

# SFSW Challenge Entry: Scripting a SIOC explorer

Benjamin Heitmann and Eyal Oren

Digital Enterprise Research Institute  
National University of Ireland, Galway  
Galway, Ireland

**Exploring SIOC data** In order to explore SIOC data from multiple online social communities in an integrated way, we have developed a SIOC explorer. It enables importing and exploring SIOC data from community sites if they expose their content by publishing it in the SIOC<sup>1</sup> format. The SIOC explorer can be found online<sup>2</sup>, and its source code can be found at the launchpad site<sup>3</sup>.

**Using the SIOC explorer** The entry page shows a list of SIOC forums in the database. In terms of SIOC each collection of posts is called a “forum”. After selecting a forum, a list of post excerpts is shown in the main column. The user can expand a specific post in order to see the full content and comments. The user can then browse posts by author or by topic or by period of time across all forums in the database. On the left side a possible list of filters is displayed, which can be used to restrict which posts get displayed.

**Benefits for the user** Firstly, different types of community sites, like weblogs, forums, mailing lists and IRC chat logs can export their content as SIOC data. This includes not only posts but also replies, information about authors and commenters and the links between all entities. The SIOC explorer uses this information to give the user a unified view on the content and structure of multiple sources.

Secondly, the SIOC format allows for richer content metadata. Topics can refer to terms from a SKOS<sup>4</sup> taxonomy, person references can point to FOAF<sup>5</sup> profiles, post titles and creation timestamps are expressed with the Dublin Core<sup>6</sup> vocabulary. The SIOC explorer allows the user to explore and navigate the data using all the available metadata.

In contrast, RSS<sup>7</sup> based news readers can only aggregate posts, and they can only use the limited metadata capabilities of RSS, like author names and category keywords.

<sup>1</sup> <http://rdfs.org/sioc/spec/>

<sup>2</sup> <http://www.activerdf.org/sioc/>

<sup>3</sup> <http://launchpad.net/sioc-ex>

<sup>4</sup> <http://www.w3.org/TR/swbp-skos-core-spec/>

<sup>5</sup> <http://xmlns.com/foaf/0.1/>

<sup>6</sup> <http://dublincore.org/>

<sup>7</sup> <http://web.resource.org/rss/1.0/>

**Capabilities of the Semantic Web** Besides allowing shared concepts between different sources, the Semantic Web allows each source to use different and evolving schemata to describe new concepts and to mix vocabulary from different ontologies. SIOC data can refer to SKOS topic descriptions or to FOAF personal profiles. In the SIOC explorer this gives the user the ability to filter posts based on fine-grained constraints:

**Example of a complex type: FOAF maker** When browsing a forum with more than one author, each of the authors is identified by an instance of `FOAF:maker`. When browsing the posts of such a forum, the user can locate the filter “maker” in the left column, click on “show details” and then select e.g. the workplace of a specific maker instance. Only posts from the maker with the specified workplace will then be displayed.

**Browsing data from foreign schemata** The SIOC explorer can handle data using vocabularies, which are not associated with SIOC data, because the navigation engine is domain agnostic and only relies on the features of RDF data. We use this feature to put e.g. the week of each post in the database using `SIOCEX:week`. SIOC posts only have a timestamp, but materialising this property using our own vocabulary allows the user to easily browse posts by week, month or year. This can be seen in the left column after selecting a forum.

**Usage of a Scripting Language** To develop the SIOC explorer, we extended the Ruby on Rails framework<sup>8</sup> with components for consuming and processing Semantic Web data. The first component is `ActiveRDF`<sup>9</sup>, which addresses the “model” mismatch and maps RDF data onto objects. The second component is `BrowseRDF`<sup>10</sup>, a faceted browsing and navigation engine that enables exploration of large Semantic Web datasets without domain-specific knowledge. The third component is a SIOC crawler which crawls, extracts, normalises and integrates SIOC data.

Each of the three components is designed to augment and integrate with Ruby on Rails. `ActiveRDF` can serve as a data layer in Ruby on Rails, replacing or augmenting the default `ActiveRecord` layer. The `BrowseRDF` navigation algorithms are implemented as a library for Ruby on Rails and provides generic navigation on top of `ActiveRDF`. The SIOC crawler uses several libraries and command-line tools which are external to Ruby, and incorporates the results into the model of the Ruby on Rails application.

Using `ActiveRDF` and our other extensions, the integration of Rails with RDF data was straightforward and the development effort was quite low compared to the benefits of using Semantic Web data. The models itself are automatically provided as virtual models, the controller (with all application logic) contains around 95 lines of code and the views contain around 100 lines of abstract HTML. The SIOC crawler consists of around 150 lines of code.

<sup>8</sup> <http://www.rubyonrails.org/>

<sup>9</sup> <http://www.activerdf.org/>

<sup>10</sup> <http://www.browserdf.com/>