Experimenting with Explorator: a Direct Manipulation Generic RDF Browser and Querying Tool

Samur F. C. de Araújo **Daniel Schwabe** **Simone D.J. Barbosa**
Informatics Department, PUC-Rio
Rua Marques de Sao Vicente, 225
{saraujo, dschwabe, simone}@inf.puc-rio.br
+ 55 21 3527-1510

ABSTRACT

In this paper we present a preliminary study with Explorator, a tool for exploring RDF data by direct manipulation. Explorator's visual user interface allows users to explore a semi-structured RDF database to both gain knowledge and answer specific questions about a domain, through browsing, search, and exploration mechanisms.

Author Keywords

Exploratory search, semantic browsing, user interface for semantic data exploration, semantic web.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

As the volume of information on the Web increases considerably, we need better tools to help us discover and make sense of the available information, as well as to seek answers to specific questions we may have.

This paper presents a preliminary study with Explorator [5], a direct manipulation tool we have developed to support the exploration of semi-structured RDF databases. Our goal is to support the users in discovering and understanding a domain, as well as in answering specific questions about the domain through browsing, search, and exploration. The work reported here extends the results described in [1] by describing additional experiments and corresponding lessons learned. In particular, comparisons between Explorator and its model with other RDF browsers is presented in that reference.

In the next section, we argue for the importance of supporting exploratory search. The third section describes Explorator's processing model. In the fourth section, we describe Explorator's direct manipulation user interface, following the interaction paradigm we deemed more adequate for the kinds of manipulation we support. The fifth section describes the user testing studies we conducted, and the final section concludes the paper with a summary of the findings and directions for future work.

EXPLORATORY SEARCH

In the hypertext field, search, navigation and browsing are terms that describe distinct processes of information retrieval. Carmel et al. [2] did an extensive study about the cognitive process of browsing and searching, and based on it we will draw the following distinctions.

- Search is the process of seeking a specific known piece of information.
- Browsing is the process of investigating a vast collection of information items in a superficial and not oriented way.
- Navigation is the oriented process to access, view or select a number of information items.

We call *information exploration* the process of seeking, learning about, and investigating a (potentially large) collection of information items through search, browsing or navigation, but not excluding other forms, in order to discover something new.

The research area called exploratory search [9] has tried to develop solutions that support information exploration. Exploratory search is applicable in situations where the user's task and the search environment have complex elements that require constant user interpretation during the exploration process. For example, how to support the user's search task when she is not familiar with the search domain, or she does not have sufficient knowledge about domain to make a query; how to support the navigation in vast information spaces, or when the

VISSW 2009, February 8–, 2009, Sanibel Island, FL, USA.

navigation, searching and browsing are not enough. Marchionini [9] made a distinction between exploratory search, lookup and search retrieval. According to him, exploratory search is based not only on lookup but also in investigation and learning. He argues that investigative search and learning search require more human iteration than a simple lookup, because these are exploratory processes that support tasks that require the cognitive and interpretative ability of user. These kinds of tasks are commonly found in the exploration of RDF databases, where the users need to identify classes and properties from the schema, in order to understand concepts, acquire knowledge and learn about the domain. In order to provide the user with an exploratory search tool that supports learning and investigative search on the semantic web, we focused on three inter-related aspects:

- Information search (how semantic data is found),
- Information manipulation (how semantic data is used),
- Information visualization (how semantic data is presented).

Understanding Semantic Data

The typical challenge when accessing an RDF repository is how do users make sense of the available data? At what level of abstraction do they think of that information? Research in Cognitive Science has shown that people's bodily experience and the way we use imaginative mechanisms are central to how we construct categories to make sense of experience [7]. Eleanor Rosch (apud Lakoff [7]) proposed that thought, in general, is organized in terms of prototypes and basic-level categories.

We follow on their footsteps and hypothesize that people, when exploring an information space in the semantic web, focus not on sentences that describe the properties of the entities in the database, but on the entities that play the roles of subjects and objects in those sentences, especially entities that would be considered members of the basic-level categories implicit in the database schema. As such, our user interface privileges the visualization and manipulation of such entities, as will be seen in the fifth section. In other words, entities would be equivalent to resources in RDF that denote "things" that people conceptualize in order to solve tasks.

An important caveat of our work at this point in our research is that we are first focusing on people who have some knowledge of the RDF data structure, and investigating whether they are able to explore the semantic space by means of the kinds of queries and operations allowed by the proposed model describe next. With positive results at this step, we shall then proceed to provide a more adequate user interface for those unfamiliar to RDF as well.

EXPLORATOR'S PROCESSING MODEL

Our experience in Web application design methods [8, 11] has shown us that it is useful to characterize the user information processing as set manipulation operations, in what has been called "set-based navigation" [8]. This view is also supported by more recent working tools such as Parallax[1]. Basically, the user is processing (browsing) information items within a set of interest; if necessary, this set is further manipulated to either remove uninteresting elements or to add additional elements of interest to the set.

We will show in the following subsections that this model can encompass classical browsing, set-based navigation as found in SHDM [8], and faceted browsing [10], as well as keyword search. The model has been more extensively described in an accompanying paper [1], and is only briefly presented here to facilitate the understanding of the studies we have conducted.

Sets

The model manipulates two kinds of sets: sets of RDF triples and sets of RDF resources. For sets of RDF resources, the usual set operations —union, intersection and difference— are available. Since RDF resources are treated as URIs, blank nodes will only be included if they are assigned to URIs, as occurs for some data stores.

When operating on sets of triples, we interpret the set operations as applying to any of the triple components, namely, subjects (S), predicates (P) or objects (O). This is equivalent to projecting a set of triples along one of its three positions, as illustrated in Figure 1. In the remainder of the paper, each position will also be called a *role* in a triple.

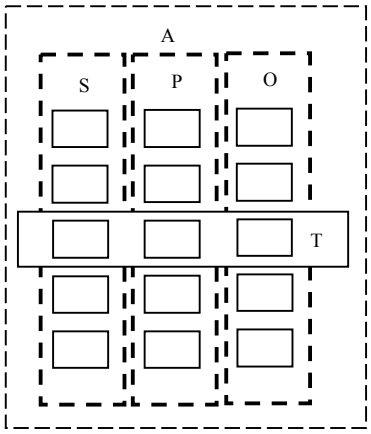


Figure 1. Triple (T), sets of resources (S, P, and O), and set of triples (A).

A triple is denoted by (s,p,o), where s, p, and o are resources. Let A be a set of triples. The set R of resources of A can be given as:

[1] http://mqlx.com/~david/parallax/index.html

$R = S \cup P \cup O$: $\forall s,p,o\ (s,p,o) \in A$ and $s \in S$ and $p \in P$ and $o \in O$.

Given the triple set A, we also have the following functions:

$S = R_s(A) = \{x \in S \mid \exists\ p,o:(x,p,o) \in A \text{ and } p,o \in R\}$
$P = R_p(A) = \{x \in P \mid \exists\ s,o:(s,x,o) \in A \text{ and } s,o \in R\}$
$O = R_o(A) = \{x \in O \mid \exists\ s,p:(s,p,x) \in A \text{ and } s,p \in R\}$

Where S is the set of all subjects, P is the set of all the predicates and O is the set of all objects in the triples of A.

Semantic Operations

Given a set of triples A, a set of resources R, and subsets S, P, and O of R ($S \subseteq R$, $P \subseteq R$, $O \subseteq R$), we can define the SPO function as follows:

- the set of all triples in A:

  $SPO(\varnothing,\varnothing,\varnothing) = \{(s,p,o) \in A \mid s,p,o \in R\}$

- the set of only the triples in A whose subject is in S:

  $SPO(S,\varnothing,\varnothing) = \{(s,p,o) \in A \mid s \in S \text{ and } p,o \in R\}$

- the set of only the triples in A whose predicate is in P:

  $SPO(\varnothing,P,\varnothing) = \{(s,p,o) \in A \mid p \in P \text{ and } s,o \in R\}$

- the set of only the triples in A whose object is in P:

  $SPO(\varnothing,\varnothing,O) = \{(s,p,o) \in A \mid s,p \in R \text{ and } o \in O\}$

- the set of only the triples in A whose subject is in S and predicate is in P:

  $SPO(S,P,\varnothing) = \{(s,p,o) \in A \mid s \in S \text{ and } p \in P \text{ and } o \in R\}$

- the set of only the triples in A whose subject is in S and object is in O:

  $SPO(S,\varnothing,O) = \{(s,p,o) \in A \mid s \in S \text{ and } p \in R \text{ and } o \in O\}$

- the set of only the triples in A whose predicate is in P and object is in O:

  $SPO(\varnothing,P,O) = \{(s,p,o) \in A \mid s \in R \text{ and } p \in P \text{ and } o \in O\}$

- the set of only the triples in A whose subject is in S, predicate is in P, and object is in set O:

  $SPO(S,P,O) = \{(s,p,o) \in A \mid s \in S \text{ and } p \in P \text{ and } o \in O\}$

The function $SPO(\varnothing,\varnothing,\varnothing)$ can be translated into the following SPARQL query:

```
SELECT ?s ?p ?o WHERE  {?s ?p ?o} .
```

For the following data:

```
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
_:a  foaf:name   "Johnny Lee Outlaw" .
_:a  foaf:mbox   <mailto:jlow@example.com> .
_:b  foaf:name   "Peter Goodguy" .
_:b  foaf:mbox   <mailto:peter@example.org> .
_:c  foaf:mbox   <mailto:carol@example.org> .
```

The query above should return all triples. On the other hand, the function $SPO(\varnothing,\{foaf:mbox\},\varnothing)$ can be translated into:

```
SELECT ?s ?p ?o WHERE  { ?s ? p ?o. Filter (p = foaf:mbox)} .
```

And this query returns all triples that have the property foaf:mbox.

It is important to note that, although in SPARQL we cannot pass arrays of resources to a query, our SPO function works with either single resources or sets of resources.

Set Operations

The model supports the following set operations:

Let $V = \{s,p,o\}$; $v, v' \in V$

Let $U_R = \{x \in U_R \mid x \in R_v(M) \text{ or } x \in R_{v'}(N)\}$
$U = (M,v) \cup (N,v') \equiv SPO(U_R,\varnothing,\varnothing)$

Let $I_R = \{x \in I_R \mid x \in R_v(M) \text{ and } x \in R_{v'}(N)\}$
$I = (M,v) \cap (N,v') \equiv SPO(I_R,\varnothing,\varnothing)$

Let $D_R = \{x \in D_R \mid x \in R_v(M) \text{ and } x \notin R_{v'}(N)\}$
$D = (M,v) - (N,v') \equiv SPO(D_R,\varnothing,\varnothing)$

The union, intersection and difference operations are calculated over sets of resources playing a certain role (v or v') in a triple. For instance, $(M,o) = R_o(M)$, i.e., represents all resources that play the role of object in the M set of triples. The operation is calculated over these sets and then resulting on the triples where the resulting set plays the role of the subject.

A simple example of how this model could be used to solve the task "find all Russian lakes" is as follows:

```
SPO(R(SPO(∅,∅,{mondial:Lake}),s),
      ∅,
      R(SPO(∅, ∅,{'Russia'}),s))
```

or

```
SPO(R(SPO(∅,∅, { mondial:Lake}),s), ∅, {mondial:Russia})
```

The following section presents Explorator's direct manipulation interface and shows how it keeps the users in control of their searching, browsing, navigating, and overall exploration of the RDF database.

EXPLORATOR'S DIRECT MANIPULATION USER INTERFACE

Direct manipulation is a user-system interaction paradigm that allows users to point at visual representations of objects and actions to carry out tasks rapidly and observe the results immediately [13]. The direct manipulation paradigm mainly consists of:

- visual presentation of the world of action: show users the available objects and actions;
- rapid, incremental, and reversible actions;

- selection by pointing, not typing; and
- continuous visual display of status.

In argument for direct manipulation, Shneiderman[13] states that first time users "are struggling to understand what they see on the display while keeping in mind their information needs. They would be distracted if they had to learn complex query languages or elaborate shape-coding rules" [13:511].

Shneiderman lists the following high-level tasks for open-ended browsing of known collections and exploration of the availability of information on a topic:

- specific fact finding (known-item search), e.g, Find the country named Russia;
- extended fact finding, e.g., What are the neighboring countries of Russia?
- open-ended browsing, e.g., Is there information about the past presidents of each country?
- exploration of availability, e.g., What geographic information is available for Brazil?

Empirical studies show that users perform better and have higher subjective satisfaction when they can view and control the search [9]. This was one of Explorator's main goals: to put users in control of their queries, and provide immediate feedback to their actions. Figure 2 illustrates the Explorator user interface:

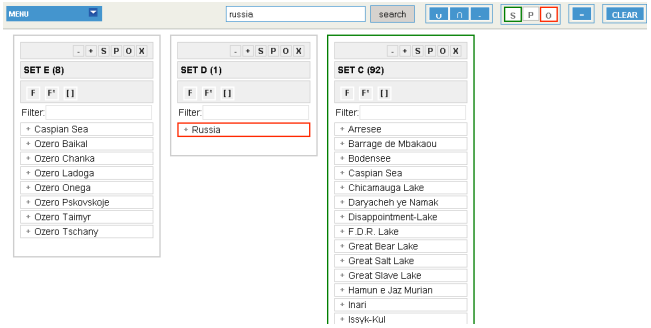


Figure 2. Snapshot of Explorator's interface.

To empower users in their exploration tasks, Explorator supports the following operations at the user interface[2]:

- searching for all resources containing a given string (using the search box in the toolbar);
- selecting a resource (e.g. Russia), by clicking on it;
- detailing a resource, by double-clicking on it to reveal all its properties, by showing all the triples where the resource is the subject;
- selecting multiple resources, by ctrl+clicking on them;
- selecting a binary operation over two sets of resources — union, intersection, and difference—, by clicking on the corresponding toolbar button;
- assigning a role —S, P or O— to a set of resources in an SPO query, by clicking on the corresponding toolbar button;
- calculating the operation result, by clicking on the [=] toolbar button; and
- changing the visualization of a set of resources, e.g. grouping them by one of the roles (S, P, O), expanding or collapsing all the triples in the set, and so on. These changes in visualization are made by clicking on toolbar buttons on the corresponding set pane.

Whereas the actual result of any of the above operations is a set of triples, the visual presentation is a set of resources. This is achieved by grouping these triples by one of the roles (S, P, O), and hiding the other triple elements until the user expands the corresponding interface widget (Figure 3).

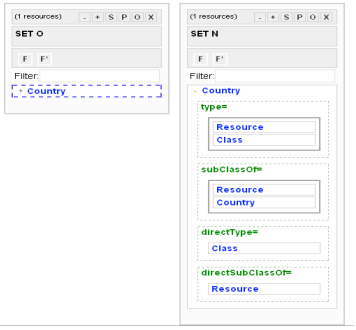


Figure 3. Two views of the resource *Country*: collapsed on the left, expanded on the right.

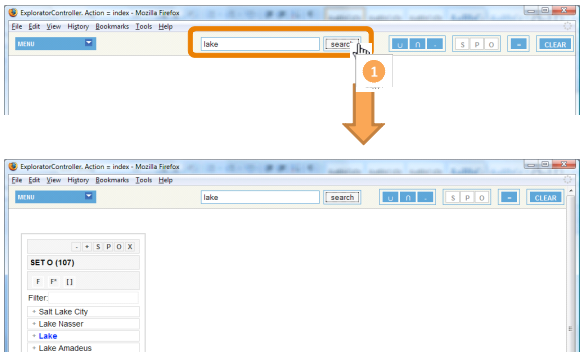
Sample Scenario

Let us now illustrate the usage of Explorator. Suppose a geographer called David needs to find all the lakes contained exclusively in Russia (and not in any other country). There are several possible ways to achieve this task; on possible way would be as follows:

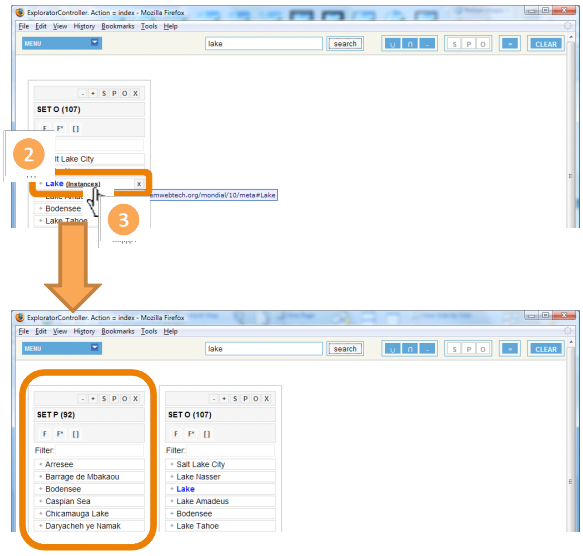
1. Find all the lakes in the database;
2. Find Russia, the country;
3. Find all the lakes in Russia obtaining a set we will call LR;
4. Find the countries that share a boundary with Russia (Russia's neighbors);
5. Find all the lakes in Russia's neighbors, obtaining a set we will call LN; and
6. Build the set of the lakes contained exclusively in Russia by calculating the difference between the previous sets: LR-LN

[2] Additional operations are supported, such as faceted navigation, among others. We present here only the operations that are relevant to the described studies.

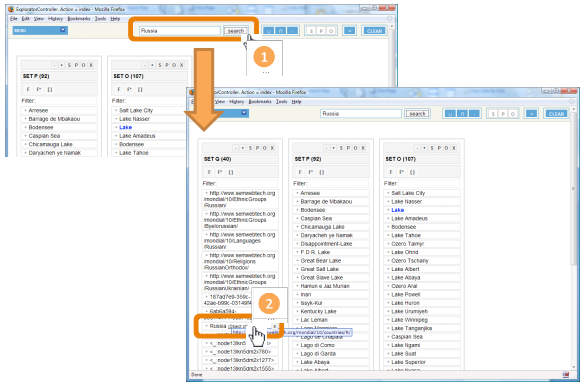
To find all the lakes in the database, David first searches for “lake”:



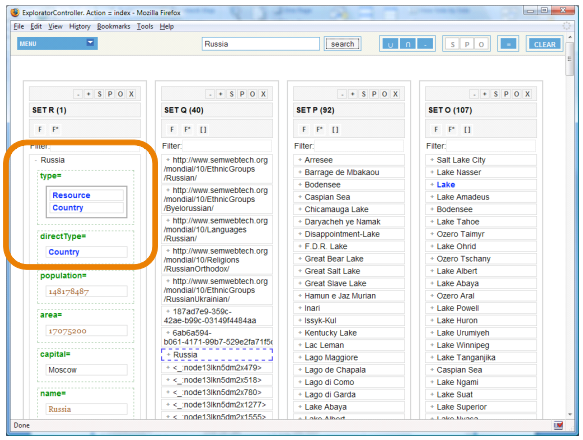
He locates the Lake class in the resulting set, and gets the set of instances of the Lake class by clicking on the Instances link, to obtain all the lakes in the database:



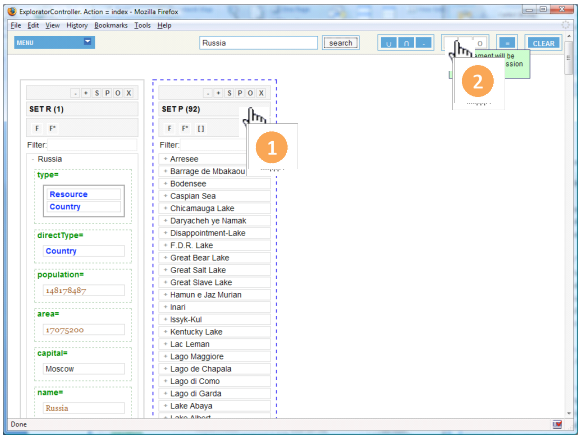
Next, to find Russia, he searches for “Russia” and locates the resource Russia in the resulting set:



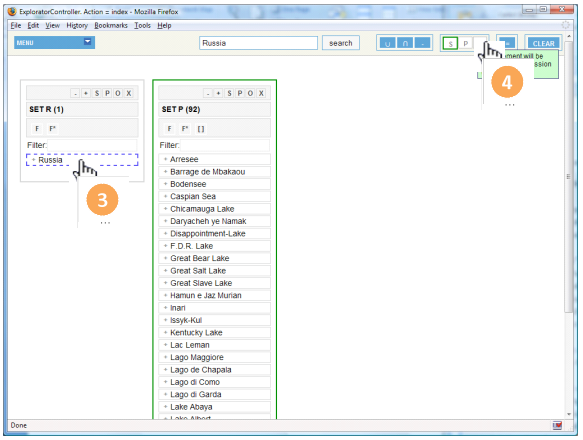
To make sure he has the right resource, David views the resource details:



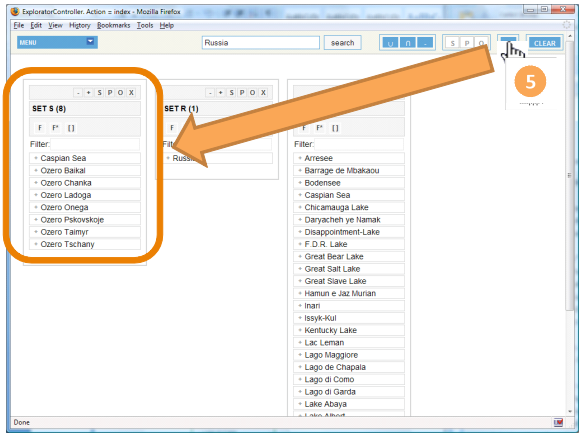
Next, to find all the lakes LR in Russia, he selects the set of all lakes and sets it as the subject of his query by clicking on the [S] toolbar button:



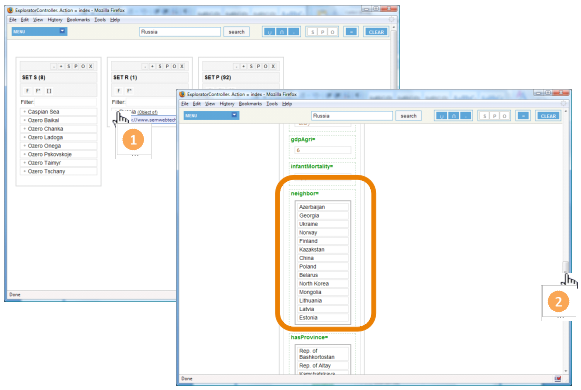
Continuing to build the query, he selects the resource Russia and sets it as the object of his query:



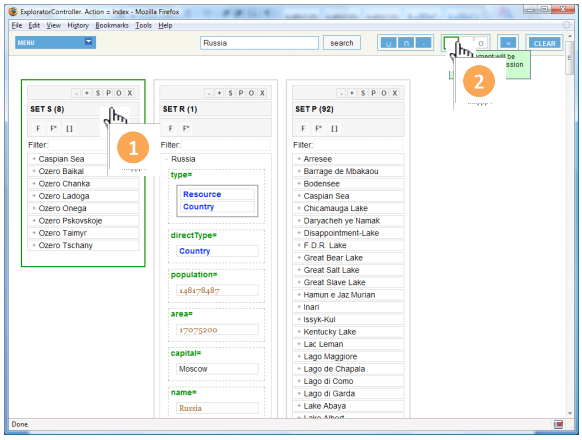
He executes the query to obtain the set of all lakes in Russia:



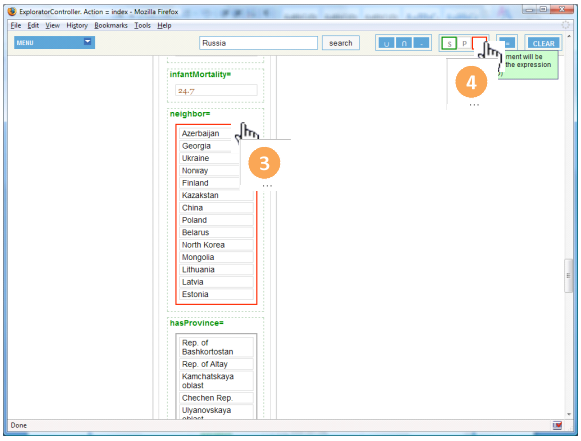
Next, to find the countries that share a boundary with Russia, he views the details of the Russia resource and locates the *neighbor* property in Russia, thereby finding its neighboring countries:



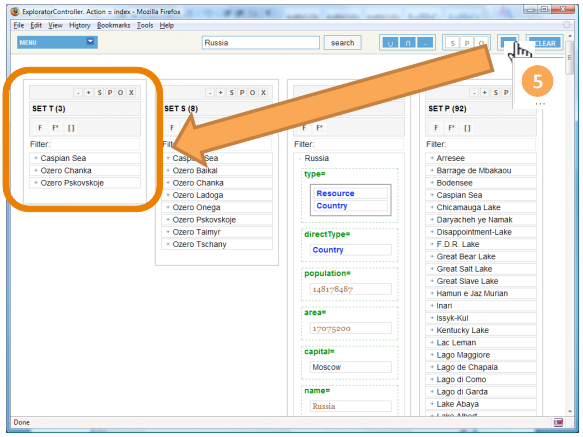
To find all the lakes in Russia's neighbors, he selects the set of Lakes in Russia and sets it as the subject of his next query:



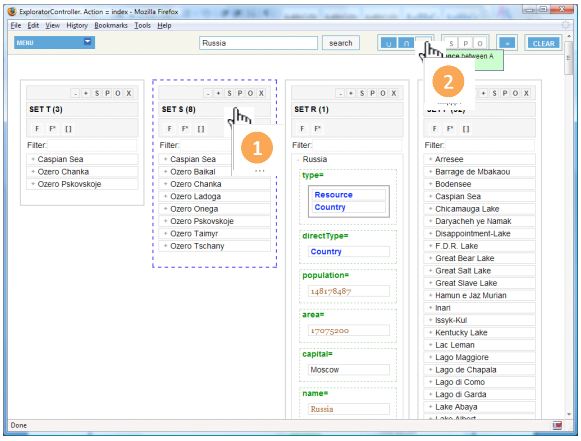
He selects the set of Russia's neighbors and sets it as the object of his query:



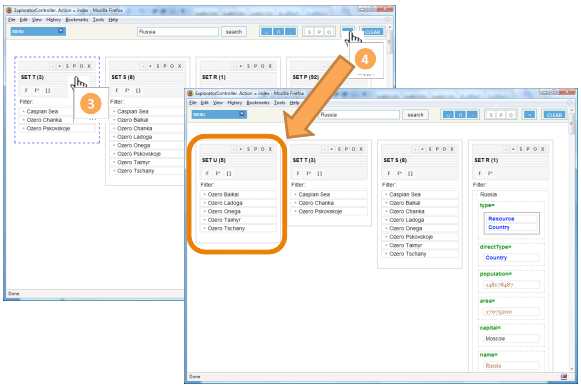
He then executes the query to find all lakes in Russia's neighboring countries:



Finally, to build the set of the lakes contained exclusively in Russia, he needs to calculate the difference between the set of lakes in Russia and the set of lakes in Russia's neighbors. To do this, he selects the first set and the difference operator:



Finally, he selects the second set (containing the lakes in Russia's neighbors) and executes the difference operation by clicking on the equal sign [=] toolbar button, thereby obtaining the desired result:



USER TESTING

We have conducted a pilot study and a small-scale experiment with Explorator to better understand the role, benefits and challenges of such a general-purpose semantic data exploration tool.

Pilot study

Six users were recruited who knew some basic concepts of the semantic web and RDF, such as the representation in <S,P,O> triples. They were provided an instructions script containing a few examples illustrating the tool usage to perform simple queries.

After going through the script, users were asked to perform a set of tasks using Explorator. Tasks 1 and 2 were performed on a database of cell phone handsets, whereas tasks 3 and 4 were performed on a database of geopolitical data, similar to the "CIA World Factbook".

- Task 1: form the set of all handsets made for Latin America that also have a WAP 2.0 browser, using the faceted navigation mechanism offered by Explorator.
- Task 2: Same as task 1, but without the faceted navigation, i.e., using the query-building mechanisms.
- Task 3: form a set with the names of the capital cities of neighboring countries of Tanzania.
- Task 4: form a set with the name of all lakes which are entirely contained within Russia.

Having completed each task, they were asked to grade the following sentences in a 5-point Likert scale:

1. I have perfectly understood the task I had to perform.
2. I found it too easy to use this tool to perform this task.
3. This kind of system would be very useful in my day-to-day activities.
4. I perfectly understood how the system works.
5. I found the interface very easy to use.
6. I noticed I could have performed this task in several alternative ways in this system.
7. For each action I took in the system, I obtained exactly what I expected.

When tabulating the results, we grouped the 2 most positive answers as "agree", and the 3 most negative answers as "disagree", obtaining the averages depicted in the following table:

Question	Agree	Disagree
1	90.91%	9.09%
2	36,36%	63,64%
3	90.91%	9.09%
4	50.00%	50.00%
5	40.91%	59.09%
6	86.36%	13.64%
7	50.00%	50.00%

In parallel with the pilot study, we inspected the Explorator's user interface. As a result of this inspection, we have decided to make some changes in the user interface, to make it more consistent and less cluttered. The resulting user interface is the one reported in this paper, and is also the version used in the experiment described next. Regarding the study planning, the pilot study revealed that it was too early to collect opinions about the system as in the proposed Likert scale. Consequently, we revised the study methodology to adopt a more qualitative approach in which we are able to gain more insight on the underlying motives of the users' actions, leaving a more quantitative study for later stages in the research.

Small-scale experiment

Due to the necessarily exploratory nature of the study at this stage, we have conducted a more in-depth qualitative study [1] with the revised user interface. We asked users to perform the same set of information exploration tasks using Explorator as in the pilot study. The users' interaction with the system was recorded using screen capture software, and their oral remarks were recorded in audio.

We have asked users to think aloud while carrying out the tasks, so as to give us insight on their thought processes [4]. At the end of the interactive session, we quickly interviewed users and posed the following questions:

- Which aspects of the user interface and interaction confused you or made you feel insecure about what you were doing and the results you were getting?
- What would you like to change in Explorator?
- What did you like the best in Explorator?

Four (4) users were recruited who knew some basic concepts of the semantic web and RDF, such as the representation in <S,P,O> triples.

Results

During the experiment, we noticed that the participants faced two separate problems in carrying out tasks. The first problem was related to the domain exploration itself, or how to discover the domain properties. The second problem was related to the participants' interaction with the user interface and with the new widgets proposed.

Regarding the first issue, we noticed that all users needed to find out the relations between classes and instances to be able to formulate their queries properly. In that process of domain exploration, all participants tried to retrieve the properties of the instances from their class. For example, some participants expanded the class Country expecting to obtain the properties of the instances of Country. However, the semantics of this operation in the tool is to display all the triples where the resource is the subject. This might work for some ontologies in which "domain" and "range" properties are declared, but this was not the case in the examples.

There was a recurring situation in which the participants made an intersection between a class and a set of instances. Ex: Lake – intersection – {Baikal, Caspian, New York, Ness, London, Paris}. When asked about what they expected, the participants said that they hoped to obtain the lakes related to those instances.

During the process of learning about the domain, some participants formulated queries such as: SPO(Russia, rdfs: property,?). When asked about this query, the participants said that they hoped to obtain all the properties of Russia. There was another recurring situation, in which the user thought in Portuguese and literally tried to translate what they had in mind into the SPO operation. A query that indicated this type of reasoning was: SPO(Lake, locatedIn, Russia). Note, in this case, that the implemented semantics is different from the one desired by the user.

Most participants had difficulties in obtaining the properties to formulate their queries. We conclude that it is vital to have a shortcut in the user interface to obtain the list of class properties. Note, however, that there actually is a widget in the interface where the user can view all the properties of an instance. Nevertheless, this widget was not accessed, perhaps because this information was not conveyed to the user in the instructions script.

Regarding the second issue, we noticed that some visual elements were not intuitive to the participants. They tended to associate the most common interface operations, such as maximize and minimize, with icons that are used today in the Windows OS, as the following testimony shows: "It would be better if the icon were equal to that of Windows" (P1). Also note that we did not provide any instructions to the participant about these newly introduced icons.

Additional observations were as follows:

- All participants began the task 1 searching for a known term. Ex.: "browser", "wap 2.0", "Latin America", "Nokia", etc. We have noticed that the user tends to use the search when looking for a known item.
- Some participants did not realize they could select the set as a whole.
- Users constantly referred to classes when intending to refer to their instances, as illustrated by the following query: SPO (Lake, locatedIn, Russia). By Lake here the users actually meant the set of lakes, and not the class itself.
- The participants expected to be able to scroll horizontally as new sets were created. However, the current scroll is vertical and this confused the participants.
- Despite the color coding of classes and properties, participants recurrently used a class instead of a property in SPO queries. However, by the end of the experiment, all participants acknowledged such differences and said to have made such mistake due to a lack of attention.
- The participants did not identify some clickable elements in the screen. One of them said, "I did not click here because the hand cursor for the mouse did not appear" (P2). We noticed that the mental model of all users reflected their familiarity with the Windows interface. Therefore, we noticed that the Explorator's widgets need to be explained to users so they can use them correctly.
- The participants successfully understood the set metaphor at the user interface, i.e., they understood that each box at the interface represented a set of resources.

CONCLUDING REMARKS

The preliminary studies have shown encouraging results. Users with only basic knowledge of RDF were able to elaborate nontrivial queries with Explorator.

We detected that the user confused the way classes and the instances were handled at the user interface. From their comments, however, we have realized they had the right intention, but in this case the user interface got in the way. This problem led us to a redesign to make it explicit whether the selection of an element at the user interface refers to the instances of the class or the class itself, maintaining the reference to the instances as the default. However, new experiments must be conducted to verify the efficiency of this proposed solution.

We also realized that the Explorator's performance had a negative impact on the user experience. It may be the case that users explored less because of the time it took to compute the queries. This issue is of the utmost importance and is being addressed for future versions.

As expected, the experiments showed us that Explorator is better suited to advanced users who have solid knowledge about RDF. Nevertheless, the experiments were

brief, so we cannot yet draw any conclusions about Explorator's learning curve.

The next step in our study will be to investigate the use of Explorator as an epistemic tool, for users to understand more about the represented data domain, as opposed to performing predefined tasks and answering specific questions. In particular, an open hypothesis is the adequacy of the RDF model to match the user's mental models – some of the collected evidence suggests that it might be too low level, which means suitable abstractions might have to be introduced.

Additional larger-scale experiments should be conducted to compare different user interface alternatives and interaction paradigms to better support both novice and expert users in exploring the semantic web. To do so, Explorator can be instrumented to remotely capture the users' actions at the user interface and on the underlying processing model.

ACKNOWLEDGMENTS

Daniel Schwabe and Simone Barbosa were partially supported by grants from CNPq.

REFERENCES

1. Araujo, S. & Schwabe, D. "Explorator: a tool for exploring RDF data through direct manipulation." *Submitted to WWW 2009*.
2. Carmel E., Crawford S. e Chen H. 1992. Browsing in Hypertext: A Cognitive Study. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22. no. 5, Sep/Oct 1992.
3. Denzin, N.K.; Lincoln, Y.S. 2005. Introduction: The Discipline and Practice of Qualitative Research. In N.K. Denzin & Y.S. Lincoln (eds.) *The SAGE Handbook of Qualitative Research, 3rd edition*. Thousand Oaks: SAGE Publications.
4. Ericsson, K.; Simon, H. (1987). Verbal reports on thinking, in C. Faerch & G. Kasper (eds.): *Introspection in Second Language Research*. Clevedon, Avon: Multilingual Matters, 24–54.
5. Explorator tool: http://www.tecweb.inf.puc-rio.br/explorator/demo (this version may already have evolved from the one reported in this paper).
6. Koenemann, J.; Belkin, N.J. 1996. A Case For Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness. *Proceedings of CHI 1996*, pp. 205-212.
7. Lakoff, G. 1987. Women, Fire, and Dangerous Things: What Categories Reveal about the Mind. *The University of Chicago Press*.
8. Lima, F.; Schwabe, D. 2003. Application Modeling for the Semantic Web, *Proceedings of LA-Web 2003*, Santiago, Chile, Nov. 2003. IEEE Press, pp. 93-102, ISBN (available at http://www.la-web.org).
9. Marchionini G. 2006. Exploratory search: From finding to understanding. *Communications of the ACM*, 49(4), 2006.
10. Oren, E.; Delbru, R.; Decker S. 2006. "Extending faceted navigation for RDF data". *5th International Semantic Web Conference*, Athens, GA, USA, November 5-9, 2006, LNCS 4273
11. Rossi, G.; Schwabe, D.; Lyardet, F. Patterns for Designing Navigable Spaces. *Proceedings of PLoP98* (Tech Report TR #WUCS-98-25, Washington University, St. Louis, MO, USA), Monticello, Illinois, USA, August 1998.
12. Schwabe, D.; Rossi, G. 1998. An object-oriented approach to web-based application design. *Theory and Practice of Object Systems (TAPOS)*, Special Issue on the Internet, v. 4#4, October, 1998, 207-225.
13. Shneiderman, B. 1998. Designing the User Interface: Strategies for Effective Human-Computer Interaction, *3rd edition. Reading, MA: Addison-Wesley*.