Parallax and Companion: Set-based Browsing for the Data Web

David F. Huynh
Metaweb Technologies Inc.
631 Howard St. Suite 400
San Francisco, CA 94105, USA
david@metaweb.com

David R. Karger
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139, USA
karger@csail.mit.edu

ABSTRACT

The arrival of the Data Web will bring an abundance of explicit semantics either complementary to or embedded within traditional web content. This body of semantics both demands and enables new interaction techniques to be introduced into the web experience. In this position paper, we proposes that the current web browsing paradigm of "one web page at a time" needs to be updated because the typical unit of web information to interact with will no longer be a whole and single web page but can be smaller and numerous bits of data. We introduce the *set-based browsing* paradigm that lets the user traverse graph-based data that will be found on the Data Web in an efficient manner, moving from a set of things to a related set of things rather than from one single thing to one single other thing. We demonstrate this paradigm as a standalone application on a web-like database and as a browser extension on existing web pages.

Categories and Subject Descriptors

H5.2 [**Information Interfaces and Presentation**]:
User Interfaces – Graphical user interfaces (GUI).
H5.4 [**Information Interfaces and Presentation**]:
Hypertext/Hypermedia – User issues.

General Terms

Design, Human Factors.

Keywords

Web browsing, faceted browsing, graph-based data, pivoting, visualization.

1 INTRODUCTION

Given a standard web browser, interacting with the Web involves looking at one web page at a time. Even when it is possible to load several web pages into several browser windows or tabs, they still remain segregated. No command—even native browser commands like Find—can act across several web pages. Hence, whatever content that typically resides within a single web page is the typical unit of web information that users can interact with. Most often, it is a document in natural language text, with hyperlinks to some other web pages.

For some information tasks, it is sufficient and perhaps even natural to deal with just one such unit of web information at a time. For example, to get a basic understanding about the economy of France, one can simply read the corresponding Wikipedia article. However, if one wants to quantitatively compare the economies of various countries previously in the French colonial empire, then one must read many articles, extract out the quantities for comparison (e.g., GDP), tabulate them in a spreadsheet, and then construct a visualization such as a bar chart to show the comparison. That is, when dealing with data in the aggregate, in order to gain a big picture from bits and pieces extracted out from several sources, the current Web and web browser fall short due to their "one web page at a time" browsing paradigm, leaving the user to flip between several web pages and carry out tedious data tabulation manually.

There is evidence that more and more structured data—as opposed to natural language text or multimedia content—is arriving on the Web. For example, microformats [2] and RDFa [5] make it possible to annotate bits of web content with explicit semantics, making them machine-processible, and data APIs from a host of web sites have spawned many data mash-up applications and widgets. This new abundance of data and data sources both makes it easier for users to retrieve bits and pieces of data from several sources (without resorting to screen scraping HTML) as well as amplifies the desire to deal with such data in the aggregate to surface trends and insights that no single source alone offers.

In this position paper, we propose that, in the future when it is the norm for web users to interact with web data on a frequent basis, the original browsing paradigm of the Web—one web page at a time—is no longer suitable because the typical unit of information to interact with is no longer a whole and single text document.

In particular, we propose a novel browsing paradigm called *set-based browsing* in which browsing to an aggregate of several units of information is a core browsing operation, to be supported as universally as the Back button in the web browser. We prototype this browsing paradigm in a web interface called Parallax on top of a web-like database and illustrate how it allows users to navigate through the data more efficiently than by traditional means. We also attempt to integrate this navigation paradigm into the web browser in the form of a browser extension called Companion, and illustrate how this new navigation paradigm might be grafted onto the current web experience.

WWW 2009, May 8–12, 2009, Madrid, Spain.
ACM 978-1-59593-654-7/07/0005.

1.1 Outline

For the rest of the paper, we first present a linguistic analogy to help explain the set-based browsing paradigm conceptually. Next in section 3, we describe Parallax, a specific concrete implementation of the paradigm. In section 4, we discuss the paradigm's concepts in general terms. Section 5 illustrates how the browser extension Companion grafts the paradigm onto the existing web browsing experience. Section 6 details the prototypes' implementation. Section 7 presents related work and section 8 discusses this direction of research and draws conclusions.

2 LINGUISTIC ANALOGY

Every human language includes proper nouns—words and phrases that when evoked, communicate the identities of specific concrete or abstract entities, e.g., IBM. URLs are the analogous proper nouns for concrete entities in the "language of the Web," e.g., http://www.ibm.com/. One person can send another person a URL to communicate exactly what he/she is looking at through his/her web browser.

Semantic Web technologies add more parts of speech to the language of the Web. URIs let us communicate the identities of *abstract* entities that might not have any representation (that is, there are no corresponding web pages). Some of these abstract entities are nouns that stand for classes of entities, e.g., **`http://www.daml.org/2001/03/daml+oil#book`** stands for books. RDF lets us describe these entities individually (e.g., specifying that a particular book was published in 1984, or specifying that every human has two biological parents) and relate them together (e.g., specifying that Austin is a city in Texas).

Human languages have evolved to increase our power of speech. For example, without proper nouns for concrete entities, to refer to something, we must stand in its vicinity and point to it. Proper nouns free us from that physical constraint. Similarly, once capable of expressing relationships between entities, we can refer to something without a proper noun by relating it to things with proper nouns, e.g., "John's mother's older sister." We can also use attributes of entities to identify them, e.g., "the tall boy with dimples."

Our languages also allow us to refer to entities en masse rather than individually: the founders of this nation, the invaders from the North, the families in grief, the flights being delayed, etc. Note the plurals in those phrases. This power of speech frees us from the need to identify individual members of a group of entities and lets us to communicate much more efficiently.

As Semantic Web technologies increase the amount of explicit semantics complementary to or embedded within traditional web content, they should also increase the analogous "power of speech" on the Web: we should be able to express our desired interactions with web content more powerfully and more efficiently. On the current Web, we can command the browser to follow only one hyperlink at a time. On the Data Web, we can tell the browser to follow several hyperlinks that can be identified together, e.g., all hyperlinks from this movie to its actors and actresses who have received acting awards. This is the essence of the set-based browsing paradigm: core browsing operations are provided for following several hyperlinks that can be identified together. The result of following one or more hyperlinks is a *set* of one or more entities, hence the name of the browsing paradigm. When only one hyperlink is followed, yielding only one entity, the experience is degraded to the original web browsing experience.

Figure 1. The user starts using Parallax by entering search keywords. Parallax responds by suggesting database entry types as well as specific entries whose names match those keywords.

3 PARALLAX DATA BROWSER

Having explained the set-based browsing paradigm conceptually through a linguistic analogy, we will now describe one particular implementation of it—the Parallax data browser—so to give the reader a more concrete picture. Then in section 4, we will generalize Parallax's specific features to general concepts.

Parallax is a web application built on top of the Freebase service [1]. It allows people to browse through the heterogeneous graph-based data that is in Freebase. The reader is encouraged to view the screencast about Parallax and try it out at http://mqlx.com/~david/parallax/.

A user starts using Parallax much like using a typical search engine—by performing a keyword search (Figure 1). Parallax responds by suggesting the types of data whose names contain the keywords the user enters (e.g., US President) as well as specific database entries whose names match. Picking a type from the list of suggestions shows all entries of that type (Figure 2). The left side of the page lists several filters that the user can use to narrow the results down further. These filters are also known as *facets* in the faceted browsing paradigm [15]. If these filters which are automatically suggested are not enough, the user can add more filters using

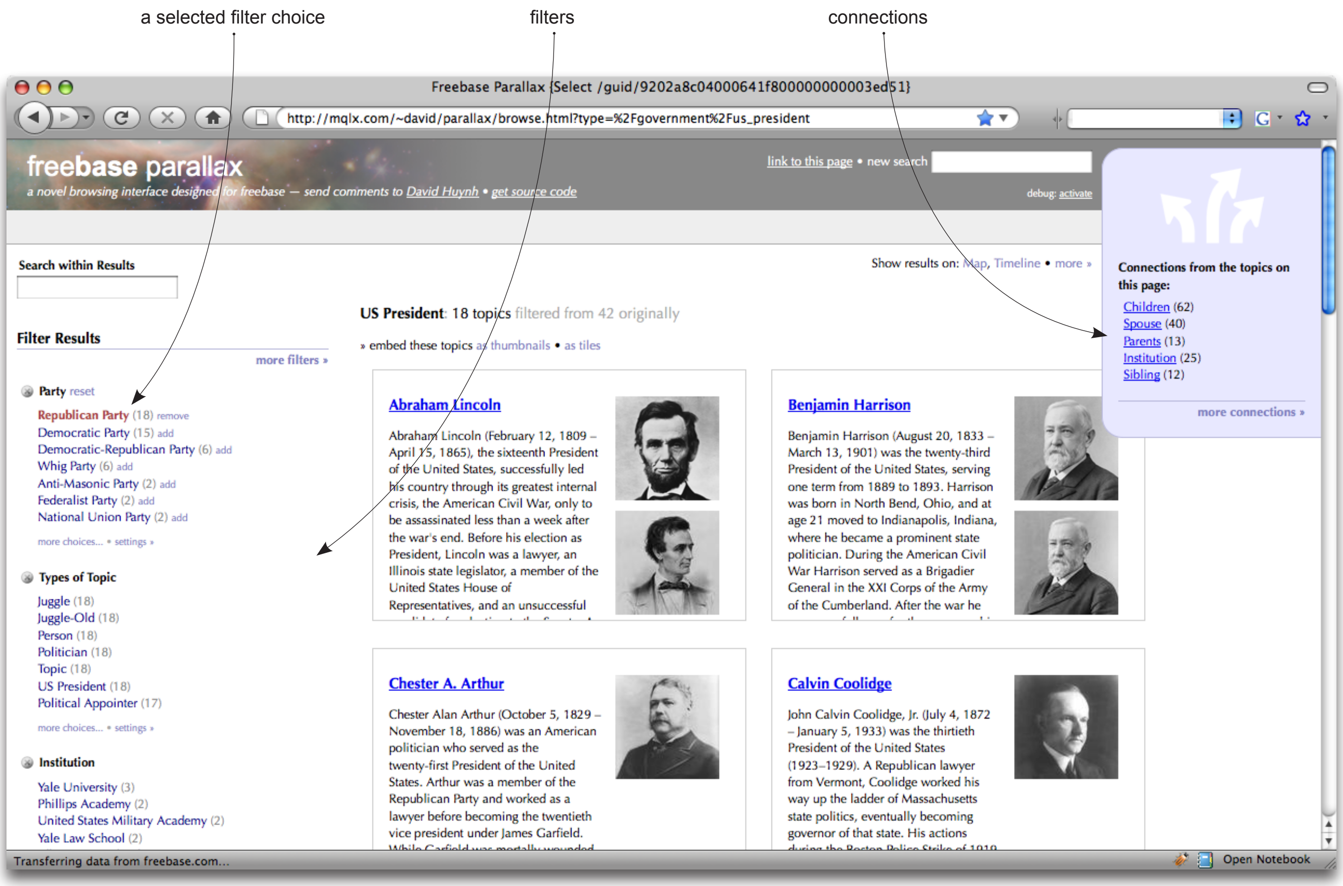


Figure 2. Once the user picks a database entry type suggested for a keyword search (Figure 1), Parallax displays all entries of that type (US President in this example). Parallax provides several filters (on the left side) for narrowing down the result set further. Here, Republican Party is selected in the first filter; this causes only 18 out of 42 presidents to be shown. The top right corner of Parallax lists several *connections*. A connection is a compound link made up of hyperlinks of the same kind of relationship. It lets the user browse from the current set of entries to another set of entries connected through that relationship.

the add more filters link. So far, Parallax behaves much like any conventional faceted browsing interface such as those for browsing online product catalogs.

3.1 Connections

The top right corner of Parallax shows a list of *connections*. A connection is a group of hyperlinks from the current result set (of database entries) to other database entries, defined by a particular relationship. For example, as shown in Figure 2, the 18 currently shown database entries about 18 presidents have hyperlinks typed "children" to 82 other database entries. Or more humanely, the 18 presidents have 82 children, and clicking on Children (82) will switch Parallax from showing the 18 presidents to showing their 82 children (Figure 3). This novel feature is a core set-based browsing operation that lets the user browse from one set of things to another related set of things.

Note that we arrived at 18 presidents by filtering 42 presidents by their political party to only the Republicans, as annotated in Figure 2. Thus, we can describe the 82 entries shown when the connection labeled Children (82) is clicked as "children of the Republican presidents." As discussed in the Linguistic Analogy section, given more explicit semantics, user interface mechanisms such as filters and connections combined allow us to perform more powerful browsing operations: such a result set as "children of the Republican presidents" cannot be retrieved using just conventional hyperlinks.

3.2 "Browse All"

After entering search keywords at the start, the user can pick a particular database entry (rather than a type) in the suggestion list to get to that particular entry. The user can also get to a single database entry (rather than a set of entries) by clicking on any thumbnail within a set of entries. Once at a particular entry, the user can get to another entry by clicking on a conventional hyperlink.

With regards to set-based browsing, there is one relevant feature in how Parallax displays a single database entry. Whenever there are several values for a particular property, such as the property Contributor in Figure 4, a connection labeled browse all is shown below the property values. Clicking on that connection switches Parallax from showing that single entry to showing all those property values together. Thus, connections lead not only from a set of entries to another set of entries, but also from a single entry to a set of entries.

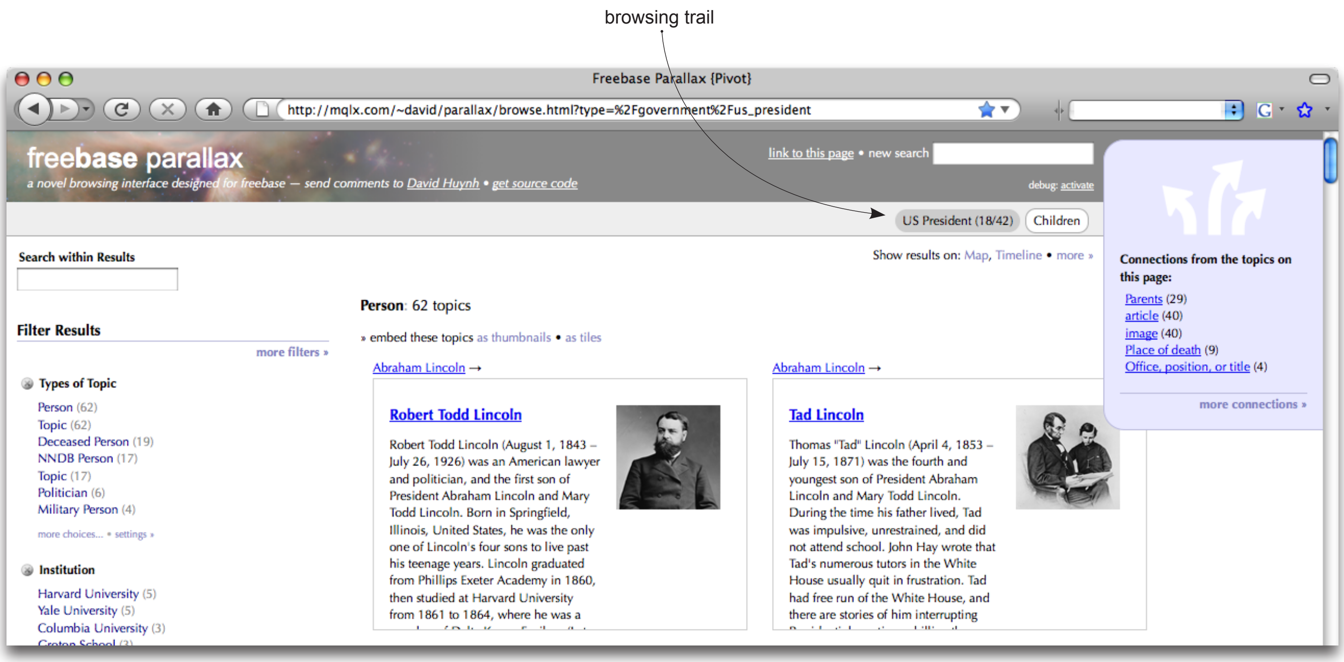
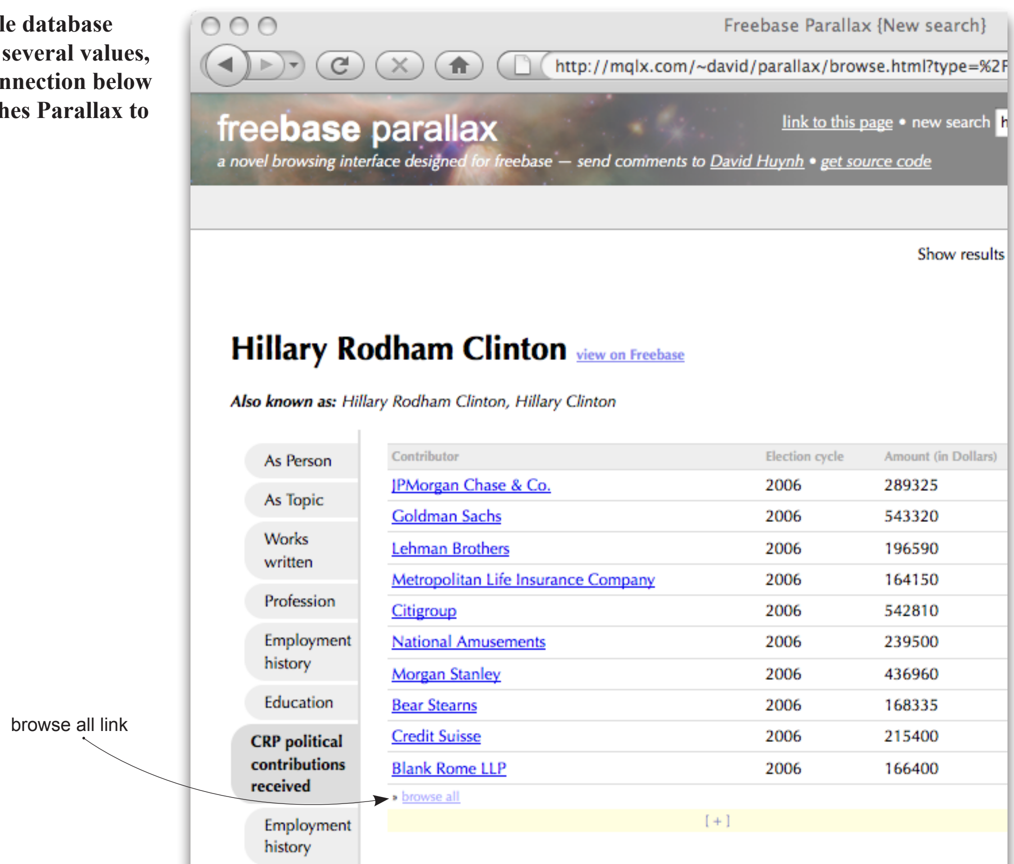


Figure 3. Clicking on a connection switches Parallax from showing one set of entries to showing another set of entries, connected to the first set through a particular relationship. A browsing trail displays all the sets of entries or single entries that the user has browsed through.

Figure 4. When displaying a single database entry, for each property that has several values, Parallax appends a browse all connection below the values. This connection switches Parallax to showing that set of values.



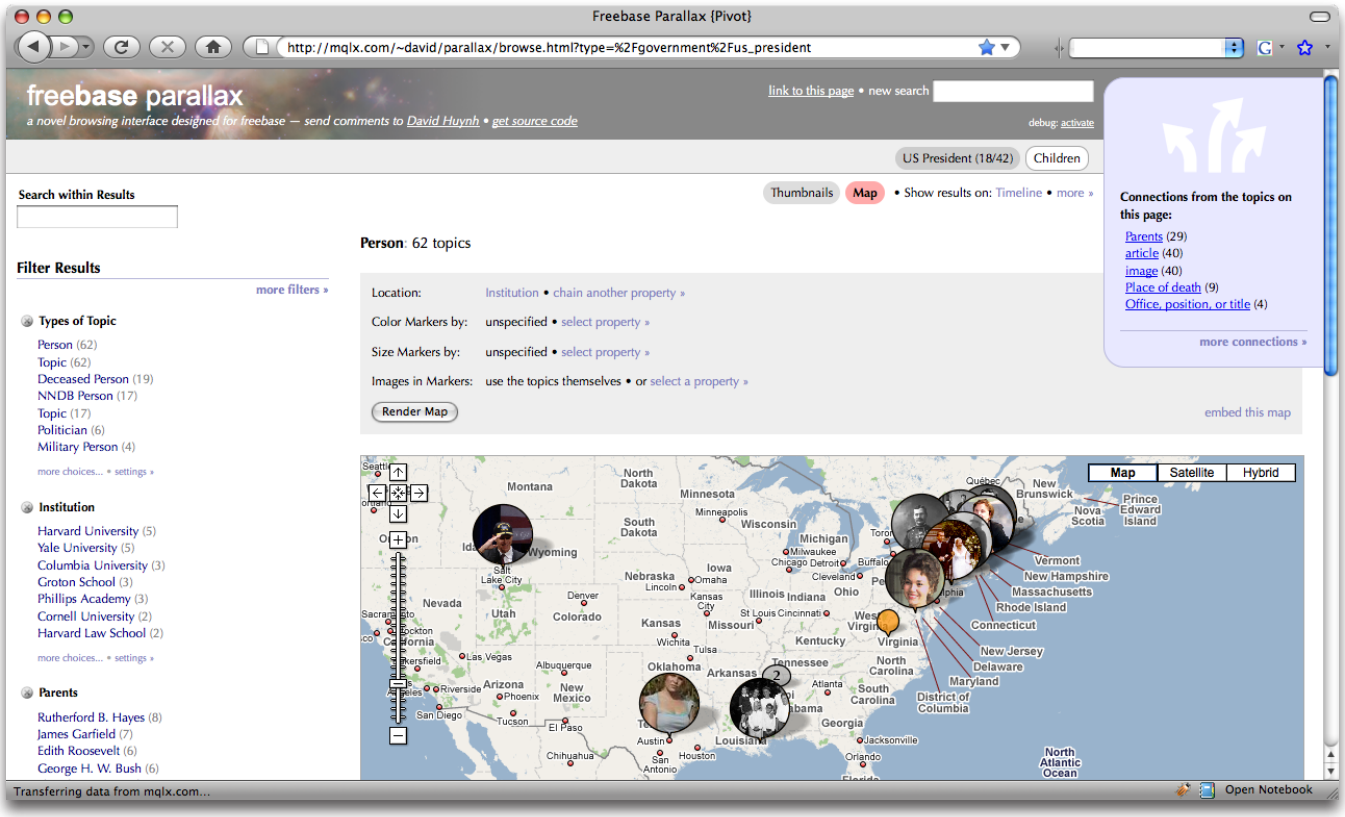


Figure 5. Because it's easy in Parallax to get to a set of database entries, there are many opportunities to construct aggregate visualizations (called *views*).

3.3 Browsing Trail

In the conventional web browser, each point in the browsing history is a single web page named by a URL. In Parallax, each point in the browsing history can be either a single database entry or a set of several entries. Figure 3 shows how Parallax presents this browsing history, also called the *browsing trail*, to the user. It is a horizontal list of *trail points*, each displaying the name of the type, the entry, or the relationship that defines that point. Each trail point can be clicked in order to switch Parallax to showing its corresponding entry or set of entries.

One trail point might influence the next. For instance, by filtering the 42 presidents down to only 18 Republicans, clicking on the connection Children (62) shows us only children of Republican presidents, not children of all presidents. When the user switches back to an earlier trail point and changes its filters (e.g., deselecting the Republican Party filter choice), subsequent trail points are affected. This is different from conventional web browsing in which interactions with an earlier web page does not typically affect any subsequent web page.

3.4 Views

The ease with which the user can browse to a set of database entries yields many opportunities for constructing aggregate visualizations of those entries, such as plotting them together on a map (Figure 5). Parallax offers many kinds of aggregate visualization (called *views*), including maps, timelines, 1D plots, 2D plots, etc. Note that these aggregate visualizations do not make sense in conventional web browsing in which you can only see one thing at a time.

4 GENERAL CONCEPTS

Having described Parallax as a specific concrete implementation of the set-based browsing paradigm, we now describe the paradigm's concepts in generality and point out the open issues that need further research.

4.1 Transitions

In the set-based browsing paradigm, users can make four types of transition as determined by the cardinalities of the source and target sets of entities: one entity-to-one entity transitions, one entity-to-many entities transitions, many-to-many transitions, and many-to-one transition. Note that as soon as we can browse from one entity to many entities, then we must be able to browse onward from many entities, hence the last two types of transition.

One-to-one transitions are the only type of transition available on the Web today, and we propose no change to it. The purpose of categorize transitions into four types is to compare the set-based browsing paradigm against the original Web browsing paradigm. We will now discuss the three other, new types of transition.

4.1.1 One-to-Many Transitions

A simple example of a one-to-many transition is browsing from a company to its employees. A more complex example is browsing from a company to those among its employees who live within five miles of the office. In the second case, the collective identity of the links being followed consists of the relationship "employee" as well as the constraint on the target entities (that they live within 5 miles of the office). In Parallax, one-to-many transitions are supported through browse all connections in conjunction with filters on the target set.

4.1.2 Many-to-Many Transitions

A simple many-to-many transition can be browsing from several artists to their artworks. A complex transition can be, while looking at several artists, browsing to artworks sold for over one million dollars by the ones among those artists who have had exhibits in San Francisco. The collective identity of the links in the second case consists of the relationship from artists to artworks (rather than, say, from artists to exhibits), the constraint on the source entities (artists with exhibits in San Francisco), and the constraint on the target entities (artworks sold for more than $1M). In Parallax, many-to-many transitions are supported through connections at the top right corner in conjunction with filters on the source and target sets.

4.1.3 Many-to-One Transitions

A many-to-one transition can be browsing from several movies to the one protagonist of the movie that won the Best Picture award. Such a transition can be made by clicking on a single hyperlink much like how every one-to-one transition is made on the current Web. However, since there are many source entities, and hence many more hyperlinks than in one-to-one transitions, it takes more work to locate the one hyperlink to click. Thus, user interface mechanisms additional to hyperlinks might be needed to facilitate many-to-one transitions. Parallax provides filters for narrowing the source set down further until the desired hyperlink can be spotted visually.

4.2 Link Selection

To follow only some hyperlinks among many available, the user must be able to select just those links through some user interface operations. Such selection can be done in a manual, direct way in which the user selects (but doesn't immediately invoke) one hyperlink after another and then invoke them all together. This is similar to how one might select several files within a file browser and then invoke a command on all of those files. While this direct selection mechanism may be familiar to most users, it can be very tedious and error-prone to use. It is easy to miss some hyperlinks that should be selected because there are so many.

Alternatively, hyperlinks can be selected by defining their collective identity, particularly by specifying constraints that they and they alone must satisfy. As each hyperlink connects a source entity to a target entity and denotes a specific relationship, three types of constraint can be used to select hyperlinks.

4.2.1 Source Constraints

A source constraint is used to narrow down the hyperlinks to only those starting from the source entities that match that constraint. In section 4.1.2, the constraint "with exhibits in San Francisco" is placed on the source entities "artists." Parallax's filters are provided for this purpose.

4.2.2 Target Constraints

A target constraint narrows down the hyperlinks to only those ending at target entities that match that constraint. In section 4.1.2, the constraint "sold for more than $1M" is placed on the target entities "artworks." Parallax's filters are also provided for this purpose.

4.2.3 Relationship Constraints

A relationship constraint acts on the meaning or the metadata of the hyperlinks. For example, we might want to follow only "made artwork" hyperlinks added into the data model after a particular date and validated by at least some number of domain experts. Parallax currently can only constrain the hyperlinks by their relationship kind (e.g., "children" hyperlinks).

4.2.4 Constraint Combinations

A set of hyperlinks can be defined collectively, and thus selected for following, by any combination of constraints. The second example in section 4.1.2 illustrates one such combination. Note that we have not found a reason why all constraints of a combination should be specified together in a single user interface operation, so it is conceivable that constraints can be applied one at a time until the transition is fully made. In Parallax, the user would filter the source set, click through a connection, and then filter the target set in order to enforce a constraint combination.

4.3 Collection Views

In the set-based browsing paradigm, it is the norm to be viewing a set of entries. This is a departure from the conventional web browsing experience of viewing one web page at a time. There is a challenge in that the browser must be able to display many things at once, but there are also opportunities for showing the aggregate data in various kinds of visualization (maps, timelines, scatter plots, etc.), making the experience even richer and more valuable for the user. We refer to these various ways in which the browser shows a set of entries as *collection views*. Although this paper is about a novel browsing paradigm, for some users, it is these collection views enabled by the browsing paradigm that are the primary value proposition.

4.4 Browsing Context

The explicit semantics within the Data Web will allow us to reference things *en masse* and by relationships rather than just individually and directly by URLs. For example, rather than just referencing the Eiffel Tower at http://en.wikipedia.org/wiki/Eiffel_Tower, using something like Parallax, we can browse to "the architects who designed skyscrapers." We can even go further to "the corporations with more than $5B in capital who own buildings in New York designed by European architects who have designed skyscrapers located in North America." In an interface as Parallax, when viewing the corporations, there is a complicated browsing context consisting of filters and connections that has led us to the current database entries. This context is displayed as the browsing trail but there can be other ways to present it. Showing the context in a comprehensible manner without taking up too much valuable screen space is a user interface design challenge. Early informal feedback from users indicates that when a user receives a permanent link to Parallax in a particular state, the user does have trouble understanding the browsing context.

5 THE COMPANION EXTENSION

Such novel concepts as those discussed in the previous section, whether or not they are useful and powerful, must be carefully introduced into the current web experience in order to achieve adoption. In this section, we describe our first attempt of grafting the set-based paradigm onto the current web experience in the form of a web browser extension called Companion.

When Companion is activated, it detects named entities within the current web page and highlights them (Figure 6). In the future when these entities are identified semantically within the web page, Companion will not have to perform named entity extraction. But for now, named entity extraction is necessary for Companion to work on the existing Web.

Companion offers filters on the right hand side to let the user narrow those detected entities down to some subset of interest. (The filters are computed from data in Freebase.) Then the user can click on a connection in the right pane to browse a related set of entities (using data from Freebase). The connection takes the user into Parallax.

From a set of entities, the user can also find recent news articles about them. Companion presents those news articles (web pages) side-by-side as miniatures (Figure 7). Hovering on a miniature displays a magnifying glass for closer inspection, and clicking on a miniature zooms into the corresponding web page, showing it at 100% and making it interactive like a normal web page. Companion can also be commanded to detect named entities within those news articles. The detected entities are shown as pins on the miniatures. Filters and connections are offered on those entities in multiple news articles much like on a single web page.

Using Companion, the user can thus traverse multiple hyperlinks from a web page to a related set of entities. Although those hyperlinks do not come from within the web page itself but rather from external sources such as Freebase, we hope to still be able to illustrate the set-based browsing paradigm within the context of an existing web page.

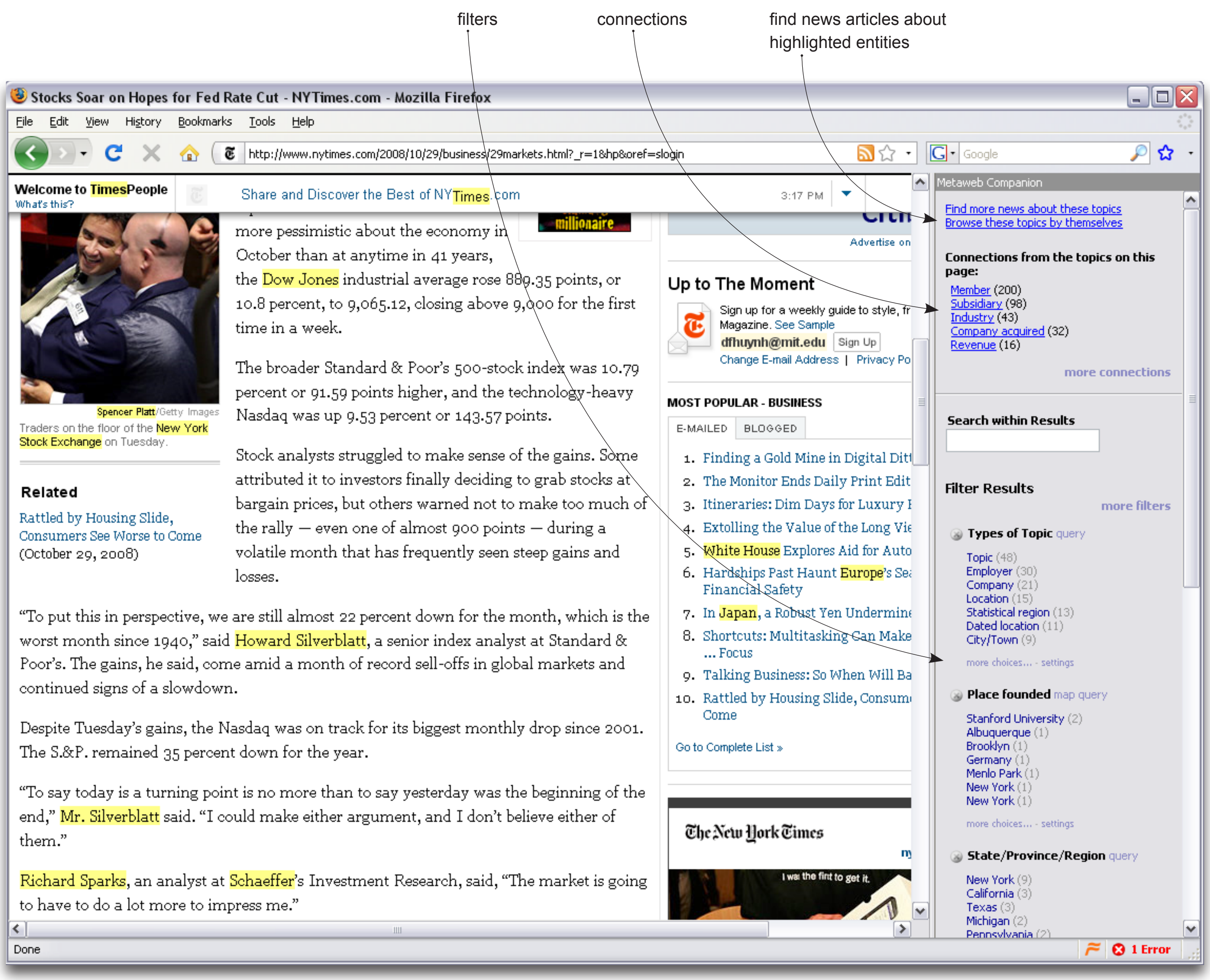


Figure 6. The Companion browser extension detects named entities within a web page and highlights them in yellow. It also offers filters on the right side to let the user narrow down to just those entities of interest. Connections are offered for browsing to other sets of entities related to these entities of interest. There is also a link for finding recent news articles about the entities of interest.

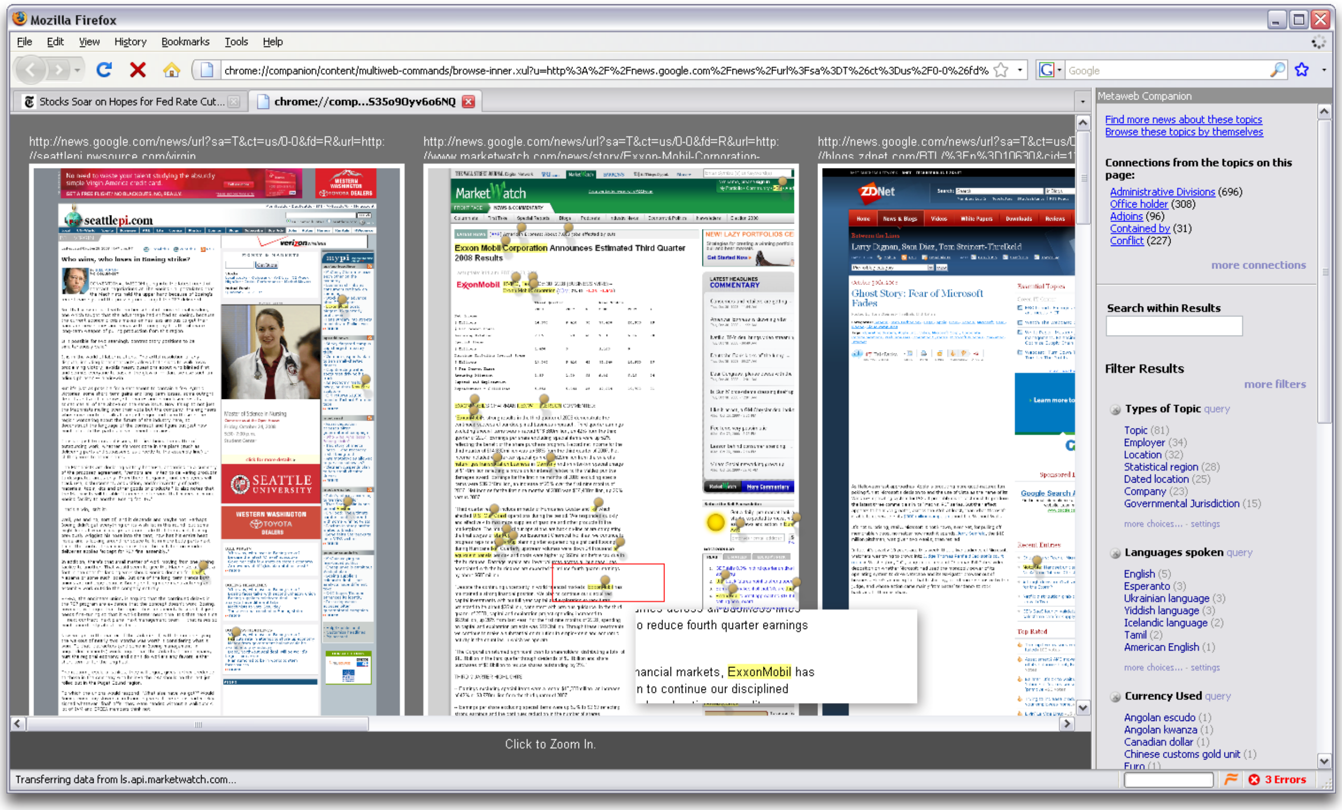


Figure 7. Companion can deal with several web pages at a time. This might be useful for exploring a set of news articles mentioning a set of topics. The web pages are shown as miniatures with detected named entities indicated with pins. Filters and connections are offered on all entities across all pages.

6 IMPLEMENTATION

In this section, we discuss the implementation of Parallax and Companion at a high-level of abstraction.

6.1 Parallax

Parallax is implemented primarily in HTML and Javascript. Some of what can be done in Javascript is instead implemented on the server simply for performance. That is, it is a client-side only web application. Parallax's code is open source and can be found at http://code.google.com/p/freebase-parallax/.

Parallax's Javascript code queries the Freebase service for data and schemas using the Metaweb Query Language (MQL) [3]. All user interaction and system states are kept on the client side.

The overall state of Parallax at any time is a structure that captures the current browsing trail, which is a list of *trail points*. Each trail point consists of a *collection definition*, a collection, several facets, and one or more views. The collection object provides access to two sets of entities: the set of entities as currently displayed (potentially with some facet choices selected), and the set of entities that would be displayed if no facet choice is selected. The first set is called the *restricted set* and the second the *total set*. The collection definition specifies how to query for the total set, and the facets augment this query further to specify how to query for the restricted set. The views render the entities in the restricted set.

There are three kinds of collection definitions. One kind consists of one or more IDs of specific entities. This kind is used whenever the user searches for a specific entity (e.g., "Hillary Clinton"), or when the user clicks on a connection within Companion and Companion invokes Parallax with the initial set of entities. Another kind of collection definition consists of a type ID. This kind is used whenever the user searches for a type of entities (e.g., "US President"). The third kind of collection definition consists of a reference to the preceding collection in the browsing trail and a property ID. This kind is used when the user follows a connection.

Connection-based collection definitions (the third kind) are what tie trail points in the browsing trail together. A one-to-many or a many-to-many transition results in a new trail point linked to the previous one, whereas a one-to-one or a many-to-one transition results in a new independent trail point.

6.2 Companion

Companion is a Firefox browser extension implemented in XUL, HTML, and Javascript. For the functionality that it shares with Parallax, it embeds and reuses Parallax's code. For the remaining functionality, it invokes the OpenCalais web service [4] for named entity extraction and Google News API for finding recent news articles.

7 RELATED WORK

7.1 Extending the Web Browsing Experience

The recognition that the original web browsing paradigm—one web page at a time—is lacking is not new. In the early days of the Web, researchers already experimented with web crawling agents that scouted ahead of the user and recommended which links to go forward given the user's past behavior, essentially giving the user farther sight beyond the current page (e.g., WebWatcher [7], Letizia [12]). Still, these agents only let the user browse forward one hyperlink at a time.

Recently, researchers have started to explore supporting semantic interactions within web pages. The browser extension Magpie [8] can perform named entity extraction on any web page and then "semantic services" can run on those extracted entities to support additional, semantic operations. One such operation is browsing to a "semantic neighbor" of an entity—another entity related to the entity in question using data from a different source. Compared to Companion, Magpie does not provide a way to browse to a *set* of semantic neighbors.

7.2 Browsing and Querying

Parallax's browsing interface can be seen as an extension of faceted browsing interfaces, the first few of which include the Flamenco system [15]. Faceted browsing has also been applied to generic semantic web data in such systems as mSpace [13], /facet [9], and Piggy Bank [10]. None of these systems allows browsing from one set to another.

Preliminary work at Endeca Technologies, Inc. [6] has been started to extend the faceted browsing paradigm to work on graph-based data (as opposed to tabular data for which faceted browsing has been designed). This work only seems to target closed-world data corpora which contain a limited number of known schemas. In contrast, Parallax works on Freebase, which contains numerous heterogeneous schemas.

Another preliminary work, the Humboldt browser [11], is experimenting with browsing linked (semantic web) data. Humboldt is designed for the user to build up a complex query involving a small number of related sets. It does support pivoting from one set of entities to another set of entities like Parallax, but pivoting in Humboldt is essentially switching between those few related sets rather than browsing forward in a large graph of data.

We also note that the concept of browsing from set to set has been explored in web application modeling research. The Object-Oriented Hypermedia Design Method (OOHDM) [14] models navigation through a data space as a sequence of "navigation contexts," each of which contains a query defining the set of things of interest. This parallels Parallax's browsing trail closely. The difference is that OOHDM does not expose a uniform user interface mechanism (like connections in Parallax) for the user to take control of browsing from one set to another. Furthermore, OOHDM is for building individual web applications, whereas in this paper, we're proposing to aim for a browsing paradigm that works over the entire Data Web.

8 DISCUSSION & CONCLUSION

In this paper, we argue that with the coming of the Data Web, the original web browsing paradigm of "one web page at a time" needs to be updated as the typical unit of information to interact with is no longer a whole and single text document, but can be a blob of data. In particular, we propose a new browsing paradigm called set-based browsing that lets the user efficiently browse through graph-based data by moving from a set of things to a related set of things, following several hyperlinks at the same time. We demonstrate this paradigm in a standalone web application called Parallax that works on top of the web-like data in Freebase. We also present a web browser extension called Companion that illustrates how this paradigm might be grafted onto the current web browsing experience by bootstrapping from named entity extraction.

While Parallax can be used to formulate complex structured queries, query formulation is not our focus in this paper. To us, query formulation implies that we are dealing with a closed, unchanging, relatively coherent database containing a limited number of known schemas and built for some specific applications in mind. Web browsing, on the other hand, has to deal with an open, constantly changing, heterogeneous, huge body of content and data authored by millions in an uncoordinated manner for no particular application in mind. We believe that user interfaces developed for one context cannot be optimal for the other.

Query formulation also implies that the user already knows all the characteristics of the final results and it is simply a matter of tweaking the query to get precisely those results. In contrast, web browsing, even with a clear information seeking purpose, often involves opportunistic walks in the information landscape. As the user does not know what relevant data exists on the Web, she has to react to intermediate results at every stage of her exploration, deciding which route to take next.

Furthermore, there might be no such thing as the final results for her purpose, but instead, it is the exploration process itself together with all of the intermediate results that she wants. This is similar to how one might actually want to walk around a city to understand its layout and enjoy its culture rather than taking a quick taxi ride to a specific destination. In case the city is not walkable and exploration by foot would be laborious, a more effective means of transportation is needed, such as the bicycle or the personal car. The set-based browsing paradigm can be seen as a more efficient means for exploring the Data Web, complementing the tradition "one web page at a time" paradigm.

Continuing with our city exploration analogy, we note that even with bikes and cars, there are times when and places where walking is more effective and desirable. In the same way, we expect that exploration of the Data Web will involve a mix of moments when the user is reading narratives in natural language text, digesting information that would be very hard to capture as rigid structured data, and moments when the user is leveraging the structured data to follow hyperlinks *en masse* or to construct custom aggregate visualizations. We see the set-based browsing paradigm as a promising first step in the search for user interface mechanisms for supporting the second kind of moments.

REFERENCES

[1] Freebase: an open, shared database of the world's knowledge. http://www.freebase.com/.
[2] Microformats. http://microformats.org/.
[3] MQL Reference Guide. http://mql.freebaseapps.com/.
[4] OpenCalais. http://www.opencalais.com/.
[5] RDFa Primer. http://www.w3.org/TR/xhtml-rdfa-primer/.

[6] Anderson, C. Record Relationship Navigation: Implications for Information Access and Discovery. HCIR Workshop 2007.

[7] Armstrong, R., D. Freitag, T. Joachims, and T. Mitchell, Web Watcher: A Learning Apprentice for the World Wide Web. AAAI Spring Symposium on Information Gathering, 1995.

[8] Domingue, J.B., M. Dzbor, and E. Motta. Magpie: Browsing and Navigating on the Semantic Web. IUI, 2004.

[9] Hildebrand, M. and L. Hardman. /facet: A Browser for Heterogeneous Semantic Web Repositories. ISWC 2006.

[10] Huynh, D., S. Mazzocchi, D. Karger. Piggy Bank: Experience the Semantic Web Inside Your Web Browser. ISWC 2005.

[11] Kobilarov, G. and I. Dickinson. Humboldt: Exploring Linked Data. Linked Data on the Web Workshop (LDOW2008) at WWW 2008.

[12] Lieberman, H. Letizia: An Agent That Assists Web Browsing. International Joint Conference on Artificial Intelligence, 1995.

[13] schraefel, m. c., D. Smith, A. Russel, A. Owens, C. Harris, and M. Wilson. The mSpace Classical Music Explorer: Improving Access to Classical Music for Real People. Music Network Open Workshop, Integration of Music in Multimedia Applications, 2005.

[14] Schwabe, D. and G. Rossi. An Object Oriented Approach to Web-Based Applications Design, New York, Theory and Practice of Object Systems (TAPOS) 4 (4) pp 207-225, Wiley & Sons, 1998, ISSN: 1074-3224.

[15] Yee, P., K. Swearingen, K. Li, and M. Hearst. Faceted Metadata for Image Search and Browsing. SIGCHI 2003.