FERASAT: a Serendipity-fostering Faceted Browser for Linked Data

Ali Khalili[1], Peter van den Besselaar[2] and Klaas Andries de Graaf[1]

[1] Department of Computer Science, Faculty of Science
[2] Department of Organization Science, Faculty of Social Sciences
Vrije Universiteit Amsterdam, the Netherlands
[a.khalili, p.a.a.vanden.besselaar, ka.de.graaf]@vu.nl

Abstract. Accidental knowledge discoveries occur most frequently during capricious and unplanned search and browsing of data. This type of undirected, random, and exploratory search and browsing of data results in Serendipity – the art of unsought finding. In our previous work we extracted a set of serendipity-fostering design features for developing intelligent user interfaces on Semantic Web and Linked Data browsing environments. The features facilitate the discovery of interesting and valuable facts in (linked) data which were not initially sought for. In this work, we present an implementation of those features called FERASAT. FERASAT provides an adaptive multigraph-based faceted browsing interface to catalyze serendipity while browsing Linked Data. FERASAT is already in use within the domain of science, technology & innovation (STI) studies to allow researchers who are not familiar with Linked Data technologies to explore heterogeneous interlinked datasets in order to observe and interpret surprising facts from the data relevant to policy and innovation studies. In addition to an analysis of the related work, we describe two STI use cases in the paper and demonstrate how different serendipity design features are addressed in those use cases.

1 Introduction

“Unless you expect the unexpected you will never find [truth], for it is hard to discover and hard to attain.” -*Heraclitus*[3]

Serendipity (known as the art of unsought finding [19]) plays an important role in the emerging field of data science, allowing the discovery of interesting and valuable facts not initially sought for. Serendipity consists of two main steps: a *surprising observation* and then a *correct interpretation*. In our previous work [13], we extracted a set of serendipity-fostering design features (cf. Table 1) which help designers and developers of Semantic Web and Linked Data browsing environments to facilitate discovery and interpretation of new, useful, and interesting facts. For example, design Feature F_{10} (in Table 1) suggests sharing of surprising observations among multiple users to increase the chance encounter.

[3] according to secondary sources

Table 1: List of the proposed serendipity-fostering design features [13].

Design Features Related to Observations	
$\mathbf{F_1}$	Make surprising observations more noticeable.
$\mathbf{F_2}$	Make errors in data more visible in order to detect successful errors easier.
$\mathbf{F_3}$	Allow inversion and contrast.
$\mathbf{F_4}$	Support randomization and disturbance.
$\mathbf{F_5}$	Allow monitoring of side-effects when interacting with data.
$\mathbf{F_6}$	Support detection and investigation of by-products.
$\mathbf{F_7}$	Support background knowledge and user contextualization.
$\mathbf{F_8}$	Support both convergent and divergent information behavior.
Design Features Related to Explanation of the Observations	
$\mathbf{F_9}$	Facilitate the explanation of surprising observations.
$\mathbf{F_{10}}$	Allow sharing of surprising observations among multiple users.
$\mathbf{F_{11}}$	Enable reasoning by analogy.
$\mathbf{F_{12}}$	Support extending the memory of user by invoking provocative reminders and relevance feedback.

A surprising observation done by user A, when correctly explained by user B, can result in positive serendipity.

In this work, we present an implementation of those serendipity-fostering features called "FERASAT"[4] (FacEted bRowser And Serendipity cATalyzer). FERASAT provides an adaptive multigraph-based faceted browsing interface to catalyze serendipity while browsing Linked Data. It is important to mention that serendipitous discovery may be facilitated but it is by definition an emergent process. With our work we aim to facilitate the process of serendipity by providing an incubator-like environment for serendipity. In other words, the FERASAT environment will increase the likelihood of serendipity, without guaranteeing it.

FERASAT is built on top of the LD-R framework [11] to support the following serendipity-fostering principles:

- *Skeuomorphic Design.* "One way of overcoming the fear of new is to make it look like the old", said Don Norman to describe the process of skeuomorphic design [15]. Skeuomorphism in the context of UI design, refers to the practice of incorporating recognizable objects which are familiar to users to decrease the cognitive load of users and to hide complexity of the underlying technology. Most of the current Linked Data browsing interfaces fall into the *Pathetic Fallacy of RDF* [9] trap where they display RDF data to the users as a graph because the underlying data model is a graph. Abstracting the complexity of RDF and graph-based data representation provides more easily-discoverable and usable interactions (i.e. more affordances) for non-Semantic Web users to serendipitously browse data scattered over multiple knowledge graphs.
- *Adaptive Design.* In order to support serendipity-fostering features, the browser UI needs to act in a proactive way to persistently customize and personalize

[4] in Persian, the term 'Ferasat' refers to the ability of intuitive knowledge acquisition.

user interfaces based on the type of data and the information needs of the end-users. Intelligent UI adaptation allows the application to act more like a human and consequently, more intelligently.

- *Component-based Design.* Component-based design is a prerequisite to realize intelligent UI adaptation. A component-based environment consists of a set of structured UI components as building blocks, that can dynamically get injected to system, replaced and customized while user is interacting with the system.

The contributions of this work are in particular:

- Design and implementation of an open-source faceted browser to facilitate serendipity while browsing Linked Data.
- An analysis of the related work in terms of the serendipity design features.
- Discussing two use cases in the domain of science, technology and innovation (STI) studies to demonstrate the capabilities of our implemented solution.

The rest of this paper is organized as following: In Section 2 we elaborate on the design and architecture of the FERASAT environment and indicate how it addresses the proposed serendipity-fostering features. Implementation details are described in Section 3. Section 4 presents two use cases on serendipitous knowledge discovery when FERASAT was used in the STI domain. In Section 5, we review the related work in the area of serendipity on Linked Data and compare FERASAT to existing Linked Data-based faceted browsers. Finally in Section 6 we conclude and briefly mention the future directions of this work.

2 FERASAT: A Trigger and Facilitator for Serendipity on Linked Data

Figure 1 depicts the architecture of FERASAT where related elements are color coded. The system provides three main modes of interaction with data namely search, browse and view. During the user interactions, based on the semantics of data and the given user context, the system adapts its behavior by rendering appropriate interactive UI components. In the following sections we describe the main architectural building blocks of the FERASAT environment together with how they support the serendipity-fostering features shown in Table 1.

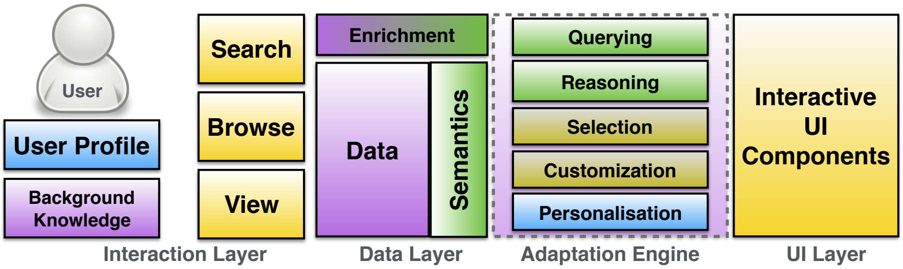


Fig. 1: Architecture of the FERASAT adaptive faceted browser.

2.1 Interaction Layer

According to the theory of “Seven Stages of Action” [15], which explains the psychology of a person while executing a task, user interactions with a system occur in two gulfs namely a *gulf of execution* and a *gulf of evaluation*. The gulf of execution focuses on allowable interactions (i.e., affordances) in the system, whereas the gulf of evaluation reflects the amount of effort that the person must exert to interpret the state of the system after an interaction. Within the FERASAT environment, interactions in the gulf of execution (e.g. interacting to invert a selected facet) are used as triggers for serendipity and interactions in the gulf of evaluation (e.g. visualization & in-detail browsing of the properties of a resource) are used to support the process of abduction.

Figure 2 shows a mock-up of the design we devised for the FERASAT faceted browser (Figure 5 depicts the actual implementation). When browsing a set of linked data which is scattered over multiple knowledge graphs (e.g. Figure 3), the first step is to identify properties of interest as semantic links to move forward and backward in the data space. The index facet lists these designated RDF properties grouped by the aspect they are addressing. In the initial state, all the RDF resources are displayed without any constraints. When a user selects a property, a new facet is generated to display the object values of the selected property together with the number of resources containing those values. The facet can be configured to employ different interactive UI components (e.g. charts, maps, etc. in Figure 2) to render the values of a selected property.

Flexible UI components support the serendipity-fostering features F_1, F_2 and F_4 (see Table 1) by allowing users to exploit multiple interactive visualizations to do surprising observations and also discover successful errors in data together with the possible explanations for their occurrence. The list of values in a facet can be shuffled to change the ordering based on a random factor or some criteria other than the default sorting criterion which is the frequency of the corresponding resources (supporting F_7). When a user selects one or more values of an active facet, a SPARQL query with the corresponding constraints is generated and executed to update the results list. Users can invert the selected values in a facet to see the results which exclude those selected values (supporting F_6). If multiple facets are active, any change to a facet will affect the remaining active facets to take into account the constraint imposed by that change (supporting F_9). Users can focus on each facet, search within its values and view in-detail characteristics of each object value (supporting both convergent and divergent information behavior presented by F_{12}). When a user is browsing a facet which was browsed before, the UI provides some reminders as pop-ups about the previous usage of that facet (supporting F_8).

The result list is the terminal facet in the system which shows the final result of the generated SPARQL query as a set of RDF resources constrained by the selected RDF properties and values. Clicking on a resource reveals the detailed characteristics of that resource (supporting F_2 to study a particular surprising observation). To further investigate the results, a user can select multiple resources and ask for the potential correlations between them (supporting F_9 to

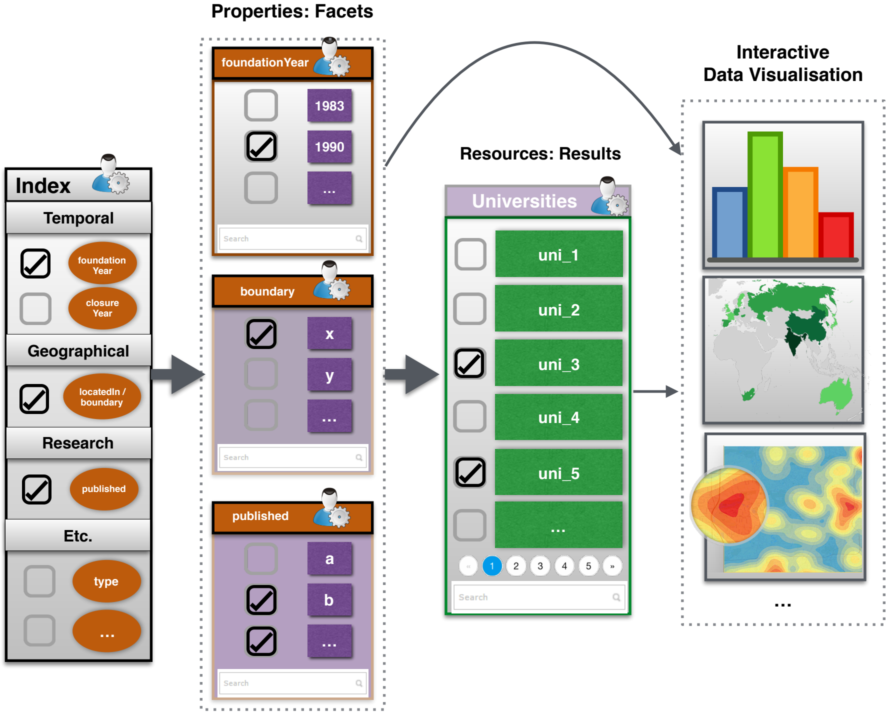


Fig. 2: A mock-up of the FERASAT adaptive faceted browser.

investigate by-products). For example, given the linked data in Figure 3, a user might want to browse and find the relation between entities of type universities which are founded in certain years AND are located in certain administrative boundaries AND have published on certain research topics. Using the faceted browser, such a query is generated in a progressive way where users can investigate the effect of each selected facet on other facets and on the results list, while traversing multiple distributed knowledge graphs (in this case, graphs that provide data about universities connected to graphs that provide values related to publications and administrative boundaries), until the full query is answered.

2.2 Data Layer

There are five different sorts of data taken into account within the FERASAT environment: 1) *user's profile data* to understand the user preferences, 2) *user's background knowledge* to consider a user's domain of interest while browsing data, 3) *original data* to be browsed, 4) *configuration data* as output of adaptation process to customize and personalize both data and UIs, 5) *complementary data* added as enrichment to original data for richer contextualization. All the above datasets are represented as single or multiple RDF graphs (e.g., Figure 3) to be ready for integration (using federated SPARQL queries) and analysis.

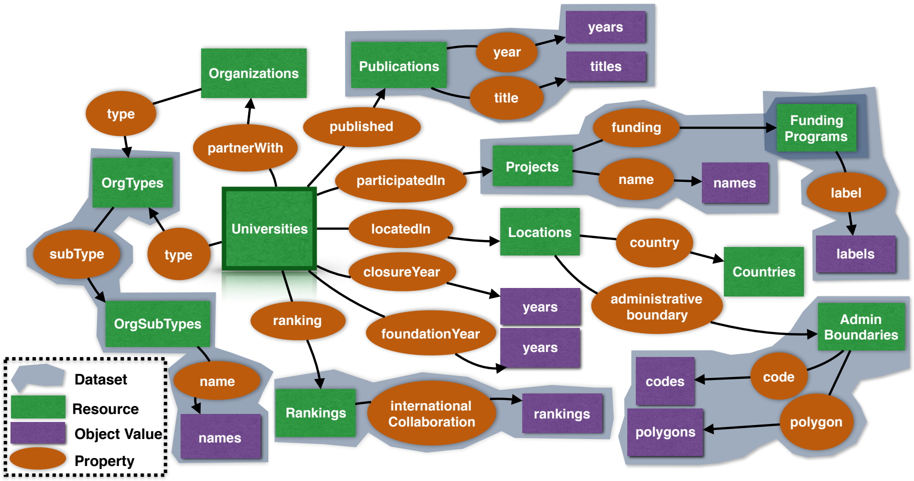


Fig. 3: An example of the Linked Data scattered over multiple graphs (datasets).

FERASAT supports resource annotation to interlink the original data with the user's background knowledge and to generate complementary data connected to the original data to be browsed (supporting F_{11} for user contextualization by giving users additional contextual facets to complement their browsing experience). At the moment, two types of annotation are supported within the system: *Named Entity Recognition* (NER) using DBpedia Spotlight[5] and *Geo-boundary-tagging* supported by open geo boundaries from OpenStreetMap and GADM[6]. There are interactive UIs embedded in the FERASAT system to interactively annotate a dataset before the browsing activity starts. Above semi-automatic annotation supports F_6 and F_8, as information diverges (by adding annotations) with possible by-products resulting from the annotations.

2.3 UI Layer

As depicted in Figure 4, there are four core component levels in an FERASAT Web application. Each core component abstracts the actions required for retrieving and updating the graph-based data and provides a basis for user-defined components to interact with Linked Data in three modes: search, browse and view. The data-flow in the system starts from the Dataset component which handles all the events related to a set of resources under a named graph identified by a URI. The next level is the Resource component which is identified by a URI and indicates the RDF resource to be described in the application. A resource includes a set of properties which are handled by the Property component. Properties can be either individual or aggregate when combining multiple features

[5] http://www.dbpedia-spotlight.org

[6] http://gadm.org

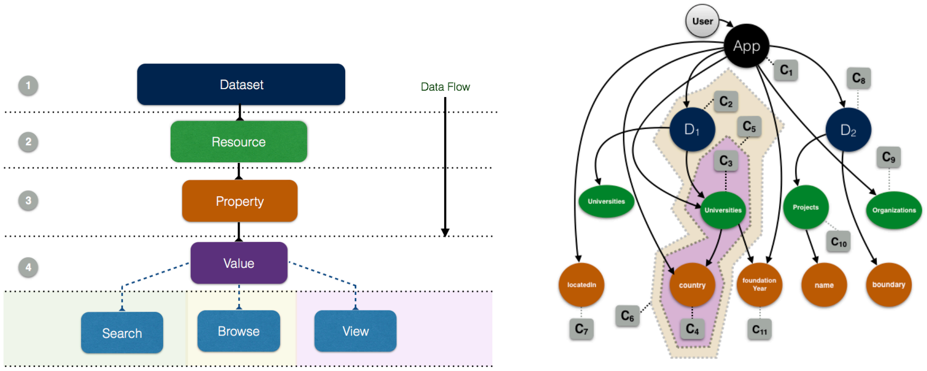


Fig. 4: (left) core reactor components, (right) example configuration hypergraph.

of a resource (e.g. a component that combines longitude and latitude properties; start date and end date properties for a date range, etc.). Each property is instantiated by an individual value or multiple values in case of an aggregate object. The value(s) of properties are controlled by the Value component. In turn, Value components invoke different components to search, browse and view the property values. Value components are terminals in the FERASAT single directional data flow where customized user-generated components (e.g. charts, maps, diagrams, etc.) can be plugged into the system.

2.4 Adaptation Engine

An adaptive UI[7] is a UI which adapts, that is, changes its layout and elements to the needs of the user or context and is similarly alterable by each user. In the context of FERASAT, we devised a particular type of adaptive UI called a *data-aware UI* [10] that a) can understand users' data and b) can interact with users accordingly. As depicted in Figure 1, FERASAT incorporates an adaptation engine to realize data-aware UIs when users interact with data. The task of adaptation engine is to make a bridge between data (enriched by semantics) and existing UI components suitable to render data. The adaptation engine includes the following core components:

- *Querying.* This part is responsible for composing, sharing and running of SPARQL queries within the FERASAT environment. FERASAT exploits the WYSIWYQ (What You See Is What You Query) model [12] to allow sharing, modification and repurposing of SPARQL queries among multiple users in a visual and interactive way (supporting F_3). It also provides a set of SPARQL query templates similar to the one discussed in [3] to find analogous resources within the same domain (partially supporting F_5).
- *Reasoning.* This is the core part of the engine where different datasets mentioned in Section 2.2 are analyzed in an integrative way to find the best strategy for data rendering and UI augmentation.

[7] http://en.wikipedia.org/wiki/Adaptive_user_interface

- *Selection.* This part allows to manually or automatically (as result of the reasoning) select or replace an existing UI component. FERASAT employs a specific notation to identify a chain of facet properties using RDF property path and federated SPARQL queries. For example, `ex:address->geo:geo` selects a property path, `ex:orcidID->[http://orcid.org]rdfs:label` allows using a property path which ends up in a named graph, and `ex:orcidID->[http://myendpoint.com>>http://orcid.org]rdfs:label` refers to a property path which is on a named graph located in a different SPARQL endpoint.[8]
- *Customization.* This part allows to manually or automatically customize an existing UI component.
- *Personalization.* This part allows to manually or automatically personalize an existing UI component. Personalization will overwrite the configurations used for customization to consider the user's context.

Figure 4 shows that the configuration process is done by traversing the hypergraph generated either manually by a user or automatically as result of the reasoning. FERASAT exploits a hierarchical permutation of the Dataset, Resource, Property, and Value components as *scopes* to select specific parts of the UI to be customized or personalized. Each scope conveys a certain level of specificity on a given context ranging from 1 (most specific: DRPV) to 15 (least specific: D). Scopes are defined by using either the URIs of named graphs, resources, and properties, or by identifying the resource types and data types. A configuration is defined as a setting which affects the way the UI components are interpreted and rendered (e.g. render a specific component for a specific RDF property or a specific RDF resource within a specific RDF graph). UI adaptation is handled by traversing the configurations for scopes, populating the configurations and overwriting the configurations when a more specific applicable scope is found.

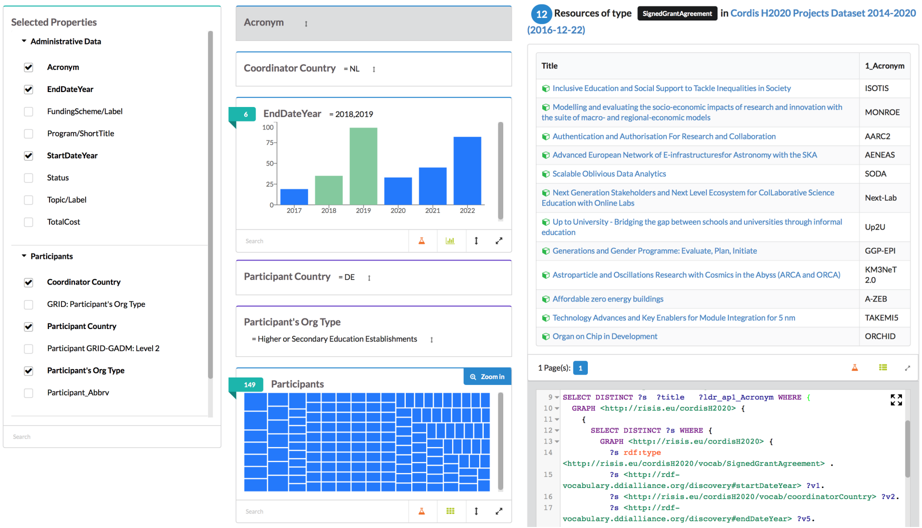
3 Implementation

FERASAT is implemented as a ReactJS component (backed by NodeJS) within the open-source Linked Data Reactor[9] (LD-R) framework and is available to download at `http://ferasat.ld-r.org` together with its documentation and demos (see Figure 5 for an screenshot of the FERASAT environment). FERASAT supports two strategies for creating UI configurations:

- *Dynamic Configurations*: This option (that is necessary for reasoning and automatic UI adaptation) uses a SPARQL endpoint to store user configurations dynamically as RDF. This is a flexible approach that allows updating user configurations on-the-fly without the need to restart or rebuild the entire application. It also facilitates sharing and querying of the user configurations.
- *Static Configuration*: For this option, users do not require to provide a read-/write SPARQL endpoint to store their configurations and can enter and

[8] More details are available at `http://ld-r.org/docs/configFacets.html`

[9] `https://github.com/ali1k/ld-r`



Selected Properties

Administrative Data

- ☑ Acronym
- ☑ EndDateYear
- □ FundingScheme/Label
- □ Program/ShortTitle
- ☑ StartDateYear
- □ Status
- □ Topic/Label
- □ TotalCost

Participants

- ☑ Coordinator Country
- □ GRID: Participant's Org Type
- ☑ Participant Country
- □ Participant GRID-GADM: Level 2
- ☑ Participant's Org Type
- □ Participant_Abbrv

Acronym

Coordinator Country = NL

EndDateYear = 2018,2019

Participant Country = DE

Participant's Org Type = Higher or Secondary Education Establishments

Participants

12 Resources of type SignedGrantAgreement in Cordis H2020 Projects Dataset 2014-2020 (2016-12-22)

Title	1_Acronym
Inclusive Education and Social Support to Tackle Inequalities in Society	ISOTIS
Modelling and evaluating the socio-economic impacts of research and innovation with the suite of macro- and regional-economic models	MONROE
Authentication and Authorisation For Research and Collaboration	AARC2
Advanced European Network of E-infrastructuresfor Astronomy with the SKA	AENEAS
Scalable Oblivious Data Analytics	SODA
Next Generation Stakeholders and Next Level Ecosystem for CollLaborative Science Education with Online Labs	Next-Lab
Up to University - Bridging the gap between schools and universities through informal education	Up2U
Generations and Gender Programme: Evaluate, Plan, Initiate	GGP-EPI
Astroparticle and Oscillations Research with Cosmics in the Abyss (ARCA and ORCA)	KM3NeT 2.0
Affordable zero energy buildings	A-ZEB
Technology Advances and Key Enablers for Module Integration for 5 nm	TAKEMI5
Organ on Chip in Development	ORCHID

1 Page(s): 1

```
SELECT DISTINCT ?s  ?title   ?ldr_ap1_Acronym WHERE {
  GRAPH <http://risis.eu/cordisH2020> {
    {
      SELECT DISTINCT ?s WHERE {
        GRAPH <http://risis.eu/cordisH2020> {
          ?s rdf:type
<http://risis.eu/cordisH2020/vocab/SignedGrantAgreement> .
          ?s <http://rdf-
vocabulary.ddialliance.org/discovery#startDateYear> ?v1.
          ?s <http://risis.eu/cordisH2020/vocab/coordinatorCountry> ?v2.
          ?s <http://rdf-
vocabulary.ddialliance.org/discovery#endDateYear> ?v5.
```

Fig. 5: An screenshot of the implemented FERASAT adaptive faceted browser.

store all their configurations locally on the application server as JSON. This options is faster than the static method (since it removes the overhead of using a triple store) and more secure, however the cost of this option is that users need to restart their application server anytime they modify the configurations. Users can combine this method with the dynamic method where the static configurations always have a higher priority in UI configuration.

4 Use Cases

FERASAT is integrated into the SMS[10] (Semantically Mapping Science) platform as the technical core within the RISIS.eu project. It is actively used to browse data related to Science, Technology & Innovation (STI) studies. A complete list of use cases is available at `http://sms.risis.eu/usecases`. In this section, we provide a brief summary of two use cases related to serendipitous knowledge discovery in the STI domain written by two social scientists who experienced browsing data on FERASAT environment while conducting research.

4.1 Analyzing change in the research/Higher Education (HE) systems

The RISIS datastore contains many datasets with information about organizations. I was mainly interested in structural change in HE systems by navigating through those datasets. The faceted browser was of great help, as it enabled me to explore the available information in a graphical form. While browsing

[10] `http://sms.risis.eu`

the datasets, I found a property “foundation year”. Selecting that property for a country, I got the frequency of new foundations of HE institutions per year (see Figure 5), and I saw immediately (F_1) a high concentration in two consecutive years: in 1986 and 1987 some 21 new HE institutions were founded in the Netherlands, on a total of 114: So some substantial changes in the HE system seem to have taken place! By selecting these two years, the list of organizations shows the names of the institutions that were founded in these two years. I could inspect the list, but also select a single institution and inspect the available information in the datastore, but also more broadly on the web, as all the organizations are also linked to their website and their Wikipedia page (F_7, F_8, F_9). So, I did not only have much numerical data in the data network, such as numbers of students and staff, but also qualitative (textual) data for further inspection. Looking at the various newly founded schools showed that these are all Universities of Applied Sciences, so the “second layer” Dutch HE institutions. By reading the historical information on their Websites, one would find out that the new founded institutions in fact are conglomerations of smaller schools into very large new institutions (F_7, F_9). This indeed can be considered as a major reform of the Dutch HE system.

A follow-up question would be whether this is a typical Dutch phenomenon, or whether similar changes have taken place in other countries (F_{11}). Belgium could be a second case to inspect, and I followed the same steps. Indeed, as the browser shows, also here we find concentrations of foundations of new HE institutions, but now in the year 1995 when 32 new HE institutions were founded in Belgium. If I select the year 1995, I get a list with the names of the newly founded institutions and can further inspect the available information on those institutions. I did not have any prior knowledge on the Belgian system (F_7), but inspecting the list of names in the results, one immediately sees that the changes probably took place in the French speaking part of Belgium, as all institution names are French, and not in the Flemish speaking part (F_6). Indeed, the two language regions have their own HE system, so this could clearly be the case.

The third example I tried was Austria, and indeed also there I detected a concentration of new institutions in 2007 - a decade after the changes in Belgium and two decades after the changes in the Netherlands. Of the total of 102 HE institutions in Austria, 15 were created in 2007 - again a percentage suggesting some form of structural change. Even if one is completely unknowledgeable about the Austrian HE system, selecting the entity type in the browser tells us (F_5) that the changes have taken place in the sector of teacher education: the newly founded HE institutions are all of type “University of Education”, ‘University College of Teacher Education”, and “Pedagogical University”. Without further investigation, one already can conclude that the changes in the Austrian system are less broad than in the Netherlands or in Belgium, where the changes seem to cover a much larger part of the HE system.

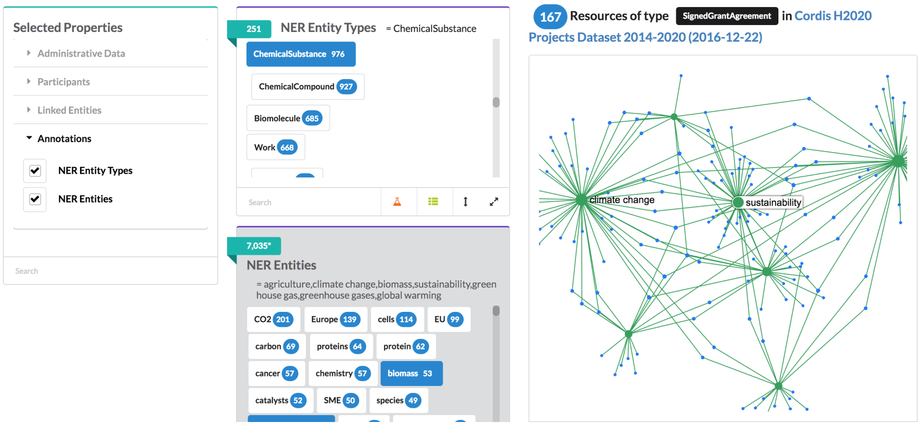


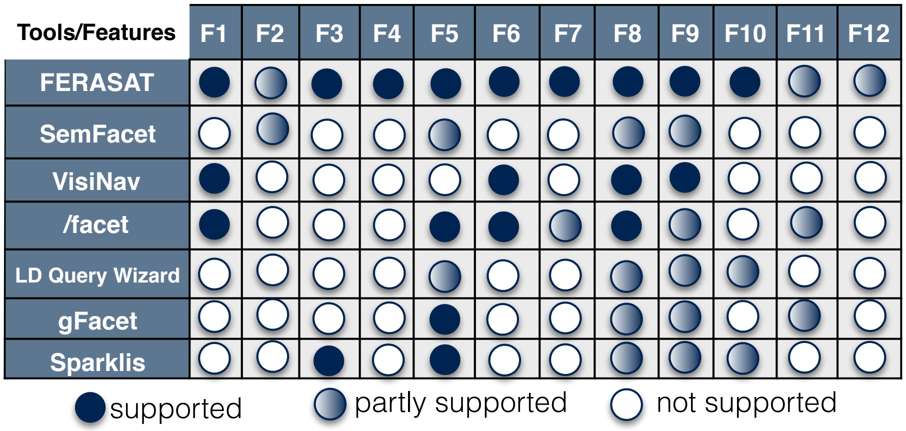
Fig. 6: Employing background knowledge to facilitate browsing of data.

4.2 Evaluating research portfolios with regards to current societal challenges

I used the faceted browser to browse CORDIS open dataset on H2020 EU projects to evaluate research portfolios. The browser showed the relevant characteristics of the projects, such as organizations involved, the organization type, and the program the project belongs to. The CORDIS dataset contains among others a text summarizing the content of the projects. Using the annotator tool helped me to extract general encyclopedic concepts from these textual descriptions and enabled me to browse data using two new facets (F_7), one for extracted terms and one for categories these terms belong to (see Figure 6). Combining extracted terms has a great advantage, as we can combine technical research terms and policy related terms to retrieve the relevant projects (F_1). This may solve the problem of finding how research links to the grand societal challenges. This is a core problem in assessing relevance of research (described in technical terms and policy related terms). Because the resulting set for a very specific topic is generally not too large, I could even manually inspect the policy-science link. As an example, I looked at chemical research in H2020 projects, related to one of the societal challenges. There are quite some water related topics in the H2020 projects. In total 22.5% of the water projects seem related to chemistry. Going a little deeper into this case shows the multidisciplinary character of the water related research in H2020, and what disciplines are more and what are less important in this portfolio (F_5, F_8, F_9).

5 Related Work

Path-finding on semantic graphs such as RDF graphs, where semantics of the relations between resources are explicitly defined, leads to discovering meaningful and insightful connections between multiple resources. That is the reason why



Tools/Features	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
FERASAT	●	◐	●	●	●	●	●	●	●	●	◐	◐
SemFacet	○	◐	○	○	◐	○	○	◐	◐	○	○	○
VisiNav	●	○	○	○	○	●	○	●	●	○	○	○
/facet	●	○	○	○	●	●	◐	●	◐	○	◐	○
LD Query Wizard	○	○	○	○	◐	○	○	◐	◐	◐	○	○
gFacet	○	○	○	○	●	○	○	◐	◐	○	◐	○
Sparklis	○	○	●	○	●	○	○	◐	◐	◐	○	○

● supported ◐ partly supported ○ not supported

Fig. 7: Comparing existing RDF facted browsers based on the proposed serendipity design features.

most of the current research work which investigates serendipity on Linked Data is focused on novel approaches for semantic traversal of RDF graphs and thereby serendipitous discovery of new related nodes.

Tools such as Everything_Is_Connected_Engine [2] and DBpedia RelFinder[11] allow serendipitous *storytelling* and *relation extraction* which benefit from path-finding on general knowledge graphs. In addition to that, domain-specific knowledge graphs enable experts to reveal *unsuspected connections* and/or *hidden analogies*. For instance, the Linked Data version of the TCGA (The Cancer Genome Atlas Database) [18] allows bio-medical experts to discover how cancer types tend to metastasize into other cancer types and to serendipitously explore linked data to see how the rheology of certain cancer types affects this metastasis. Furthermore, serendipitous recommendation realized by LOD paths-based techniques has been incorporated into the design of many personalized systems to minimize blind spots in information delivery. For example, in [14], a serendipity-powered TV recommender using BBC programs dataset is presented.

There are also several frameworks, tools and related works in the area of browsing (semi) structured data which do not claim explicitly for a contribution in terms of serendipity (cf. Figure 7):

In [16] a generic guideline is proposed for designing expressive exploration environments over semi-structured data which organizes the design space of exploration tools in a three-layer architecture: Data Access, Functional, and Interaction/Interface. Except for the adaptation engine, their proposed architecture fits very well with the technical architecture of the FERASAT.

SemFacet [1] is a faceted search tool enhanced by the Semantic Web technologies to allow browsing of interlinked documents. SemFacet is implemented on top of a fragment of Yago and DBpedia abstracts. Similar to the pivot change fea-

[11] `http://www.visualdataweb.org/relfinder.php`

ture in FERASAT, SemFacet allows refocusing results to support F_8. Although SemFacet exploits ontology-based reasoning for generation of facets and queries, no user-contextualization is supported. The main advantages of FERASAT over SemFacet are supports for customized interactive facet visualizations and enabling federated SPARQL queries over multiple knowledge graphs tailored based on the user context. *VisiNav* [5] is another linked data navigation system which combines features such as keyword search, object focus, path traversal and facet selection to browse web of data with a large variance. Although VisiNav provides some mechanisms to address the issue of *naked objects* (i.e. objects that are displayed without type-specific styling), it does not provide any personalized integrated view on distributed knowledge graphs. In our opinion, VisiNav acts more as a tool for traversing web of data rather than a direct knowledge discovery tool. *\facet* [7] is a linked data faceted browser very similar to FERASAT but with limited capabilities to share the generated queries, adapt the results based on user context and to invert and randomize the facets for increasing the chance encounter. \facet enables multi-type browsing experience and allows adapting the dynamically generated facets based on their RDF relations. It also allows users to create facet specifications and build facet dependent visualizations and interactions to make surprising observations more noticeable. *Linked Data Query Wizard* [8] is a linked data browsing UI, heavily dependent on RDF Data Cube standard, which turns graph-based data into a tabular interface with support for search and filtering to facilitate exploring linked data. Although converting graphs to interactive spreadsheet tables increases the learnability of UI for users, it also results in limited capabilities for serendipity by limiting the flexibility of information visualization related to certain dimensions of data. *gFacet* [6] is a graph-based faceted browser which allows users to build their facets of interest on the fly. It enable users to perform a pivot operation and switch a facet to a result list. Color coded facets and their relationships facilitate explaining the surprising observations. However, no mechanism for sharing the query results, inverting and randomizing values is offered. *Sparklis* [4] is a query-based faceted search UI that uses the expressivity of natural language to facilitate browsing Linked Data and understanding the generated query. It does not exploit any interactive visualizations in the facets to make surprising observations more noticeable. Also using only a single facet on a single knowledge graph, to browse data, makes the divergent information behavior difficult to achieve, though it increases the expressiveness and scalability of the traversed paths.

To the best of our knowledge, the related work in the domain of Linked Open Data where other aspects of serendipity than mere semantic path-finding are addressed, is quite scarce. The closest to our work is [3] which is based on SPARQL querying perspective where authors propose a query modification process to support serendipity features F_{11}: analogy, F_1: surprising observation, F_3: inversion, and F_4: disturbance. There are also similar querying-based approaches in the domain of data mining [17] to enhance the expressiveness of query languages to allow queries like "Find all the customers with deviant transactions." Or to offer meta-query languages ("Find me a pattern that connects something about writers" backgrounds and the characters in their novels). What distinguishes our approach from the above work is our more comprehensive investigation of serendipity design features and their implications on linked data faceted browsing environments for fostering serendipity on LOD.

6 Conclusions and Future Work

In this paper we presented a novel faceted browsing environment on Linked Data called FERASAT which aims to foster accidental knowledge discovery while browsing data scattered over multiple knowledge graphs. We reviewed the related work and showcased the applicability of FERASAT in the domain of science, technology and innovation by describing two relevant use cases.

As future work, we plan to implement more serendipity-fostering strategies within our faceted browser environment, in particular for detecting successful errors and performing cross-domain analogical reasoning. We also envisage to evaluate the usability of our implementation using a rigorous evaluation framework and also to extend its application to other domains such as life sciences.

7 Acknowledgement

We would like to thank our colleagues from the Knowledge Representation and Reasoning research group at Vrije Universiteit Amsterdam for their helpful comments during the development of our Linked Data-based faceted browser. This work was supported by a grant from the European Union's 7th Framework Programme provided for the project RISIS (GA no. 313082).

References

1. M. Arenas, B. C. Grau, E. Kharlamov, S. Marciuska, D. Zheleznyakov, and E. Jiménez-Ruiz. Semfacet: semantic faceted search over yago. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, pages 123–126, 2014.
2. L. De Vocht, S. Coppens, R. Verborgh, M. Vander Sande, E. Mannens, and R. Van de Walle. Discovering meaningful connections between resources in the web of data. In *LDOW*, 2013.
3. J. S. A. Eichler, M. A. Casanova, A. L. Furtado, L. A. P. P. L. Lívia Ruback, G. R. Lopes, B. P. Nunes, A. Raffaetà, and C. Renso. Searching linked data with a twist of serendipity. In *CAISE2017*, 2017.
4. S. Ferré. Expressive and scalable query-based faceted search over sparql endpoints. In *International Semantic Web Conference*, pages 438–453. Springer, 2014.
5. A. Harth. Visinav: A system for visual search and navigation on web data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):348–354, 2010.
6. P. Heim, T. Ertl, and J. Ziegler. Facet graphs: Complex semantic querying made easy. *The Semantic Web: Research and Applications*, pages 288–302, 2010.

7. M. Hildebrand, J. van Ossenbruggen, and L. Hardman. /facet: A browser for heterogeneous semantic web repositories. *The Semantic Web-ISWC 2006*, pages 272–285, 2006.
8. P. Hoefler, M. Granitzer, E. E. Veas, and C. Seifert. Linked data query wizard: A novel interface for accessing sparql endpoints. In *LDOW*, 2014.
9. D. Karger and M. Schraefel. The pathetic fallacy of rdf. SWUI, 2006.
10. A. Khalili and K. A. de Graaf. Linked data reactor: Towards data-aware user interfaces. In *SEMANTiCS 2017*. ACM, 2017.
11. A. Khalili, A. Loizou, and F. van Harmelen. Adaptive linked data-driven web components: Building flexible and reusable semantic web interfaces. In *ESWC2016*, pages 677–692, 2016.
12. A. Khalili and A. Meroño-Peñuela. WYSIWYQ - what you see is what you query. In V. Ivanova, P. Lambrix, S. Lohmann, and C. Pesquita, editors, *Proceedings of the Third International Workshop on Visualization and Interaction for Ontologies and Linked Data co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 22, 2017.*, volume 1947 of *CEUR Workshop Proceedings*, pages 123–130. CEUR-WS.org, 2017.
13. A. Khalili, P. van Andel, P. V. den Besselaar, and K. A. de Graaf. Fostering serendipitous knowledge discovery using an adaptive multigraph-based faceted browser. In Ó. Corcho, K. Janowicz, G. Rizzo, I. Tiddi, and D. Garijo, editors, *Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017*, pages 15:1–15:4. ACM, 2017.
14. V. Maccatrozzo, M. Terstall, L. Aroyo, and G. Schreiber. Sirup: Serendipity in recommendations via user perceptions. In *IUI2017*, pages 35–44. ACM, 2017.
15. D. A. Norman. *The Design of Everyday Things: Revised and Expanded Edition.* Basic Books, Inc., New York, NY, USA, 2013.
16. T. Nunes and D. Schwabe. Towards the design of expressive data exploration environments. In *Proceedings of the Third International Workshop on Visualization and Interaction for Ontologies and Linked Data co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 22, 2017.*, pages 60–74, 2017.
17. N. Ramakrishnan and A. Y. Grama. Data mining: From serendipity to science. *Computer*, 32(8):34–37, 1999.
18. M. Saleem, M. R. Kamdar, A. Iqbal, S. Sampath, H. F. Deus, and A.-C. Ngonga. Fostering serendipity through big linked data. *ISWC SemWeb Challenge*, 2013.
19. P. van Andel. Anatomy of the unsought finding. serendipity: Origin, history, domains, traditions, appearances, patterns and programmability. *Br J Philos Sci*, 45(2):631–648, June 1994.