

# Falconer: Once SIOC Meets Semantic Search Engine\*

Gang Wu  
School of Computer Science  
and Engineering  
Southeast University  
Nanjing 210096, P.R.China  
wu@seu.edu.cn

Mengdong Yang  
School of Computer Science  
and Engineering  
Southeast University  
Nanjing 210096, P.R.China  
mdyang@seu.edu.cn

Ke Wu  
School of Computer Science  
and Engineering  
Southeast University  
Nanjing 210096, P.R.China  
wuke09@gmail.com

Guilin Qi  
School of Computer Science  
and Engineering  
Southeast University  
Nanjing 210096, P.R.China  
gqi@seu.edu.cn

Yuzhong Qu  
State Key Laboratory for Novel  
Software Technology  
Nanjing University  
Nanjing 210093, P.R.China  
yzqu@nju.edu.cn

## ABSTRACT

Falconer<sup>1</sup> is a semantic Web search engine enhanced SIOC (Semantically-Interlinked Online Communities) application, which is designed to demonstrate the ability of accelerating the creation and reuse process of semantic Web data with easy-to-use user interfaces. In this process, semantic Web search engines feed existing semantic data into the SIOC framework, where new semantic data are composed by the community and indexed again by those search engines. Compared to existing social (semantic) Web applications, Falconer inherently conforms to SIOC specification. It provides semantic search engine based user registration suggestion, friends auto-discovery, and semantic annotation for forum post content. Another distinctive feature is that it enables users to subscribe any resource having a URI as the topic they are interested in. The relationships among users, topics, and posts are further visualized for analyzing the topic trends in the community. As all semantic data are formatted in RDF and RDFa, they can be queried with SPARQL query language.

## Categories and Subject Descriptors

H.m [Information Systems]: Miscellaneous

## General Terms

Design

## Keywords

SIOC, semantic search engine, social semantic Web

## 1. INTRODUCTION

As announced by the Linking Open Data community project, there already exist billions of semantic data available on the Web. However, it is hard for common users to realize how

\*This is supported by the National Natural Science Foundation of China under Grant No.60903010 and the Natural Science Foundation of Jiangsu Province under Grant No.BK2009268

<sup>1</sup><http://iws.seu.edu.cn/services/Falconer/>

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.  
ACM 978-1-60558-799-8/10/04.

to utilize these semantic data. Today, the semantic Web is placed in such an awkward position.

In fact, what are familiar to common users are the social Web applications, like Twitter, Facebook, YouTube, Wikipedia, MySpace, etc [8]. The social Web is increasingly becoming an ‘authority’ traffic source on the internet<sup>2</sup>. The main reason is that the social Web magically arouses people’s passion on creating and sharing information through the Web. Furthermore, easy-to-use user interfaces remove the technical barriers from ordinary people. However, machines are unable to understand the large scale data produced on the social Web for lacking of semantic data.

Obviously, the social semantic Web, which combines the intelligence from the semantic Web and the popularity from the social Web [4], could be a solution. Typical applications include semantic wikis semantic blogs and semantic tagging and folksonomies.

Although much effort has been made in social semantic Web, there is still at least one problem unsolved: making fully and freely use of existing large scale linked data on the semantic Web. Semantic data used in most current semantic wikis, blogs, and tagging systems are usually restricted to specific pre-existing ontologies or data sources, or user generated ontologies. Take Freebase<sup>3</sup> as an example, available data sources besides those contributed by community collaboration are only Wikipedia, Chemoz, NNDB, and MusicBrainz. It is challenging to promote the utilization of existing large scale semantic data in the social semantic Web.

In this work, we design and implement the Falconer, a semantic Web search engine enhanced SIOC<sup>4</sup> application. SIOC stands for Semantic Interlinked Online Communities. It is an ontology for describing data from online community sites, like blogs, forums, bbs, mailing lists and newsgroups [3]. SIOC makes it easier to construct a social semantic Web application. Falconer is a demo forum site inherently conforming to the SIOC ontology. In order to help common users play freely and fully with any reachable information on

<sup>2</sup>[http://weblogs.hitwise.com/sandra-hanchard/2009/09/government\\_sites\\_receive\\_more.html](http://weblogs.hitwise.com/sandra-hanchard/2009/09/government_sites_receive_more.html)

<sup>3</sup><http://www.freebase.com>, a collaborative knowledge base powered by semantic Web technologies.

<sup>4</sup><http://rdfs.org/sioc/spec>

the semantic Web, we employ a semantic Web search engine to feed the semantic data. Semantic Web search engines “allow more expressive queries over information integrated from multiple sources, and return specific information about entities, for example people, locations, news items” [5]. The specific semantic search engine we choose is Falcons<sup>5</sup> [7]. The most distinctive feature of Falcons is the query-dependent snippet for each result object, which may directly answer the user’s question [7]. Falcons also provides a set of REST API for developers to access its search services, which can greatly simplify the implementation of Falconer. Hence, the meaning of the name, “Falconer”, can be explained as a “Falcons enhanced republic”.

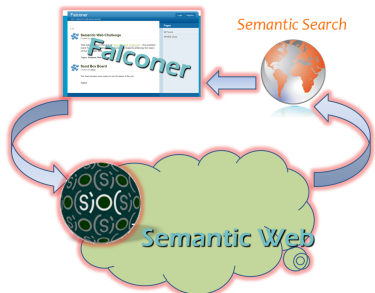


Figure 1: Semantic data creation and reuse

In Falconer, the creation and reuse of semantic data form an ecosystem as shown in Figure 1. On the one hand, under the framework of SIOC, newly created resources, e.g. users, posts, and topics, are formatted in RDF and hence indexable for Falcons. On the other hand, users can freely and easily refer to any resources retrievable from Falcons in the content of the posts. The cycle of creation and reuse of semantic Web data is made easier in this way. The general idea of Falconer can be applied to any social semantic Web applications.

## 2. FEATURES OF FALCONER

### 2.1 Semantic Search Engine based Features

#### 2.1.1 User Registration Suggestion

As the Friend of a Friend (FOAF) project states, FOAF makes people conveniently share, transfer, extend, merge, and reuse their profile information, relationships and activities on the Web. That means it is reasonable and feasible to identify a person in the physical world with a FOAF description on the Web. However, such machine-readable information has not produced satisfactory effect. For example, we are still suffering from the boring and repetitive process of providing similar personal information during the registration on every social website. In Falconer, we provide a simple solution with the help of Falcons search engine which has indexed FOAF documents crawled on the semantic Web.

As shown in Figure 2, there is a “Search” button near the “Name” field. When a user fills the field and press the button, it will trigger a search action through Falcons (shown at left bottom of Figure 2). Once the user chooses a URI referring her/his FOAF from the returned result, it will fill the registration form automatically (shown at right top of Figure 2). If the user submits the registration form, it will generate a new `sio:User` associating with the FOAF URI through the `sio:account_of` property. If nothing relevant found, the user

can still get a FOAF profile together with the newly generated `sio:User` account just by filling the form manually. In this way, Falconer alleviates the registration by utilizing FOAF information retrieved from Falcons.

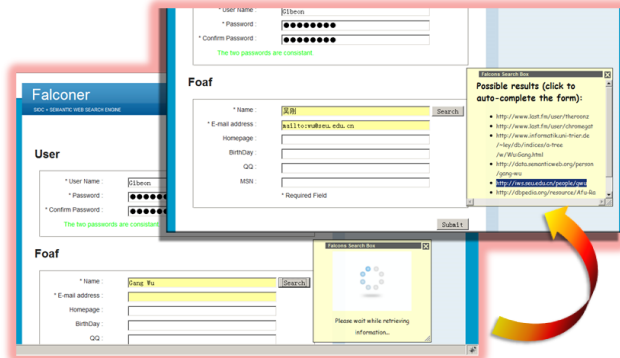


Figure 2: User Registration Suggestion

#### 2.1.2 Friends Auto-discovery

Since the relationships between people are also embedded in FOAF, we can automatically discover the registered friends of a user following his corresponding FOAF Person resource’s `foaf:knows` property. As much as we know, such friends auto-discovery feature is not available for known social Web applications. In those applications, users have to provide private information like email addresses or IM accounts and corresponding passwords in order to find out their friends. This feature is integrated in the profile page of a user where her/his friends will be listed automatically. Note that the discovery is unidirectional according to the characteristics of `foaf:knows`. It means that user *A* may claim that she/he knows user *B*, while *B* may not accept the relationship.

Falconer provides a scheme to help users create new friendships within the system, if they are not provided directly in users’ FOAF. After login, a user can make any other users as her/his friend by clicking the image hyperlink “✓ I KNOW THIS PERSON”. Friends added in this way will be listed in the user’s profile page together with those friends got from `foaf:knows` as well because Falconer has already created the `foaf:knows` between the user and her/his friend.

#### 2.1.3 Semantic Annotation Tool

Tagging is a typical characteristics of today’s social Web applications. The advantage of a tagging system is that it is more flexible to classify items compared with hierarchical classification systems using controlled vocabulary. However, the tag is usually represented in keywords to which people may assign quite different semantics. Hence, phenomena of homonyms and synonyms are very common in tagging systems. Since the semantic Web is proposed to explicitly define the meaning (semantics) of information and services on the Web, annotating keywords with semantic Web techniques may solve the ambiguity to a certain extent. Although semantic annotation has been well studied, Falconer is different from the existing semantic annotation tools used in other social semantic Web applications.

First, with the help of Falcons, users can freely and fully make use of billions of existing semantic data to annotate their own information. As a demo forum site, Falconer provides users space to post their discussions. Users can an-

<sup>5</sup><http://iws.seu.edu.cn/services/falcons>

notate any keywords in the content part of the post with a URI. The URI is generated by sending the keywords to Falcons as a query, and then selecting from the query results returned. Therefore, there is no limitation to the semantic annotation except the scale of semantic data indexed by Falcons and the length of users' post.

Second, the process of annotation is more operable for common users. No prior knowledge of semantic Web is required. This is supported by the powerful query-dependent snippet feature of Falcons. Each search result object is described with a snippet, which may directly answer the user's question in mind. There are only three steps for the annotation: 1) Select the keywords; 2) Click to search with Falcons; 3) Select a proper URI. After the above three steps, the annotation will process automatically in the background and not interfere the post editing process in the foreground.

The semantic annotation is optional. Users can ignore it, and use Falconer as a general forum. The main difference between Falconer and semantic wikis is that Falconer focuses on instance data rather than creating concepts.

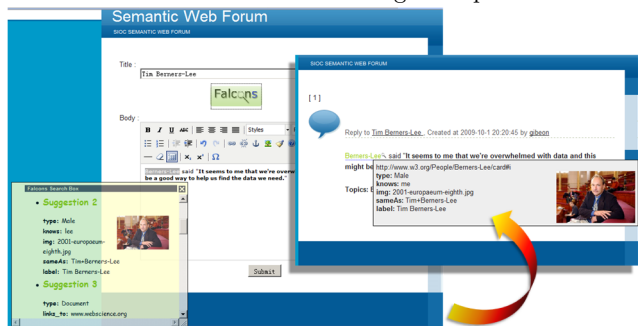


Figure 3: Semantic Annotation Tool

In Figure 3, we present an example where a post about the conversation<sup>6</sup> with Tim Berners-Lee in 2007 was created. In the content part, we selected 'Berners-Lee' to annotate, because we thought the topic of the post is mainly about him. A small notice window popped after we clicked the button "Falcons" in which all search results were listed. We selected the second suggestion as shown in the left scene, and then we got the right scene after submitting the post and reviewing it. A gray color snippet window will show if we move mouse over the annotation.

## 2.2 URI Subscription

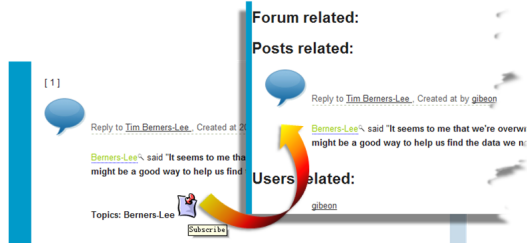


Figure 4: Post Topics

In Falconer, each annotation is automatically taken as a topic of the post. That means there is a `sioctopic` property linking from the post to the URI of the annotation. In Figure 4, we can find that the topics related to the post are listed just under the content part (shown in left bottom).

<sup>6</sup>[http://www.businessweek.com/technology/content/apr2007/tc20070409\\_961951.htm](http://www.businessweek.com/technology/content/apr2007/tc20070409_961951.htm)

There is a "Subscribe" icon near each topic. The purpose of the "Subscribe" is to provide users a method for watching the updates of a topic including "Who subscribed the topic" and "Which posts or forums also have the topic". It is like the idea of Web feed which enables subscribed users to get newly updated content from a URL to HTML pages or other kinds of media. The difference is that users of Falconer subscribe a general URI. Once a topic is subscribed, it will be listed in "My Topics". Detailed information related to the topic will be shown as the right top of Figure 4 if the topic is further selected from "My Topics".

Now, we can exemplify the ability of Falconer to solve the ambiguity problems in annotation. Suppose in a post about "American Tiger", we have annotated "Jaguar" with URI <http://dbpedia.org/resource/Jaguar> whose type is "Mammal101861778". And in another post about "Jaguar Cars", we have annotated the same words "Jaguar", but with different URI [http://dbpedia.org/resource/Jaguar\\_Cars](http://dbpedia.org/resource/Jaguar_Cars) whose type is "Company108058098". Obviously this is a homonym problem. Though the same keyword "Jaguar" is used, they are different items. In figure 5, there are two topics named "Jaguar" listed which lead to different topic information pages. For synonyms disambiguation, we use another example about "Apple mobile" whose URI is <http://dbpedia.org/resource/IPhone>. Suppose it is annotated with keyword "iPhone" in one post, and in "Apple mobile" in another post. We can only find one item about "Apple mobile" in the list, rather than two items one for "iPhone" and the other for "Apple mobile". When exploring the detail of the item, we can find the two posts that use different keywords.

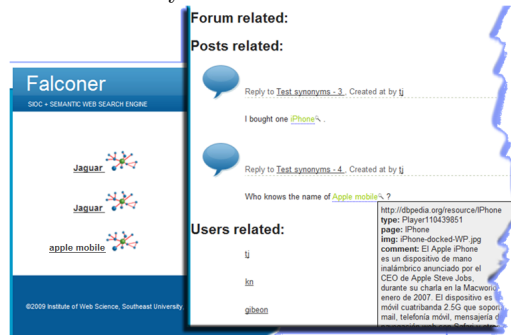


Figure 5: Annotation Disambiguation

## 2.3 SIOC Relation Visualization

Falconer is based on the SIOC Core Ontology which defines standard online community concepts, such as Post, User, Forum, etc. and their relations. As the evolution of such social semantic Web, the relations will become more complicated. Falconer provides a visualization interface to help users understand the relations among User, Post, Forum, and annotated topics. The rendering engine is implemented with RaVis<sup>7</sup> library for Adobe Flex.

# 3. IMPLEMENTATIONS OF FALCONER

## 3.1 Architecture

Figure 6 shows the architecture of Falconer, which is a Struts 2 framework in general. The noticeable difference is that a couple of semantic technologies are employed. In order to efficiently store and query RDF format SIOC data,

<sup>7</sup><http://code.google.com/p/birdeye/wiki/RaVis>

openRDF Sesame 2.0 [2] is chosen as the persistent storage. At the same time, Falconer plugs a semantic Web search engine, Falcons, to support the set of semantic search engine based features. As using Hibernate to provide object/relational persistence and query service in traditional Struts framework, Falconer uses OpenRDF Elmo<sup>8</sup> 1.5 to perform the object/RDF mapping. The RDF format semantic data from both the Sesame storage and the Falcons semantic search engine could be accessed easily through Elmo.

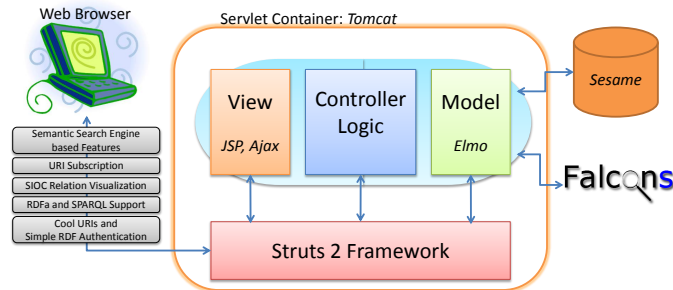


Figure 6: Architecture of Falconer

Based on this architecture, Falconer implements the distinctive features introduced above, i.e., User Registration Suggestion, Friends Auto-discovery, Semantic Annotation Tool, URI Subscription, and SIOC Relation Visualization. Although some other technologies used in Falconer are featureless, they conform to the state of the art Web specifications, like RDFa, SPARQL, and Cool URIs, and reflect the mentality of designing social semantic Web applications.

## 3.2 Specifications Conformance

### 3.2.1 SIOC

According to the introduction on SIOC project website, most of the applications create SIOC data by exporting and transforming data in traditional social Web applications rather than creating SIOC data natively. In Falconer, SIOC objects are created and queried through Elmo API, and directly stored as RDF in Sesame. As an RDF repository, Sesame ensures the efficiency and flexibility for accessing RDF by avoiding unnecessary overhead of transforming.

### 3.2.2 RDFa

As described in Section 2.2, each annotation is automatically taken as a topic of the post by creating a `sioctopic` property linking from the post to the annotation. Within the content part of the post, Falconer generates some RDFa [1] format data as well. In current version, each annotation keyword is surrounded with an HTML tag `A` that has an RDFa `@about` attribute specifying the topic's URI.

### 3.2.3 SPARQL

Since all semantic data created by Falconer are described in RDF or RDFa format, they can be queried with the SPARQL [6] query language. Currently, Falconer integrate a simple user interface with a textarea for users inputting their SPARQL queries, and returning XML style results.

### 3.2.4 Simple RDF Authentication

As we know, the user privacy is vital to social Web applications. Hence, an effective and efficient authentication is the

<sup>8</sup><http://www.openrdf.org/doc/elmo/1.5/>

first line of security for the user privacy. OpenId<sup>9</sup> is a widely used solution for this purpose which is an open, decentralized standard for authenticating users that can be used for access control [10]. However, as analyzed in [9], it may not be an appropriate one in the field of social semantic Web for several reasons. One dominant reason is that OpenId requires an identity server which is a vulnerable control point and may increase the cost of communication.

In Falconer, we employ a simpler, RESTful method with fewer point failure and fewer points of control authentication scheme similar to RDFAuth [9]. The differences include: First, Falconer uses the MD5 signature of user's password to identify the user identity which is not as security as PGP<sup>10</sup> method used in RDFAuth; Second, an ontology<sup>11</sup> is introduced to describe the authentication scheme, while RDFAuth stores the PGP public key in user's FOAF document. Considering that FOAF information may come from outside Falconer, the instance of the ontology can be created without changing the original FOAF document.

## 4. CONCLUSIONS

Falconer is a social semantic Web application with several interesting features. It helps end-users freely and fully utilizing large scale semantic data to semantically annotate their own content, which will be indexed by semantic Web search engines again. Thus, the creation and reuse of semantic data forms an ecosystem accelerating the development of semantic Web. Furthermore, enhanced by semantic Web search engines, Falconer is scalable. Interesting features of user registration suggestion, friends auto-discovery, semantic annotation tool, and URI Subscription are few reported in existing papers and applications. Other features, like multi-lingual support, multiple device support, and the close relationship between Falconer and social semantic Web, doubtless makes the application background more practical.

## 5. REFERENCES

- [1] B. Adida and M. Birbeck. RDFa Primer, October 2008. <http://www.w3.org/TR/xhtml-rdfa-primer/>.
- [2] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema.
- [3] DERI Ireland. The SIOC wiki pages, April 2009. [http://wiki.sioc-project.org/index.php/Main\\_Page](http://wiki.sioc-project.org/index.php/Main_Page).
- [4] T. Gruber. Collective Knowledge Systems: Where the Social Web meets the Semantic Web. *Journal of Web Semantics*, 6(1):4–13, February 2008.
- [5] A. Harth, A. Hogan, J. Umbrich, and S. Decker. Building a Semantic Web Search Engine: Challenges and Solutions. In *Proceedings of 3rd XTech Conference*, Dublin, Ireland, 2008.
- [6] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF, January 2008. <http://www.w3.org/TR/rdf-sparql-query/>.
- [7] Y. Qu, G. Cheng, H. Wu, W. Ge, and X. Zhang. Seeking Knowledge with Falcons. In *Semantic Web Challenge 2008*, Karlsruhe, Germany, 2008.
- [8] A. Shakya and H. Takeda. Information Sharing on the Social Semantic Web. In *Proceedings of the second NEA-JC Workshop on Current and Future Technologies*, Tokyo, Japan, 2008.
- [9] H. Story. RDFAuth: sketch of a buzzword compliant authentication protocol, March 2008. [http://blogs.sun.com/bblfish/entry/rdfauth\\_sketch\\_of\\_a\\_buzzword](http://blogs.sun.com/bblfish/entry/rdfauth_sketch_of_a_buzzword).
- [10] Wikipedia. Openid, 2009. <http://en.wikipedia.org/wiki/OpenID>.

<sup>9</sup><http://openid.net/>

<sup>10</sup>[http://en.wikipedia.org/wiki/Pretty\\_Good\\_Privacy](http://en.wikipedia.org/wiki/Pretty_Good_Privacy)

<sup>11</sup><http://iws.seu.edu.cn/services/falconer/ns>