Exploratory querying of SPARQL endpoints in space and time

Simon Scheider [b,*], Auriol Degbelo [a] Rob Lemmens [c] Corné van Elzakker [c] Peter Zimmerhof [a]
Nemanja Kostic [a] Jim Jones [a] Gautam Banhatti [a]
[a] *University of Münster, Institute for Geoinformatics, Münster, DE*
E-mail: {degbelo, peterzimmerhof, kostic_nemanja, jim.jones, g_banh02}@uni-muenster.de
[b] *ETH Zürich, Inst. f. Kartografie u. Geoinformation, Zürich, CH*
E-mail: sscheider@ethz.ch
[c] *University of Twente, Faculty of Geo-Information Science and Earth Observation (ITC), Enschede, NL*
E-mail: {r.l.g.lemmens, c.vanelzakker}@utwente.nl

Abstract. The linked data Web provides a simple and flexible way of accessing information resources in a self-descriptive format. This offers a realistic chance of perforating existing data silos. However, in order to do so, space, time and other semantic concepts need to function as dimensions for effectively exploring, querying and filtering contents. While triple stores, SPARQL endpoints, and RDF were designed for machine access, large burdens are still placed on a user to simultaneously explore and query the contents of a given endpoint according to these dimensions. First, one has to know the semantic concepts and the type of knowledge contained in an endpoint a-priori in order to query content effectively. Second, one has to be able to write and understand SPARQL and RDF. And third, one has to understand complex data type literals for space and time. In this article, we propose a way to deal with these challenges by interactive visual query construction, i.e., by letting query results feedback into both (space-time) exploration and filtering, and thus enabling exploratory querying. We propose design principles for SPEX (Spatio-temporal content explorer), a tool which helps people unfamiliar with the content of SPARQL endpoints or their syntax to explore the latter in space and time. In a preliminary user study on a repository of historical maps, we found that our feedback principles were effective, however, that successful question answering still requires improvements regarding space-time filtering, vocabulary explanation and the linking of space-time windows with other displays.

Keywords: SPARQL endpoints, visual query formulation, spatial and temporal linked data, exploratory querying, historical maps

1. Motivation

Data repositories often remain black boxes, for lay persons and database experts alike and regardless of whether the repository in question is a traditional relational database, a Web-based fast indexing and search system such as Elasticsearch[1], or an RDF[2] triple store[3]. This is surprising, since all this technology is rather mature and was designed for exactly that: helping people retrieve particular information.

Space and time are primary dimensions for structuring and searching information [18], and thus can help open up such data silos. Map-like query interfaces can play a key role in retrieval tasks [29] and are easily adopted by users [17]. In addition, a very large proportion of data, not only from traditional sources [11] but also from the Semantic Web[4], has a (potentially complex) space/time component, or can easily be linked to

[*] Corresponding author

[1] `http://www.elasticsearch.org`

[2] Resource Description Framework, `http://www.w3.org/RDF/`

[3] See [36] for a short introduction to triple stores.

[4] Compare the representative sample of linked geospatial data crawled from the Web and described in `http://stko.geog.ucsb.edu/location_linked_data`.

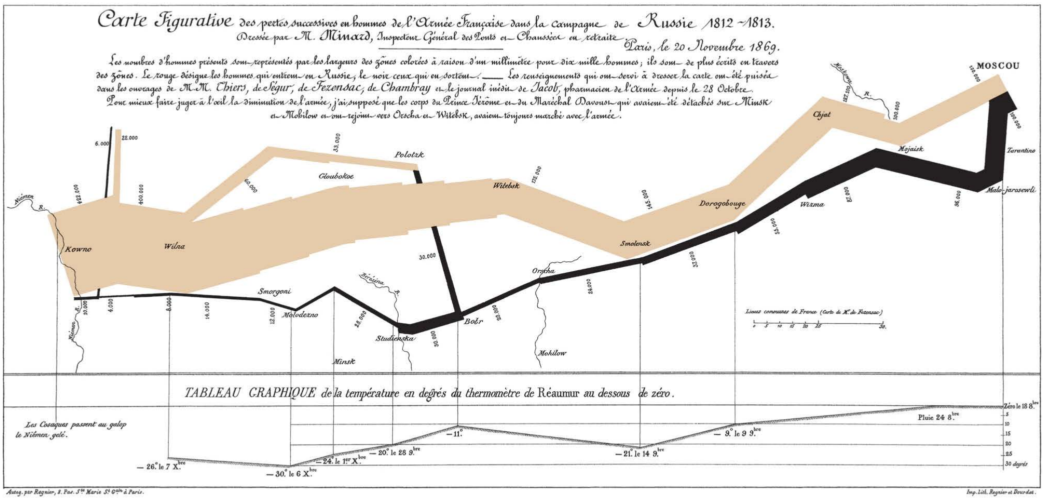

Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zones colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zones. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de MM. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Niémen R.
Kowno
Wilna
Gloubokoe
Polotzk
Witebsk
Smolensk
Dorogobouge
Chjat
Mojaisk
Tarantino
Malo-jarosewli
Wizma
Moskowa R.
Smorgoni
Molodezno
Minsk
Studienska
Bérézina R.
Bobr
Orscha
Mohilow
422.000
400.000
6.000
22.000
60.000
33.000
175.000
145.000
127.100
100.000
100.000
96.000
87.000
55.000
37.000
24.000
20.000
50.000
28.000
12.000
14.000
8.000
4.000
10.000
30.000
Lieues communes de France (Carte de M.r de Fezensac)
0 5 10 15 20 25 50.
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
— 26.° le 7 X.bre
— 30.° le 6 X.bre
— 24.° le 1.er X.bre
— 20.° le 28 9.bre
— 11.°
— 21.° le 14 9.bre
— 9.° le 9 9.bre
Pluie 24 8.bre
Zéro le 18 8.bre
5
10
15
20
25
30 degrés
Autog. par Regnier, 8. Pas. S.te Marie S.t G.ain à Paris.
Imp. Lith. Regnier et Dourdet.

Fig. 1. Charles Minard's map from 1861 about Napoleon's 1812 march on Moscow. How could non-experts query for its complex spatio-temporal content in a way which goes beyond keywords [34]?

such (e.g., based on OpenStreetMap [40]). Still, how space and time can be used together with other content dimensions in order to open up the black box of a repository to non-expert users remains a central question of research [20,32].

Linked open data, even though it changes the conditions for and approaches to retrieval of spatio-temporal and other information in positive ways [41,26], places also particular burdens on a user. Think about an historian or librarian who wants to find maps which contain information about Napoleon's invasion to Russia [34] (Figure 1). Suppose he or she is given a list of URIs of SPARQL[5] endpoints[6] containing RDF descriptions of large collections of historical maps [25]. How to proceed from there? Even if the SPARQL syntax is known, even if text search and linked data browsers are available, and even if the historian knows where to search on a map and when to search in time, how could the historian possibly know how to pose a query to the endpoint in order to get what he or she wants?

In this article, we suggest that a major challenge in answering these questions is a dilemma which has to do with the *lacking integration of exploration and querying along all dimensions of space, time and theme*, and correspondingly, with the lacking possibility of simultaneously learning about the information needed while specifying it.

Think again about our example in Figure 1. Piecewise data exploration (e.g. faceted browsing) does not allow users to perform complex searches and does not scale across larger datasets which cannot be explored manually. There are two traditional kinds of counter measures: either, one ignores data concepts and exploits text search, or one makes use of data concepts and formulates queries. From a user's viewpoint, text search is simple but also imprecise. For example, searching for "Napoleon" or "Russia" will not lead to any success, because the intended map (cf. Figure 1) happens to have the digitized title "Carte Figurative des pertes successives en hommes de l'armé française dans la campagne de Russie 1812-1813". A database or linked data query may identify the right map because it works independently from keyword labels. However, it remains unclear, first of all, how to formulate the right query, and secondly, which kinds of relations and concepts are required and available in order to formulate the right query [34]. Furthermore, in our case, the historian might want to filter results by a certain spatial area in Russia and a certain time interval (1812-1813) and thus needs to handle georeferenced historical map regions with complex polygon shapes [34]). This requires specific *interaction strategies* to account for resolution, facilitate map overview, zooming, as well as geometric comparison and filtering.

All this unfortunately presupposes a lot of expertise, and systems which make strong assumptions about technical skills and a-priori knowledge strongly limit the set of potential users. On the other hand, a query

[5] `http://www.w3.org/TR/rdf-sparql-query/`

[6] RESTful RDF repository interfaces for sending SPARQL queries on the Web. `http://labs.mondeca.com/sparqlendpointsstatus/` is a list of available endpoints.

system that allows users inexperienced with repository contents and linked data to acquire such knowledge needs to support them in *interacting* with concepts and data in novel ways. That is, at the end of the day, successful queries require exploration and scalable exploration requires queries. While similar insights have recently motivated *exploratory query* [22] and, more generally, *exploratory search* [31,43] as fields in information retrieval, it remains an open question how Semantic Web technology and how space and time can be used in order to realize it.

In this article, we focus on the following research questions:

1. How can we integrate exploration and querying of linked spatio-temporal data in a tightly closed loop, such that people inexperienced with content of a repository, its syntax or data types are still put in a position to answer complex questions about it?
2. What can be the role of spatio-temporal interfaces in such exploratory query systems?

We experience a lack of systematic and empirically tested design principles for linked data tools that combine query with exploration on the three dimensions of space, time and theme[7]. Furthermore, there is a corresponding lack of tools for users unfamiliar with the syntax of linked data repositories, as we will argue below.

In seeking for answers, we first discuss corresponding challenges and related work (Section 2), and then suggest a set of design principles (Section 3) for a SPARQL query and exploration tool called SPEX (Section 4). Other tools are compared against these principles. A historical maps scenario (Section 5) serves to test the tool in a preliminary user study (Section 6). The scenario builds on earlier work in the map library area [34,33] illustrating the requirements of handling complex space-time queries[8]. While an empirical *validation* is still future work, the study provides first insights on how users unfamiliar with linked data make use of these principles to answer complex questions about a repository, on the role of spatio-temporal interfaces, and on the main open problems regarding the tool.

2. Joining linked spatio-temporal data exploration and retrieval

Joining exploration with querying parallels the old dilemma of joining the learning of concepts with their specification: How to specify something if that something is basically unknown? This dilemma may have stimulated the relational schema in classical database research. A relational schema [9] allows users to specify a piece of data without knowing all the details (e.g. its particular value). However, for this purpose, users need to explore the schema. *Exploration*, in our sense, is something that people do when looking at and manipulating a display in order to learn about something, e.g., when looking up a table schema in a relational database or a digital map. *Specification* is something that people do when they construct retrieval requests, e.g., by specifying a “select” query. *Retrieval* is something that only machines do. However, it often requires specification as input and may have exploration applied to its output.

Since linked data is inherently self-descriptive, it can be used to do both specification and learning about concepts at the same time. Technically, *linked metadata are simply linked data*, and as such are not hidden in the data structure [26], in contrast to a relational database schema[9].

This principle allows users to perform exploration and retrieval in a *closed loop* (see Figure 2): Finding meta-data is simply done in terms of a certain kind of query (a query for classes and relations, called *classify* here). This information is suggested in a display and can be explored. Exploration feeds back into the construction of an instantiation query (e.g. a query for instances of classes and relations), which retrieves explorable results that can feedback into another query, and so on. How does space and time fit into this loop? Since space and time can be handled in terms of linked data types (compare Section 2.4 below), one only needs appropriate display (e.g. map), exploration (e.g. zoom), query construction and query strategies (e.g. spatial filters) which take advantage of the space-time semantics, and which smoothly fit into ordinary linked data interaction strategies that might follow up in the loop.

In the following, we discuss this challenge and review related work.

[7] What we have in mind here are studies like [27].

[8] The tool and underlying principles suggested in this paper are novel and not constrained to that particular domain.

[9] Syntactically, table schemas are different from table contents and thus need special syntax for look up.

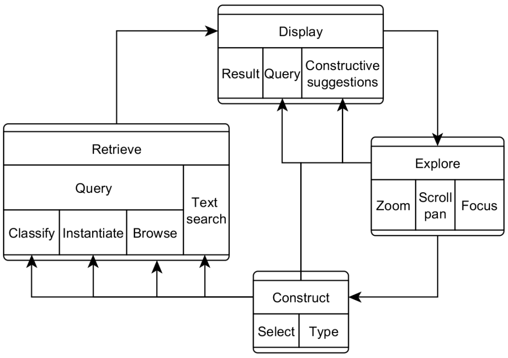


Fig. 2. Schema of chaining exploration and retrieval strategies via displays and query construction. With linked data, this loop can be performed both on meta-data as well as data, and on space as well as on time.

2.1. Retrieval strategies

Catarci et al. [8] distinguished top-down (moving from general to specific) from browsing strategies (moving along neighborhoods of data as well as concepts). In the Semantic Web, top-down query strategies are either instantiations of concepts (finding all instances that satisfy an RDF class description) or browsing of hierarchies of RDF classes and RDF properties (Figure 2). Browsing can also be done on an instance level (browsing is simply a special case of querying, namely querying for neighborhoods of resources in an RDF graph). In Figure 2, we furthermore added the bottom up query strategy "classify" (finding classes or properties for certain instances, including the entire domain of instances), as well as direct text-based search of concepts (which is done in current Web search engines).

2.2. Exploration strategies

Shneiderman [37] distinguished a number of exploration tasks (not only for visualization purposes), namely:

1. overview of the collection
2. zoom in on items of interest
3. details-on-demand for items
4. view relationships between items

Zoom, overview, inspection of details and relationships can be realized by focusing attention on parts of a display, scrolling, panning and zooming in our schema (Figure 2). Note that for all of these exploration tasks, one can choose a top-down, bottom up, browsing or a text retrieval strategy, and for a certain retrieval strategy, one can choose any of the exploration strategies mentioned above. For example, one can search for overview concepts based on their text label or browse through them or query for them. And one can get an overview or visually "zoom into" a concept hierarchy which was itself retrieved by text search or a browser. Retrieval strategies are therefore orthogonal to Shneiderman's exploration tasks.

Many linked open data (LOD) exploration and visualization tools provide an overview of an endpoint as well as details on demand [10]. Since the linked data structure is a graph, graph based visualization techniques seem obvious to address exploration. However, as Schraefel and Karger argue [35], *great big graphs* have severe drawbacks for visualizing linked data sets because what they exactly not provide are overview, zoom and details-on-demand. Instead, a mix of graphical options which fit these tasks and corresponding data types are more adequate. Currently discussed techniques in the Semantic Web range from menus, treemaps and sitemaps or simple lists, cartographic overviews, to facets and pivoting (concatenating facets) [7]. One major challenge is to combine those elements on-the-fly to fit a certain purpose or data set [35].

2.3. Query construction strategies

A visual query system needs to support query formulation [8] by letting users select visual data representation elements and manipulating them. Each possible manipulation needs to translate into a syntactical operation in the formal query language. Even though the choice of a visual query interface depends on the query language (i.e., SPARQL), the development in the Semantic Web illustrates a wide variety of query construction approaches.

While *faceted browsing* is useful to learn about concepts and data, it is very restricted in expressive power regarding SPARQL, at least in its pure form. One basically moves a local focus on a node and its immediate property neighborhood from one node to another, selecting values for property ranges. This is well suited for exploration of neighborhoods, but not expressive enough for arbitrary data retrieval. The idea of *pivoting*, i.e., of concatenating facets, allows users to formulate more complex queries, because one can restart a faceted search from the results of another search [7].

We find a lot of *linked data browsers* which support significant subsets of SPARQL, such as LESS

[3], RelFinder, gFacet [15], RDF Gravity[10], Tabulator[11], and Rhizomer[12] [10]. From these, RelFinder and gFacet directly work on SPARQL endpoints and can be used without a-priori knowledge about content, based on autosuggestion and feedback. RelFinder is restricted to instance based queries without variables. gFacet can already be considered a visual query tool, as it follows a visual graph-pattern strategy very similar to the design principles proposed in this paper (compare Section 4.1).

Visual or text-based query languages afford the formulation of expressive queries either by *typing text* or by *selecting* visual objects (Figure 2). In general, one can [8] (1) construct schematic paths (and graphs), (2) compose concepts and subqueries (3) query by examples, or (4) select quality ranges (e.g. by maps and time sliders). Semantic Web developers have suggested graphical SPARQL query languages (such as RDF-GL [16] or [44]) which help users generate SPARQL code more conveniently, covering large parts of graph pattern construction, composition and filtering. However, these languages require also some a-priori understanding of SPARQL. Many visual SPARQL clients have been built in the last years. NITELIGHT [39], IsaViz (SPARQLViz), ViziQuer [44], OpenLink iSPARQL, Sgvizler [38], Visualbox [12] and SparqlFilterFlow [13] can be directly used as clients on SPARQL endpoints. From these, iSPARQL, SPARQLViz, Visualbox and Sgvizler require substantial a-priori knowledge about SPARQL or contained vocabularies for building a meaningful query. ViziQuer, NITELIGHT and SparqlFilterFlow, in contrast, have a form of overview and suggestion tool for available vocabularies and datasets in an endpoint as well as substantial feedback and support in building queries. ViziQuer and NITELIGHT, however, seem to be made primarily for tech-users and focus less on data exploration. An interesting interactive query tool suitable for inexperienced users which gives feedback on data satisfying a query in terms of a "flow" chart is SparqlFilterFlow [13]. However, all of these SPARQL tools do not handle complex space time literals in queries.

2.4. Handling of space and time in exploratory querying

The insight of [35] was that every kind of linked data needs a mix of visualization tools appropriate for the task. One way to handle this is to exploit the different kinds of semantics that linked data offers. A good example for this strategy is spatio-temporal data: one can treat the semantics of spatio-temporal data types (RDF typed literals) with special visualization and query panes, while treating ordinary RDF semantics in a different way. In RDF, linked spatio-temporal data can be encoded as GeoSPARQL [4] and OWL Time[13]. For example, the time and space extent of a historical map can be represented as follows [34]:

```
@prefix time:<http://www.w3.org/2006/time#> .
@prefix sf:<http://www.opengis.net/ont/sf#> .
@prefix geo:<http://www.opengis.net/ont/geosparql#> .
@prefix maps:<http://www.geographicknowledge.de/vocab/maps#> .

:map maps:mapsTime "1840"^^xsd:gYear;
        maps:mapsArea _:g.
_:g  geo:asWKT "<http://www.opengis.net/def/crs/EPSG/0/4326>
        POLYGON((9.874690102339652 52.25156096729222,
        9.874324681594004 52.126487663211606,
        10.07547489355107 52.1268449901813,
        10.073392224324136 52.252405987705664,
        9.874690102339652 52.25156096729222))"
        ^^sf:wktLiteral.
```

In how far is exploratory querying of such data supported by linked data technology? In recent years, we have seen a considerable progress in adopting spatial functionality for linked data. Still, there remains a technical gap between GIS, which is based on layer and geometry oriented data models (and corresponding query and visualization tools), and linked data [26, 21,14,19]. While many triple stores now support spatial queries of varying complexity[14], we know of only two data type standards (GeoSPARQL and stRDF) and corresponding triple stores which explicitly adopt the existing geospatial geometry standard[15] [4,28]. A few exploration tools for linked geospatial data have been proposed so far [1,5,24,23]. Some linked data visualization tools and browsers support geographic map interaction (such as gFacet [15], LGD Browser [40], LDVizWiz and CODE wizard).

[10] http://semweb.salzburgresearch.at/apps/rdf-gravity/

[11] http://www.w3.org/2005/ajar/tab

[12] http://rhizomik.net/html/rhizomer/

[13] http://www.w3.org/2006/time

[14] Ranging from GraphDB's geo-spatial extension (http://owlim.ontotext.com/display/GraphDB6/GraphDB-SE+Geo-spatial+Extensions) supporting (W3 Geo RDF) point geometries, to Virtuoso's (7.1+) support for complex WKT geometries (http://docs.openlinksw.com/virtuoso/sqlrefgeospatial.html.

[15] i.e. the *simple features* standard of the Open Geospatial Consortium (OGC) http://en.wikipedia.org/wiki/Simple_Features.

However, we do not know of any tools that specifically exploit the use of maps and time sliders in supporting SPARQL query formulation, and of any user studies which cover this aspect. To us, it seems that a particular challenge lies in clarifying to users the conceptual link as well as the difference between the kinds of semantics involved, namely between spatial geometries, time intervals (as literals) and ordinary linked data concepts. Correspondingly, it remains unclear how space-time exploration and filtering can be intuitively integrated into query pattern construction, especially for non-tech users. For example, how should a user decide what to do if in principle, both query pattern expansion and space-time filtering are possible to pose a query, for example, for maps about "Africa", either via map extent or via instance name (see Section 6)?

3. Design principles

Users unfamiliar with a linked data repository should be supported in finding complex patterns of information of interest about some subject which is hidden in the repository. Mandel [30, chapter 5] suggested three general interface design principles which can be concretized according to this task as follows:

1. The interface should place users into control ..
    - Offer users query manipulation overview, allowing them to navigate through a query and to undo each step.
    - Immediately feed-back results into all displays. In this way, users can explore and learn the function of all manipulations based on feedback (see principles 3.3 and 3.2 below).
    - Avoid any menus which could change a display modus (*modeless design*) and instead use context menus and (mouse) foci for exploration.
    - Automatically parse datatypes into appropriate displays.
2. ..it should reduce their memory load ..
    - Always suggest (only) the local space of meaningful query construction possibilities (see explanation in Section 3.2).
    - Progressively disclose options.
    - Use visual cues for possible actions.
3. ..and it should be consistent.
    - Be predictable and exploratory.
    - Use a consistent symbology.

Based on these general objectives, we suggest the following design principles for an exploratory spatio-temporal query interface for linked data.

3.1. The syntax is in the visual construction of a query pattern

Syntactical rules for SPARQL need to be enforced by visual construction. This means that bad syntax is avoided upfront by offering only those construction possibilities that correspond to a valid SPARQL query. Furthermore, every construction possibility needs to be visually indicated and needs to be explained. Users should be able to redo every step and reload a query pattern (a state of construction). Visual cues (e.g. colors) should identify every kind of syntactical element of the SPARQL graph pattern in a consistent manner (variable nodes, constant nodes, property edges, subpatterns...). Furthermore, users should be able to *navigate and focus on parts of a query* in order to modify it. Selection of suggestions should be possible via context menus locally on selected foci.

3.2. Feedback 1: Constructive suggestions are grounded in non-empty results

In formulating a query, search or exploration techniques serve as a starting point to learn about contents. However, during query construction, both techniques can also be used to constantly *map out the space of meaningful query construction steps*. For example, the space of meaningful predicates to choose from reduces dramatically if a subject or object resource is fixed. The current query restricts the possible ways how to proceed, which can feedback into *suggestion tools* (Figure 2) that map out the space of possible actions. Therefore, constructive suggestions should always entail a *meaningful* (non-empty) query result. Otherwise, they should not be displayed. For this purpose one needs a *vocabulary suggester* which takes into account the current part of a query in focus and offers exploration/selection of only those vocabulary or data terms which will produce non-empty query results. In this way, it becomes possible to avoid frustrating query attempts and unnecessary browsing. At the same time, users can immediately experience the coverage of data which is available in a repository.

3.3. *Feedback 2: Results feedback into the construction*

Results should be reused in the construction. That is, people should be able to use results directly to filter or restrict a query. This reduces the memory load for users and furthermore allows them to quickly generate smaller result sets which can be visually explored more efficiently.

3.4. *Typed literals (space and time) are hidden and automatically handled by "display-filters"*

An example of this latter principle are displays for typed literals. Typed literals should be hidden from users and should be handled by special *display-filter* diagrams. That is, displaying and filtering results should be allowed in the same diagram. This has the advantage that results can be used for exploration as well as for setting of filters. Spatial geometries should be discovered automatically and visualized in maps, and temporal elements in calendars and time sliders. Simple filtering can be performed based on the current extent of the respective windows in time and space. In order to satisfy principle 3.2, filtering should only be possible if variables are "spatio-temporally-enabled", i.e., if corresponding literals are present.

3.5. *Resources are explorable, i.e. automatically labeled and visually linked across displays*

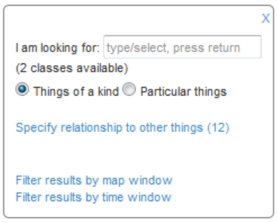
URIs should never be shown to users and only be used as HTML hyperlinks of result items. Instead, all RDF resources should be displayed by their label, and, if not available, by a prefix abbreviation or a label generated from an URI sub-string. Furthermore, result items should be visually linked across different displays, e.g., by highlighting of mouse events, in order to allow users identifying them across displays.

4. Spatio-temporal Content Explorer (SPEX)

The design principles discussed above were implemented in SPEX, a Web client written in Javascript. We used object-oriented-programming techniques in Javascript in order to divide the code into meaningful parts. Furthermore, we used evolutionary prototyping in order to quickly build a running system with the principles implemented in a preliminary way. We refined this prototype and added further functionalities after the first heuristic evaluation. The SPEX code is available for exploration and reuse at `https://github.com/lodum/SPEX`. For a working version of the application, see `http://geographicknowledge.de/spex/`. Note the current version is not suitable for Chrome browsers and was developed for Firefox.

The layout of the user interface is shown in Figure 3 and is divided into three window parts, namely the *query pane* (upper left), two *display-filters* for space and time (upper right) and the *results pane* (lower left).

The construction of queries is possible in the query pane, through the use of context menus on nodes (see Figure 4). There are two kinds of nodes (Figure 5): variable nodes (nodes which represent sets of instances) and instant nodes (nodes which represent particular instances). The former are identified by a question mark inside the node, the latter are empty. Variable nodes can be restricted to an RDF class and are labeled by the name of that class. Otherwise they are labeled by a variable name. Instance nodes are labeled by the particular resource they denote. Spatial or temporal enabling of nodes is depicted by green vs. blue colors.


x

I am looking for: type/select, press return

(2 classes available)

● Things of a kind ○ Particular things

Specify relationship to other things (12)

Filter results by map window

Filter results by time window

Fig. 4. Context menu on a selected node. It offers node restrictions to things of a kind (i.e. classes) or particular things (i.e. instances), and can be used to specify relationships to/from other things and to carry out space/time filtering on a node.

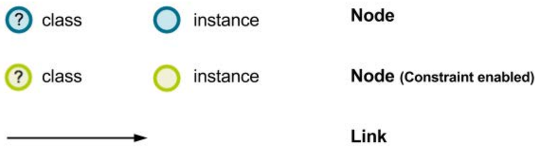
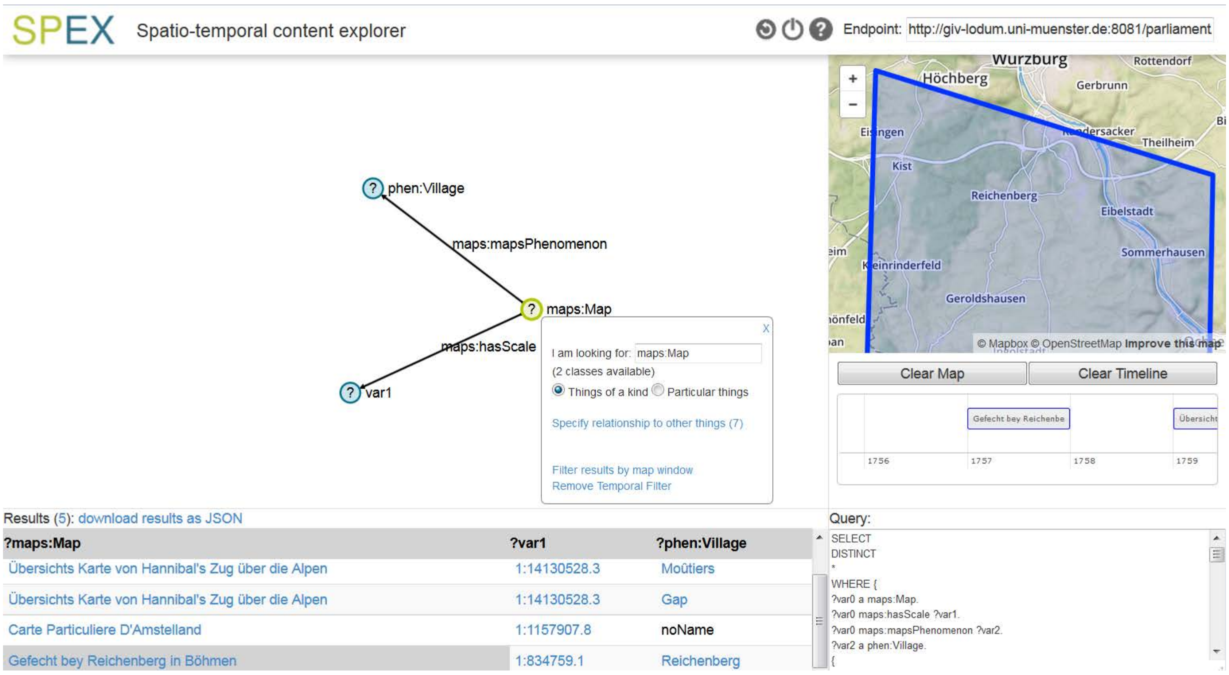


Fig. 5. SPEX visual query constructor elements.

The different *kinds of user interaction* with SPEX, which are explained in the following, are depicted in the hypothetical flowchart of Figure 6, labeled with abbreviations. The query pane is initialized with a single unlabeled variable node, a question mark which



SPEX Spatio-temporal content explorer

Endpoint: http://giv-lodum.uni-muenster.de:8081/parliament

I am looking for: maps:Map
(2 classes available)
● Things of a kind ○ Particular things
Specify relationship to other things (7)
Filter results by map window
Remove Temporal Filter

Results (5): download results as JSON

?maps:Map	?var1	?phen:Village
Übersichts Karte von Hannibal's Zug über die Alpen	1:14130528.3	Moûtiers
Übersichts Karte von Hannibal's Zug über die Alpen	1:14130528.3	Gap
Carte Particuliere D'Amstelland	1:1157907.8	noName
Gefecht bey Reichenberg in Böhmen	1:834759.1	Reichenberg

Query:

```
SELECT
DISTINCT
*
WHERE {
?var0 a maps:Map.
?var0 maps:hasScale ?var1.
?var0 maps:mapsPhenomenon ?var2.
?var2 a phen:Village.
{
```

Fig. 3. The SPEX layout consists of three visual panes (query, results, and display-filters). The query pane (upper left) is used to construct query patterns, and space-time filter panes (upper right) are used to set spatio-temporal constraints on nodes. In this example, we queried for historical maps that show periods in the 18th century, their map scale and the villages they represent. Clicking on a result (lower left, e.g. "Gefecht bey Reichenberg in Böhmen") zooms the space and time windows accordingly.

stands for an unspecified question. A user can add constraints (triple patterns and filters) by selecting a node (SN, Fig. 6) and an option from the node's context menu: (1) either he or she can text search (TN, Fig. 6) or scroll through available classes or instants (SNL, Fig. 6) connected to this node and submit an instantiation query by pressing return. Class restrictions translate into `?var rdf:type Class` statements, and instance restrictions into filter statements of the form: `FILTER (?var = <URI>)`. Or, (2) he or she can specify in- or out-going relationships (AL-TL-SLL, Fig. 6) (i.e., triple patterns of the form `?var property ?var1`) to the node by text searching and scrolling through a list of available properties connected to the node. This adds an arrow pointing to a new variable node to the query pane (Fig. 3). Or, (3) he or she can set a spatial or temporal constraint on the node (SC, Fig. 6), adding special filters to the query[16]. A spatial or temporal constraint can only be set if a node is spatially/temporally enabled, i.e., it is linked to corresponding literals (based on implicit background queries). The constructive process assures that only valid SPARQL can be generated. Suggester queries running in the background (class *Suggester*) make sure suggested queries are always non-empty. They filter out domain vocabulary RDF terms (including classes, properties and instance names) that are contained in the endpoint and linked to a node, excluding syntactical RDF terms. Result RDF resources are clickable (dereferencing the corresponding URI) (BW, Fig. 6) and selectable (highlighting and zooming) (IO, Fig. 6) and automatically labeled and displayed in terms of a scrollable table (ER, Fig. 6). The number of results is shown above the table, and so empty result sets can be explicitly discovered. Furthermore, results are automatically projected to time line events and map window objects that are visually linked to table items (via mouse over highlighting and zooming), provided that corresponding literals are present. Map and time windows are both zoomable

[16] Map window and time slider window can be zoomed and used to filter out results based on whether linked literal values (e.g., WKT geometries in GeoSPARQL [4] or time instances/intervals in OWL-Time (`http://www.w3.org/TR/owl-time`)) are contained in a window or not. The spatial filtering is currently done inside the SPEX client in order to remain independent from endpoint technology. In the future, we intend to add server side spatial filtering for performance improvement.

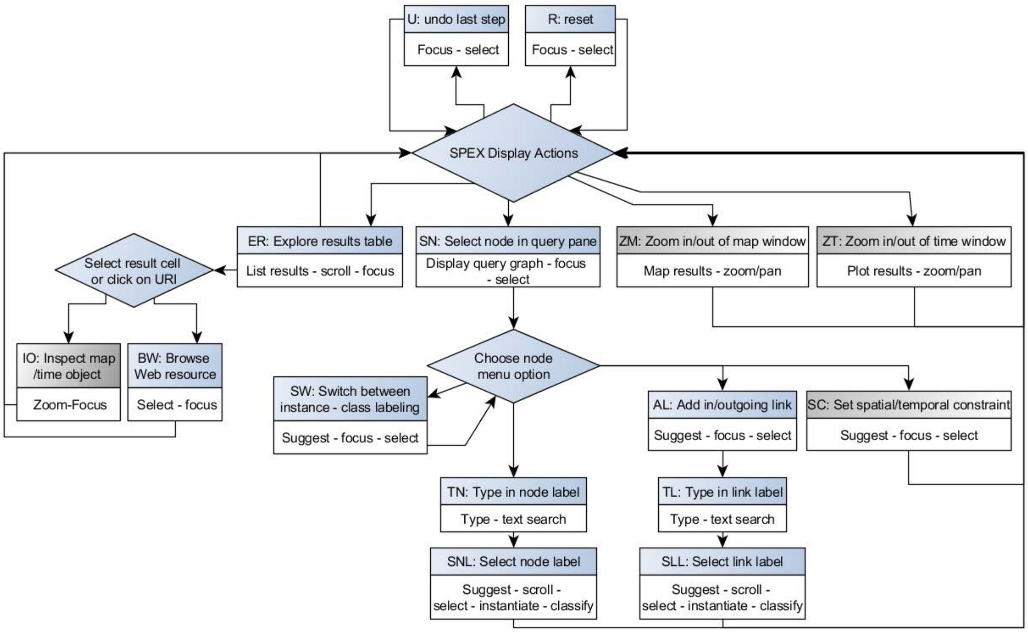

U: undo last step
Focus - select
R: reset
Focus - select
SPEX Display Actions
ER: Explore results table
List results - scroll - focus
SN: Select node in query pane
Display query graph - focus - select
ZM: Zoom in/out of map window
Map results - zoom/pan
ZT: Zoom in/out of time window
Plot results - zoom/pan
Select result cell or click on URI
IO: Inspect map /time object
Zoom-Focus
BW: Browse Web resource
Select - focus
Choose node menu option
SW: Switch between instance - class labeling
Suggest - focus - select
AL: Add in/outgoing link
Suggest - focus - select
SC: Set spatial/temporal constraint
Suggest - focus - select
TN: Type in node label
Type - text search
TL: Type in link label
Type - text search
SNL: Select node label
Suggest - scroll - select - instantiate - classify
SLL: Select link label
Suggest - scroll - select - instantiate - classify

Fig. 6. Flow chart of visual query construction and exploration possibilities in SPEX. Each process step is represented by a box and a 2-letter short name and is further subdivided into display - explore - construct and retrieval actions (compare Figure 2). Dark grey boxes denote actions with spatio-temporal displays.

and panable[17] (ZM/ZT, Fig. 6). This allows multidimensional exploration of results (Principles 3.5 and 3.3) and at the same time serves as an interface for window-based filtering.

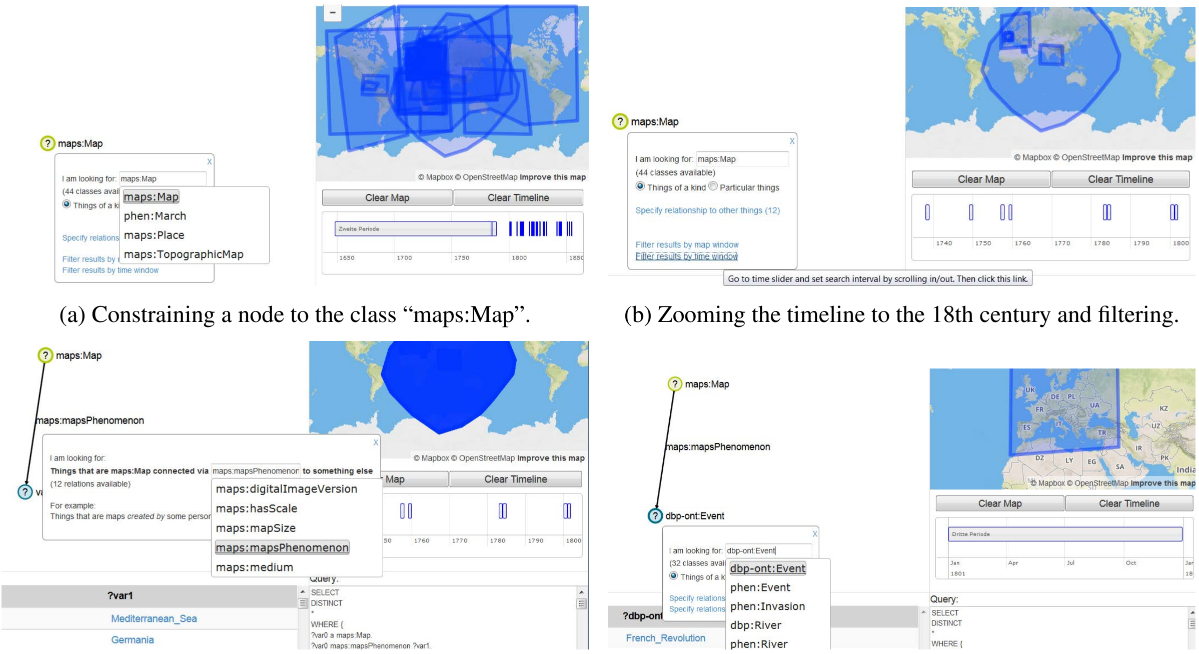
To illustrate the functionality, consider a query for maps of the 18th century that show some important historical event, such as the French Revolution (compare Figure 7). After constraining the empty node to the class "maps:Map" (7a), one can zoom the time line to the 18th century and filter results by the time window (7b). Then one can add a property link to the phenomena represented by maps (7c) and constrain the new node by the class "dpb-ont:Event" (7d).

The current development covers only a subset of SPARQL. It allows the construction of *connected conjunctive triple patterns* and focuses on SELECT queries with '*', since CONSTRUCT, ASK queries and the distinction of bound and unbound variables are not needed for exploration purposes[18]. It does not yet allow users to deal with *graph patterns*, and thus does not support the formulation of disjunctive (via UNION) and OPTIONAL patterns. This was not required in our scenario (see Section 5) but could be added in the future by allowing the manipulation of corresponding sub-query-panes (compare [44]). FILTER expressions are supported in two ways: (i) In terms of display-filters for selected typed literals (space, time), according to our design principle 3.4; or (ii) As query constraints on variables denoting individual resources, which enables turning a variable node into an individual node. Note that our focus was not on expressive power, but on integrating content, space and time exploration with querying.

While SPEX implements all our design principles to some extent, it is only a prototype so far. There is much room for improvement (see the discussion in Section 6), especially through combining further exploration and query strategies in different displays. Furthermore, our user study also revealed that it still requires a con-

[17] Based on Leaflet (`http://leafletjs.com/`) with OSM background layer.

[18] In an exploratory query system, information on the satisfiability of queries (ASK) comes for free (through data feedback mechanisms, see principle 3.3 and 3.2). Furthermore, CONSTRUCT queries and unbound variables do not support data exploration.



(a) Constraining a node to the class “maps:Map”.

(b) Zooming the timeline to the 18th century and filtering.

(c) Adding a link from maps to the phenomena they represent.

(d) Constraining link by the class “dbp-ont:Event”

Fig. 7. Querying for maps of the 18th century that show some important historical event (such as the French revolution).

siderable time for those users to get acquainted with the functionality (see Section 6).

4.1. *Functional tool comparison*

In Table 1, we compared a number of SPARQL tools in the Semantic Web with respect to the SPEX design principles and the main requirements in this paper, namely target group and query expressivity required for our scenario (to be introduced in Section 5).

In summary, it seems that the tools do not fully correspond to these principles and requirements, mainly because they lack query expressivity (in particular space-time querying), are designed rather for experienced users or do not support feedback or exploration. However, Rhizomer, ViziQuer and especially gFacet (compare Section 2) seem to have a significant overlap.

VizQuer has a class and property suggester which is grounded in results, very similar to the one we implemented in SPEX, and thus can prevent users from building non-meaningful queries. However, it has a complex expert-oriented user interface which still allows forming non-meaningful queries, and results are not feed-backed into construction and exploration.

Rhizomer is more than a faceted browser in that it supports expressive queries with variables, properties and classes, supplies only grounded suggestions and is made for exploration of lay persons. In Rhizomer, arbitrary class restrictions require filtering of values of the facet “rdf:type”, so that a user needs to understand the RDF syntax to some extent. Rhizomer’s browser and pivoting-based visualisation strategy also makes it difficult to get an *visual overview of a query*: One needs to move along a link (pivot) or read a natural language-like description of the current query. In addition, the pivoting option is visually separated from the facets, which makes it difficult to learn how the current query can be extended.

gFacet [15] is another faceted browser which links hierarchical facets (facets that can be logically combined) with graph-pattern visualization. Facets in gFacet correspond to class nodes in SPEX, which can be expanded via properties linking to new facets. This provides a visual overview of the query. Since expansion is restricted to locally available properties, query construction is also grounded. Information can be filtered by selecting elements in a facet, which broadcasts into other facets of the graph. This provides feedback of results into a query. Query results are expressed as the set of “valid” elements in each facet, which is the set of elements satisfying all selections and property edges. Nodes cannot only be facets, but also ge-

Table 1
Qualitative comparison of SPARQL tools against our principles

Criterion	NITELIGHT	SPARQLViz	ViziQuer	iSPARQL	Rhizomer	RelFinder	gFacet
Validity of visual construction (3.1)	yes	yes	yes	yes	yes	yes	yes
Vocabulary Suggestion (3.2)	yes	no	yes	no	yes	yes	yes
Suggestion grounded in results (3.2)	no	no	yes	no	yes	yes	yes
Results feedback into construction (3.3)	no	no	no	no	yes	no	yes
Handling of space and time in queries (3.4)	no	no	no	no	no	no	partially
Exploration support (3.5)	(yes)	no	no	no	yes	yes	yes
Target groups	experienced	experienced	experienced	experienced	inexperienced	inexperienced	inexperienced
Query expressivity suitable for scenario	no	no	no	no	no	no	no

ographic maps or other displays, and the latter can be used both for displaying and selecting geometric points. Thus, compared to SPEX, gFacet seems to be a tool with comparable functional scope regarding our principles (Table 1). What seems missing, however, is a support for spatio-temporal queries and geometries/temporal entities beyond points.

Tools like gFacet might be candidates for a comparative evaluation in the future. However, this was out of scope in this preliminary user study, as explained below.

5. Scenario: Exploring a repository of historical maps

We tested our approach in terms of a query and exploration scenario where users inexperienced with linked data should answer a number of questions about a linked data repository which contains descriptions of various historical maps. The repository has been built in the course of previous work on linked data map georeferencing and encoding [34,33] and will be used in a corresponding library service. Even though the scenario is domain specific, it illustrates how complex space-time footprints and corresponding queries can be essential for non-expert linked data users, such as historians, and thus serves to make the case for similar kinds of linked spatio-temporal data.

The task was to explore a SPARQL endpoint[19] about historical maps from the University library in Münster[20]. We asked subjects, all of them inexperienced regarding SPARQL and RDF, to answer the following list of questions about this map repository (correct answers given in brackets):

1. Maps with lakes
   (a) How many maps represent lakes? (22)
   (b) List those represented lakes that have a name (Lamak lake, Lac du Castillon, Lac du Chambon, Lac de Serre-Poncon)
2. Maps with cities
   (a) How many historical maps represent cities which are in Poland? (1)
   (b) Show these cities in the browser ("Stettin")
3. Maps with military conflicts
   (a) How many historical maps represent military conflicts? (2)
   (b) Which years do these maps represent? (1801, 1812)
4. Maps about Africa

[19] `http://giv-lodum.uni-muenster.de:8081/parliament/sparql`

[20] `http://sammlungen.ulb.uni-muenster.de/nav/classification/116654`

   (a) How many maps depict Africa? (5)
   (b) How many of them were created by ”Karl Ferdinand Weiland”? (1)
   (c) Which years do these maps represent (1841)?

5. Maps about Europe in the 18th century
   (a) How many maps show historical periods in the 18th century (6)
   (b) Which one of these has the largest map scale (largest scale ratio or smallest divisor)? (“Gefecht bey Reichenberg in Böhmen”)
   (c) Name a village which is represented in this map (“Reichenberg”)

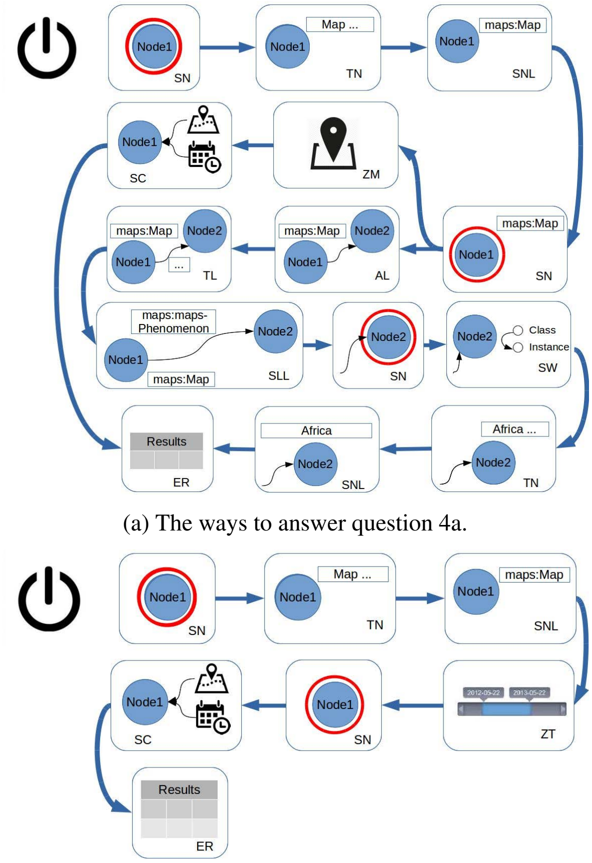
These questions [34] correspond to knowledge that is (at least in principle) of interest to historians and historical cartographers, but cannot be posed in current library search systems [33]. Questions increase in complexity, and even though they are partially repetitive and build on each other in order to ease learning, it requires all of the functionality of SPEX to answer them. The different sub-questions (a, b, c...) cover different aspects, i.e., different kinds of triple pattern, spatio-temporal query strategies and also different sets of processes in Figure 6. Questions where chosen such that it is nearly impossible to answer them by simple text search and that it is difficult to answer them by piecewise exploration (for example by counting in a large result list), but instead require a combination of query and exploration, where queries involve relationship and class instantiation as well as classification. Approximately half of the questions require handling of space and time to some extent. For these reasons, they represent a scenario which is both realistic and challenging for people unfamiliar with SPARQL, space-time data or the repository.

In SPEX, the tasks can be most efficiently solved by the following procedures (compare process steps in the flowchart model of Figure 6):

```
Q 1a ``How many maps represent lakes?'': SN[
    node1]->TN[Map]->SNL[maps:Map]->SN[node
    1]->AL->TL->SLL[maps:mapsPhenomenon]->SN[
    node 2]->TN[Lake]->SNL[phen:Lake]->ER
    ->[``22'']
Q 1b ``List those represented lakes that have
     a name'': ->ER->[``Lamak lake, Lac du
    Castillon, Lac du Chambon, Lac de Serre-
    Poncon'']
Q 2a ``How many maps represent cities which
    are in Poland?'': ->R->SN[node1]->TN[Map
    ]->SNL[maps:Map]->SN[node 1]->AL->TL->SLL
    [maps:mapsPhenomenon]->SN[node 2]->TN[
    City]->SNL[phen:City]->SN[node 2]->AL->TL
    ->SLL[dbp:country]->SN[node2]->SW[
    instance labels]->TN[Poland]->SNL[dbp:
    Poland]->ER->[``1'']
Q 2b ``Show these cities in a browser'': ->BW
    [dbp:Stettin]
Q 3a ``How many historical maps represent
    military conflicts?'': ->R->SN[node1]->TN
    [Map]->SNL[maps:Map]->SN[node 1]->AL->TL
    ->SLL[maps:mapsPhenomenon]->SN[node 2]->
    TN[Conflict]->SNL[phen:MilitaryConflict
    ]->ER[``2'']
Q 3b ``Which years do these maps represent
    ?'':->IO->[``1801, 1812'']
Q 4a ``How many maps depict Africa?'': ->R->
    SN[node1]->TN[Map]->SNL[maps:Map]->SN[
    node 1]->AL->TL->SLL[maps:mapsPhenomenon
    ]->SN[node 2]->SW[instance labels]->TN[
    Africa]->SNL[Africa]->ER[``5'']
            [alternative: using ZM to Africa]
Q 4b ``How many of them were created by Karl
    Ferdinand Weiland?'': ->SN[node 1]->AL->
    TL->SLL[dct:creator]->SN[node 3]->SW[
    instance labels]->TN[Weiland]->SNL[Karl
    Ferdinand Weiland]->ER[``1'']
Q 4c ``Which years do these maps represent
    ?'': ->ZT->[``1841'']
Q 5a ``How many maps show historical periods
    in the 18th century?'': ->R->SN[node1]->
    TN[Map]->SNL[maps:Map]->ZT[1700-1800]->SN
    [node 1]->SC->ER->[``6'']
Q 5b ``Which one of these has the largest
    maps scale?'': ->SN[node 1]->AL->TL[Scale
    ]->SLL[maps:hasScale]->ER->[``Gefecht bey
     Reichenberg in Boehmen'']
Q 5c ``Name a village which is represented in
     this map'': ->SN[node 1]->AL->TL->SLL[
    maps:mapsPhenomenon]->SN[node 2]->TN[
    Village]->SNL[phen:Village]->ER->[``
    Reichenberg'']
```

Listing 1: Flowchart based procedures for expected solutions.

To illustrate, let us take a closer look at questions 4a and 5a. In 4a (“How many maps depict Africa?”), the user starts with a reset (Figure 8a) , selects the empty start node (SN), types (TN[“Map”]) and selects the class of “maps” (SNL[maps:Map]) to restrict the node to maps. Then he adds a link (AL), types (TL) and selects the property that links maps to what they represent (SLL[maps:mapsPhenomenon]). Then he selects the newly introduced node (SN), changes it from “class” to “instance” mode, types and selects the instance “Africa” (SNL) and looks at the results (ER) to count the maps in the table. Alternatively, after restricting node 1 to maps, he could also zoom the map window (ZM) to where Africa is and filter the query by



(a) The ways to answer question 4a.

(b) The way to answer question 5a.

Fig. 8. Expected solutions for questions 4a and 5a.

the map window (SC) correspondingly. In 5a ("How many maps show historical periods in the 18th century?"), after reseting (Figure 8b), the user selects the empty node (SN), types (TN) and selects the map class (SNL[maps:Map]), and then zooms the timeline to the 18th century (ZT). Then he selects node 1 (SN), filters the query by the timeline (SC) and explores the results (ER).

6. Evaluation

6.1. *User study design*

We tested the functionality of SPEX by observing how users performed on the tasks in Section 5. Note that we did neither do a quantitative evaluation of SPEX usability nor an empirical comparison. For our purpose, the functional differences between existing tools, especially regarding space-time functionality, seemed too large to allow for comparisons which exactly involve these functions (compare results in Table 1). Furthermore, our study was preliminary and aimed at discovering in how far the design principles are in effective use, as well as the remaining problems with the tool in order to reach a maturity sufficient for usability tests. While this does not yet allow drawing representative conclusions, it already gives fundamental insights regarding the general workings of the tool and regarding our research questions, which is a necessary basis for further prototype improvement and larger tests.

The SPEX tool was tested with potential users in two stages: in the first stage a heuristic evaluation was done with an expert. The outcomes of this evaluation were used to adjust and improve the information in the "Help file[21]", the formulation of the tasks (see section 5) and some aspects of the interface. Thereafter, the tool was tested with seven subjects (one of them only in a pre-test) who are familiar with the problem of searching for geographic information on the basis of particular criteria. Two subjects are employed as librarian / information specialist in two different universities, two other test persons are currently doing PhD research in the geo domain and the last two test persons are scientists / lecturers at the Faculty ITC of the University of Twente.

The test persons were first asked to familiarize themselves with the SPEX tool through the "Help file" (which was printed on paper) and SPEX displayed on the screen. They were allowed, and even encouraged, to interact with the interface to try to better understand its functioning and design. Thereafter, the test persons were asked to execute the tasks as described in Section 5. One test person took 94 minutes to go through the Help file and familiarize himself with the tool. The other subjects took less time, but still 45 minutes or more. For the execution of the tasks, usually 30 to 45 minutes were needed, although not all test persons could complete all the tasks.

The use and user issues of the SPEX prototype were discovered through qualitative user research in which a mix of methods was applied. Such a mixed approach is quite common in current user research, as different methods lead to different and complementary information about the interaction of a subject with the tool or interface [6,2,42]. We observed the

[21] `http://www.geographicknowledge.de/spex/help.html`

test persons while they were going through the “Help file” and while they were executing their tasks with SPEX. We did that by asking them to constantly think aloud (audio recorded), by screen and event logging (mouse clicks, etc.), by eye movement registration and by video recording their facial expressions. All these recordings were synchronously made, stored and analyzed in a Tobii X60 hardware and Tobii Studio 3.2.1 software envionment, installed in the cartographic usability laboratory of the Faculty ITC of the University of Twente.

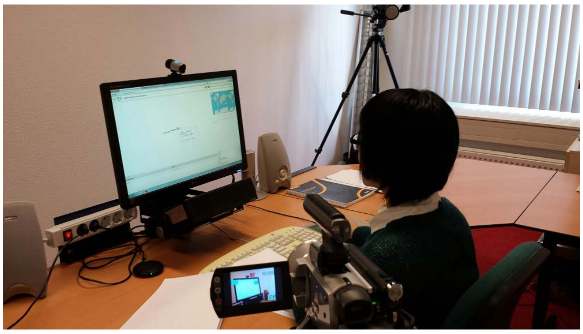


Fig. 9. The configuration of screen, eye tracking and video recording with one test person at ITC.

The 8 - 10 hours recordings (in total) were analyzed in a qualitative way by matching the hypothetical SPEX interaction model (Figure 6) with the thinking aloud and activity protocols, and by studying the usage of the space time interfaces in greater detail with gaze maps based on eye-tracking.

The execution of the user tests went fine although some test persons had difficulties with thinking aloud (they had to be reminded to do so every now and then by the research leader who was in the same room all the time) and the eye movement registration of some subjects sometimes also suffered from the fact that at the end of the long sessions the test persons substantially moved their chairs and bodies. The research leader did not interfere with this in order not to disturb the think aloud process. All in all, the recordings resulted in a substantial and valuable amount of research data. At the end of each test session, the research leader briefly interviewed each subject and asked them about their general satisfaction with SPEX. The execution of the user tests went fine although some test persons had difficulties with thinking aloud (they had to be reminded to do so every now and then by the research leader who was in the same room all the time) and the eye movement registration of some subjects sometimes also suffered from the fact that at the end of the long sessions the test persons substantially moved their chairs and bodies. The research leader did not interfere with this in order not to disturb the think aloud process. All in all, the recordings resulted in a substantial and valuable amount of research data. At the end of each test session, the research leader briefly interviewed each subject and asked them about their general satisfaction with SPEX.

6.2. *Discussion of general results*

For each user, we recorded the path they were following in terms of our flowchart from Figure 6 to see which exploration-query steps they were performing, and which of them led to a success regarding our tasks, which did not, and why.

A synopsis of successes and failures is given in Table 2: “success” means that the question was answered correctly and as expected; “failed” means that the answer and the way to the solution were incorrect; “success (quasi)” means the way of solution was entirely correct but the answer was not, and this only because of the influence of previous failures (dependencies between answers); “partial” means that the question was not answered correctly, but a significant part of the way to the solution was correct; “skipped” means that the question was skipped for some reason (e.g., because it depends on predecessors which were not answered, or because participants were exhausted or simply forgot to give an answer). The addition of “(a)” for “alternative” means that the test person used an alternative (unforeseen) but in principle valid way to the solution. We also added the solution ratio, which is the proportion of successfully answered questions (either success or quasi success) divided by the number of answers given (without skipped questions).

What can be seen from this table can be summarized as follows:

1. SPEX in principle allows non-expert users of linked data to answer almost all these questions about an unknown repository (compare test person 1 with a success ratio of 6/8). There is a small learning phase during the first question, after which most people’s performance stabilizes quickly. The most difficult questions, however, required a lot of reasoning and iterative trials by all test persons. Question 5 (the most difficult one) was obviously too difficult for beginners.

Table 2

Successes and failures of test persons in finding answers to questions from Section 5

User	Q 1a	Q 1b	Q 2a	Q 2b	Q 3a	Q 3b	Q 4a	Q 4b	Q 4c	Q 5a	Q 5b	Q 5c	sol. rat.
1	success	success	partial	skipped	success	success	partial	success (quasi)	success (quasi)	partial	skipped	skipped	6/8 (0.75)
2	partial	success	success	success	success	skipped	partial	skipped	skipped	failed	skipped	skipped	4/7 (0.57)
3	partial	success (a)	success	success	partial	skipped	success	success (a)	skipped	partial	failed	skipped	5/9 (0.55)
4	partial	success (quasi)	success (a)	success	success (a)	partial	failed	skipped	skipped	failed	skipped	skipped	4/8 (0.5)
5	partial	partial	success (a)	success	partial	skipped	failed	success (quasi)	failed	skipped	skipped	skipped	3/8 (0.38)
6	partial	success (quasi)	partial	skipped	skipped	skipped	skipped	skipped	skipped	skipped	skipped	skipped	1/3 (0.33)
sol. rat.	1/6 (0.16)	5/6 (0.83)	4/6 (0.66)	4/4 (1.00)	3/5 (0.6)	1/2 (0.5)	1/4 (0.25)	3/3 (1.00)	1/2 (0.5)	0/4 (0)	0/1 (0)		

2. Complete failures seldom occur (in 14% of all trials). Persons chose at least partially valid ways to an answer (in 85%). Some of them even figured out creative ways to a solution that we did not anticipate (see discussion below for examples).
3. Furthermore, taking into account that the difficulty and level of complexity of our questions was considerably high right from the start and increased a lot towards the end, we notice that the level of successes is still quite high (52%). However, a considerable *exhaustion* of users can be observed towards the end in terms of a significant increase of skipped trials as well as a decrease of successes.

Listing 2 shows an excerpt of our flowchart based recordings for test person 1 in Table 2. Explanation text in square brackets was added by us, and quotes were uttered by the test person.

```
Q 3a ``How many historical maps represent
    military conflicts?'': R->SN->SNL[Map]->
    SN->AL->SLL[mapsPhenomenon, ``Military
    conflict is a phenomenon, probably
    '']->[``Strange that those arrows are
    always pointing into a different
    direction'' (visual arrows direction vs.
    text direction)] ->SN[selects phenomenon
    node]->SNL[``Military Conflict'']-> [
    correct answer: 2]
Q 3b ``Which years do these maps represent
    ?'':  ->ER->IO[inspects time events]->[
    correct answer: 1801, 1812]
Q 5a ``How many maps show historical periods
    in the 18th century?'': R[restart with
    new window]-> SN->SNL[Maps]->[``And now
    the 18th century. Oh I should query that
     ..'']->SN-AL-SLL[does not find property,
    ``What does time mean?'', select date]->
    ER[``Is date the one?'' ``I don't know
    how to do this. I can do it with the
    timeline. But what does the timeline tell
     me? Is it the creation date?'']-> U[
    undoes; would like to have a delete
    function in the node]->SN->ZT[scrolls
    timeline to 18th century; clicks on
    events and wonders what happens] -> SW[
    confused; ``I am doing too many things in
     one window here'']->AL->SLL[``has a
    period. That's what I am looking for'',
    but only finds date, select data again]->
    ER[``I only get one. There is nothing
    that is called a 'period'.'']
```

Listing 2: Flowchart based recording of test person 1.

Comparing Listing 2 with Listing 1 illustrates how usability problems can be spotted immediately. For example, the user in Listing 1 does not succeed in querying with the timeline for Q 5a because he is confused about how the temporal values in the time line relate to query results (creation date?) and whether or not he should rather try a temporal query with a graph query pattern (using the “date” property).

In the following, we will discuss the general insights from our analysis. We start with our observations regarding the functioning of our design principles:

– People made effective use of all visual query graph construction possibilities. Choosing node labels and adding visual edges seems in general to translate well into the questions of the task (provided, however, that people understand the meaning of vocabulary terms). All people used visual link construction as well as node labeling to conceptualize the question and to solve their query task. Test person 1 explicitly formulated questions in terms of node-link query patterns, even though linked data principles were unknown to this person.

- People used result exploration, Web browsing as well as map and timeline exploration in order to answer their questions in ways that were not foreseen by us (“alternative solution”). They used many trials to figure out different ways to reach an equivalent goal (and indeed, “many ways lead to Rome”), e.g.: Using map geometry sizes to figure out map scales; using Web link information (DBpedia) to find information about associated times; Using two filters on space and time and the property “hasScale” to answer questions 5a and 5b, which in principle is a correct (but unexpected) answer.
- People ran through many query-exploration cycles iteratively (which is not time consuming because SPEX is modeless, i.e., works without menu hierarchies, and has an undo/reset button). This allowed them to quickly learn something new which they could use in the next iteration, i.e., they could make iterative learning progress. For example, test person 1 figured out that pressing return is necessary after input in his first cycle, and then used this in all subsequent cycles. Another test person figured out (after several unsuccessful trials with browsing map descriptions on the Web) that map scales can also be compared via geometry sizes in the map window, which lead to an alternative solution.
- People explored term meanings using term suggestions and result feedback. For example, test person 1 explored the meaning of the term “Topographic Map” by checking whether results are a subset of “Map”.

The following list contains main difficulties users were confronted with when using the current version of the tool on the given tasks. We classified them according to the causes of their confusion or task failure and added possible cures:

- Difficulties with understanding linked data vocabularies and example (historical map) data models
    - ∗ People had difficulties in guessing (without any explanation) the conceptual data model underlying our scenario which treats maps as well as their contents as linked nodes. For example, one test person largely failed to understand that in the repository, maps are entities different from lakes and countries to be selected by a separate node and in need of linking to the latter. This issue has probably to do with the particularly unusual way of modeling map contents by linked data.
    - ∗ People had difficulties with abstract and unclear categories such as “hasPhenomenon” in the property list. For example, test person 1 subsumed “lake” under “phenomenon” but not “country” in the case of Poland and therefore failed to answer question 2. Testers also had trouble understanding the difference between *dc:date* and the word “period” as used in question 5a), compare Listing 2. This is a consequence of the particular vocabulary terms used and a lack of their explanation in the current version of SPEX. Understanding vocabularies can be improved by adding vocabulary exploration or explanation tools. For example, a simple measure would be to load RDF term comments from the Web and show them to a user whenever a term needs to be selected.
    - ∗ Prefixes occurring in front of terms were disturbing (test person 1: “dct: I have no idea what that is”). Vocabulary prefixes should be explained (or removed).
- Difficulties with the context menu options
    - ∗ The query terminology used in context menus was difficult to understand. For example, the distinction between things of a kind and particular things (test person 1: “Why do I have to make the decision?”) could be removed.
    - ∗ People were easily confused about the necessity of pressing return to go to the next step (extra cognitive effort).
    - ∗ Node context menu options were not offered in sequential steps but all together at all times. For example, label selection was offered also after a label was selected. This made it hard to think about what step was next (test person 1: “I would not expect to see maps here in text form if I have selected the label map already.” “There should be a difference between what

you have left behind and what your next task is going to be".)

6.3. The role of space-time in exploratory querying

We did an additional analysis including eye-tracking results in order to learn in how far and how successfully the space-time functionality was used to solve exploratory querying tasks.

Table 3 shows a synopsis of the usage of the space time windows regarding SPEX functionality as in Figure 6: inspection of query result objects in space time windows (IO), zooming of map (ZM) and time line (ZT) and the use of space-time filtering (SC). What can be seen from this table is that the space-time interface was used for roughly half of the questions (especially Questions 3b, 4a/c, and 5a). Compared to what we would consider meaningful given our expected solutions (last row in Table 3), it appears that candidates 0-4 (including the three most successful ones) closely matched our expectations. That is, these users, including our pre-test (0), used space-time windows in a way which can be considered appropriate for the corresponding task, even if they failed to answer the question, and provided they did not skip the task for some reason. However, the less successful candidates (4,5,6) also made considerably less use of the space-time windows. User 6 used the map window for Q 2a, which can be considered appropriate (as Poland can be searched in the map), even though not leading to success. According to our recordings, user 4 obviously thought that the map was only illustrative and thus did not consider it as a tool, while user 5 for some reason only focused on query pattern construction. Usage in general was mostly explorative. Regarding successes, if we compare table 3 with 2, we see that the questions requiring space-time interfaces (Q 3b, 4c and 5a) were a bit less successfully answered than overall (2 successes (25%), 3 partial (37,5%) and 3 failures (37,5%) out of 8 trials). However, these questions were also among the more difficult ones in terms of general complexity. Quite a few people skipped these tasks. Concerning Q 3b, our recordings reveal that people skipped sometimes because they thought the solution was obvious after they tried out the time slider. Thus the solution ratio might be actually higher.

If we take a look at the gaze plots (Figures 10), we see clear differences in the way users approached their tasks, which stands in a certain relation to their respective success: For Q 4a, user 1 (Fig. 10a) made heavy use of the time and space window, almost neglecting the query pattern construction, while user 4 (Fig. 10b), in contrast, almost neglected space-time windows. User 1 partially solved the task, while user 4 completely failed. User 3 (Figure 10c) used space-time windows and query pattern construction in a more balanced way and successfully solved the task. Figure 10d shows how user 5 tried to answer Q 3b with hardly any time exploration, which is not possible. Thus, the gaze differences between people show not only a different awareness of the possibilities that these windows offer, they also illustrate different strategies and partially reveal the causes for success or failure.

In analyzing the user test, we still discovered a number of problems with the current space-time interface, and addressing these could greatly improve this result. In particular, the space-time filter functionality (SC), which was very relevant in Q 4a and 5a (compare Table 3), was not used very often and not very successfully compared to space-time exploration. In general, people said that the map window should be more prominent and should be better explained. People found it difficult to decide whether they should treat time and space constraints via filters or via relationships in the query window. They tried out both and were usually exhausted when one option fails and did not try the other option any more. Furthermore, the meaning of the link between map and time objects and results remained unclear. Do the events on the time line represent what is shown in a map or when the map was created (compare Listing 2)? Links to space time literals are currently not shown to the user. Finally, cluttered map and time objects makes space time windows less useful.

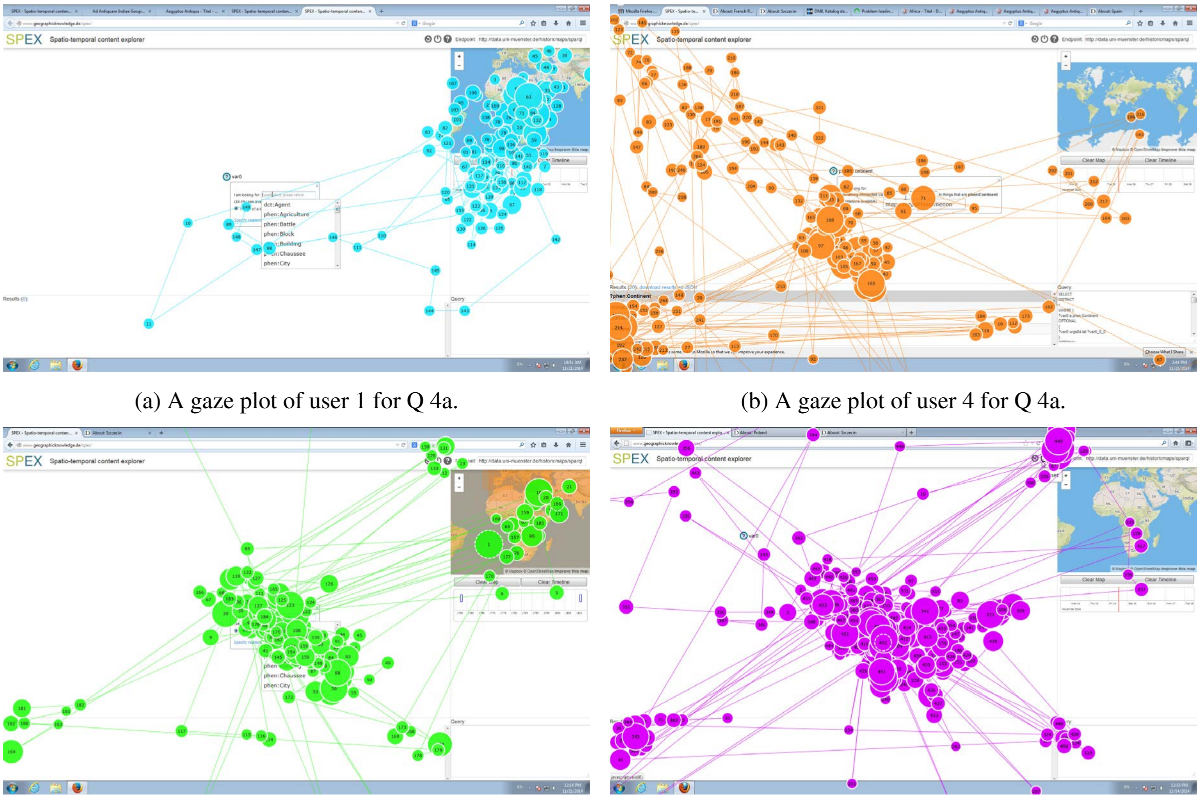
6.4. Summary

In summary, the preliminary study shows that our design, i.e., the query, exploration and feedback strategies on which it is based, were effectively used in a large number of cases. It also highlights the frequent and purposeful use of space-time windows, as well as the different roles they can play in solution strategies. *Space-time exploration* was purposefully and rather naturally adopted for answering questions with linked data in almost all relevant cases, while *space-time filtering* obviously requires more explanation and needs to be further improved. One way would be to filter nodes automatically when zooming, without requiring a user to set a filter constraint. Successful users seemed to adopt a "balanced" strategy, combining space-time interaction with query construction whenever appro-

Table 3

Usage of space-time functionality in exploratory querying compared to our expectation *exp.*. Function labels as in Figure 6. User 0 was a pre-test.

User	Q 1a	Q 1b	Q 2a	Q 2b	Q 3a	Q 3b	Q 4a	Q 4b	Q 4c	Q 5a	Q 5b	Q 5c
0	no	no	no	no	no	IO	SC	no	no	ZT/SC	ZM	skipped
1	no	no	no	no	no	IO	ZM	no	IO/ZT	ZT	skipped	skipped
2	no	no	no	no	no	ZT	no	skipped	skipped	no	skipped	skipped
3	no	no	no	no	no	skipped	ZM	no	ZT	IO/SC/ZT	no	skipped
4	no	no	no	no	no	no	no	skipped	skipped	no	skipped	skipped
5	no	no	no	no	no	skipped	no	no	no	skipped	skipped	skipped
6	no	no	ZM/ZT	skipped	skipped	skipped	skipped	no	skipped	skipped	skipped	skipped
exp.	no	no	ZM	no	no	IO/ZT	ZM/SC	no	IO/ZT	ZT/SC	IO/ZM	IO/ZM



(a) A gaze plot of user 1 for Q 4a.

(b) A gaze plot of user 4 for Q 4a.

(c) A gaze plot of user 3 for Q 4a.

(d) A gaze plot of user 5 for Q 3b.

Fig. 10. Gaze plots of different users for questions Q 3b and Q 4a.

priate. The question is how such a balanced strategy can be supported. The reasons for usability problems were in general mostly located either outside of the scope of the intended functionality of SPEX (e.g., supporting the understanding of vocabularies and domain models was not a goal of our design), or they were related to implementation choices that are not based on our design principles (compare Section 3). That is, the discovered problems might exist precisely because the principles were not yet implemented with enough consequence: For example, the syntactical choice between class and instance based queries was driven by our understanding of RDF but contradicts principle 3.1, because it is neither explained nor even necessary. The missing semantic link between results and map/time objects and missing exploration of result name spaces contradict principle 3.5. And showing a menu option also after its use contradicts the principle of progressive disclosure of options (Compare Section 3).

Overall, users said in the interviews that the tool offers an interesting way of searching. Especially the time window was considered a useful part. This fits with our observations that users frequently went through query and exploration cycles to approach a valid solution, even if they failed in the end. These cycles were enabled by our design strategies. For these reasons, we conclude that the study supports the general functioning of our design principles and the way they integrate space and time, even though it highlights also clear opportunities for improvement of the tool.

7. Conclusion

In current linked data retrieval systems, query and exploration are often not tightly integrated and do not handle space and time in both query and exploration. This prevents users unfamiliar with a given repository from taking advantage of the self-explanatory power of linked data and the intuitive retrieval possibilities of maps and time sliders in order to find out about interesting repository content by themselves. In this paper, we have proposed design principles for exploratory querying of SPARQL endpoints in space and time, used them to compare existing tools, implemented them in terms of the SPEX tool, and tested them in a preliminary user study. The latter was based on a number of questions about a historical maps repository which require complex query construction and interaction with space-time interfaces.

Results show that the tightly closed query and exploration loops of our design enable novices on linked data to pose and correctly answer complex queries about a repository they would not be able to answer otherwise. Modeless design and immediate feedback of query on exploration and vice versa enable users to iteratively explore query functionality, data content and vocabulary meaning, as well as to find alternative solution paths for a given question. Space and time windows were used whenever appropriate and mostly for exploratory purposes. Our analysis furthermore revealed rather diverse solution strategies. However, we also found that successful question answering still requires a simplification of the space-time filtering interface, a better understanding of vocabularies, improved usability of context menus, and a better linking of maps/time sliders with other displays.

In the future, we plan to address these issues by corresponding developments as indicated in this paper. Furthermore, there are options to improve both the variety of query and exploration methods as well as the scalability of SPEX. The latter requires mainly server side support for spatio-temporal queries.

Acknowledgments

This work has been funded by the German Research Foundation (DFG) through the *Linked Data for eScience Services* (LIFE) project (KU 1368/11-1). We thank our project partners, the Münster University Library (ULB), for their constant support of this work. Furthermore, we thank all participants of our user test for letting us peek over their shoulders.

References

[1] Alonen, M., Kauppinen, T., Suominen, O., Hyvönen, E.: Exploring the Linked University Data with visualization tools. In: Cimiano, P., Fernández, M., Lopez, V., Schlobach, S., Völker, J. (eds.) The Semantic Web: ESWC 2013 Satellite Events. pp. 204–208. Springer, Montpellier, France (2013)

[2] e Amin, F., Alghamdi, A.S.: A mixed method study on usability evaluation of smartphone web browsers. Journal of Internet Technology 15(5), 783–792 (2014)

[3] Auer, S., Doehring, R., Dietzold, S.: Less - template-based syndication and presentation of linked data. In: Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part II. pp. 211–224. ESWC'10, Springer-Verlag, Berlin, Heidelberg (2010)

[4] Battle, R., Kolas, D.: Enabling the geospatial Semantic Web with Parliament and GeoSPARQL. Semantic Web 3(4), 355–370 (2012)

[5] Becker, C., Bizer, C.: Exploring the geospatial semantic web with dbpedia mobile. J. Web Sem. 7(4), 278–286 (2009)

[6] Bleisch, S.B.: Evaluating the appropriateness of visually combining quantitative data representations with 3D desktop virtual environments using mixed methods. Ph.D. thesis, City University, London (2011)

[7] Brunetti, J.M., García, R., Auer, S.: From overview to facets and pivoting for interactive exploration of semantic web data. Int. J. Semantic Web Inf. Syst. 9(1), 1–20 (2013)

[8] Catarci, T., Costabile, M.F., Levialdi, S., Batini, C.: Visual query systems for databases: A survey. J. Vis. Lang. Comput. 8(2), 215–260 (1997)

[9] Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM 13(6), 377–387 (Jun 1970)

[10] Dadzie, A.S., Rowe, M.: Approaches to visualising linked data: A survey. Semantic Web 2(2), 89–124 (2011)

[11] Franklin, C., Hane, P.: An introduction to GIS: linking maps to databases. Database 3(4), 333–354 (2012)

[12] Graves, A.: Creation of visualizations based on linked data. In: Proceedings of the 3rd International Conference on Web In-

telligence, Mining and Semantics. pp. 41:1–41:12. WIMS '13, ACM, New York, NY, USA (2013)

[13] Haag, F., Lohmann, S., Bold, S., Ertl, T.: Visual sparql querying based on extended filter/flow graphs. In: Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces. pp. 305–312. ACM (2014)

[14] Hart, G., Dolbear, C.: Linked Data: A Geographic Perspective. CRC Press, Boca Raton (2013)

[15] Heim, P., Ziegler, J., Lohmann, S.: gfacet: A browser for the web of data. In: Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSWâĂŹ08). vol. 417, pp. 49–58 (2008)

[16] Hogenboom, F., Milea, V., Frasincar, F., Kaymak, U.: Rdf-gl: a sparql-based graphical query language for rdf. In: Emergent Web Intelligence: Advanced Information Retrieval, pp. 87–116. Springer (2010)

[17] Hornbæk, K., Frøkjær, E.: Do thematic maps improve information retrieval. In: Human-Computer Interaction (INTERACT 99). pp. 179–186 (1999)

[18] Janowicz, K.: The role of space and time for knowledge organization on the semantic web. Semantic Web 1(1-2), 25–32 (2010)

[19] Janowicz, K., Scheider, S., Pehle, T., Hart, G.: Geospatial semantics and linked spatiotemporal data - past, present, and future. Semantic Web 3(4), 321–332 (2012)

[20] Jones, C.B., Purves, R.S.: Geographical information retrieval. Int. J. Geogr. Inf. Sci. 22(3), 219–228 (2008)

[21] Jones, J., Kuhn, W., Keßler, C., Scheider, S.: Making the web of data available via web feature services. In: Huerta, J., Schade, S., Granell, C. (eds.) Connecting a Digital Europe Through Location and Place, pp. 341–361. Lecture Notes in Geoinformation and Cartography, Springer International Publishing (2014)

[22] Kadlag, A., Wanjari, A., Freire, J., Haritsa, J.: Supporting exploratory queries in databases. In: Lee, Y., Li, J., Whang, K.Y., Lee, D. (eds.) Database Systems for Advanced Applications, Lecture Notes in Computer Science, vol. 2973, pp. 594–605. Springer Berlin Heidelberg (2004)

[23] Kauppinen, T., de Espindola, G., Jones, J., Sanchez, A., Gräler, B., Bartoschek, T.: Linked Brazilian Amazon Rainforest Data. Semantic Web Journal 5(2) (2014)

[24] Keßler, C., Janowicz, K., Kauppinen, T.: spatial@linkedscience - exploring the research field of giscience with linked data. In: Xiao, N., Kwan, M.P., Goodchild, M., Shekhar, S. (eds.) Geographic Information Science, Lecture Notes in Computer Science, vol. 7478, pp. 102–115. Springer Berlin Heidelberg (2012)

[25] Kovarsky, J.: Searching for early maps: Use of online library catalogs. ACMLA Bulletin (138) (2011)

[26] Kuhn, W., Kauppinen, T., Janowicz, K.: Linked data - A paradigm shift for geographic information science. In: Geographic Information Science - 8th International Conference, GIScience 2014, Vienna, Austria, September 24-26, 2014. Proceedings. pp. 173–186 (2014)

[27] Kveladze, I., Kraak, M.J., van Elzakker, C.P.: A methodological framework for researching the usability of the space-time cube. The Cartographic Journal 50(3), 201–210 (2013)

[28] Kyzirakos, K., Karpathiotakis, M., Koubarakis, M.: Strabon: a semantic geospatial dbms. In: The Semantic Web–ISWC 2012, pp. 295–311. Springer (2012)

[29] Larson, R.: Geographic information retrieval and digital libraries. In: Agosti, M., Borbinha, J., Kapidakis, S., Papatheodorou, C., Tsakonas, G. (eds.) Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science, vol. 5714, pp. 461–464. Springer Berlin Heidelberg (2009)

[30] Mandel, T.: The Elements of User Interface Design. John Wiley & Sons, Inc., New York, NY, USA (1997)

[31] Marchionini, G.: Exploratory search: From finding to understanding. Commun. ACM 49(4), 41–46 (Apr 2006)

[32] Purves, R.S., Clough, P., Jones, C.B., Arampatzis, A., Bucher, B., Finch, D., Fu, G.: The design and implementation of spirit: a spatially aware search engine for information retrieval on the internet. International J. of Geographical Information Science 21(7), 717–745 (2007)

[33] Scheider, S., Degbelo, A., Kuhn, W., Przibytzin, H.: Content and context description - how linked spatio-temporal data enables novel information services for libraries. gis.Science (2014), forthcoming

[34] Scheider, S., Jones, J., Sánchez, A., Keßler, C.: Encoding and querying historic map content. In: Huerta, J., Schade, S., Granell, C. (eds.) Connecting a Digital Europe Through Location and Place, pp. 251–273. Lecture Notes in Geoinformation and Cartography, Springer International Publishing (2014)

[35] Schraefel, M., Karger, D.: The pathetic fallacy of rdf. In: International workshop on the semantic web and user interaction (SWUI). vol. 2006 (2006)

[36] Sequeda, J.: Introduction to: Triplestores [`http://semanticweb.com/introduction-to-triplestores_b34996`; last accessed: December 03, 2014] (2013)

[37] Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Proceedings of the 1996 IEEE Symposium on Visual Languages. pp. 336–343 (1996)

[38] Skjæveland, M.: Sgvizler: A javascript wrapper for easy visualization of sparql result sets. In: Extended Semantic Web Conference (2012)

[39] Smart, P.R., Russell, A., Braines, D., Kalfoglou, Y., Bao, J., Shadbolt, N.R.: A visual approach to semantic query design using a web-based graphical query designer. In: Gangemi, A., Euzenat, J. (eds.) Knowledge Engineering: Practice and Patterns (EKAW 2008). pp. 275–291. Springer-Verlag Berlin Heidelberg, Acitrezza, Italy (2008)

[40] Stadler, C., Lehmann, J., Höffner, K., Auer, S.: Linkedgeodata: A core for a web of spatial open data. Semant. web 3(4), 333–354 (2012)

[41] Usbeck, R.: Combining linked data and statistical information retrieval. In: The Semantic Web: Trends and Challenges, pp. 845–854. Springer (2014)

[42] Venkatesh, V., Brown, S.A., Bala, H.: Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. MIS quarterly 37(1), 21–54 (2013)

[43] White, R.W., Roth, R.A.: Exploratory Search: Beyond the Query-Response Paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan & Claypool Publishers (2009)

[44] Zviedris, M., Barzdins, G.: Viziquer: A tool to explore and query SPARQL endpoints. In: The Semanic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011, Proceedings, Part II. pp. 441–445 (2011)