansourian, and Lars Harrie. Towards knowledge-based geospatial data integration and visualization: A case of visualizing urban bicycling suitability. *IEEE Access*, 8:85473–85489, 2020.

[HMBB17] Barbara Hofer, Stephan Mäs, Johannes Brauner, and Lars Bernard. Towards a knowledge base to support geoprocessing workflow development. *International Journal of Geographical Information Science*, 31:694–716, 4 2017.

[HP18] Olaf Hartig and Jorge Pérez. Semantics and complexity of graphql. pages 1155–1164. Association for Computing Machinery, Inc, 4 2018.

[HSoM91] Ivan Herman, Ieee Computer Society, Guy Melanc Ë on, and M Scott Marshall. Graph drawing by force-directed placement. *Software Practice And Experience*, 21:1129–1164, 1991.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[HSSC08] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. 1 2008.

[Ing96] Peter Ingwersen. Cognitive perspectives of information retrieval interaction: Elements of a cognitive ir theory. *Journal of Documentation*, 52:3–50, 1996.

[IS98] Carol L Barry I and Linda Schamber. Users' criteria for relevance evaluation: A cross-situational comparison. 34:219–236, 1998.

[JG22] Markus Jobst and Georg Gartner. Accessing spatial knowledge networks with maps. *International Cartographic Conference*, 8, 2022.

[JT22] Markus Jobst and Fischer Tatjana. Approaching a common conscious dataspace from a data provider perspective – requirements and perspectives. pages 333–343, 2022.

[KAF14] Yoed N. Kenett, David Anaki, and Miriam Faust. Investigating the structure of semantic networks in low and high creative persons. *Frontiers in Human Neuroscience*, 8, 6 2014.

[Ken18] Alexander J. Kent. Form follows feedback: Rethinking cartographic communication. *Westminster Papers in Communication and Culture*, 13:96–112, 10 2018.

[KF19] Yoed N. Kenett and Miriam Faust. A semantic network cartography of the creative mind. *Trends in Cognitive Sciences*, 23:271–274, 4 2019.

[KPS19] Ales Komarek, Jakub Pavlik, and Vladimir Sobeslav. Time-based change visualization techniques survey. *Journal of Engineering and Applied Sciences*, 14:10538–10543, 2019.

[Li20] Zhilin. Li. Information theory of cartography: A framework for entropy-based cartographic communication theory. volume 43, pages 11–16. International Society for Photogrammetry and Remote Sensing, 8 2020.

[Mat22] Mathworks. Directed and Undirected Graphs. https://www.mathworks.com/help/matlab/math/directed-and-undirected-graphs.html, 2022. [Online; accessed 19-July-2022].

[MBK] Krishna Mangaladevi, Wouter Beek, and Tobias Kuhn. Understanding knowledge networks. pages 38–49.

[Med62] Sarnoff A Mednick. The associative basis of the creative process. *Psychological Review*, 69:220–232, 1962.

[MGP+04] Alan M Maceachren, Mark Gahegan, William Pike, Isaac Brewer, Guoray Cai, Eugene Lengerich, and Frank Hardisty. Geovisualization for knowledge construction and decision support. 2004.

[MHC+22] Gengchen Mai, Weiming Huang, Ling Cai, Rui Zhu, and Ni Lao. Narrative cartography with knowledge graphs. *Journal of Geovisualization and Spatial Analysis*, 6, 6 2022.

[MM07] David Mountain and Andrew MacFarlane. Geographic information retrieval in a mobile environment: Evaluating the needs of mobile individuals. *Journal of Information Science*, 33:515–530, 10 2007.

[NMTG16] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. volume 104, pages 11–33. Institute of Electrical and Electronics Engineers Inc., 1 2016.

[Num] NumPy. NumPy. `https://numpy.org`. [Online; accessed 10-July-2022].

[OR21] Marco Oliveri and Tumasch Reichenbacher. Suitability of colour hue, value, and transparency for geographic relevance encoding in mobile maps. *Abstracts of the ICA*, 3:1–2, 12 2021.

[ove21] Stack overflow. Developer Survey 2021. `https://insights.stackoverflow.com/survey/2021#overview`, 2021. [Online; accessed 25-July-2022].

[PGT10] Akshara Pande, Manjari Gupta, and A K Tripathi. Dnit-a new approach for design pattern detection. pages 545–550, 2010.

[Pur98] Helen Purchase. Which aesthetic has the greatest effect on human understanding? pages 248–261, 1998.

[RA97] Amy J. Ruggles and Marc P. Armstrong. Toward a conceptual framework for the cartographic visualization of network information. *Cartographica*, 34:33–47, 1997.

[RDS11] Tumasch Reichenbacher and Stefano De Sabbata. Geographic relevance – different notions of geographies and relevancies. *SIGSPATIAL special issue on Geographic Information Retrieval*, 3, 01 2011.

[RP95] Arthur H Robinson and Barbara Bartz Petchenik. The map as a communication system. pages 7–15, 1995.

[RSPF16] Tumasch Reichenbacher, Stefano De Sabbata, Ross S. Purves, and Sara I. Fabrikant. Assessing geographic relevance for mobile search: A computational model and its validation via crowdsourcing. *Journal of the Association for Information Science and Technology*, 67:2620–2634, 11 2016.

[Sar96a] Tefko Saracevic. Modeling interaction in information retrieval (ir): a review and proposal. volume 33, pages 3–9, 1996.

[Sar96b] Tefko Saracevic. Relevance reconsidered. pages 201–218. School of Communication, Information and Library Science, 10 1996.

[SFMJ10] Zareen Syed, Tim Finin, Varish Mulwad, and Anupam Joshi. Exploiting a web of semantic data for interpreting tables. pages 26–27, 2010.

[SFV19] A L Sayeth Saabith, Mmm Fareez, and T Vinothraj. Related papers python current trend applications-an overview popular web development frameworks in python. *International Journal of Advance Engineering and Research Development*, 6, 2019.

[SMR15] Stefano De Sabbata, Stefano Mizzaro, and Tumasch Reichenbacher. Geographic dimensions of relevance. *Journal of Documentation*, 71:650–666, 2015.

[SSJK02] Kee Soh, Tonya L Smith-Jackson, and Brian M Kleiner. Developing outdoor map design guidelines using a real-world wayfinding task, 2002.

[SW86] Dan Sperber and Deirdre Wilson. *Relevance : Communication and Cognition.* 1986.

[SZRM07] O. Swienty, M. Zhang, T. Reichenbacher, and L. Meng. Establishing a neurocognition-based taxonomy of graphical variables for attention-guiding geovisualisation. volume 6751, page 675109. SPIE, 6 2007.

[TS19] Rajesh Tamilmani and Emmanuel Stefanakis. Modelling and analysis of semantically enriched simplified trajectories using graph databases. *Advances in Cartography and GIScience of the ICA*, 1:1–8, 7 2019.

[TVSV18] Ruben Taelman, Miel Vander Sande, and Ruben Verborgh. GraphQL-LD: Linked Data querying with GraphQL. In *Proceedings of the 17th International Semantic Web Conference: Posters and Demos*, October 2018.

[UMS+09] P. Utrilla, C. Mazo, M. C. Sopena, M. Martínez-Bea, and R. Domingo. A palaeolithic map from 13,660 calbp: engraved stone blocks from the late magdalenian in abauntz cave (navarra, spain). *Journal of Human Evolution*, 57:99–111, 2009.

[VU18] Dalia E. Varanka and E. Lynn Usery. The map as knowledge base. *International Journal of Cartography*, 4:201–223, 5 2018.

[Wil73] Patrick Wilson. Situational relevance. *Inform Stor Retr*, 9:457–471, 1973.

[WLoSEoPL09] Eric. Wohlstadter, ACM Digital Library., ACM Special Interest Group on Software Engineering., and ACM Special Interest Group on Programming Languages. *Proceedings of the 8th workshop on Aspects, components, and patterns for infrastructure software.* ACM, 2009.

[WS95] Deirdre Wilson and Dan Sperber. *Relevance Theory.* Handbook of Pragmatics. Oxford, 1995.

[Zha14] Ziqi Zhang. Towards efficient and effective semantic table interpretation. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, pages 487–502, Cham, 2014. Springer International Publishing.

Appendix

Figure 1: SeMaptics Flexible Layout

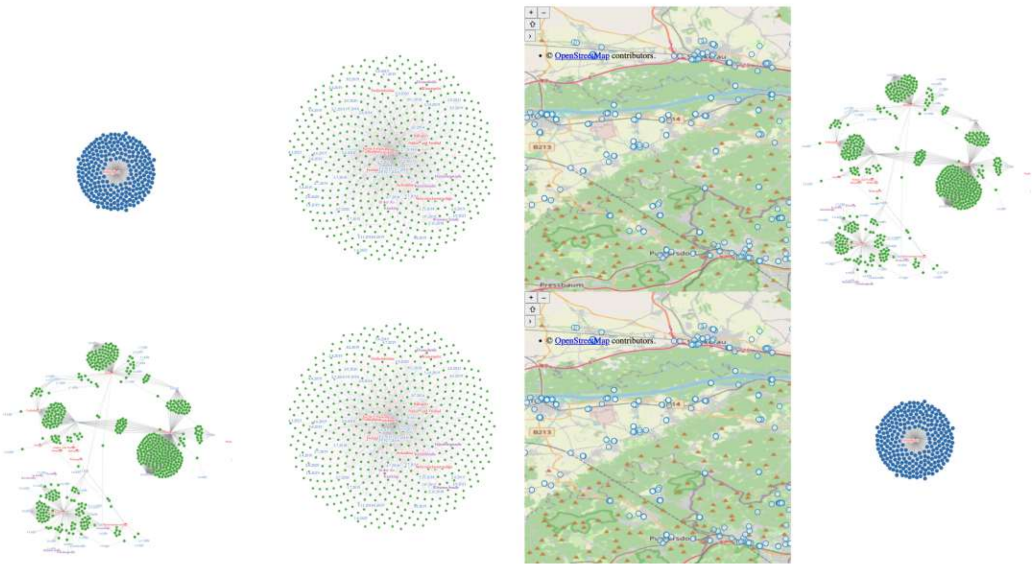


Figure 2: SeMaptics Flexible Layout

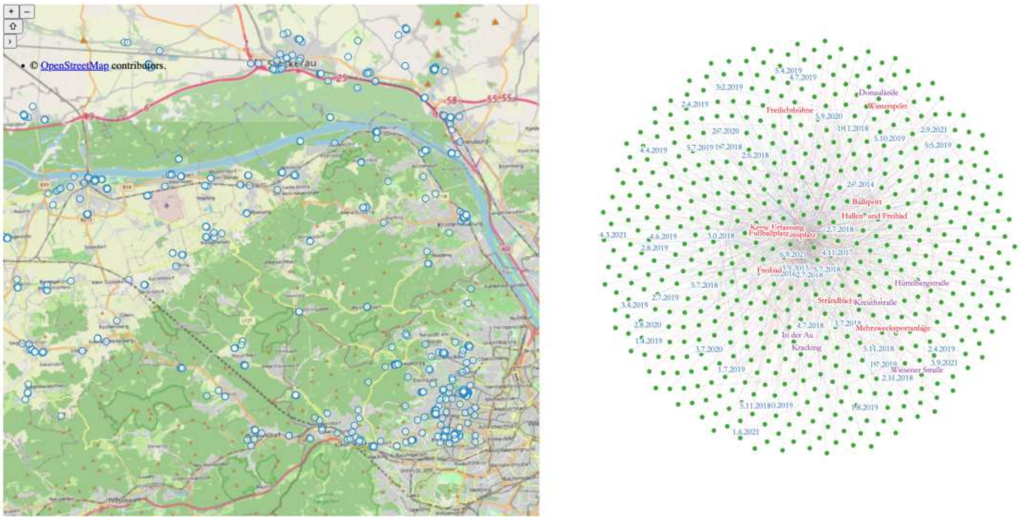


Figure 3: SeMaptics Supports other projections such as EPSG:31255

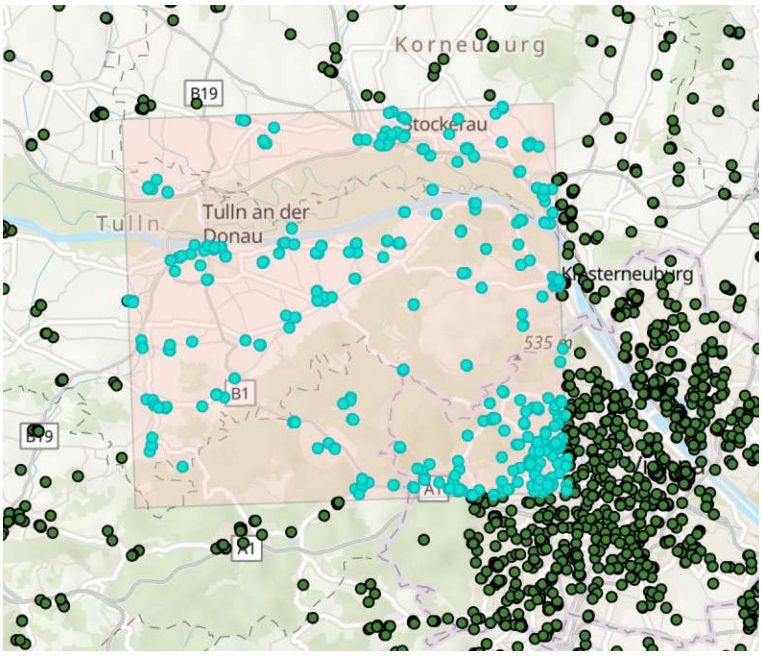


Figure 4: SeMaptics Data selection – Visualized on ArcGISPro