Towards Semantically-Interlinked Online Communities

John G. Breslin, Andreas Harth, Uldis Bojars, and Stefan Decker

Digital Enterprise Research Institute (DERI), Galway, Ireland
`firstname.lastname@deri.org`

Abstract. Online community sites have replaced the traditional means of keeping a community informed via libraries and publishing. At present, online communities are islands that are not interlinked. We describe different types of online communities and tools that are currently used to build and support such communities. Ontologies and Semantic Web technologies offer an upgrade path to providing more complex services. Fusing information and inferring links between the various applications and types of information provides relevant insights that make the available information on the Internet more valuable. We present the SIOC ontology which combines terms from vocabularies that already exist with new terms needed to describe the relationships between concepts in the realm of online community sites.

1   Introduction

At the moment, most online communities are islands that are not linked. Sites are hosted on stand-alone systems that cannot be interconnected due to application and interface differences. Parallel discussions on interrelated topics may exist on a number of sites, but their users are unaware of that. There is a huge amount of related information that could be harnessed across such online communities, from similar member profile details to common-topic discussion forums.

The goal of SIOC[1] (Semantically-Interlinked Online Communities) is to interconnect these online communities. Community sites can include many discussion primitives, such as bulletin boards, weblogs and mailing lists, which we have grouped under the concept of forum.

SIOC will facilitate the location of related and relevant information; by searching on one forum, the ontology and interface will allow users to find information on forums from other sites that use a SIOC-based system architecture. Other uses include cross-site querying, topic-related searches, and the importing of SIOC data into other systems, for example, using an email program to browse data imported from a SIOC-enabled site. Therefore, SIOC tries to overcome the serious limitations of current sites in making information accessible to their users in an efficient manner [6].

---

[1] http://rdfs.org/sioc/

A part of the task of linking on-line communities is to suggest additional information related to any given forum and forum entry. One approach would be to perform a search on, for example, post title, author, date, keywords or the full post text in community sites. Existing Internet search engines locate the information by performing a keyword search on a full-text index of Internet resources. Some search engines try to improve the quality of search results by analysing the link structure of web resources. But even with these improvements, search engines lack an understanding of the information being searched for and return a high number of irrelevant results. In this paper, we try to solve this problem by narrowing the scope of a search to a set of interlinked community sites and by describing the information in a machine-readable form using the SIOC ontology.

In a typical usage scenario, a user is searching for information on, for example, installing broadband on a Linux-based PC in their house in Galway. There is a post A discussing local ISPs on site 1, a bulletin board dedicated to Galway, that references (on the HTML level) both a Usenet post B comparing broadband modems and a mailing list post C detailing how to install broadband on Linux. Previously the user would have had to traverse three sites to find the relevant information. However, by making use of the SIOC ontology and remote RDF querying, a search for broadband on the Galway bulletin board will also yield the relevant text from the interlinked Usenet and mailing list posts B and C.

There are some challenges for SIOC. The grand challenge is adoption by community sites, i.e. how can the users be enticed to make use of the SIOC ontology. By using concepts that can be easily understood by site administrators, and by providing properties that are automatically created by an end-user, the SIOC ontology can be adopted in a useful way. A second challenge is how best to use SIOC with existing ontologies. This can be partially solved by mappings and interfaces to commonly-used ontologies such as Dublin Core[2], FOAF[3] and RSS 1.0[4]. Another challenge is how SIOC will scale. If there are more sites to query, then there are more potential relevant results, but also longer response times and higher loads on the participating community sites. We will keep the scaling challenge in mind when creating a future architecture for an interconnected system of community sites.

The main contributions of this paper are the development of the SIOC ontology and mappings to other RDF vocabularies, and a prototype to produce SIOC metadata from a community weblog. These contributions will be detailed as follows. In section 2, we describe the SIOC ontology for linking information both within and between community sites using RDF data, and demonstrate how to map to other existing vocabularies (e.g., FOAF, RSS) and formats (email, XHTML, etc.). In section 3, we will discuss the exchange of SIOC instances by exporting and importing to web-based and legacy discussion systems as well as

---

[2] http://purl.org/dc/elements/1.1/
[3] http://xmlns.com/foaf/0.1/
[4] http://purl.org/rss/1.0/

RDF stores. Section 4 will describe some usages of the created instances, and related work will be discussed in section 5. Section 6 concludes the paper.

2   Ontology

In this section we present the SIOC ontology. The ontology consists of two major parts: first, it contains classes and properties that describe discussion forums and posts in online community sites. The ontology is available online[5]. Second, it includes mappings that relate SIOC to existing vocabularies such as FOAF and RSS.

   We have identifed the main concepts in online communities as Site, Forum, Post, Event, Group and User. These are shown in Figure 1. While similar parent concepts are found in other ontologies, it is the relationships, sub-classes and properties of these concepts in the arena of online discussion methods that make SIOC unique and provide use cases that were not previously possible.

2.1   Main Classes

We list the major classes that are used in the SIOC ontology, and describe their usage in more detail.

Site. is the location of an online community or set of communities, with users in groups creating posts on a set of forums. While an individual forum or group of forums are usually hosted on a centralised site, in the future the concept of a "site" may be extended (for example, a topic thread could be formed by posts in a distributed forum on a peer-to-peer environment).

Forum. can be thought of as a channel or discussion area on which posts are made. A forum can be linked to the site that hosts it. Forums will usually discuss a certain topic or set of related topics, or they may contain discussions entirely devoted to a certain community group or organisation. A forum will have a moderator who can veto or edit posts before or after they appear in the forum. Forums may have a set of subscribed users who are notified when new posts are made. The hierarchy of forums can be defined in terms of parents and children, allowing the creation of structures conforming to topic categories as defined by the site administrator. Examples of forums include mailing lists, online bulletin boards, Usenet newsgroups and weblogs.

Post. is an article or message posted by a user to a forum. A series of posts may be threaded if they share a common subject and are connected by parent and child relationships. Posts will have content and may also have attached files, which can be edited or deleted by the moderator of the forum that contains the post.

Event. is a virtual or real-world event with a single or multiple participants. Examples include meet-ups associated with a particular user or set of users,

---

[5] http://rdfs.org/sioc/ns#

Aggregator
Bulletin_Board
Mailing_List
Private_Message_Box
Usenet_Newsgroup
Weblog

Site

description
has_admin
 -istrator
host_of
name
related_to
topic

Event

description
finishes_at
has_organiser
has_participant
location
name
starts_at
topic

has_host

subClassOf

Forum

closed_at
container_of
description
has_host
has_moderator
has_parent
has_subscriber
name
parent_of
topic

has_
parent

has_
organiser

has_
participant

has_
administrator

Group

description
has_member
name
participant_of
topic

has_
subscriber

has_
member

Document

has_
container

has_
moderator

Person

subClassOf

subClassOf

has_creator

subClassOf

Post

closed_at
content
created_at
has_container
has_creator
has_parent
has_sibling
ip_address
links_to
modified_at
note
parent_of
related_to
sibling_of
subject
topic
views

has_
parent

User

administrator_of
avatar
created_at
creator_of
description
knows
location
member_of
modified_at
moderator_of
name
organiser_of
participant_of
subscriber_of
topic

subClassOf

knows

Administrator
Guest
Moderator
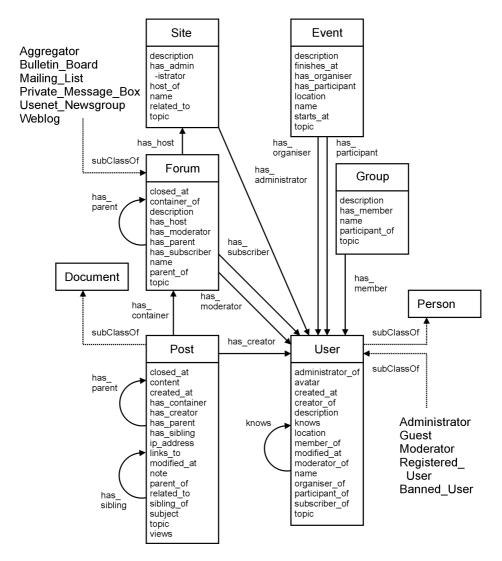Registered_
 User
Banned_User

has_
sibling

Fig. 1. Overview of classes and properties used in SIOC

a meeting for subscribers of a certain community forum, or private task
reminders to a single user.

Group. is a set of members or users of a community site who have a common
role, purpose or interest. While a group of users may be a single community
that is linked to a certain forum, they may also be a set of users who perform
a certain role, for example, moderators or administrators.

User. is a person who is a member of an online community. They are connected
to posts that they create or edit, to forums that they are subscribed to
or moderate, to sites that they administer, to other users that they know,

and to events that they organise or participate in. Users can be grouped for purposes of allowing access to certain forums or enhanced community site features (weblogs, webmail, etc.).

2.2   Important Properties

In the next paragraphs, we describe properties of SIOC concepts that are important for extracting meaning from and for interlinking online community sites.

topic. A topic definition applies to most of the concepts defined above, and topic metadata can be a useful way to match documents and people to each other.

While it may be more difficult to require a user to assign a topic to a post at creation time, it is more likely that a forum will have an associated topic or set of topics that can be propagated to the posts it contains. Similarly, users or groups can define topics of interest when their profiles are created or modified.

In order to enable the location of related information between the community sites, a common categorisation system has to be used. On large scale, general interest community sites, topics may be quite broad and a general categorisation such as the DMOZ[6] category hierarchy may be used.

On specialised sites, which may have a very specific category hierarchy, generic categorisation systems are not suitable because they are too broad and may not have the necessary level of detail. For these sites, we propose to define a category hierarchy in the SKOS framework [7] and to create mappings between these concepts and a common category system. In future work, SKOS may be used to describe all category schemes and mappings between them, but the lack of generic taxonomies expressed in SKOS (since it is in an early adoption phase) makes its current use difficult.

A proper use of topics can lead to many interesting scenarios in community sites. For example, a user has defined certain topics of interest on registering an account, after which forums matching those topics are suggested to the user.

views. The views property represents the number of times a particular post or user profile has been viewed. This is an example of where content is automatically created by an end-user, and can increase the content's importance in terms of searching. For example, a user creates a query across a set of SIOC-enabled sites, and is returned a list of subjects and extracts from certain posts, sorted by the popularity of the post, as indicated by the views property.

has_sibling. A recent development in online discussion methods is an article or post that appears in multiple blogs, or has been copied from one forum to another relevant forum. In SIOC, we can treat these copies of posts as siblings of each other if we think of the posts as non-identical twins that share

---

[6] http://dmoz.org/

most characteristics but differ in some manner. We can avoid duplication of common data in the creation of siblings by linking to the new sibling, the instance of which only contains the changed properties (in the example, has_container and topic would change). A sibling might also be a version of a post in another language.

closed. The closed property applies to posts in a threaded topic, but can also be used for forums. It specifies the date and time that the post or forum was closed. A closed property for posts is a useful for two reasons. Firstly, it is used to specify that a particular post can have no more children. Secondly, it gives us details of when the closure occurred, and can therefore be used to determine how relevant in time a discussion or set of discussions may be.

has_creator. The has_creator property links a post to the user profile of its author. Thus, we can follow the link from the post to the creator and locate the other posts by the same person. The community can be seen as a network of posts with users linked to each post, and there is also a network of other posts created by a given user stemming from there. We can use the information in community sites to locate more contributions by the given author.

knows. The knows property is a basic property to show the structure of social networks inside community sites. Who knows whom is the basic property used for describing social network sites and provides information about the links between community users. One of the options to locate relevant information on a given topic is to search for information, not in the full scope of the knowledge base, but in a subset of posts accessed by a person or friends of that person. There are three possible types of knows links: linking to a user inside the same community, to a user of other SIOC-enabled communities, or to other resources outside SIOC.

2.3 Mappings

One of the main functions of SIOC is to provide a means for exchanging community instance data. Since there are already a considerable number of classes and properties defined in RDF on the Web, we provide mappings in RDFS and OWL to allow the import and export of SIOC instance data in different vocabularies. Therefore, we can leverage the instance data that is already available.

We provide different kinds of mappings in RDFS for import and export using rdfs:subClassOf and rdfs:subPropertyOf, and also mappings in OWL using owl:equivalentProperty and owl:equivalentClass together with other OWL constructs. The mappings to various other RDF vocabularies are online[7]. In Table 1, we show how classes in FOAF, RSS, and various email vocabularies correspond to SIOC classes. Mappings of properties are described in a similar manner.

Carrying out the mappings requires a reasoning engine. Because of the various open issues with regard to OWL reasoning, we split our mappings into two parts. One part defines mappings in RDFS, which is somewhat limited in expressiveness

---

[7] http://rdfs.org/sioc/mappings

Table 1. Selected SIOC mappings

SIOC	FOAF	RSS	Email	Atom
Site	–	–	–	–
Forum	–	channel	–	feed
Post	Document	item	body	entry
User	Person	–	(from, to, cc)	(author, name)

but there exist scalable reasoning engines that allow for reasoning of class and property hierarchies and classification. A second part is encoded in OWL and describes more complex mapping constructs. At the current stage, we assume that the mappings are carried out on community sites that export or import data, but in theory the mappings can be completely decoupled.

Since mappings in SIOC are not only restricted to ontologies, we provide means to extract information from simple data structures. For example, we might want to map from XML documents into the SIOC ontology using XSL stylesheets[8]. For that purpose, we provide an XSL stylesheet to extract data from XHTML documents to create a SIOC Document instance. In the generic stylesheet, titles, images, and hyperlinks are extracted from Web pages, somewhat similar to how GRDDL[9] is used to extract information from XHTML documents.

Similarly, an XSL stylesheet can be used that maps from the XML-based RSS formats (0.9x and 2.0) to RSS 1.0, and from there we have RDF mappings to SIOC. Also, we have created a stylesheet that maps Atom[10] to SIOC, and this is used for importing Atom files into SIOC. A mapping from SIOC to Atom for data export requires a combination of queries against RDF data with an Atom template where the appropriate values can be filled in.

3   Exchanging Instances

The core use of SIOC will be in the exchange of instance data between sites. In the following, we elaborate on how the exchange, both importing and exporting data, can be carried out. We show how wrappers can help to achieve export functionality, either based on exporting documents containing the information or by rewriting queries. Another solution for incorporating the "document-based" wrapping into a more sophisticated query infrastructure is to mirror the exported and converted RDF documents in an RDF data store and thus allow for performing queries. We present a third solution, possibly for newly-developed applications, which uses a native RDF repository to store and retrieve statements, making import and export straightforward.

---

[8] http://www.w3.org/TR/xslt
[9] http://www.w3.org/2004/01/rdxh/spec
[10] http://www.atomenabled.org/

3.1   Wrappers to Existing Tools

Wrappers will allow us to export instances of community site concepts such as forums or posts in RDF format. They can also allow us to import SIOC instances to other non-SIOC systems. While there are many possible kinds of community sites for which wrappers could be developed, we will limit discussion to some of them, divided into two categories - legacy systems that do not use HTTP as a transport protocol, and web-based systems that can be accessed via HTTP.

Legacy Systems. A large number of systems preceding the current Web are still deployed and widely used on the Internet. Email is used for exchanging messages and files in an asynchronous way, Internet Relay Chat (IRC) is widely used for synchronous communication, and Usenet is still used to exchange messages. Therefore, to really capture a large amount of data currently exchanged in online communities on the Internet, these legacy systems and protocols need to be considered for SIOC.

In contrast to web-based systems, where we just need to translate the data, we need to employ protocol wrappers for legacy protocols to HTTP. For example, for email we need to translate the data representation format from RFC822[11] to SIOC, and provide a wrapper to the access protocol for email stores (usually POP3[12] or IMAP4[13]). Wrappers can be either quite simple (just a dump of the entire data set) or have some "intelligence" that allows for rewriting queries posed over HTTP into the original data format and access protocol. If we also provide importing facilities, for example into a mailing list, then we are building a gateway between a SIOC site and the mailing list.

The email export wrapper accepts a conjunctive query over HTTP GET and returns the results in SIOC. Parameters such as which posts to retrieve, the time duration for results to be returned, etc. are encoded into the query. Certain predicates can be used to restrict the set of posts to retrieve (such as modified_at > 2004-02-10). In a next step, the query is parsed and translated into IMAP4 to send to the original data source. The original data source then returns the results in RFC822 format, which is then translated back into RDF and returned to the original caller via HTTP. We have implemented the wrapper and the mapping using the Java programming language.

For imports, the email wrapper can receive sioc:Posts via HTTP PUT. Parameters needed for executing the mail sending process are also submitted via a conjunctive query to have the same interface for both GET and PUT. The posts are then translated into the RFC822 format that is suitable for sending via SMTP. The wrapper can then return a status code indicating that the addition of data was completed correctly. The import part of the wrapper still has to be implemented.

---

[11] http://www.ietf.org/rfc/rfc822.txt
[12] http://www.ietf.org/rfc/rfc1939.txt
[13] http://www.ietf.org/rfc/rfc1730.txt

Interfacing with IRC requires a different approach than wrapping email since the "data representation language" in IRC channel is just free-form text. In IRC, so-called "bots" are responsible for the exchange of data. A very simple bot just logs all utterings in an IRC channel and stores them persistently. More complex bots can understand a defined syntax and perform actions based on the commands issued. Also, some bots understand either a simple query syntax or conjunctive queries that are posed inside the IRC channel. One bot we are providing is logging the channel and recording URIs similar to the chump bot[14]. The content that is accumulated is made available in RDF via query over HTTP.

In addition to data that can be auto-generated from the existing sources, a wrapper has to provide additional information which has to be manually added, such as descriptions about mailing lists in sioc:Forum or general information in sioc:Site.

Web-Based Systems. Providing mappings from web-based systems is somewhat easier than mapping from legacy systems since protocol translation is not needed here.

We will discuss three kind of community sites using web-based systems - bulletin boards, weblogs and social networking sites. All these systems are based on content management systems with different complexity levels. Therefore exporting and importing information from and to such systems can be accomplished by adding wrapper interfaces to the existing content management systems.

For bulletin boards, some export functionality is already available (e.g. FOAF from vB [15] and phpBB [16], RSS from phpBB [17]). Most bulletin board systems use a LAMP (Linux, Apache, MySQL, PHP/Perl) architecture, and a wrapper to export data from these systems will use existing Perl and PHP libraries such as XML_FOAF, Magpie RSS, etc. However, most existing wrappers don't export their data in SIOC, and only provide a document-based export functionality rather than a query interface.

Weblogs usually are small scale systems consisting of one or more contributors and a community of readers. Most weblog engines already have RSS export functionality and there are some experimental implementations of export of other metadata, such as the Wordpress FOAF plugin [18]. Since the majority of these engines are open source software, it is straightforward to modify existing export functions to generate SIOC metadata. Import interfaces can be created in a similar way, allowing weblogs to import SIOC data. One of the use cases for SIOC import is replicating post entries among weblogs and community sites.

Social networking sites are based around the concept of persons and the relations between them. At the same time, many social networking sites are imple-

---

[14] http://usefulinc.com/chump/

[15] http://www.vbulletin.org/forum/showthread.php?t=66434

[16] http://www.phpbb.com/phpBB/viewtopic.php?p=1088960

[17] http://www.phpbb.com/phpBB/viewtopic.php?t=144548

[18] http://www.wasab.dk/morten/blog/archives/2004/07/05/wordpress-plugin-foaf-output

menting other functionality, such as bulletin boards or forums. There are existing implementations of FOAF metadata exports of user profiles on ecademy.com and Tribe.net. Similarly for bulletin boards, wrappers to export SIOC metadata on posts and forums can be created using existing Perl and PHP libraries. However, many social networking sites are members-only and are not viewable to the outside world, which raises a question of privacy and trust regarding the information exported from these sites. The issue of privacy can be partially addressed by exchanging sensitive information only among a closed network of trusted community sites.

The main challenge for using SIOC with web-based systems are not in the technical implementation of SIOC wrappers, but rather in the wide adoption of the SIOC ontology to gain incentives for people to provide data and tools for SIOC.
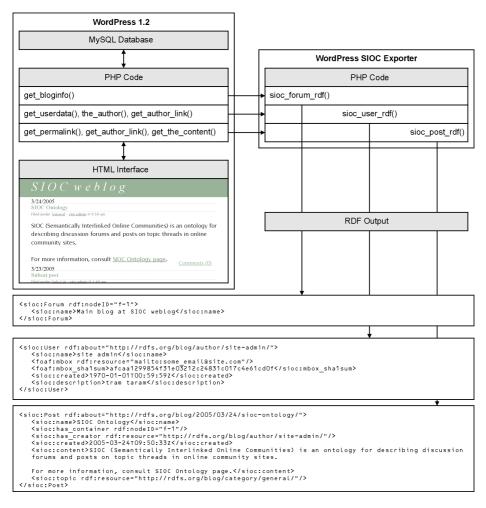

Fig. 2. SIOC metadata export from WordPress

By making SIOC data available through exports, we are encouraging the adoption of SIOC concepts. To this end, we have created a SIOC metadata export facility[19] for the WordPress weblog engine. This makes use of existing WordPress PHP functions to access the information about posts, users and forums (weblog channels) from the underlying relational database. SIOC metadata in RDF is generated for each concept instance. The export process is illustrated by example in Figure 2. Other export facilities are being written for the bulletin board systems phpBB and vBulletin, and the content management system Drupal.

3.2   Mirror Data in RDF Store

Most of the web-based wrappers just provide simple document-based export facilities. Replacing the simple wrappers with full-featured wrappers that are capable of query rewriting takes time. Since our goal is to make SIOC data available for query and to entice people to use SIOC now, we need a method to allow querying of the information that sites publish in flat files.

A solution to provide query facilities for sites that have only simple data export facilities is to replicate the information in a data store that can process queries. Queries are then answered from the replica. The replica is updated either by a scutter - an RDF crawler that traverses rdfs:seeAlso links - that periodically crawls the data, or by the original site that pushes updates and changes automatically into the mirror store once the data changes. If the data is exported in a format other than SIOC, then the system also needs to include a component that carries out the mappings from the vocabulary that is used to export data into SIOC.

Replicating the contents of the entire site from the relational database to an RDF store may work initially and create an easy upgrade path. However, in the longer term, storing and integrating data in a native RDF repository is the desirable solution.

3.3   Native RDF Store

The previous two subsections discussed tasks that concerned querying existing sites and their content. We will now describe how newly architectured sites can make use of a native RDF repository to store their data.

Exporting data is quite simple because RDF does not restrict you in the way data can be expressed. On the flip side, the flexibility of RDF creates a problem when importing data into systems with a fixed schema. Issues arise here, for example, when an application is importing data using a given schema, and certain mandatory data is missing.

Since community sites provide access to complex structures of information with different types, it is natural to store that information in RDF directly. Repositories such as Jena2 [10], Sesame [3], Redland [1], or YARS [5] can be used to store and retrieve the data. With an RDF store as the data repository,

---

importing and exporting information is straightforward, and also data integration tasks can be facilitated. An API similar to the RDF NetAPI [9] can be used as well. The route we chose for SIOC is to use a restful interface that uses HTTP methods such as PUT and DELETE for adding and removing data.

We can use an RDF repository as the data store and build the application functionality on top of the repository in a way that is flexible in regards to the schema. The user interface should also function when pieces of data are missing, since we cannot control which data (added or removed from the underlying RDF store) is agnostic to any schema definition.

4   Using SIOC Data

Given the ontology, the mappings, and the wrappers, we are now able to pose queries and add data to individual SIOC sites.

4.1   Browsing

Once we have made the data available using a common query infrastructure, we can use various user interfaces to navigate SIOC data. The simplest solution is to use a mapping from SIOC to a data format where client programs already exist. For example, SIOC data can be mapped to email and then read in any email program. Also, a mapping from SIOC to RSS allows us to navigate a subset of SIOC information inside a regular RSS news reader. Since SIOC has a richer data model than RSS, some information will be lost during the conversion.

Another approach is to use existing RDF browsers such as BrownSauce [20] to view arbitrary RDF data. Leveraging the full potential of SIOC requires the provision of custom programs and user interfaces specially tailored towards SIOC.

However, since most programs are already providing browsing facilities for their underlying data structures, implementing import facilities for those programs allows the seamless integration of data without the need for new user interfaces.

4.2   Query

Representing data in SIOC enables users to pose structural queries against the collected data rather than just having keyword search. An implication of structural queries is that you get precise answers as a result, and not just pieces of documents that match the keyword.

Until now, we have only considered querying one community site in isolation. However, since sites are linked together, we might want to perform queries across similar community sites that all share some connections.

---

[20] http://brownsauce.sourceforge.net/

One central problem in P2P networks is how to route queries [8]. We plan to exploit the link structure that connects forums or sites to route queries. The forum and site linkage inside SIOC makes it easier to do routing than in general-purpose peer-to-peer networks, since we have some (human-created) links that can be exploited. We expect a scale-free behaviour of these links once SIOC is widely used in practice.

By building the infrastructure for distributing queries into the different site management software or wrappers, we can perform queries without any central components. As a result, querying inside an intranet will be simple and already integrated into the tools used to manage the different community sites inside an organisation, such as mailing lists or forums.

4.3   Locating Related Information

Querying the community sites for information on demand is not the only model of end-user interaction. Another way to enhance the end-user experience is to prepare the data in advance, at creation time of a post.

Once a new post is created in a community site and the SIOC information is available, this site then queries the network of community sites to find related posts. A query is performed based on the post metadata, such as other posts by this person or other posts in the set of the post's topics.

After the information about related resources is received, the community site stores this information using a related_to property. Information about the resources the article links to is also extracted from the post body and stored in a links_to property. These properties can then be reused by other users of SIOC data and by SIOC and RDF browsers to browse forum entries and navigate through the web of interlinked posts, independent of the underlying site structure that the forums and posts are hosted on.

The results of this information retrieval model are the enhanced functionality added to community sites, and better scalability since the information is prepared in advance.

5   Related Work

Harvest is an early system [2] that can be used to gather information from diverse repositories to build, search, and replicate indexes, and to cache objects as they are retrieved across the Internet. Harvest uses the Summary Object Interchange Format (SOIF) to exchange metadata about resources. In contrast, SIOC uses RDF as the exchange format and allows for mappings between different vocabularies, which is not envisioned in SOIF. The various Harvest subsystems are arranged in a hierarchical fashion, similar to the Domain Name System. We do not have any specified way of accessing resources in SIOC, but intend to apply database techniques for query processing and integration.

Various approaches for data integration on the Web, such as data representation languages, structural information retrieval, and query processing, are

surveyed in [4]. The survey also describes the warehousing approach to data integration that aggregates all information at one central site. However, advanced database techniques have failed so far to surface on the Web. SIOC is a first step in providing a common vocabulary for data representation across online communities. In further work, we plan to apply usable techniques from the database community to web data integration problems.

At the moment, RDF Site Summary (RSS 1.0) is widely used in weblog systems and news sites. RSS 1.0 defines a lightweight vocabulary for syndicating news items, but is used for all sorts of data exchange. Although RSS works well in practice, there are several issues: firstly, only the last "n" news items are typically exported in RSS. There is no standardised way of accessing older posts. Secondly, there is an issue with regard to updates. Different vocabularies have different update semantics: where RSS usually provides a stream of news items that should be accumulated over time, changes in FOAF files mean that the previous version should be replaced by the current. Because vocabularies can be mixed in the same file, determining what update semantics to apply for a certain file is difficult. Thirdly, although there exists a large number of extensions, none of the advanced functionality of RSS is widely deployed, since tools lack support for creating and using the extensions. RSS is widely adopted in certain areas, such as weblogs, but is not used in a wider context such as bulletin boards, mailing lists, Usenet, wikis, etc.

Also, TrackBack[21] is a system implemented by many blogging tools that allows a weblog article to be linked to the followup articles. This is achieved by sending a summary and metadata of the new article to the weblog containing the original article, and adding this information to the original article. Linking together cross-site conversations is a step in the direction of semantically-interlinked online communities; however there are limitations to TrackBack. Firstly, it is being used in a very limited number of weblog entries and in most implementations the author has to manually enter the TrackBack address. Secondly, it only connects two individual instances of posts, not reflecting the links to the community and, in the case of archived post entries, the readers may even be unaware of the existence of this new link. Thirdly, TrackBack does not have a machine readable representation that would allow one to export its link semantics in RDF, to aggregate the resulting information and reuse it to identify related post entries.

6   Conclusion

We have presented the SIOC ontology and various mappings to and from other vocabularies that are already deployed on the Web. We have described how instance data in SIOC can be exchanged among online community sites. Our initial SIOC ontology can also be used to enable more complex use cases, for

---

[21] http://www.movabletype.org/docs/mttrackback.html

example cross-site structural queries, and integration based on the warehousing approach.

To tackle the challenge of adoption, we have provided an upgrade path that allows a gradual migration from existing systems to semantically-enabled sites. For combination with other ontologies, we have presented mappings to and from SIOC that allow the export and import of SIOC data using existing systems and tools. We have developed a prototype SIOC exporter for a weblog engine, and several more are in development. In the future, we intend to exploit the characteristics of intra- and inter-site links to guide query routing in a P2P-like environment.

References

1. D. Beckett. The Design and Implementation of the Redland RDF Application Framework. *Computer Networks*, 39(5):577–588, 2002.
2. C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz. The Harvest information discovery and access system. *Computer Networks and ISDN Systems*, 28(1–2):119–125, 1995.
3. J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *International Semantic Web Conference*, pages 54–68, 2002.
4. D. Florescu, A. Y. Levy, and A. O. Mendelzon. Database Techniques for the World-Wide Web: A Survey. *SIGMOD Record*, 27(3):59–74, 1998.
5. A. Harth and S. Decker. Yet Another RDF Store: Complete Index Structures for Storing Semantic Web Data With Contexts. *DERI Technical Report*, 2004.
6. R. Lara, S.-K. Han, H. Lausen, M. Stollberg, Y. Ding, and D. Fensel. An Evaluation of Semantic Web Portals. In *IADIS Applied Computing International Conference 2004, Lisbon, Portugal, March 23-26*, 2004.
7. A. J. Miles, N. Rogers, and D. Beckett. SKOS Core RDF Vocabulary. 2004. `http://www.w3.org/2004/02/skos/core/`.
8. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P networking infrastructure based on RDF. In *WWW*, pages 604–615, 2002.
9. A. Seaborne. An RDF NetAPI. In *International Semantic Web Conference*, pages 399–403, 2002.
10. K. Wilkinson, C. Sayers, H. A. Kuno, and D. Reynolds. Efficient RDF Storage and Retrieval in Jena2. In *Proceedings of SWDB'03, The first International Workshop on Semantic Web and Databases, Co-located with VLDB 2003*, pages 131–150, 2003.