Enhancing the Usefulness of Open Governmental Data with Linked Data Viewing Techniques

Conference Paper · January 2019

5 authors, including:

Stanislav Ronzhin
University of Twente
6 PUBLICATIONS   **2** CITATIONS

SEE PROFILE

Erwin Folmer
University of Twente
60 PUBLICATIONS   **105** CITATIONS

SEE PROFILE

Wouter Gerardus Beek
Vrije Universiteit Amsterdam
51 PUBLICATIONS   **253** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Kadaster Data Platform View project

IWEI2013 View project

Enhancing the Usefulness of Open Governmental Data with Linked Data Viewing Techniques

E. Folmer
Kadaster & University of Twente
erwin.folmer@kadaster.nl

W. Beek
Kadaster, University of Twente,
Triply & VU University Amsterdam
wouter@triply.cc

L. Rietveld
Kadaster & Triply
laurens@triply.cc

S. Ronzhin
Kadaster & University of Twente
s.ronzhin@utwente.nl

R. Geerling
Kadaster & Saxion
rutgergeerling91@gmail.com

D. den Haan
Kadaster & University of Twente
d.denhaan@utwente.nl

Abstract

Open Governmental Data publishing has had mixed success. While many governmental bodies are publishing an increasing number of datasets online, the potential usefulness is rather low. This paper describes action research conducted within the context of the Dutch Cadastre's open data platform. We start by observing contemporary (Dutch) Open Data platforms and observe that dataset reuse is not always realized. We introduce Linked Open Data, which promises to deliver solutions to the lack of Open Data reuse. In the process of implementing Linked Data in practice, we observe that users face a knowledge and skill and that contemporary Linked Open Data tooling is often unable to properly advertise the usefulness of datasets to potential users, thereby hampering reuse. We therefore develop four components for Linked Data viewing to enhance the current situation, making it easier to observe what a dataset is about and which potential use cases it could serve.

1. Introduction

An increasing number of governmental organizations is publishing Open Data online [24]. However merely publishing datasets online does not guarantee use [22, 23, 24]. The Land Registry and Mapping Agency of the Netherlands ('Kadaster' in Dutch) publishes large authoritative geospatial datasets, including several key registers of the Dutch Government. This includes a detailed description of the full topography of the Netherlands, as well as registrations of all the addresses and buildings in the Netherlands. These data assets are published in the online PDOK data catalogue (https://data.pdok.nl). PDOK is a data publication service that exposes over 130 geospatial datasets form various Dutch governmental institutes. Together, these datasets include descriptions of

hundreds of millions of geospatial objects. On a yearly basis, PDOK receives billions of hits (2.153.892.039 hits in Q1 of 2018 alone), emphasizing the popularity of the platform and the data on it. However, if we further analyse these hits, it is seen that from the 130 datasets only 5 are responsible for 84% of the total number of hits [9].

The number one dataset, the Web Map version of the official Topographical Map of the Netherlands (BRT Achtergrondkaart), is responsible for 34% of the total number of hits (726.868.918 hits for Q1 of 2018), followed by the Building and Address register (BAG) which is responsible for 25% of the total number of hits (537.541.269 hits). therefore, over half (59%) of the total number of hits is caused by these two datasets alone, which shows that publishing as much datasets as possible does not necessarily improve Open Data use. Another example is the official Open Data platform of the Dutch government: data.overheid.nl. In total, over 12,000 datasets are published as Open Data on that platform, yet only 82 datasets are classified as 'high value datasets'.

Moreover, if these 82 datasets are further inspected, only a handful of the datasets found on data.overheid.nl cover the entirety of the Netherlands and are regularly updated, e.g. the National Commercial Register.

Therefore, quantity should not be the priority of an Open Data platform. Instead, the focus should be placed on publishing datasets that have high value and (re)usability for users [23]. Also, platforms should improve the accessibility and usability of their open data [22, 23], e.g. by creating functionalities and services.

Linked Open Data [13] provides promises of increased accessibility, usability and value of open data [1]. By representing the data in a standardized way, different components can be used for publishing, storing, retrieving, reusing, integrating and analysing

HICSS

the data [1]. The implementation of Linked Open Data and the adoption by users is however hindered by barriers [19, 20, 21] which will be addressed in the next section.

2. Approach

The new Kadaster Data Platform (KDP) is using Linked Open Data to improve the usefulness of the datasets it is publishing. Data is not only published for the Dutch Cadastre, but also as a shared service for other Dutch governmental organisations. E.g., the first Linked Open Data release of the spatial-statistical dataset of Dutch neighbourhoods ("Kerncijfers wijken en buurten" in Dutch) published by the Dutch Central Bureau of Statistics (https://cbs.nl).

Unfortunately, Linked Data is not an out-of-the-box solution that can be directly applied in the organization. For this reason, KDP is using action research [21] to implement Linked Data support over time.

Linked Data is a collection of best practices on how to publish data on the Web [4,12]. The idea of Linked Data is that data is published on the Web, so that it can be explored by both persons and machines. Rather than being stored in a traditional relational database, Linked Data is stored in a graph-based data model, typically indexed by a triple store, using standardized serialization formats like Turtle and RDF/XML [12]. The use of URIs/IRIs as identifiers in the data allows for the creation of links between datasets, providing context to the data, and thereby improving its understandability and usability [1]. Furthermore, it allows for the discovery of new data by potential users. SPARQL is a standardized query language that can be used to answer complicated questions over one or more Linked Datasets. In the case of the Kadaster Data Platform, the Linked Data is also Open Data. Tim Berners-Lee has created the 5-star Linked Open Data model, to indicate which criteria must be met by Linked Open Data:

*	Available on the web under an open license.
**	Available as machine-readable structured data
***	Available as machine-readable data, but in a non-proprietary format (e.g. CSV instead of XSLT)
****	Using open web standards (IRIs for identifiers, RDF for data model, SPARQL for querying)
*****	Linked to other Linked Open Data on the web

In practice, there is a clear distinction and a big implementation gap between the first three and the last two stars [5] because it requires the use of Linked data. Also, the adoption of Linked (Open) Data is relatively slow [19]. The main barrier is the lack of knowledge and skills of users [20, 21]. In addition, users are often unable to find data(sets) of interest, since it is difficult to relate published datasets to their concrete use case [19]. Finally, SPARQL is a versatile and expressive query language, but also has a steep learning curve [20].
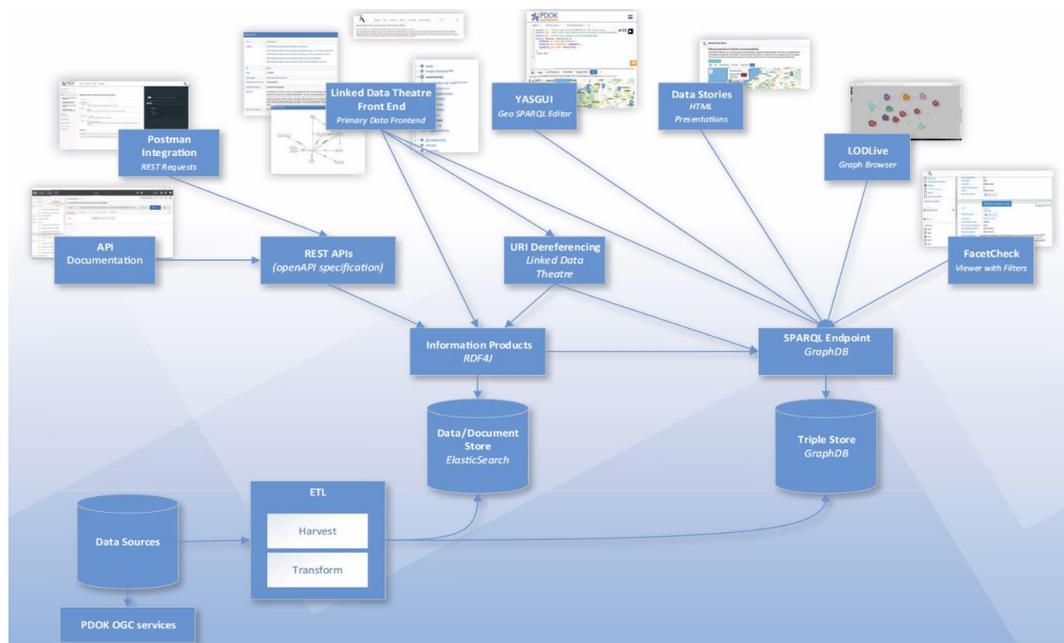

Figure 1. The Kadaster Dataplatform Architecture

To improve upon the above described situation, the KDP has developed four methods for Linked Data viewing:

- Data Stories that provide an overview of interesting queries that can be performed over a Linked Data and that present use cases.
- FacetCheck that allows users to browse a Linked Dataset by interacting with a set of UI facets.
- 3D visualization of SPARQL result sets makes it easier to interpret complex geospatial data.
- Integration of Linked Data within existing Business Intelligence (BI) tools allows data to be visualized and viewed in various ways.

3. Architecture

The Data Stories and FacetCheck components are part of the larger Kadaster Data Platform (KDP) architecture (Figure 1). The Dutch Cadastre currently publishes the majority of its 130 Open Datasets by using one of the GIS-specific formats that are standardized by the Open Geospatial Consortium (OGC). These formats, e.g., GML, are popular with GIS specialists, but are not used on a wider scale. Specifically, these geo standards are not used on the web. In the new Kadaster Data Platform (KDP) these existing formats will therefore be extended upon, by also offering Linked Open Data variants (i.e., RDF) and queryable REST APIs over the same data. To effectuate this process, an Extract, Transform and Load (ETL) procedure was designed that allows existing data assets to be automatically and incrementally transformed and loaded into an RDF triple store and a document store.

Based on these newly created Linked Data access points, it is possible to define novel 'Information Products', i.e., specific APIs and/or applications that are created with a specific business goal in mind. Because all data is semantically described as Linked Data, it is relatively easy to combine various datasets into one Information Product. The Information Product consists of a set of integrated Linked Data queries, that are exposed through a REST API that uses the OpenAPI specification. This is also where the main cost saving property of Linked Data resides: it significantly reduces the cost of integrating heterogeneous datasets with the purpose of generating new APIs. This is particularly useful when there are multiple business goals that need to be covered at the same time, and/or when business goals change over time. The content of the integrated RDF triple store is exposed through a SPARQL endpoint. On top of this endpoint, the KDP has implemented various front-end functionalities.

Specifically, the following three Linked Data browsing paradigms were introduced earlier [2]: tabular browsing, hierarchical browsing, and graph navigation.

Tabular browsing is a simple yet popular way for browsing database content, which displays records in rows and properties in columns. In addition to record-oriented tabular browsing, hierarchical browsing makes use of the tree structure of the concept and property hierarchies to display the various classes and properties that are present in the data. As such, a hierarchical browser gives the user a quick overview of the main classes and properties that are in a dataset. Hierarchical browsing works well for gaining an understanding of a concept schema. Both the tabular and hierarchical browser are implemented by the Open Source project Linked Data Theatre (https://github.com/architolk/Linked-Data-Theatre) to which the Dutch Cadastre is a main contributor.

Graph navigation uses the graph-shape of the RDF data model to display concepts and instances as nodes, and properties as edges between those nodes. Graph navigation was observed to work well for explorative browsing, e.g., it allows the discovery of links to other datasets. For graph navigation the existing Open Source tool LODLive (http://en.lodlive.it) is used. In addition to these three existing data browsing approaches, the KDP also includes an advanced SPARQL query editor with added support for GeoSPARQL queries and geospatial visualisations of query result sets [3].

4. Data Stories

Since Linked Data is a relatively new technology for most users, many of them are unaware of the potential that can be unlocked. Users are observed to have difficulty with determining whether a Linked Dataset is useful for their own use case. With the browsing features described in Section 3, a first step towards becoming familiar with a new dataset is to browse through that dataset's metadata description.

A second step consists of browsing through the dataset-specific data model, i.e., the concept and property hierarchies. Unfortunately, this approach is relatively complicated, since it requires a user to be able to identify the usefulness of a dataset based on the concepts it contains. For many users, a concept hierarchy does not immediately translate into potential

use cases of the data. Furthermore, a dataset may contain a large number of concepts and/or properties. In such cases, a user may miss those parts of the data model that are most important to their use case.

To bridge the gap between (a) the vast but implicit potential that a Linked Dataset encapsulates, and (b) the specific and often more explicit use cases a prototypical user may have in mind, we have developed *Data Stories*. A Data Story allows a specific use case to be explained to a potential user through a sequence of data examples, that are connected by an overarching story. To be as generic as possible, the data examples that compose a Data Story are visualizations of SPARQL result sets. This ensures that the components of a Data Story are declarative (how the data is obtained is encoded in the SPARQL query), reproducible (the query is recomputed when the Data Story is generated), and modifiable (advanced users can click a button to open the SPARQL query view, where the query can be altered and rerun).

A Data Story allows the original data publishers to emphasize the potential use cases that they envision for their dataset. This includes their ability to highlight interesting aspects of the dataset itself, e.g., interesting objects and/or interesting relationships between objects, as well as interesting ways in which the data can be combined with other Linked Data sources (e.g., DBpedia). A Data Story can be thought of a 'advertisement tool' for data. It consists of a textual description/explanation of the story line, interspersed with SPARQL queries. When a story is read, the SPARQL queries are executed in sequence, and their result sets are displayed inline. During the creation of a Data Story, the writer can choose to visualize the results of queries in tables, diagrams/charts, pivot tables, widget galleries, or geo-spatial maps. As such, many different types of information, e.g., geographical and statistical information, can be combined to tell an engaging story with data. An example of such a multi-modal combination of data visualization techniques is a thematic map, in which a statistical property is used to colour the regions of a map.

While it is possible to create diagrams with statistics programs, and thematic maps with GIS toolkits, the queries in Data Stories are encoded in a standardized query language and executed within a regular web browser. For each of the displayed query results, an advanced user can open a corresponding query editor that contains the query itself. When a query is changed, the results of the change are calculated on the spot, making the elements of a Data Story more interactive/modifiable than their read-only counterparts from regular web articles. Various

examples of Data Stories can be found in the KDP Labs environment (https://data.labs.pdok.nl/stories).
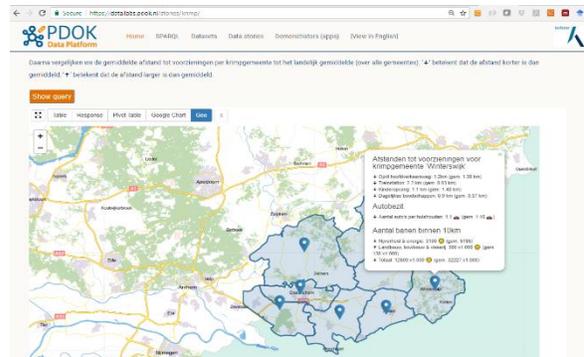

Figure 2: Screenshot from the depopulation dataset

5. FacetCheck

Since Linked Data does not have a static schema, each dataset can be structured in a different way. This provides great flexibility to the data publisher and allows for a wide variety of datasets to be published with high semantic detail, without requiring the introduction of non-standardized and/or domain-specific constructs. Unfortunately, on the side of the data consumer this great flexibility makes it more difficult to understand how a specific dataset is structured, and how it can be queried.

The problem of querying an unfamiliar schema is already 'solved' on today's web by faceted browsers. For example, when a customer wants to buy a television, many online stores allow customers to search for a television based on various properties such as minimum rating, price, weight, screen resolution, and screen size. Customers are able to express a relatively complicated SQL query by interacting with various widgets (check boxes and sliders) within the web UI.
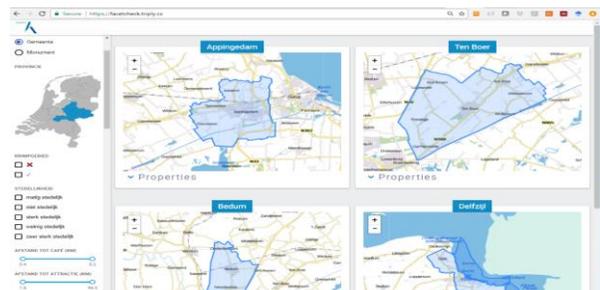

Figure 3. FacetCheck showing the Dutch neighbourhood dataset

Faceted browsers can easily be created when the database schema is stable: an application developer can create widgets that correspond to query filters. Selecting multiple facets results in a (conjunctive) composite query over the set of data entities.

As such, creating a faceted browser is a relatively expensive and time-consuming process since it requires non-trivial development effort for each database. With Linked Data, the properties in the database are described in semantic terms. For example, standards-compliant Linked Data specifies the domain and range types for each property. Based on this semantic description, the faceted browser widgets can be generated automatically.

FacetCheck is a specific implementation that maps semantic descriptions onto UI widgets and underlying SPARQL sub-queries. The FacetCheck UI consists of two components: the left-hand side of the screen containing the various widgets, while the right-hand side of the screen displays the entities that match the specified filters. When making selections within the FacetCheck UI, a SPARQL query is automatically assembled out of the sub-queries associated with a widget. The entities that adhere to the specified query are retrieved and displayed on the right side of the screen for the user. An instance is also displayed by a compositional widget. The components of an entity widget are determined by the direct properties that the corresponding entity has in the database. (This is sometimes referred to as the 'Concise Bounded description' of an entity.) Based in the displayed entity widgets, the users can decide whether the results are wat they wanted, or whether (other) widgets need to be set, or changed, to improve the results. Since FacetCheck allows for the automatic generation of selection- and entity widgets, it is relatively easy to create a FacetCheck browser over a specific Linked Dataset.

5.1 Case study: Dutch neighbourhoods

An example configuration of FacetCheck can be used online (https://facetcheck.triply.cc). Currently configurations for several KDP datasets exist, including one over the spatial-statistical dataset of Dutch neighbourhoods ("Kerncijfers wijken en buurten" in Dutch). This dataset links geospatial data assets of the Land Registry and Mapping Agency to statistical data from the National Statistics Office (CBS). In November 2017, two data journalists were invited to express their interests in Dutch neighbourhood data. They were interested in data about depopulated areas and specifically economic and/or social trends in those areas. Together with the data journalists, several Data Stories were created, and FacetCheck was used to find interesting filter criteria for identifying depopulation areas.

The data story includes multiple queries that show various characteristics of depopulation areas, such as the average distance to public transport, car ownership, and access to jobs (Figure 2).

By pressing the orange "Show Query" button (Figure 2), the user can verify the query, and with a bit of SPARQL knowledge, the query can also be adapted. For instance, the specific depopulation areas the query retrieves can be changed with a small edit. Now we focus on the job market participation rate in shrink areas. The query results in Figure 3 show that access to work in the northern depopulation areas is below the national average.

By looking at the results, and zooming in, we learn that only 2 out of 9 depopulation areas have lower than average employment rates. Additionally, we see that there are areas without depopulation (such as Rotterdam and The Hague) that have lower employment rates. In combination with the depopulation Data Story, the FacetCheck browser was used to filter depopulation areas based on various criteria. In Figure 3, the left-hand side of the screen shows the filters that are based on the properties in the dataset. By scrolling, over 100 data properties can be selected through a map, a slider or a checkbox list. The right-hand side shows the widgets for 4 of the currently selected neighbourhoods.

6. 3D visualisation

3D environments allow for advanced spatial navigation and visualisation but have traditionally provided limited support for performing non-spatial data analysis operations like filtering, joining, and integrating data on-the-fly. Linked Open Data provides advanced support for performing filters and joins over datasets that can be dynamically combined through SPARQL federation. Unfortunately, Linked Data results often lack intuitive visualisation capabilities, making it relatively difficult for a data analyst to interpret the data. This section discusses an integration of 3D visualisation into the read-evaluate-print-loop of SPARQL query execution.

Because of the complementary nature of the two approaches, the combination of 3D GIS and Linked Open Data provides ample potential for data analysis use cases. Unfortunately, not that much prior work on

truly combining 3D GIS and Linked Open Data has been performed. There existing prior work on semantically describing 3D objects in Linked Data [11], and some viewers are able to display (part of) a Linked Dataset within a 3D viewer [4]. However, what is currently lacking is 3D content that is formatted in a standards-compliant way, is accessed through standardized means, and is visualized in a 3D environment.

6.1. The SPARQL Query REPL

Performing complicated data analyses is akin to programming, in the sense that a complex query is not constructed all at once. Rather, query construction is a highly iterative process that consists of repeatedly changing the query until it gives the required result. In programming, this process is widely known as the read-evaluate-print-loop (REPL) is a well-known concept. In data analysis, we observe a similar process:

1. The SPARQL endpoint reads a query (preferably) a SPARQL editor with syntax highlighting and auto-cmpletion functionality.
2. If the read query is grammatically correct, it is evaulated against the triple store. (preferably with a standards-cpmliant endpoint)
3. A SPARQL result set is retunred to the client. These results can be visualized (e.g. on maps or a diagram).
4. With the visualisation, the user can determine whether (part of) the query has to be changed (starting the loop at step 1).

This read-evaluated-print-loop (REPL) principle is implemented by YASGUI, an integrated SPARQL editor and result set visualizer [3] that is developed by Triply (https://triply.cc) and used as a component by many Open Source projects and data publishers. In collaboration with the Kadaster Dataplatform, YASGUI was extended to support GeoSPARQL, the OGC-standardised GIS extension to SPARQL [14]. With this extended support it is possible to query for geospatial relationships, return them in a standard-compliant result set formal, and automatically display them on a 2D Leaflet map [3].

6.2. Benefits of 3D SPARQL

While YASGUI was extended in 2017 to automatically visualize 2D geospatial information on a Leaflet map, no 3D geospatial support was available. In fact, 3D results were treated in exactly the same way as 2D results: the altitude was simply not processed.

At the same time, it is possible to identify several generic benefits of adding 3D support to the REPL principle:

1. 3D visualisation mimics the real world more closely than 2D. 3D visualisations are therefore more powerful in engaging users.
2. Using 3D, multiple attributes can be displayed for the same area. E.g., displaying average income as height.
3. 3D environments allow for the display of multiple views on data. Rather than 3D maps, full 3D environments allow for full six degrees of view. Allowing to display more information about an object (e.g. a building).

In addition to these generic benefits, several use cases were found in which 3D support is not only convenient, but also necessary in order to allow query results to be interpreted correctly. Indeed, the correct interpretation of intermediate query results is required to be able to make the correct edits for the next iteration of the query:

1. With 2D visualisation, buildings that contain multiple administrative entities (e.g. an office containing businesses that own a single floor, or apartments in an apartment block) are displayed on top of each other. Some information is then lost
2. Certain datasets also have height dimensions. E.g., drone no-fly zones often also have a certain height limit for drones. This cannot be displayed in 2D.
3. Height values are crucial for emergency services: fire brigades often need to know the height of a building to determine the number of floors, how many apartments there are etc. Also 3D models of buildings can show them were entrances are, on what height etc.

6.3 Implementation

To integrate 3D support in the SPARQL REPL, we will first take a look at the read component, which consists of the data that is stored in the triple store and the query that is written in order to be evaluated over that data. Even though the GeoSPARQL standard does not mention 3D specifically, the datatypes, relations,

and functions it defines can also be applied to 3D shapes.

Figure 4 shows an example of a small RDF graph that encodes a 3D geometry. It contains a node representing a particular building, together with a triple that asserts that this building is a feature. Second, the graph contains a node that represent the geometry of that building, and a relationship between the feature and the geometry. Third, a node represents a particular serialisation of the geometry. In this case, a serialisation in Well Known Text (WKT).
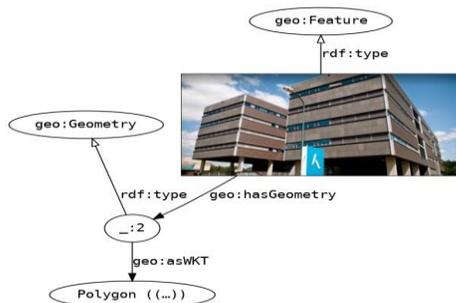

Figure 4. RDF graph with 3D geometry

Such a serialisation starts with a keyword that indicates the kind of shape involved and is followed by nested lists of spatial coordinates. When writing a SPARQL query, the data analyst is able to retrieve the data in various ways. The analyst can retrieve the feature based on some other criteria (e.g., the address of the building), and then also retrieve its geometry and shape. Alternatively, the data analyst may first retrieve the shape based on some geospatial criterion (e.g., proximity to a point of interest), in order to subsequently retrieve the geometry and feature.

With respect to the evaluate component, i.e., the triple store, it is important to choose one that supports 3D. Unfortunately, there are no adequate options for this on the current market. While most triple stores allow 3D geometries to be stored, some do not allow them to be retrieved through SPARQL. Specifically, such triple stores will actively remove the Z coordinate from 3D shapes. This is worse than not supporting 3D at all, since that would at least leave the plain WKT string intact. When 3D information is actively purged from SPARQL results, it is impossible for YASGUI to display the data correctly. Other triple stores do preserve Z coordinates, but do not support the GeoSPARQL vocabulary. Some triple stores do support geospatial filters and relations, but non-standardized, custom-tailored notation. The very few triple stores that do support GeoSPARQL notation do not always apply effective indexing on geometries,

resulting in poor performance for some, especially large, queries.

The last component that must be present to add 3D support to the SPARQL REPL is the print or visualisation component. Firstly, when YASGUI receives a query result set from the triple store, it must know how to interpret 3D shapes. We focus here on the most common SPARQL SELECT query form. A SELECT query returns results in terms of a fixed number of columns that correspond to a sequence of projection variables. Multiple query results amount to multiple sequences or rows of bindings of RDF terms to these projection variables. Whenever an RDF term in such a binding has the standardized datatype: IRI geo:wktLiteral, YASGUI is instructed that a 3D shape is present. Secondly, YASGUI must be able to visualize the detected 3D shapes within a 3D environment. Previously, automatic visualisation of 2D shapes was implement by including a plug-in that is based on the Open Source Leaflet library (http://leafletjs.com). For the current extension, a plug-in is added that is based on the Open Source Cesium library (https://cesiumjs.org). Cesium is not directly able to interpret the WKT formatted serialisations that are present in SPARQL result sets, but it is easy to transform WKT serialisations into GeoJSON, or another format that is supported by Cesium. Besides the ability to display 3D shapes in Cesium, the YASGUI plug-in includes additional support for colouring 3D shapes and for displaying labels. These labels can be displayed within the 3D environment itself (for simple textual labels) and/or in an HTML overlay (for complex labels that can include mark-up and media). At this moment, very few Linked
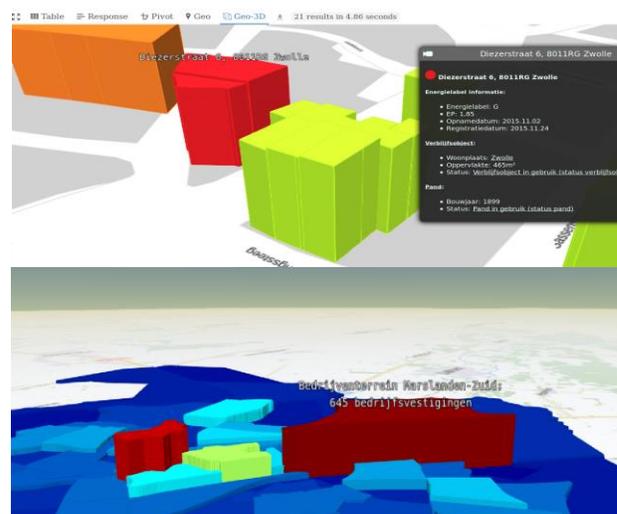

Figure 5. 3D visualisations

Datasets contain 3D shapes that are represented by WKT literals and GeoSPARQL properties. As such, the impact of the SPARQL extension would have been quite small. However, there is a lot of 2D Linked Data encoded in datasets today.

The plug-in therefore adds specific support for visualizing 2D shapes with an added height property. The height variable can be bound within a SPARQL query, either based on a query variable or by simply binding the height variable to a static value that will display all shapes at the same height.

6.4 Examples of use

In this section we present some concrete example of using 3D visualisation support within the YASGUI REPL. Figure 5 (top) shows the result of retrieving the energy labels (expressing energy consumption) of a street in the city of Zwolle. Since the result set contains 3D geometries, these are automatically drawn in the 3D viewer. In our SPARQL query, we are not only binding the geometries of the buildings, but also their energy labels mapped to their respective colour codes. Now it is immediately identifiable which building has a certain energy label. When a building is selected, its textual label (the binding of ?varName in the SPARQL projection) is shown inside the 3D environment, hovering over the building. In addition, the building's HTML labels is shown in the panel on the right-hand side. The HTML snippet in this panel contains additional information about the selected building, such as its Cadastral identifier, it's current status (occupied or not) and use (residence or business). It also contains more information about the energy labels, including when the measurement was performed.

Figure 5 shows the result of retrieving the number of businesses for each neighbourhood in the city of Zwolle. In the SPARQL query, we bind the 2D shape of each neighbourhood to the projection variable ?var, and bind the (normalized) number of businesses to the projection variables ?varColor and ?varHeight. The height of the shapes now expresses the number of businesses. This is an example of a query where the Linked Data only contains 2D shapes, but the query visualisation is still able to display 3D.

7. Visualisation with BI tools

Data driven organisations want certainty that their data is reliable before it is used in decision making. Decisions are often made on management information,

shown in business intelligence tools (BI tools). The Gartner Magic Quadrant for Analytics and Business Intelligence Platforms compares BI tools, considering multiple factors. The leaders on the current Gartner Magic Quadrant are Tableau, PowerBI and Qlik.

Business Intelligence (BI) software is a collection of decision support technologies enterprises aimed at enabling knowledge workers such as executives, managers, and analysts to make better and faster decisions [6]. The data on which business intelligence tasks are performed often come from different internal and external sources. This data varies in quality, format and consistency. The preparation of the different datasets before analysis is called the Extract-Transform-Load (ETL) process. The transformed data is traditionally stored in a relational data warehouse.

When Linked Data can be analysed and visualized in such business intelligence tools, the best of both worlds can be combined. With Linked Data, it is possible to combine a large variety of data and query data at the source. Business Intelligence tools serve as an optimal GUI for the visualisation of these data. The data would no longer need to be copied and extracted to data warehouses and could be analysed and visualised directly from the source. For end-users who want to use the data in business intelligence tools, the Linked Data technology will become much more accessible. Business intelligence tools can serve as a 'Killer App' for Linked Data and give the Linked Data technology a boost.

There are two possible approaches to combine Linked Data and business intelligence: the analysis-oriented and the modeling-oriented approach [10]. In the case of the analysis-oriented approach, an ETL process takes place as usual. The data is queried via a SPARQL query and the results are loaded. In that case, the connection is made directly on the (linked) data source. In the other approach, the analysis is conducted directly on the Linked Data without an ETL process beforehand. This approach seems more effective but needs a complex cube model to conduct the analyses.

The analysis-oriented approach can be achieved with the help of connectors. Two applicable solutions that focus on the analysis-oriented approach are the Tableau Web Data Connector and the ODBC connector. The Web Data Connector is offered exclusively by the business intelligence tool Tableau.

The data.world platform has developed its own Web Data Connector [7]. The data.world platform focuses mainly on the semantic web and offers users the possibility to store and query data in (Linked) Open Data formats (e.g. RDF). It is also possible to

access external endpoints (federated) via data.world and to display the results. The data.world connector allows users to visualise query results in Tableau [7].

Another type of connection that can be used to visualize Linked Data is an Open DataBase Connectivity (ODBC). Tableau, PowerBI and Qlik all offer the possibility to set up a connection with an ODBC source. However, Virtuoso is the only triple store that offers the possibility to set up an ODBC connection. Users can query federated external endpoints and to collect the desired data in the same way as with the Web Data Connector.

For the modeling-oriented approach, several frameworks have been published. These frameworks describe how Linked Data becomes compatible with business intelligence tools. The most of these frameworks are based on the RDF Data Cube Vocabulary (QB) and the expanding RDF vocabulary QB4OLAP [15]. SPARQLytics [17], GeoSemOLAP [16] and SETL [18] are three examples of these frameworks that have developed a workflow and / or architecture to bring both worlds together. The frameworks focus on mapping data in Linked Data formats to multidimensional analytics. A reusable framework that can be used to carry out multidimensional analyses directly on the Linked Data source would provide the most value for analysing and visualising Linked Data. For this reason, it is therefore highly recommended to closely monitor developments relating to the modeling-oriented approach.

7.1 Examples of use

With the FacetCheck UI, users can analyse and visualize spatial-statistical data of Dutch neighbourhoods. The connectors mentioned in the analysis-oriented approach can be used to visualize this data in Tableau. This example of use shows the possibilities of the data.world Tableau Web Data Connector. With a SPARQL query, the user can select the preferred facets. The data.world platform shows an 'Open in app'-button that generates the correct link for the Tableau Web Data Connector. From Tableau, the connection can be set up with the data.world server through the Web Data Connector. The results of the SPARQL query now appear in Tableau.

However, all spatial Linked Data is described in Well-known Text (WKT) format, a mark-up language not supported by Tableau. The National Statistics Office (CBS) publish Shapefiles (SHP) for the Dutch neighbourhoods. By connecting this neighbourhood Shapefile with the spatial-statistical dataset of Dutch

neighbourhoods, the neighbourhood polygons can be plotted in Tableau. The alternative is to develop a parser that makes it possible to convert the WKT format into a Tableau supported format.

Figure 6 shows maps in Tableau for the province of Overijssel. The first map shows the percentage of houses per neighbourhood built after 2000. The percentage of houses increases from yellow to red. Just like FacetCheck, it is possible to combine criteria. The third map shows all the neighbourhoods in Overijssel where at least 50% of the households have children and where a day care centre is up to 2 kilometres away.

The visualisations have been made without copying and loading the data to tableau. The data is queried at the source by means of a SPARQL query.
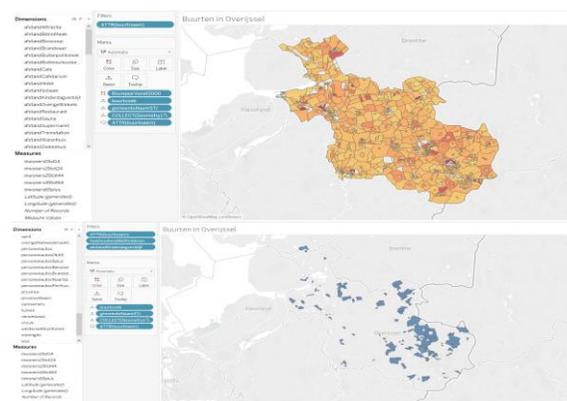

Figure 6: Map visualisations in Tableau

8. Conclusion

This paper has discussed four components for Linked Open Data viewing. The components Data Stories and FacetCheck have already proven to be valuable for the Kadaster Data Platform (KDP). Data Stories illustrates that it is possible to make Linked Datasets accessible in a better way, by advertising concrete use cases. FacetCheck allows users to find the data they need by using intuitive facets, rather than requiring them to write elaborate SPARQL queries.

The components 3D visualization and BI integration are currently under development and require additional research. 3D visualization illustrates that the visualization of complex geospatial Linked Data can be significantly improved. Integration with BI tooling holds the promise of carrying over the benefits of existing BI tools to Linked Data. While this paper shows that it is possible to visualize Linked Data in Tableau with the help of software connectors, further research is needed to improve performance and

scalability. In the optimal situation, BI tooling would have native support for Linked Data.

Over the last couple of years, Linked Open Data has seen a relatively slow adoption speed which may in part be due to the lack of functionalities supporting the usefulness of Open Data. This indicates that further Open governmental Data adoption requires Linked Data theory to be further integrated into its practical context of use. Distinctive and functional browsing and viewing components like Data Stories and FacetCheck contribute to this end.

9. References

[1] Bauer, F., & Kaltenböck, M. (2011). Linked Open Data : The Essentials. Vienna. Retrieved 17 July 2018 from http://www.semantic-web.at/LOD-TheEssentials.pdf.

[2] Beek, W., & Folmer, E. (2017). An Integrated Approach for Linked Data Browsing. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Vol. 42), pp. 35-38.

[3] Beek, W., Folmer, E., Rietveld, L., & Walker, J. (2017). GeoYASGUI: The GeoSPARQL Query Editor and Result Set Visualizer. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Vol. 42), pp. 39-42.

[4] Bosca, A., Bonino, D., Pellegrino, P. (2005). OntoSphere: More Than a 3D Ontology Visualisation Tool. In Italian *Semantic Web Workshop: Applications and Perspectives (SWAP).*

[5] Ronzhin, S., Folmer, E., Lemmens, R. 2018. Technological aspects of (linked) open data. Chapter 9 In: B. van Loenen, G. Vancauwenberghe, and J. Crompvoets (eds.), Open Data Exposed. Springer (forthcoming).

[6] Chaudhuri, S., Dayal, U., & Narasayya, V. (2011). An overview of business intelligence technology. Communications of the ACM, 54(8), 88-98.

[7] data.world. (2018). Tableau Web Data Connector. Retrieved 12 June 2018, from https://help.data.world/hc/en-us/articles/115010298907-Tableau-Web-Data-Connector

[8] Gartner. (2018). Magic Quadrant for Analytics and Business Intelligence Platforms. Opgehaald van Gartner.com: https://www.gartner.com/doc/reprints?id=1-4RVOBDE&ct=180226&st=sb

[9] Kadaster (2018). Rapportages. Retrieved 12 June 2018 from https://www.pdok.nl/nl/actueel/rapportage.

[10] Laborie, S., Ravat, F., Song, J., & Teste, O. (2015). Combining Business Intelligence with Semantic Web: Overview and Challenges.

[11] Pittarello, F., De Faveri, A. (2006). Semantic Description of 3D Environments: A Proposal Based on Web Standards. *Proceedings of the 11th International Conference on 3D Web Technology*, pp. 85-95.

[12] W3C (2016). LinkedData. Retrieved 12 June 2018 from https://www.w3.org/wiki/LinkedData.

[13] Janowicz, K., Van Harmelen, F., Hendler, J. A., & Hitzler, P. (2014). Why the data train needs semantic rails. AI Magazine http://corescholar.libraries.wright.edu/cse/169.

[14] Battle, R., & Kolas, D. (2011). GeoSPARQL: Enabling a Geospatial Semantic Web.

[15] Etcheverry, L., & Vaisman, A. (2012). QB4OLAP: A New Vocabulary for OLAP Cubes on the Semantic Web.

[16] Gür, N., Nielsen, J., Hose, K., & Pedersen, T. (2017). GeoSemOLAP: Geospatial OLAP on the Semantic Web.

[17] Nath, R., Hose, K., Pedersen, T., & Romero, O. (2017). SETL: A programmable semantic extract-transform-load framework for semantic data warehouses.

[18] Rudolf, M., Voigt, H., & Lehner, W. (2017). SPARQLytics: Multidimensional Analytics for RDF.

[19] Joo, J. (2011). Adoption of Semantic Web from the perspective of technology innovation: A grounded theory approach. International journal of human-computer studies, 69(3), 139-154.

[20] Klímek, J., Škoda, P., & Nečaský, M. Survey of tools for Linked Data consumption. Semantic Web, (Preprint), 1-57.

[21] Baskerville, Richard, and Michael D. Myers. "Special Issue on Action Research in Information Systems: Making IS Research Relevant to Practice." *MIS Quarterly*, vol. 28, no. 3, 2004, pp. 329–335.

[22] Ruijer, E., Grimmelikhuijsen, S., Hogan, M., Enzerink, S., Ojo, A., & Meijer, A. (2017). Connecting societal issues, users and data. Scenario-based design of open data platforms. Government Information Quarterly, 34(3), 470-480.

[23] Zuiderwijk, A., Janssen, M., Choenni, S., Meijer, R., & Alibaks, R. S. (2012). Socio-technical Impediments of Open Data. Electronic Journal of e-Government, 10(2).

[24] Janssen, M., Charalabidis, Y., & Zuiderwijk, A. (2012). Benefits, adoption barriers and myths of open data and open government. Information systems management, 29(4), 258-268.