# Facets and Pivoting for Flexible and Usable Linked Data Exploration

Josep Maria Brunetti, Rosa Gil, and Roberto García

Universitat de Lleida, Jaume II, 69. 25001 Lleida, Spain
{josepmbrunetti,rgil,rgarcia}@diei.udl.cat

**Abstract.** The success of Open Data initiatives has increased the amount of data available on the Web. Unfortunately, most of this data is only available in raw tabular form, what makes analysis and reuse quite difficult for non-experts. Linked Data principles allow for a more sophisticated approach by making explicit both the structure and semantics of the data. However, from the end-user viewpoint, they continue to be monolithic files completely opaque or difficult to explore by making tedious semantic queries. Our objective is to facilitate the user to grasp what kind of entities are in the dataset, how they are interrelated, which are their main properties and values, etc. Rhizomer is a tool for data publishing whose interface provides a set of components borrowed from Information Architecture (IA) that facilitate awareness of the dataset at hand. It automatically generates navigation menus and facets based on the kinds of things in the dataset and how they are described through metadata properties and values. Moreover, motivated by recent tests with end-users, it also provides the possibility to pivot among the faceted views created for each class of resources in the dataset.

**Keywords:** Semantic Web, Linked Data, Human-Computer Interaction, usability, interaction

## 1 Introduction

Though the amount of available following the Linked Data principles [2] is rapidly increasing, from the end-user perspective, the view continues to be that the available datasets are monolithic files completely opaque, or that can only be explored using complex semantic queries. The objective should be now to try to make all this data more usable so users facing a dataset can easily grasp

In: C. Unger, P. Cimiano, V. Lopez, E. Motta, P. Buitelaar, R. Cyganiak (eds.): Proceedings of Interacting with Linked Data (ILD 2012), Workshop co-located with the 9th Extended Semantic Web Conference, Heraklion, Greece, 28-05-2012, published at http://ceur-ws.org

what kind of entities are in there, how they are interrelated, what are their main properties and values, etc.

This way, users that are not experts in Semantic Web technologies can search, browse and analyse the data. This will increase the awareness of the data currently available in the Web and also facilitate the development of new and innovative applications on top of it. The overall outcome is that available data increases its impact and the society as a whole benefits more from data openness.

The best approach to make a dataset more usable to a wider range of users is to use some sort of Web publishing tool. At least, this kind of tools usually provides a HTML rendering for each resource in the dataset. Each HTML page lists all the properties for the corresponding resource. Pages are interlinked based on the connections among resources and the user can follow HTML links to browse through them. However, this feature is only useful if the user has some a priori knowledge about the dataset, especially the URI identifier of a given resource. There is no way to get at least an overview of the kind of resources in the dataset.

Our proposal is to draw from the experience accumulated in the Information Architecture (IA) domain [14] and reuse and adapt existing IA components to provide this kind of guide to users. These components are well known to Web users, as they are present in most web pages: navigation bars, facets, sitemaps, breadcrumbs, etc.

This approach is being applied in Rhizomer, a tool capable of publishing a Linked Data dataset while facilitating user awareness of its content. It is also being evaluated with end-users as part of a User Centred Design development process. Iterative evaluations have motivated and guided the introduction of new features, like pivoting, and validated improvements in the usability.

The rest of this paper is organised as follows. First, the related work is presented in Subsection 1.1. Then, the approach is introduced in Section 2, detailing the Information Architecture components used to implement it. In Section 3, an improvement is presented, motivated by a first round of evaluations with end-users. It is the possibility to use facets to pivot around the resource types in the dataset. An evaluation of this feature is also included in this section. Finally, conclusions and future plans are presented in Section 4.

### 1.1 Related Work

Publishing and presenting Linked Data in an accessible way for users has been addressed by several projects. The first tool that comes to mind when trying to realize what a dataset is about is a Semantic Web browser. Dadzie et al. [6] analyzed some of those projects finding that most of them are designed only for tech-users and they do not provide an overview of the data.

Browsers are especially useful when dealing with a dataset published as Linked Data because they provide a smooth browsing experience through the graph, e.g. Disco [4] or Tabulator [3]. However, they do not provide additional support for getting a broader view of the dataset being browsed, just a view on the current resource. With Explorator [1] it is possible to browse a dataset available through a semantic queries service. Though Explorator makes it possible to browse the dataset by combining search, facets or operations on sets of resources, it is still difficult to get a broader view on the dataset other than a list of all the classes or properties used.

In some cases it is also possible to get more informative components like facets, e.g. /facet [10]. However, in some cases, facets are pre-computed and just available for a given dataset as in the case of the DBPedia Faceted Browser [9].

Visor [15] is a generic data explorer that allows pivoting to related instances. Despite it provides a hierarchical overview of the dataset, it is complicated to filter resources because there isnt any faceted interface. Parallax [11] also provides the possibility to pivot to related types and a faceted browser, but it works only with the Freebase dataset.

Other alternatives are Content Management Systems (CMS) or Wikis with semantic capabilities. Some mainstream CMSs and wiki systems have started to incorporate semantic technologies. The most significant case is the last version of Drupal[1] that provides features such as semantic metadata storage, querying, importation or rendering. The same applies to semantic wikis, such as Semantic MediaWiki [12], that makes it possible to mix wiki mark-up with semantic annotations. However, semantic CMSs and wikis are intended more for content creation than for importing and exploring existing ones.

There are also specialised tools that publish existing datasets as Linked Data. For instance, Pubby[2] builds a Linked Data frontend with dereferenceable URIs for the subjects in the served dataset and content negotiation. It also features a metadata extension that provides provenance information. However, in both cases, the frontends provided are just like those Semantic Web browsers have.

To conclude, it is also possible to consider platforms for semantic data storage and publishing like OpenLink Virtuoso[3]. In addition to the data store, there is an HTML frontend that provide a basic faceted view for the datasets. However, its interface is not intuitive at all and like with previous tools, the support for broader awareness of the dataset structure is very limited.

Consequently, existing tools make it very difficult for non-technical users to explore a dataset, realize what kind of resources there are, what properties they have and how they are related.

## 2 Approach

Our starting point is the fundamental set of tasks for data analysis proposed by Shneiderman [17]. Next, we present each task associated to the interaction pattern chosen to satisfy it and Information Architecture [14] component selected to implement the pattern:

[1] Drupal 7, `http://drupal.org/drupal-7.0`

[2] Pubby, `http://www4.wiwiss.fu-berlin.de/pubby/`

[3] `http://virtuoso.openlinksw.com`

- **Overview:** get the full picture of the data set at hand. At this stage we propose to apply the Global Navigation interaction pattern[4]. In Information Architecture terms, it corresponds to the navigation bars users are used to see at the top or on the right of web sites.
- **Zoom & Filter:** zoom in on items of interest and filter out uninteresting items. Here the proposal is Faceted Navigation[5]. Once we have zoomed by selecting the kind of things we are interested in from the navigation bar, facets are the Information Architecture components that help users filter out those that are not interesting.
- **Details:** after zooming and filtering the user arrives to the concrete resources of interest. At this point, the user can get the details for those resources, which in the case of Linked Data is to get the properties for the resources plus those properties pointing to them. This is related to the Details on Demand[6] interaction pattern and can be implemented as a simple list of properties and values of the resource of interest or as a specific visualisation tailored to the kind of resource at hand, e.g. a map for geo-located resources.

Our proposal is to elaborate these interaction patterns in the context of Linked Data. We have chosen them because they are simple and very common so users are very confortable with them. They are currently part of the culture about how information is structured in the Web so they have been deeply studied in Information Architecture (IA) domain.

We are currently testing all these interaction patterns in a Linked Data publishing tool called Rhizomer[7]. It features navigation bars automatically generated and maintained starting from the underlying thesaurus and ontologies, and how they are structured and instantiated. Navigation menus are described in Section 2.1. A similar approach is followed for generating facets for each kind of entity in the data set. Facets are described in Section 2.2. Fig. 1 is a screen capture of Rhizomer showing at the top the navigation menu and on the left the generated facets.

### 2.1 Navigation Menus

Navigation menus, in the case of websites, let users navigate through different sections and pages of the site. They tend to be the only consistent navigation element, being present on every page of the site.

Traditionally, user-centred design techniques like Card Sorting [18] are used to develop the navigation menus of web sites. This technique requires a lot of time and effort from developers and most of this effort is wasted as soon as the structure of the menu is established and fixed in a menu that becomes something

[4] `http://www.welie.com/patterns/showPattern.php?patternID=main-navigation`

[5] `http://www.welie.com/patterns/showPattern.php?patternID=faceted-navigation`

[6] `http://www.welie.com/patterns/showPattern.php?patternID=details-on-demand`

[7] `http://rhizomik.net/rhizomer/`

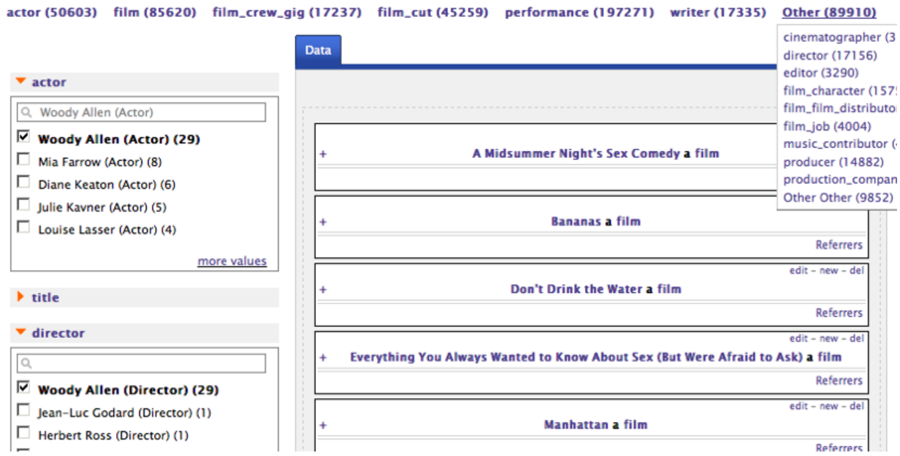

actor (50603) film (85620) film_crew_gig (17237) film_cut (45259) performance (197271) writer (17335) Other (89910)
cinematographer (3
director (17156)
editor (3290)
film_character (157
film_film_distributo
film_job (4004)
music_contributor (
producer (14882)
production_compan
Other Other (9852)
Data
actor
Woody Allen (Actor)
☑ Woody Allen (Actor) (29)
□ Mia Farrow (Actor) (8)
□ Diane Keaton (Actor) (6)
□ Julie Kavner (Actor) (5)
□ Louise Lasser (Actor) (4)
more values
title
director
☑ Woody Allen (Director) (29)
□ Jean-Luc Godard (Director) (1)
□ Herbert Ross (Director) (1)
+ A Midsummer Night's Sex Comedy a film
+ Bananas a film
Referrers
edit - new - del
+ Don't Drink the Water a film
Referrers
edit - new - del
+ Everything You Always Wanted to Know About Sex (But Were Afraid to Ask) a film
Referrers
edit - new - del
+ Manhattan a film
Referrers

**Fig. 1.** Screen capture of Rhizomer, at the top there is the navigation menu and on the left the generated facets.

static. If new kinds of items are introduced or a part of the content becomes more relevant, the Card Sorting should be repeated, at least in part.

The opportunity in the case of web sites build on top of semantic data is to automate part of the process of generation and maintenance of the navigation menus. This is possible because semantic data is structured by thesaurus and ontologies, which hierarchically organise the kinds of things described in the dataset. They specify all the classes or concept[8] but also which subjects belong to each class or are related to each concept.

Consequently, if there are few members of some kind, or there are not at all, it should be less relevant in the menu bar. On the contrary, those that do have a lot of members should be shown prominently in the menu bar. To do this, we obtain the hierarchy of all kinds of entities and apply inference rules to get their members. Then, the hierarchy is flattened to the amount of levels required because this component can generate both global and local menus, i.e. a menu for the whole dataset or for a subset of it. The site administrator can also configure some parameters: the number of levels in the menu, the number of items in each level, the order of items (alphabetical or by number of instances) and a list of classes or concepts to omit.

According to these parameters, this component generates the menu applying a recursive algorithm that mainly performs two operations: Split the concepts or classes with a large amount of members in their narrower related concepts or subclasses. Group those with few members into a broader concept or superclass.

This approach makes it possible to show the user the navigation bar that best fits the data in the dataset at that particular moment. For instance, if the dataset changes from containing mainly data about projects to mainly about

[8] SKOS Simple Knowledge Organization System, `http://www.w3.org/2004/02/skos/`

publications, the menu would change accordingly to show more prominently the part of the dataset structure about publications. More details about the implementation of navigation menus are available from [7].

### 2.2 Facets

Users dont always know exactly what they are looking for and, sometimes, they dont even know what its name is. Other times, they are unfamiliar with the domain or they want to learn about a topic. This is particularly true when facing Semantic Web datasets. In these cases, exploratory search is a strategy that allows users to refine their search by successive iterations. An exploratory interface such as faceted browsing allows users to find information without a priori knowledge of its schema.

With navigation menus we can make the user aware of the hierarchical structure of a dataset but, once they choose the class of things they are interested in, they face the barrier of not knowing how they are described. In other words, what are the main properties that describe them, which ones are the more relevant for that particular kind of things, the range of values they have in that particular case, etc. Faceted navigation is an exploratory technique for navigating a collection of elements in multiple ways, rather than a single and pre-determined order. Facet browser interfaces provide a user-friendly way to navigate through a wide range of data collections.

Traditional facet browsers relied on manual identification of the facets and on previous knowledge of the target domain. When dealing with semantic data, it is possible to automate this process and a semantic faceted browser should be able to handle any RDF dataset without any configuration. Since Linked Data facilitates integrating data from different of sources, we cant assume a single fixed schema for all data. Consequently, a Linked Data faceted browser should be scalable and generic, not depending on a particular dataset structure.

In traditional Web, facet browsers are developed to navigate through homogeneous data and facets are fixed. This conflicts with Linked Data, where data is too diverse to use a single set of facets: facets that make sense for one type of entity could be inappropriate for other types. Moreover, when new data is added the system should be able to add new facets at run time.

To build the facets, and to keep them updated, what Rhizomer does is to perform queries for each class in the dataset ontologies that retrieve all the properties their members have, the different values each property has and the cardinality for each value, i.e. how many times that property for that class takes that value.

Facets are calculated and stored in a data structure. They are then updated whenever the dataset is edited through Rhizomer. They are also updated, but just a local copy associated to a user session, when the user starts browsing and selecting values for different facets. In this case, the set of instances used for facets generation is constrained by the choices made so far and the facets are recalculated for that constrained set of results obtained so far. Those facets that are no longer relevant, i.e. no result uses them, are removed from the facets set. More details about the implementation of facets are available from [7].

## 3 Pivoting

From the point of view of OLAP systems, pivoting or rotation is described as an operation producing a change in the dimensional orientation of data. For instance, if data is initially aggregated by Product, Location and Date, by pivoting, the user can aggregate, for instance, by Location, Date and Product.

However, for richer data models pivot navigation is a way to restart a search from the results of a first search [16]. Usually, the type of resources to be browsed (e.g. book, car, paper...) remains fixed in a faceted browsing application. However, when pivoting is added to faceted navigation, it allows switching the type of displayed entities based on relations to the current result set. For instance, a user who is filtering films using film facets, e.g. director is Woody Allen, then pivots on actors. As a result of this action, the user will see now all actors in the result list, who are related to any film in the previous filtered list and continue filtering but now using actor facets, e.g. country is Spain.

Here, it is possible to establish an interesting analogy between pivoting and natural language. Indeed, the query above can be rephrased as Show actors whose country is Spain, which have acted in films directed by Woody Allen. The idea of pivot is reflected by the fact that the set of Spanish actors in the main sentence also appears in the relative sentence as the relative pronoun which. The relative pronoun point to the facet to browse for a pivot, in this case acted in.

Pivot steps can be repeated, e.g. pivot on countries from actors and filters continent is Europe, after removing the previous country is Spain filter from actors. Each pivot step corresponds to a nested relative sentences, such as Show European countries, where an actor, which has acted in a Woody Allen film, has been born. We have profited from this resemblance to natural language to generate more usable breadcrumbs that help users contextualise their exploration and know why they are getting the list of results that they are looking at as a result of their filtering and pivoting steps so far.

Next subsection illustrates the importance of offering pivoting to users exploring Linked Data, as shown in a preliminary evaluation of Rhizomer with end-users. The implementation details and the results of a new evaluation iteration are then presented in the following subsections.

### 3.1 Motivation

Pivoting is not a common feature of existing Linked Data exploration tools. To our knowledge, the only active tools capable of providing a faceted view and pivoting on semantic data are Parallax[9] and Virtuoso Faceted Browser[10]. Parallax

[9] `http://www.freebase.com/labs/parallax/`

[10] `http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtuosoFacetsWebService`

is constrained to the Freebase dataset though there is an attempt to make it capable to work on top of any SPARQL endpoint called SParallax[11]. However, in both cases, they lack a mechanism to provide an overview of the dataset and they have some performance issues, especially in the case of SParallax. Facets and pivoting in Virtuoso Faceted Browser are not so evident and quite difficult to use, though it also offers a breadcrumbs system based on the natural language analogy.

However, during the last round of user testing it became quite evident that a mechanism like pivoting was required. Evaluations are conduced at the UsabiliLAB[12] and inspired by the Rapid Iterative Testing and Evaluation (RITE) Method [13], testing with small groups as part of an iterative development process. To register sessions we used Morae Recorder and Morae Observer to analyse user test data. For the usability test metrics we chose effectiveness (percentage of tasks completed, workload) and efficiency (time to complete a task).

For testing, we used a real dataset called the Linked Movie Data Base (LinkedMDB)[13]. We chose this dataset because movies is a well know topic for most people and quite appealing. LinkedMDB is generated from the Internet Movie Database[14] (IMDB).

Six participants were selected, with a unique profile characterized by good knowledge of information technology, limited knowledge about Semantic Web technologies and interest in movies. The reason we chose this profile is because Rhizomer is intended for users interested in working with data, used to information technologies but not necessarily aware of Semantic Web and Linked Data.

The test facilitator proposed users, among others, the following task: Find three films where Woody Allen is director and actor at the same time. The full findings derived from the test are available from [5], while the ones relevant to motivating the inclusion of the pivot operation are summarised next:

- **100%** of participants needed in at least one occasion the guidance of the facilitator to successfully complete the task.
- **100%** of the participants began the navigation from actors instead than from movies (the directors where not so evident in the navigation bar because there are not so many instances than for other classes and they appear as a second level option in the menus).

When analysing the evaluation results, it became evident that the fact users started from actors was the reason why they required assistance, they arrived at a dead-end after filtering actors by name to just Woody Allen. There was no way to switch to his set of films and then filter it using the director facet.

A short path to this problem might be to add for each resource, e.g. Woody Allen (Actor), a link for each facet to the set of resources that can be reached through it, e.g. a link to all the films where Woody Allen has acted. However,

[11] `http://sparallax.deri.ie`

[12] UsabiliLAB, `http://griho.udl.cat/en/infraestructures/usabililab.html`

[13] LinkedMDB by O. Hassanzadeh and M. Consens, `http://linkedmdb.org`

[14] IMDB database, `http://www.imdb.com`

this just works for particular instances and the objective is to make it also work for sets of resources, e.g. all the films by a Spanish actor. This motivated the development of a pivot operation that can be performed even before filtering has reduced the set of explored resources to just one.

Moreover, another conclusion from the evaluation was that navigation should be better contextualised. The interface should provide more mechanisms to inform the user where she/he is, where she/he can go and where she/he has been. For that, the proposal is to integrate some kind of breadcrumbs that summarise the navigation steps though navigation menus and facets.

The previous findings motivated a next round of work with Rhizomer. The focus was now on incorporating the pivot action and also on improving contextualisation by including breadcrumbs, as detailed in the next section.

### 3.2 Implementation

To build facets that support pivoting, we first distinguish three types of properties:

- **Datatype properties:** properties whose values are RDF literals or data types from XML Schema. It is not possible to pivot on these properties but recognising them allows displaying them with specialised facet types, e.g. a histogram facet for numbers or a calendar one for dates. These specialised facets are still work in progress.
- **Object properties:** properties whose values reference other resources. These properties were treated, prior to the introduction of pivoting, as facets with literal values, where the values were resources labels. It continues to be possible to filter a set of resources based on the labels of the referenced resources, e.g. filter films through the actor facet based on the actors labels. However, pivots makes also possible to switch to the set of actors and perform a more detailed filtering based on actors facets.
- **Inverse properties:** in some cases, a dataset has a property between resource types modelled just in one way. For instance, each resource of type Film has the property actor, but the resources of type Actor do not have the inverse property to relate them with the films they have appeared in. When inverse properties are not explicit in a dataset, they are detected and facets are generated following the same approach than for explicit object properties. Consequently, it is possible to pivot through explicit object properties and also through implicit inverse object properties. This increases the flexibility of the exploration as more choices are available to the user dead-ends are avoided, like exploring actors and not being able to pivot to films because the property from actors to films is not explicitly stated in the dataset.

Each facet consists of a list of the five most used values and a text search box, which suggests possible matches. There is also the possibility to see the rest of values and choose from them. In the case of object properties and inverse properties we provide the option to pivot to their related types, marked as an arrow icon as shown in Fig. 2. Thus, users can construct complex queries that include different types of resources.

We keep a list of the selected values for each facet and also information for every pivoted step realized. This information is used to build the corresponding SPARQL queries to get results. For example, the SPARQL query when browsing films whose director is Woody Allen is:

```
SELECT DISTINCT ?r1 WHERE {
 ?r1 a <http://data.linkedmdb.org/resource/movie/film> .
 ?r1 <http://data.linkedmdb.org/resource/movie/director>
     <http://data.linkedmdb.org/resource/director/8501> }
```

For those facets that allow pivoting to another type we save the following information: the initial type, the property used to pivot and the target type. In the previous example, when user pivots from Film to Actor:

- Initial type: `http://data.linkedmdb.org/resource/movie/film`
- Pivoting property: `http://data.linkedmdb.org/resource/movie/actor`
- Target type: `http://data.linkedmdb.org/resource/movie/actor`

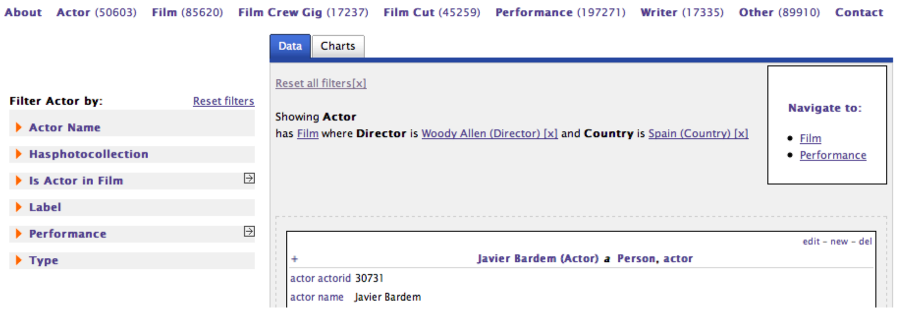
When pivoting to another type, the restrictions applied to the first one are propagated to the pivoted type through the property used for pivoting and linked using variables, marked in bold in the following SPARQL example after pivoting from films to actors:

```
SELECT DISTINCT ?r2 WHERE {
 ?r2 a <http://data.linkedmdb.org/resource/movie/actor>.
 ?r1 a <http://data.linkedmdb.org/resource/movie/film>.
 ?r1 <http://data.linkedmdb.org/resource/movie/director>
     <http://data.linkedmdb.org/resource/director/8501>.
 ?r1 <http://data.linkedmdb.org/resource/movie/actor> ?r2.
 ?r1 a <http://data.linkedmdb.org/resource/movie/film> }
```

To generate breadcrumbs we use also the restrictions used to build queries. Breadcrumbs show the path that the user has followed to arrive to that set of results. Users can use them to remove filters and pivoted steps. We have used natural language to generate them because we think this is the easiest way to understand the filtering and pivoting steps, as already introduced in Section 3. Fig. 2 shows, in the central part, the breadcrumbs generated for this example.

### 3.3 Evaluation

This section summarises the results of the first round of user testing after integrating the pivoting functionality into Rhizomer. The evaluation approach is inspired by the Rapid Iterative Testing and Evaluation (RITE) Method [13], testing with small groups as part of an iterative development process, but tests are performed much more frequently.



Showing **Actor**
has Film where **Director** is Woody Allen (Director) [x] and **Country** is Spain (Country) [x]

**Javier Bardem (Actor) a Person, actor**

actor actorid 30731
actor name Javier Bardem

**Fig. 2.** Capture showing the pivoting enhacements: breadcrumbs as natural language in the middle, pivot-able facets with arrow icon and Navigate to box with pivoting options

The aim of the test presented in this section was mainly to validate that the introduction of pivoting solves the problems highlighted in the previous evaluation, described in Section 3.1. Six users that where not involved in the previous evaluation rounds were recruited and one of the tasks, Task 2, was identical to one of the tasks used in previous rounds.

**Table 1.** Tasks posed to users during the pivoting testing round

| | |
|---|---|
| Task 1 | Find 5 films with "Orlando Bloom" as actor. |
| Task 2 | Find 5 films with "Clint Eastwood" both as director and actor. |
| Task 3 | Who has directed more films in countries located in "Oceania". |

This is the task that is going to be used to test if pivoting has improved efficiency and efficacy. The complete list of tasks is presented in Table 1 and the efficiency results are shown in Table 2. More details about previous evaluations are available from [8].

The first finding has been that the introduction of pivoting has corresponded to a great increase of efficiency, with a 30% reduction in the mean time necessary to complete Task 2. However, the most promising outcome is that the biggest improvement has been in the reduction of the maximum time on tasks, 57% improvement.

This is related with the fact that, thanks to pivoting, all users where able to find their path to solve the task without requiring assistance. Contrary to pre-pivoting tests, where most users got lost when trying to complete the tasks starting from actor or director instead of from film, with pivoting all users were able to complete the task independently of their starting point without assistance. Consequently, the maximum time was reduced significantly.

**Table 2.** Efficiency for the tasks in Table 1 plus comparison with pre-pivoting results for Task 2

| Time on Task (minutes) | **Task 2 with pivoting** | **Task 2 pre-pivoting** | **Improvement** | Task 1 with pivoting | Task 3 with pivoting |
|---|---|---|---|---|---|
| Minimum | **0.89** | **1.05** | **15%** | 1.00 | 1.99 |
| Maximum | **2.23** | **5.23** | **57%** | 4.53 | 4.50 |
| Mean | **1.69** | **2.41** | **30%** | 1.61 | 3.43 |
| Standard Dev. | **0.57** | **1.49** | **62%** | 1.21 | 0.96 |

However, there is still a lot of room for improvement. The following issues were detected and some proposals to solve them are presented in the conclusions and future work section:

- It was users for users to identify the pivoting button associated to object property of inverse property facets. Moreover, the box labelled Navigate to, that also contained the list of facets that provided pivoting was far from the facets, in the right side while facets are in the left side, and hard to identify. Finally, some users thought that following one of this pivoting links supposed starting from zero the exploration from the target class, loosing all the restrictions done so far through facets.
- Users also experienced many contextualisation problems, not being clear for them what were they seeing at the screen. The breadcrumbs helped solving this once the users realised they were available but this took some time for most of them.

## 4 Conclusions and Future Work

After some rounds of development and testing with end-users, Rhizomer is capable of publishing Linked Data while facilitating user awareness of what is in the dataset. Awareness is accomplished by components borrowed from the Information Architecture discipline and generated automatically from the dataset structure and ontologies. They are navigation bars, which show the main kinds of items in the dataset, and facets, which show the more significant properties for different kinds of items and their values.

Our preliminary tests with users show that Rhizomer facilitates the exploration of the published datasets, and they have also highlighted many issues. The last functionality addition, motivated during a previous evaluation round, is pivoting. The possibility of allow users move from faceted exploration of a particular class of resource, e.g. Actors, to another class of related resources, e.g. Film, while preserving the previous filters, provides a great level of flexibility to the interaction and avoid dead-ends due to the way the data is structured.

Pivoting has allowed reducing the mean time to complete a particular task, thus improving efficiency, by a 30%. Moreover, the maximum time has been reduced by almost a 60% as a result of the fact that now, contrary to the test prior to the introduction of pivoting, all users were able of completing the task without assistance.

The remaining issues, spotted during the last evaluation, are mainly related with the fact that the interface components providing pivoting are not so evident for users. Moreover, they suppose a conceptual shift that should be mitigated. For instance, some users saw pivoting as restarting exploration for a new class of resources.

One possible way to overcome these limitations of pivoting is to try to integrate it with facets, so users do not need to take their attention from facets, and also to make it clearer that the filtering done so far is not going to be lost. One way to do that, to be implemented and tested, is to present pivoting as a way of doing advanced filtering on a facet.

This way, users can start doing classical filtering using the labels of the facet values. For instance, filter the actors for a film using the actor facet and the actor labels. However, if they get stuck because they need a more sophisticated filter, make pivoting available as a way of attaining advanced filtering. This got a promising way to go as a result of Task 3 used in the evaluation presented in Section 3.3. Users looked for films from countries in Oceania but got stuck because the country facet showed just country names. Some of them pointed out that they needed to filter the countries by continent but did not see pivoting as the solution to their problem. They idea is that we can link their need of performing advanced filtering on properties of the entities associated to a facet to pivoting.

Finally, another issue is related with contextualisation. Though natural language-inspired breadcrumbs have been seen by users as very useful, they should get more prominent in the user interface because it took too much time to users to spot them. The idea in this case is to increase breadcrumbs size and use colours that highlight the breadcrumbs and also the entities involved, i.e. the names of the classes and facets involved.

## References

1. Araujo S., Schwabe D., Barbosa S.: Experimenting with Explorator: a Direct Manipulation Generic RDF Browser and Querying Tool. In: Proc. of VISSW 2009
2. Berners-Lee, T.: Linked Data. Design Issues (2009) `http://www.w3.org/DesignIssues/LinkedData.html`
3. Berners-Lee, T. Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., Sheets, D. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In: Proc. of SWUI 06 at ISWC 2006
4. Bojars, U., Passant, A., Giasson, F., Breslin, J. G.: An Architecture to Discover and Query Decentralized RDF Data. In: Proc. of SFSW 2007, CEUR Workshop Proceedings 248 (2007)
5. Brunetti, J.M., García, R.: Information Architecture Automatization for the Semantic Web. In: Proc. of the 13th IFIP TC 13 International Conference on Human-Computer Interaction (2011) 410–413

6. Dadzie, A.-S., Rowe, M.: Approaches to Visualising Linked Data: A Survey. The Semantic Web Journal – Special Call for Semantic Web surveys and applications **2**(2) (2011) 89–124
7. García, R., Brunetti, J.M., López-Muzás, A., Gimeno, J.M., Gil, R.: Publishing and interacting with linked data. In: Proc. of the International Conference on Web Intelligence, Mining and Semantics (2011) 18:1–18:12
8. García, R., Gimeno, J.M., Perdrix, F., Gil, R., Oliva, M., López, J.M., Pascual, A., Sendín, M.: Building a Usable and Accessible Semantic Web Interaction Platform. In: Proc. of WWW 2010 143–167
9. Hahn, R., Bizer, B., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., Scheel, U.: Faceted Wikipedia Search. In: Proc. of BIS10
10. Hildebrand, M., Ossenbruggen, J., and Hardman, L.: /facet: A Browser for Heterogeneous Semantic Web Repositories. In: Proc. of The Semantic Web at ISWC 2006, 272–285
11. Huynh, D. F., Karger, D. R.: Parallax and companion: Set-based browsing for the data web. In: WProc. of WWW 2009
12. Krötzsch, M., Vrandečić, D., Völkel, M.: Semantic MediaWiki. In: Proc. of ISWC 2006, LNCS 4273 (2006) 935–942
13. Medlock, M.C., Wixon, D., Terrano, M., Romero, R.L., Fulton, B.: Using the RITE method to improve products: A definition and a case study. In: Proc. of the Usability Professionals Association (2002)
14. Morville, P., Rosenfeld, L: Information Architecture for the World Wide Web. O'Reilly Media (2006)
15. Popov, I., schraefel, m., Hall, W., Shadbolt, N.: Connecting the Dots: A Multi-pivot Approach to Data Exploration. In: Proc. of ISWC 2011, 23–27
16. Sacco, G.M., Tzitzikas, Y. (eds.): Dynamic taxonomies and faceted search: theory, practice, and experience. Springer (2009)
17. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: Proc. of the 1996 IEEE Symposium on Visual Languages (VL'96) 336–343
18. Spencer, D.: Card Sorting. Rosenfeld Media (2009)