Web Semantics: Science, Services and Agents on the World Wide Web 8 (2010) 182–189

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Invited paper

Reflect: A practical approach to web semantics

Seán I. O'Donoghue[a,*], Heiko Horn[a,b], Evangelos Pafilis[a], Sven Haag[a], Michael Kuhn[a], Venkata P. Satagopam[a], Reinhard Schneider[a], Lars J. Jensen[a,b]

[a] European Molecular Biology Laboratory, Heidelberg, Germany
[b] NNF Center for Protein Research, University of Copenhagen, Denmark

ARTICLE INFO

Article history:
Received 6 August 2009
Received in revised form
12 November 2009
Accepted 25 March 2010
Available online 29 April 2010

Keywords:
Augmented browsing
Semantic annotation
Named entity recognition
Genes
Proteins
Chemicals

ABSTRACT

To date, adding semantic capabilities to web content usually requires considerable server-side re-engineering, thus only a tiny fraction of all web content currently has semantic annotations. Recently, we announced Reflect (http://reflect.ws), a free service that takes a more practical approach: Reflect uses augmented browsing to allow end-users to add systematic semantic annotations to any web-page in real-time, typically within seconds. In this paper we describe the tagging process in detail and show how further entity types can be added to Reflect; we also describe how publishers and content providers can access Reflect programmatically using SOAP, REST (HTTP post), and JavaScript. Usage of Reflect has grown rapidly within the life sciences, and while currently only genes, protein and small molecule names are tagged, we plan to soon expand the scope to include a much broader range of terms (e.g., Wikipedia entries). The popularity of Reflect demonstrates the use and feasibility of letting end-users decide how and when to add semantic annotations. Ultimately, 'semantics is in the eye of the end-user', hence we believe end-user approaches such as Reflect will become increasingly important in semantic web technologies.

1. Introduction

A common situation facing anyone reading text on a web-page is coming across names or concepts and wanting to know more details. In some cases, the reader wants only to quickly check the definition of the name or concept, whereas in other cases, the reader would like to navigate to other web-pages showing more detailed information about the name or concept.

Currently, faced with this situation, a reader typically executes the workflow: copy, paste, and Google. This approach usually works well enough, however some publishers simplify this process by pre-tagging names and concepts. For example, iHOP [1] provides access to a large body of the biomedical literature in which the names of genes, proteins, and other biological keywords have been systematically tagged. Such tags can help the reader comprehend scientific content more rapidly and completely.

In many cases, it would be useful if such systematic semantic tags were available for any web-page. This is especially true for complex, fast-changing technical or scientific fields with a rapid growth in the number of entities. For example, in the life sciences, there are millions of fundamental entities (genes, chemicals, pathways, etc.). This has long since overwhelmed the ability of an individual scientist to be aware of all entities. Moreover, the intricate web of interconnections between entities leads to the situation that even an expert in a focused research area can encounter unfamiliar entities on a daily basis when keeping up-to-date with the latest research literature.

Semantic tagging of an entity is only part of the story: equally important is the information that is accessed when the user clicks on a tag. In the past, entity tags were almost always simple hyperlinks to web-pages showing source data entries. Increasingly, however, entity tags are not hyperlinks but scripts that create a small popup window. A key advantage of using popups is that users can see basic information about an entity in the context of the current web-page, without having to navigate away to other pages. If needed, hyperlinks to more detailed information can be provided on the popup.

However, not all users want to see the same information about an entity. For example, a chemist may like to easily navigate from the name of a chemical to the 2D chemical structure, to information about bioactivity, or to other detailed information. For many non-scientists, such information could be very confusing: when they see the name of a chemical in a web-page, they would probably prefer to access a short text description explaining, in lay terms, what the chemical is typically used for.

* Corresponding author at: European Molecular Biology Laboratory (EMBL), Meyerhofstr. 1, 69117 Heidelberg, Germany. Tel.: +49 6221 387 8463; fax: +49 6221 387 517.
E-mail addresses: sean.odonoghue@embl.de (S.I. O'Donoghue), hornheiko@gmail.com (H. Horn), vagpafilis@googlemail.com (E. Pafilis), reflect@sven-haag.de (S. Haag), mkuhn@embl.de (M. Kuhn), satagopa@embl.de (V.P. Satagopam), reinhard.schneider@embl-heidelberg.de (R. Schneider), lars.juhl.jensen@cpr.ku.dk (L.J. Jensen).

1570-8268/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.websem.2010.03.003

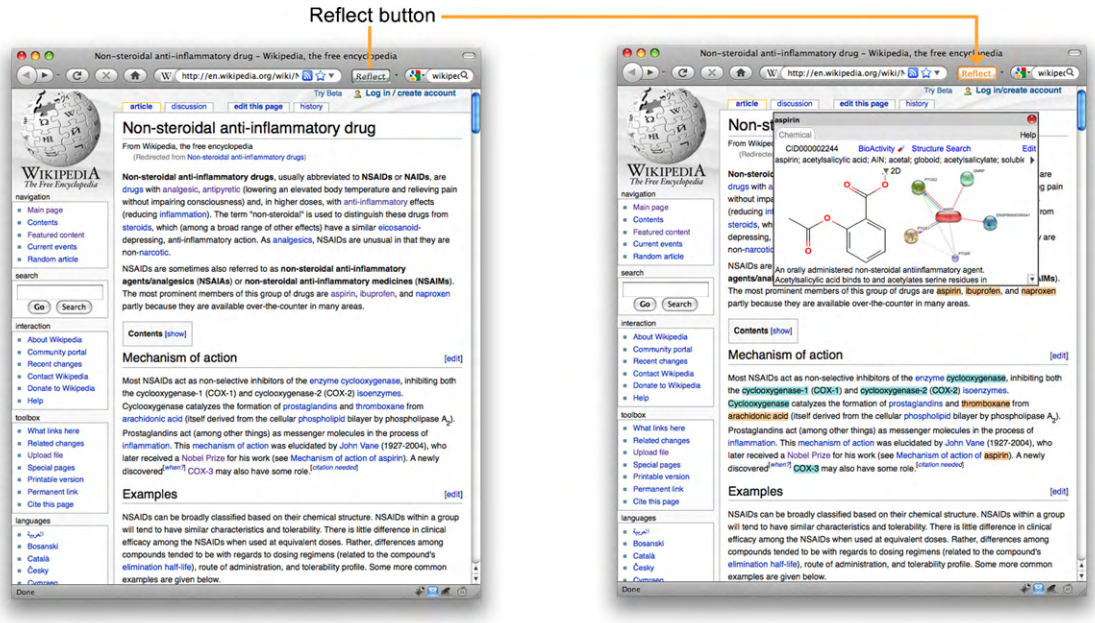

Fig. 1. Example of augmented browsing with Reflect. A web-page is shown before (left) and after (right) it has been augmented or modified by clicking on the Reflect button, a plug-in available for Firefox or Internet Explorer. Reflect tags the names of small molecules, genes, and proteins but otherwise leaves the web-page unchanged. Clicking on a tagged name (e.g., 'aspirin', right image) opens a popup giving access to more detailed information (e.g., the 2D structure of aspirin), without needing to navigate away from the current web-page.

Providing such enhancements to web content is one goal of the 'semantic web' initiative, but this goal remains largely unrealized in spite of very active research [2]. Much of the research in this area has focused on the development of technologies, such as RDF (Resource Description Framework), that are designed to be used primarily server-side by publishers. What options are available for end-users who would like to use semantic enhancements in web-pages they regularly view today? The server-side approach taken by most semantic web developments offers end-users little other than hope that mainstream publishers and service providers will eventually be systematically adopt and apply these methods. However, the slow pace in adopting semantic technologies over the last 10 years suggests that we will be waiting a very long time before all publishers provide systematically tagged content, and further provide popups that can be tailored to each user's requirements. In this work we explore a more practical approach, available and working today, that directly empowers end-users to systematically tag any web-page. This alternative approach rests on two key technologies: augmented browsing, and real-time tagging.

Augmented browsing is an emerging technology that allows end-users to automatically augment or improve the information in web-pages visited while browsing. A popular example of an augmented browsing technology is the Firefox add-on Greasemonkey (http://greasespot.net), which provides a general infrastructure making on-the-fly changes to web-pages. There are a rapidly growing number of such tools, with a wide variety of ways to modify web-pages, changing the page from the way the publisher originally intended. In principle, augmented browsing tools could modify the appearance or content of web-page beyond recognition. In practice, most such tools to date introduce only very minor changes, such as removing advertisement or adding semantic tags to a document, but otherwise leaving the formatting untouched (e.g., Fig. 1). When an end-user chooses to install such a tool, they are effectively personalizing how they view web-pages.

Within the life sciences, several specialized augmented browsing systems have been developed. One such tool, ChemGM [3], tags small molecule names and has popups showing 2D structures; however, tagging is rather slow, taking about 1 min to tag a web-page containing a full-length scientific paper. Another tool, Concept Web Linker (http://tinyurl.com/conceptweblinker) tags a broader range of bio-entities, again requiring about a minute to tag one page. The Concept Web Linker popups show less specific information, and to reach more specific information, such as protein sequences, the user needs to navigate through a series of web-pages, in some cases browsing complex ontologies. A related system, Cohse [4], has even broader scope – it enables users to choose many different ontologies, including outside the life-sciences. Currently, however, the publicly accessible versions of Cohse provide only very limited functionality, and using the life-science ontologies provided does not allow direct navigation to specific information, such as sequences.

Requiring a wait of a minute or more to tag a web-page will discourage many users. To become widely used, we believe that such methods need to achieve a 'real-time' tagging speed, by which we mean the ability to tag a document significantly faster than the time taken to transfer it over the Internet. At this speed, tagging adds only a small delay that end-users are much more likely to accept. In addition to speed, tagging also needs to be accurate: for biochemical entities, the accuracy of automated tagging has recently improved significantly [5], and such methods are now routinely used for a wide variety of text mining applications [6].

We recently published a brief announcement of the Reflect service [7], a new, free community resource that combines real-time tagging with augmented browsing (Fig. 1). Reflect was designed with a strong focus on ease of installation and ease of use. Currently, Reflect tags gene, protein, and small molecule names, and provides popups with summary information designed for biologists and chemists. In the present paper, we describe in detail the methods Reflect uses to implement real-time tagging and augmented browsing. We also describe how the Reflect dictionary is structured, how it can be extended, and how publishers can access Reflect programmatically to provide systematically tagged web content to their subscribers. Finally, we report on end-user usage, adoption, and feedback about Reflect.

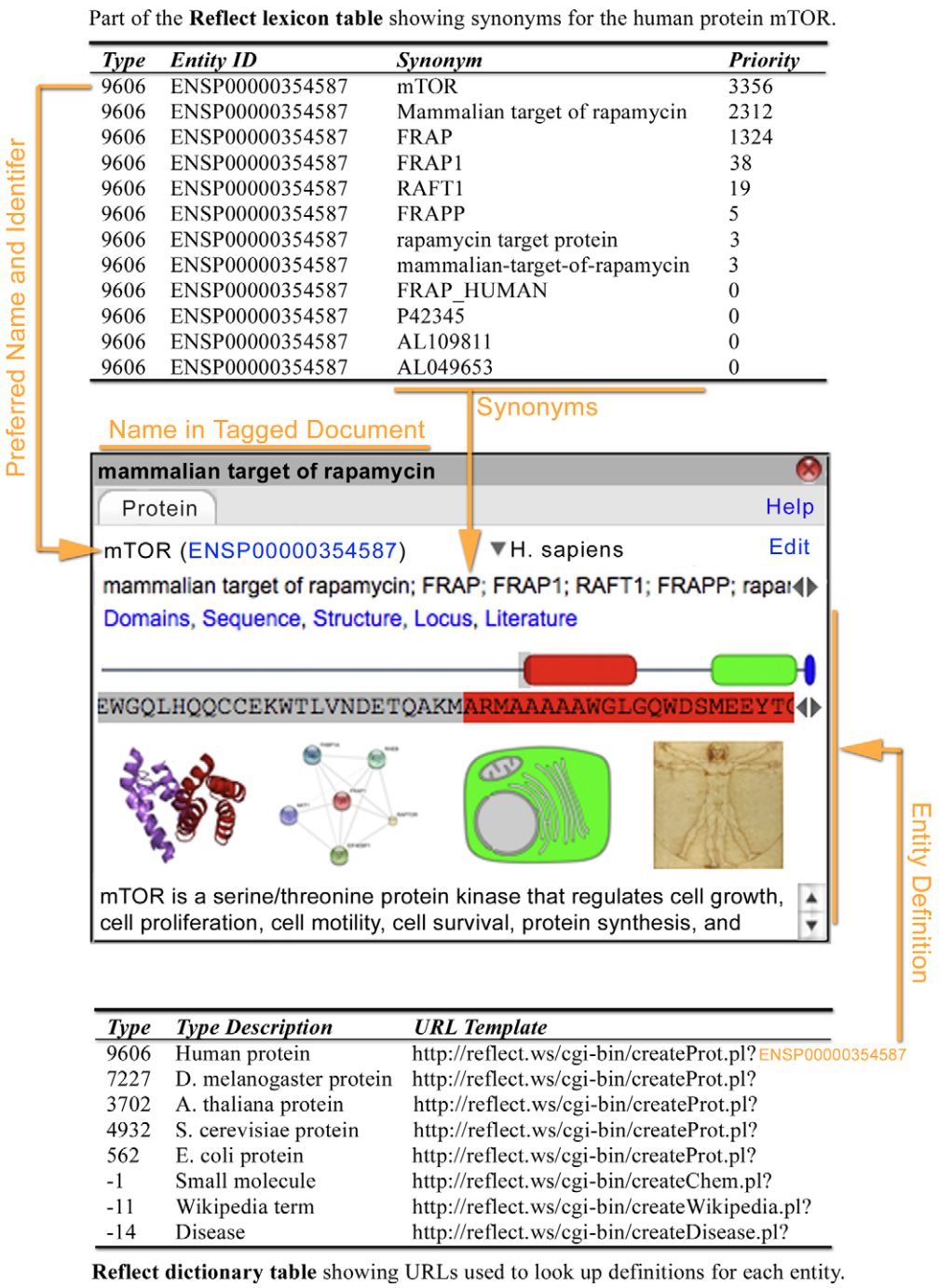

Fig. 2. Reflect lexicon and dictionary. Each type of entity in Reflect has a lexicon (top) that maps each synonym to entity identifiers and assigns a priority. The highest priority indicates the preferred entity name, and the rest are used to rank the popup synonym list (middle). Each entity type also has a dictionary service (bottom) that maps each identifier to a definition in HTML format for display on the entity popup (middle).

2. System and architecture

2.1. Reflect lexicon and dictionary

We created a lexicon of protein and small molecule synonyms by merging the lexicons from the STRING [11] and STITCH [12] databases. The Reflect lexicon currently contains over 2.6 million proteins from 640 organisms and 7.4 million small molecules. The lexicon lists all synonyms for each entity, and maps them to a unique entity identifier. We re-used existing identifiers from source databases, e.g., PubChem identifiers [8] for small molecules, Ensembl identifiers [9] for human proteins, FlyBase identifiers [10] for fly proteins, SGD identifiers [11] for budding yeast proteins, TAIR identifiers [12] for *Arabidopsis* proteins, and RefSeq identifiers [13] for prokaryotic proteins. Both proteins and small molecules have equivalent entries in several databases. In the Reflect lexicon, we included identifiers from a range of these databases as additional synonyms. The lexicon of synonyms was then expanded even further to include orthographic variants of each synonym, e.g., hyphenation characters were replaced with space characters, and visa versa. To enable fast tagging, the expanded lexicon was then loaded into a Perl hash table, with synonyms as keys, and

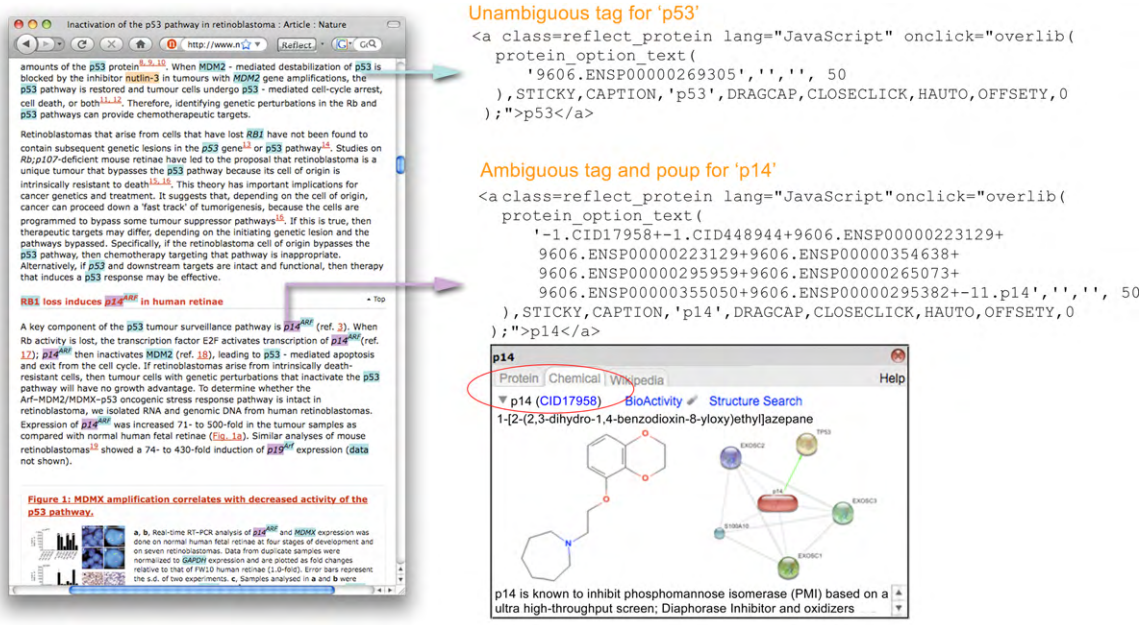

Fig. 3. Reflect tags, popups, and ambiguity. The HTML code shown is used by Reflect to launch the popup. Each tag lists all matching entities in the Reflect dictionary, specified by the entity type (e.g., 9606, indicating a human protein) plus the entity identifier (e.g., ENSP00000269305, indicating the human protein p53). In this case 'p53' matches unambiguously to a single protein, while 'p14' matches several protein, small molecules, and to a Wikipedia entry (only available on the beta server), indicated on the popup by multiple tabs (as highlighted). In addition, 'p14' matches two distinct small molecules, indicated on the popup by a drop-down menu (as highlighted).

entity identifiers as values. This hash currently requires 44 GB of random-access memory. Fig. 2 shows the structure of the lexicon, and illustrates how each entity in the lexicon is connected to a 'dictionary' web-service that delivers a definition of the entity in HTML format that can be displayed directly on the popup (Section 2.4). Adding further entity type is straightforward, requiring only a lexicon and a dictionary service with the same structure.

2.2. Tagging service

Tagging requests are managed by a daemon thread with a single hash containing the complete lexicon. The daemon can accept documents in HTML or text format, and also a URL, in which case the document is fetched by the Reflect server. The daemon does a two-pass scan of each document, first to find organism names (needed to map protein names to a specific protein entity), then a second pass to tag all terms in the document that match entries in the Reflect lexicon. Leftmost-longest-matching is used for up to five words, testing each combination against the lexicon hash. Recognized entity names that occur in the text portion of the HTML are then substituted with tags that, upon click or mouse-over events, call a JavaScript function to generate the summary popups (Fig. 3). Reflect does not change existing HTML tags or attributes, hence it preserves the original document layout. When the document is returned to the user's browser, the only noticeable difference is that entity names are now highlighted.

2.3. User interfaces

As described previously [7], we constructed two kinds of end-user interfaces to the Reflect tagging service: one is a web-page that allows the user to enter a URL and view the 'reflected' page in an iframe. The second kind of interface is via plug-ins built both for Firefox and Internet Explorer; these plug-ins use XML-based User Interface Language and Document Object Model events to tag entities in a web-page in a user's browser without changing the overall document layout or the apparent URL. Communication between browser and the Reflect server occurs via XMLHttpRequest objects.

When the user clicks on the tag of a recognized entity, a small popup window appears via the overLib JavaScript library (http://tinyurl.com/overlib). The popup is then populated with detailed content supplied mostly by CGIs on the Reflect server. For proteins and genes, the popup shows a list of synonyms from the Reflect lexicon, omitting database identifiers and trivial orthographic variations. For proteins, the popup also shows the complete amino acid sequence, the domain structure from SMART [14], an image showing the five most significant interaction partners from STITCH [15], the best matching 3D structure from PDBsum [16], a visualization of subcellular location, and an image of the organism taken from iTOL [17]. Clicking on most of these features opens a new browser window or tab showing more detailed information. Similarly, clicking on 'Locus' opens the corresponding gene entry, and clicking on 'Literature' opens all related Medline abstracts in iHOP [1]. Dragging the mouse on the domain graphical view scrolls through the sequence, and hovering over a domain shows its name. For small molecules, the popup shows the 2D structure from PubChem [8], and an image of the five most significant interactions from STITCH. The summary popup is the primary user interface in Reflect and considerable effort was spent to ensure that the popup provides a useful summary of the frequently needed information presented in an intuitive and easy-to-use fashion, while using a minimal screen space.

2.4. Data privacy

The Reflect server maintains a standard Apache log of IP access information, and in the future we plan to improve the service using information derived from these logs and from the URLs of 'reflected' documents. However we do not use or keep the content of tagged documents. When using the plug-in and the API, document tagging takes place only in random-access memory, so document content is never written to disk. Currently, Reflect does not support HTTPS, although we plan to add this later.

3. Implementation and results

3.1. Tagging speed

The current Reflect sever can tag a full-length scientific paper of 10,000 words in about 0.3 s. A more typically sized web document, say 550 words, takes about 75 ms. Tagging is almost always faster than transferring a document to and from the Reflect server – generally pages can be tagged and returned within a few seconds via standard broadband. The tagging speed is determined by hash lookup time, and so it is independent of dictionary size.

3.2. User interfaces

Reflect can be used directly from http://reflect.ws by simply typing or pasting a URL into the text input box on that web-page and pressing the 'Reflect' button. The Reflect server then retrieves the HTML document, tags it, and returns a tagged version to the user's browser. Note that this will only work for URLs that are publicly accessible.

A more convenient way to use Reflect is to install it as a plug-in into Firefox or Internet Explorer. The plug-in adds a button to the user's browser: pressing this button sends the currently viewed HTML document to the Reflect server, where it is tagged and returned. Thus, with the plug-in, users can 'reflect' any web-page that they can access. The Firefox plug-in provides an option to automatically tag all web-pages viewed, effectively enabling semantic annotation for the whole web.

Currently, Reflect tags genes, proteins, and small molecule names. Clicking on a tagged name opens a popup showing a concise summary of information about the given small molecule (Fig. 3, bottom right) or protein (Fig. 2, middle), as well as listing other synonyms. When a tagged name is ambiguous, the popup shows all found matches and allows the user to disambiguate the name by choosing which of the possibilities is most appropriate (Fig. 3). Currently, three levels of ambiguity are shown: first, a name may match both a protein and a small molecule, in which case Reflect shows both possibilities on separate tabs. Secondly, a name may match to several genes within the same organism, in which case Reflect shows all matching genes in a pull-down menu. Thirdly, for gene and protein names it is often ambiguous which organism is intended in the HTML document; Reflect shows a list of possible organisms, derived from the default organism (initially set to human, can be changed using the Firefox plug-in) plus organisms mentioned in the document. In the near future, we plan to show a fourth level of ambiguity, where users will be able to select splice variants for each gene.

3.3. JavaScript interface

We have implemented several programmatic interfaces to Reflect: the simplest of these lets publishers add a Reflect button directly to any web-page, simply by adding a JavaScript library and one line of HTML. When the end-user presses this button the web-page is replaced by a reflected version. This works only for web-pages that are publicly accessible. Below is an HTML page that implements this method (see also http://reflect.ws/reflect_by_javascript_example.html):

```
<html>
  <head>
    <script
      type="text/javascript"
      src="http://reflect.ws/script/reflect.js">
    </script>
  </head>
  <body>
     Press button to reflect protein and chemical names, e.g., p53 and estrogen.
     <input
        type="image" value="Reflect" alt="Reflect" title="Reflect this page"
        src="http://reflect.ws/images/reflect_22px_off.png"
        onclick="Reflect.reflectByJS();"
     />
  </body>
</html>
```

3.4. Proxy interface

Publishers wishing to add a Reflect button to web-pages that are not publicly accessible can do so by installing a simple Reflect proxy (e.g., see http://reflect.ws/reflect_by_proxy.cgi). When the end-user presses this button, the page is sent to the proxy, which contacts the Reflect API and returns a reflected version of the page. The proxy must be hosted on the same server as the web-page. The absolute or relative path name of the proxy can be specified as a parameter to the 'reflectByProxy' script. Below is an HTML page that implements this method (see also http://reflect.ws/reflect_by_proxy_example.html):

```
<html>
  <head>
    <script
      type="text/javascript"
      src="http://reflect.ws/script/reflect.js">
    </script>
  </head>
  <body>
    Press button to reflect protein and chemical names, e.g., p53 and estrogen.
    <input
      type="image" value=" Reflect" alt="Reflect" title="Reflect this page"
      src="http://reflect.ws/images/reflect_22px_off.png"
      onclick="Reflect.reflectByProxy('/cgi-bin/reflect_by_proxy.cgi');"
    />
  </body>
</html>
```

3.5. Reflect API

The Reflect API allows more precise control of how a document is tagged. The API can be accessed via SOAP (http://reflect.ws/SOAP_API.html) and also via REST (http://reflect.ws/REST_API.html) using HTTP 'post'. Below is a Perl example that uses HTTP 'post' to tag small molecule and protein names in a sample HTML document:

```
#!/usr/bin/perl

use LWP::Simple::Post qw(post post_xml);

$input = "<html><head></head><body>p53 actin</body></html>";

$response = post(
     'http://reflect.ws/REST/GetHTML',
     "document=$input"
);

print $response;
```

3.6. Usage and feedback

We announced the launch of the Reflect service at various seminars in 2009, and in a published announcement in June 2009 [7]. By October 2009, the Reflect plug-in had been downloaded over 30,000 times, and several organizations have begun accessing Reflect programmatically to tag text corpora. The average server load was over 3000 documents tagged per day.

We have also collected considerable qualitative feedback from end-users; they frequently told us that they are impressed with the ease of use, and that they find the information and hyperlinks in the popups to be very useful. Many end-users commented specifically that the synonyms list on the popup was especially useful. In a commonly reported scenario, an end-user would open the Reflect popup for an unfamiliar protein name, only to discover, from reading the synonym list, that the protein was one they already knew by a different name. Reflect helped these users see this connection, and thus understand the document, significantly faster than they would have done otherwise.

Several end-users also commented specifically on the usefulness of the information on the protein popup about amino acid sequence and domains. These users reported that, while reading the latest literature, they often used Reflect to look up proteins mentioned in documents. From the information in the popup, they could rapidly decide if a given protein was potentially interesting or not for their research, and if it was, they copied part of the sequence and domain information on popup and used it directly for the next step in their analysis pipeline. For these users, Reflect greatly accelerated part of their daily workflow.

By far the most common negative feedback concerned the rate of false positive and negative tags, which end-users reported were sometimes confusing and frustrating. This is a well-known issue that invariably arises with methods that automatically recognize named entities in text. Based on this feedback, we have given top priority to improving tagging accuracy in future versions of Reflect (see Section 4).

4. Discussion

4.1. Growth in usage

The number of Reflect plug-in downloads has increased continuously since we launched the Reflect service, and even more rapidly since our first publication about Reflect appeared recently. Part of this growth in usage can be accounted for by presentations that we have given about Reflect. However, the total number of plug-in downloads prior to the first published announcement [7] was over 10,000, a much larger number than the cumulative audience at our presentations. This suggests that Reflect usage has grown largely by word-of-mouth, i.e., scientists recommend Reflect to their peers.

Part of this 'viral' growth pattern can be attributed to our decision to design Reflect to be fast and simple to install and to use. However, we believe another significant factor is that the benefit Reflect brings is obvious and easily communicated. The rapid growth in usage also implies that Reflect is addressing needs that are currently unmet for many scientists: based on user feedback, the principal needs met by Reflect were the ability to easily go from an entity name in a web-page to a list of synonyms, as well as to other specific information about the entity (e.g., the amino acid sequence and domain structure of a protein). User feedback indicates that Reflect can meet these needs in a way that significantly improves the daily workflow of many life scientists, removing several manual steps they would otherwise repeat many times each day.

Encouraged by these results, we are planning to extend Reflect by adding further entity types such as disease, cell lines, and mutations. We further plan to extend Reflect beyond the life science, e.g., incorporating a wide selection of terms from Wikipedia. Extending the lexicon will not slow down tagging, since the hash lookup speed is independent of hash size.

4.2. Implications for web semantics

Reflect adds semantic information to web-pages, although in a different manner to traditional semantic web approaches like RDF. These traditional approaches add rich and powerful semantic-based capabilities, but require considerable re-engineering of servers and content databases, and hence are currently used in only a tiny fraction of all web content. In contrast to these 'depth-first' approaches, Reflect is 'breadth-first', providing semantic annotations that may be less powerful, but are available today, systematically applied for any web content.

In addition to breadth-first coverage, approaches such as Reflect have another advantage in that their strong end-user focus enables them to directly address diversity of end-user requirements for semantic annotation. For example, as discussed in Section 1, the Reflect small molecule popup may be useful for chemist, while a non-scientist would probably prefer to access only a short text description. Two chemists might differ in the specific data they wish to see on the popup. An end-user interested in the stock market may want to go from company names to financial details, whereas other users may want only a brief description of the company. To summarize, we could say that semantics are in the eyes of the end-user.

The traditional semantic web approach assumes that adding semantic capabilities is the responsibility of publishers and content providers. A key difficulty with this approach is that it requires publishers to anticipate the many, diverse ways that end-users would like to use their content. In contrast, the popularity

of Reflect and of social bookmarking services such as Delicious (http://delicious.com) demonstrate the usefulness and feasibility of semantic annotations initiated by end-users. We believe that many similar tools will be developed in the near future, some tailored to specific end-user interests and requirements. In addition, it is likely that individual tools will themselves increasingly allow customization in how they augment web-pages, e.g., the Reflect popups are currently not customizable, but we plan to add such capabilities. Overall, such tools increasingly will allow end-users to choose and personalize how they view web-pages.

At the same time as real-time tagging and augmented browsing technologies will increase, representing a new direction for semantic web technologies, traditional sever-side semantic annotation is also likely to increase. In fact, these two approaches can be synergistic, for example the Reflect API provides a simple system that allows life science publishers to deliver pre-tagged content directly to end-users. In the near future, both sever-side and end-user initiated semantic annotation have an increasing role, and eventually will probably interact. Regarding how these interactions would be structured, it is probably too early to do more than speculate.

4.3. Future improvements

Feedback from users of Reflect indicated that its main perceived weakness is the current rate of false positive and negative tags. One possible strategy for improving tagging accuracy would be to use more sophisticated methods for recognizing entity names, e.g., natural language processing and machine learning. Such methods have been the subject of intense research efforts that has lead to significant improvements in accuracy [5]. However, when we compared the recall and precision of Reflect's tagging of protein names with a range of such methods [7], we found that Reflect had median or better performance. Moreover, these more sophisticated methods are generally far too slow for real-time tagging.

We are current working on an alternative strategy that will enable users to manually correct both false positive and false negative tags by directly updating the Reflect dictionary. This feature will enable specific terms used within a document to be semantically annotated by the user community, in contrast to systems such Delicious that allow only the entire document to be annotated. Similar approaches based on collaborative content-editing have been successfully used in the life sciences (e.g., Gene Ontology [18], see http://tinyurl.com/go-edit) and are likely to increase.

In the near future we also plan to include Wikipedia terms in the dictionary, thus broadening the scope of Reflect beyond the life sciences.

Reflect was designed to help end-users browsing the web by tagging HTML pages, however it can also be used with other document types, e.g., Microsoft Office documents or PDF, by first converting to HTML then 'reflecting'. Conversion can be often be done by using a 'Save As...' command, or by dedicated document converters. A recently developed extension to Reflect called OnTheFly [19] streamlines this process, automatically converting MS Office and PDF documents to HTML, 'reflecting' the HTML documents, and returning the tagged HTML documents to the end-user. In the future we plan to integrate these document conversion services into the main Reflect server.

4.4. Conclusions and perspectives

Reflect is a publicly funded, free service for the scientific community. In its present form, Reflect is a useful tool for life scientists, helping them interpret, visualize, and connect knowledge during their daily work. We plan to extend the scope of Reflect considerably, and we welcome collaboration proposals for adding further entity types, as well as proposals from publishers and data providers interested in programmatic access to Reflect. The evident popularity of Reflect demonstrates the feasibly of real-time semantic tagging and of allowing end-users to choose how to semantic annotate their web content. This, in turn, suggests a new direction for web semantics in the future.

Acknowledgements

The work was partly funded by the European Molecular Biology Laboratory, by the European Union Framework Programme 6 grant 'TAMAHUD' (LSHC-CT-2007-037472, in part), and by the Novo Nordisk Foundation Center for Protein Research.

References

[1] R. Hoffmann, A. Valencia, A gene network for navigating the literature, Nat. Genet. 36 (7) (2004) 664.
[2] N. Shadbolt, T. Berners-Lee, W. Hall, The semantic web revisited, IEEE Intell. Syst. 21 (3) (2006) 96–101.
[3] E.L. Willighagen, N.M. O'Boyle, H. Gopalakrishnan, D. Jiao, R. Guha, C. Steinbeck, D.J. Wild, Userscripts for the life sciences, BMC Bioinformatics 8 (2007) 487.
[4] S.K. Bechhofer, R.D. Stevens, P.W. Lord, Ontology driven dynamic linking of biology resources, Pac. Symp. Biocomput. (2005) 79–90.
[5] L. Smith, L.K. Tanabe, R.J. Ando, C.J. Kuo, I.F. Chung, C.N. Hsu, Y.S. Lin, R. Klinger, C.M. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C.A. Struble, R.J. Povinelli, A. Vlachos, W.A. Baumgartner Jr., L. Hunter, B. Carpenter, R.T. Tsai, H.J. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans, C. Blaschke, R. Torres, M. Neves, P. Nakov, A. Divoli, M. Mana-Lopez, J. Mata, W.J. Wilbur, Overview of BioCreative II gene mention recognition, Genome Biol. 9 (Suppl. 2) (2008) S2.
[6] M. Krallinger, A. Valencia, L. Hirschman, Linking genes to literature: text mining, information extraction, and retrieval applications for biology, Genome Biol. 9 (Suppl. 2) (2008) S8.
[7] E. Pafilis, S.I. O'Donoghue, L.J. Jensen, M. Kuhn, N.P. Brown, R. Schneider, Reflect: augmented browsing for the life scientist, Nat. Biotechnol. 27 (6) (2009) 308–310.
[8] D.L. Wheeler, T. Barrett, D.A. Benson, S.H. Bryant, K. Canese, V. Chetvernin, D.M. Church, M. Dicuccio, R. Edgar, S. Federhen, M. Feolo, L.Y. Geer, W. Helmberg, Y. Kapustin, O. Khovayko, D. Landsman, D.J. Lipman, T.L. Madden, D.R. Maglott, V. Miller, J. Ostell, K.D. Pruitt, G.D. Schuler, M. Shumway, E. Sequeira, S.T. Sherry, K. Sirotkin, A. Souvorov, G. Starchenko, R.L. Tatusov, T.A. Tatusova, L. Wagner, E. Yaschenko, Database resources of the National Center for Biotechnology Information, Nucleic Acids Res. 36 (Database Issue) (2008) D13–D21.
[9] E. Birney, D. Andrews, M. Caccamo, Y. Chen, L. Clarke, G. Coates, T. Cox, F. Cunningham, V. Curwen, T. Cutts, T. Down, R. Durbin, X.M. Fernandez-Suarez, P. Flicek, S. Graf, M. Hammond, J. Herrero, K. Howe, V. Iyer, K. Jekosch, A. Kahari, A. Kasprzyk, D. Keefe, F. Kokocinski, E. Kulesha, D. London, I. Longden, C. Melsopp, P. Meidl, B. Overduin, A. Parker, G. Proctor, A. Prlic, M. Rae, D. Rios, S. Redmond, M. Schuster, I. Sealy, S. Searle, J. Severin, G. Slater, D. Smedley, J. Smith, A. Stabenau, J. Stalker, S. Trevanion, A. Ureta-Vidal, J. Vogel, S. White, C. Woodwark, T.J. Hubbard, Ensembl 2006, Nucleic Acids Res. 34 (Database Issue) (2006) D556–D561.
[10] S. Tweedie, M. Ashburner, K. Falls, P. Leyland, P. McQuilton, S. Marygold, G. Millburn, D. Osumi-Sutherland, A. Schroeder, R. Seal, H. Zhang, FlyBase: enhancing *Drosophila* gene ontology annotations, Nucleic Acids Res. 37 (Database Issue) (2009) D555–D559.
[11] J.M. Cherry, C. Ball, S. Weng, G. Juvik, R. Schmidt, C. Adler, B. Dunn, S. Dwight, L. Riles, R.K. Mortimer, D. Botstein, Genetic and physical maps of *Saccharomyces cerevisiae*, Nature 387 (Suppl. (6632)) (1997) 67–73.
[12] D. Swarbreck, C. Wilks, P. Lamesch, T.Z. Berardini, M. Garcia-Hernandez, H. Foerster, D. Li, T. Meyer, R. Muller, L. Ploetz, A. Radenbaugh, S. Singh, V. Swing, C. Tissier, P. Zhang, E. Huala, The Arabidopsis Information Resource (TAIR): gene structure and function annotation, Nucleic Acids Res. 36 (Database Issue) (2008) D1009–1014.
[13] K.D. Pruitt, T. Tatusova, D.R. Maglott, NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins, Nucleic Acids Res. 35 (Database Issue) (2007) D61–D65.
[14] I. Letunic, R.R. Copley, B. Pils, S. Pinkert, J. Schultz, P. Bork, SMART 5: domains in the context of genomes and networks, Nucleic Acids Res. 34 (Database Issue) (2006) D257–D260.
[15] M. Kuhn, C. von Mering, M. Campillos, L.J. Jensen, P. Bork, STITCH: interaction networks of chemicals and proteins, Nucleic Acids Res. 36 (Database Issue) (2008) D684–688.
[16] R.A. Laskowski, PDBsum: summaries and analyses of PDB structures, Nucleic Acids Res. 29 (1) (2001) 221–222.
[17] I. Letunic, P. Bork, Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation, Bioinformatics 23 (1) (2007) 127–128.
[18] M.A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G.M. Rubin, J.A. Blake, C. Bult, M. Dolan, H. Drabkin, J.T. Eppig, D.P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J.M. Cherry, K.R. Christie, M.C. Costanzo, S.S. Dwight, S. Engel, D.G. Fisk, J.E.

Hirschman, E.L. Hong, R.S. Nash, A. Sethuraman, C.L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi, S.Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E.M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, N. de la Cruz, P. Tonellato, P. Jaiswal, R. White, The Gene Ontology (GO) database and informatics resource, Nucleic Acids Res. 32 (Database Issue) (2004) D258–D261.

[19] G.A. Pavlopoulos, E. Pafilis, M. Kuhn, S.D. Hooper, R. Schneider, OnTheFly: a tool for automated document-based text annotation, data linking and network generation, Bioinformatics 25 (7) (2009) 977–978.