SRG: A Digital Document-Enhanced Service Oriented Research Grid

Geoffrey C. Fox, Ahmet Fatih Mustacoglu, Ahmet E. Topcu, Aurel Cami

***Abstract*—We describe an ongoing project building a system that consists of tools and services for supporting Cyberinfrastructure based scientific research. This system, called the Semantic Research Grid (SRG), integrates a number of existing online research tools (social bookmarking, academic search, scientific databases, journal and conference content management systems) and aims to develop added-value community-building tools that leverage the semantic analysis of digital documents. We discuss the design, the overall architecture, and the current state of the implementation of SRG, and provide a roadmap of the future work in this project.**

***Index Terms*—Cyberinfrastructure based scientific research, annotation, academic search, scientific databases**

I. Introduction

In recent years there has been a rapid development of tools and services aimed at fostering online collaboration and sharing between users and communities. Blogs (blogger.com, Google Blog), Wikis (Wikipedia, WikiWikiWeb, Wikitravel), Social Networking Tools (MySpace, LinkedIn), Social Bookmarking Tools (del.icio.us, Flickr, YouTube), Syndication Feed Aggregators (Netvibes, YourLiveWire) and other related tools are quickly being embraced by an expanding user base. The term "Web 2.0" [1] is now a widely accepted term representing this wave of new Web-based tools and the belief that they indicate a qualitative change in today's Web. This change is also apparent in the domain of scientific research, with the recent creation of a number of online tools that enable the annotation and sharing of scientific content, such as CiteULike [2], Connotea [3], and Bibsonomy [4].

These developments overlap with ongoing efforts to exploit Grid architectures based on Web services [5] for supporting international scientific and engineering research teams by enabling the sharing of large data and compute resources (i.e., creating a Cyberinfrastructure for e-Science [6, 7]).

Significant advances have also taken place in the areas of digital libraries and academic search. Domain specific academic search tools, such as CiteSeer [8], or general ones, such as Google Scholar [9], have enabled open, fast and easy access to vast online repositories of linked scientific documents.

Despite such important developments, there remains a great need for research tools geared toward niche communities of researchers. For example, currently there is no fast and reliable way to collect and analyze all the papers of a research group; a search in Google Scholar for the publications of our research lab (Community Grids Lab) will return only about 20% of the desired content [10]. Similarly, there is no easy way to find all publications that focus on a very narrow topic, say all or almost all the papers discussing a particular chemical compound. Moreover, the new tools for annotating scholarly papers (CiteULike, Connotea) are currently detached from the capabilities provided by other research tools, such as academic search tools. Finally, there is a wealth of information contained in numerous field specific scientific databases, such as PubMed, or PubChem, which also remains largely outside the scope of automated tools for scholarly research.

In this paper, we describe a project that is motivated by the above concerns and aims to develop a community-centric platform of tools and services that integrate the major existing annotation tools, academic search tools, and scientific databases into the Cyberinfrastructure based scholarly research. These tools and services, collectively called the Semantic Research Grid (SRG), will be backed by databases which store user and community specific data and metadata and will be configured into three applications: (1) A model for scientific research which links both traditional simulations and observational analysis to the data mining of existing scientific documents; (2) A model for a journal web site supporting both readers and the editorial function; (3) A model for a natural collection of related documents such as those of a research group or those of a conference.

The rest of this paper is organized as follows: Section 2 gives an overview of the existing online tools that form the basis of SRG and explains how they are used in this system. Section 3 describes the design principles and the overall architecture of SRG, expounds the various technologies and software packages used in developing this system, and details the current state of its implementation. Section 4 presents a roadmap of the future work in this project.

Manuscript received November 28, 2006.

Geoffrey C. Fox is with Indiana University, Bloomington, IN 47404, USA (phone: 812-219-4643; fax: 812-856-7972; e-mail: gcf@indiana.edu).

Ahmet Fatih Mustacoglu is with Indiana University, Bloomington, IN 47404, USA (e-mail: amustaco@cs.indiana.edu).

Ahmet E. Topcu, is with Indiana University, Bloomington, IN 47404, USA (e-mail: atopcu@cs.indiana.edu).

Aurel Cami is with Indiana University, Bloomington, IN 47404, USA (e-mail: acami@indiana.edu).

II. OVERVIEW OF EXISTING TOOLS

A. Annotation Tools

Perhaps, the best known annotation (or, social bookmarking) web site is *del.icio.us* (henceforth referred to as Delicious), a tool designed to enable the annotation and sharing of URLs. A number of other annotation tools are now in widespread use; they support annotation and sharing of a variety of resources, such as photos (Flickr), videos (YouTube), books (LibraryThing) and goals (43things). In particular, there are several online tools specializing in the annotation of scholarly publications, including Connotea, CiteULike, and Bibsonomy [11]. The core service offered by these annotation tools is the capability that allows users to quickly annotate their favorite resources (URLs, photos, or citations) using a small number of *tags* (keywords) and to share their tagged content with other users.

Tagging represents a significant shift in the *metadata creation* methodology. Traditionally, metadata creation has been handled by: (a) specialized professionals working with complex categorization schemes; or (b) the authors of scholarly content. Both of these methods suffer from various problems [12]. Among the cited shortcomings of professional metadata creation are the complexity and the lack of scalability of cataloguing systems, especially when applied to the vast amount of data in today's Web. Author metadata creation is vulnerable to inadequate, or purposefully inaccurate descriptions by authors. The new approach of metadata creation, namely *tagging*, puts the task of metadata creation in the hands of general users. This practice of collaborative categorization (which is now commonly referred to as *folksonomy* [13]) aims to harness the collective intelligence of a large number of people. It has met with widespread acceptance by the Web users, as shown by the sharp increase in the number of subscribers to such tools. Recently, there have been preliminary attempts to look into the cognitive underpinnings of the popularity of tagging [14] and some dynamic discussions about the bottom-up tagging versus top-down categorization trade-off [15, 16]. While tagging remains a new practice whose long-term benefits are not yet well-understood, some of its advantages and disadvantages have been already pointed out [13]. Among the benefits of tagging are: (a) the ease of use and access of the tagging tools; (b) the ease of discovering new content; (c) the support for the creation of niche communities. The shortcomings include: (i) the lack of a standard set of keywords; (ii) the difficulty of dealing with misspelling errors, synonyms, and acronyms, which are commonly found in tagging; (iii) the difficulty of inferring hierarchical relationships between tags (i.e., creating a taxonomy).

Each social bookmarking tool can be described in terms of: (a) A model of *data* and *metadata* adopted by the tool. (b) A *user interface* that allows users and groups to subscribe to the service, manage their tagged content, share it with other users, and discover new content; (c) An *input/output interface* that allows the data and metadata to be exported to various formats or applications, and enables programmatic interaction with the system. In the accompanying technical report [17] we give a detailed description of the above features for Delicious, CiteULike, Connotea, and Bibsonomy.

B. Academic Search Tools

The advent of the World Wide Web has led to the creation of a number of digital databases of scientific content. These databases use one of two main data acquisition methods: (i) manual insertion by volunteers (e.g., DBLP) (ii) automated harvesting by crawling open-access databases, home pages of authors, web sites of the publication venues, and so on (e.g., CiteSeer). Both methods may be complemented with user submissions.

In this project, we focus on the major open-access academic search tools that use automated methods of acquiring and analyzing scientific documents. These tools are discussed next.

CiteSeer: CiteSeer was introduced in 1997 by Giles et al. [8]. As the first tool in this category, CiteSeer is probably also the best known, especially in the field of Computer Science, which is its specialization domain. The core feature of CiteSeer is *Automated Citation Indexing*, a method for the automated extraction, parsing and indexing of the citations contained in a paper and of the context of these citations in the paper's body. CiteSeer has pioneered a number of techniques for the automated extraction of document metadata, including front-end metadata such as title, author names, author affiliations, abstract, and back-end metadata, such as acknowledgements, and citations to other papers. The algorithms used by CiteSeer are generally based on carefully crafted *heuristics* and/or *machine learning* techniques. Recently, it was estimated that CiteSeer covers about 24% of papers in Computer Science and it was pointed out that the use of automated methods for harvesting documents has led to a bias toward papers with 3 or more authors [18]. To deal with issues such as increasing query latency and degradation of system stability, as well as to improve the interoperability of the system, CiteSeer has recently announced the design of a new version of the system, called CiteSeer^X [19].

Google Scholar: Google Scholar (GS) first became public in 2004. The methods for collecting and analyzing documents used by GS are similar to those of CiteSeer. Note that CiteSeer is both a search system and a digital library having currently more than 800,000 full-text documents in its repository, while GS is a search system which attempts to find and display the URLs that point to the full-text versions of the query results. Unlike CiteSeer, GS aspires to be a "single place to find scholarly materials" covering "all research areas, and all sources" [20]. GS has been generally lauded for the open, fast and easy access it provides to vast collections of digital academic documents. There has also been significant criticism towards GS, especially from librarians. The major criticism has to do with: (i) *scope* (GS does not declare which publishers it currently covers; at the same time it is known it does not cover some major publishers, such as Elsevier, American Chemical Society, and Emerald [20, 21]); (ii)

coverage (GS does not provide full coverage of the articles from the publishers that seem to be covered [20, 21]); (iii) *accuracy* (its metadata extraction algorithms are not very precise, leading to duplicate records, unreliable citation counts, etc. [21]).

Windows Live Academic: Windows Live Academic (WLA) is the latest addition in the area of open-access academic search tools; it became public in 2006. Its objectives are similar to those of GS, but unlike GS it has revealed the list of the covered publishers and venues. The initial version of this tool, has been shown to suffer from the same issues of *coverage* and *accuracy* discussed above for GS [22]. Another drawback of WLA is that, unlike CiteSeer and GS, it does not yet provide *citation indexing*.

We achieve integration with the above academic search tools by building wrappers around them. It has been suggested [23, 24] that, for specific user categories, the "one stop shopping" or "one size fits all" approach of GS and WLA can't be an alternative to specially crafted portals integrating data from various sources. We share this belief and envision that these tools will have two main roles in the usage scenarios of our system: (1) They will be used to *seed the creation of a community* (e.g., the papers of a research group, the papers on a chemical compound, etc.). These seeds will then be expanded and refined by our community-building tools and linked with the annotation tools. (2) They will be used to *extract the citation count* of scientific papers. Due to their global nature, GS and WLA are uniquely positioned for providing this kind of service, which is analogous to the "back-linking" capability offered by the general-purpose search engines. We anticipate that such counts will also need to be refined by community-specific tools.

C. Scientific Databases

Several excellent open-access scientific databases, such as PubMed, PubChem, and Science.gov, have been created over the years. These databases constitute the "deep Web" and have been estimated to contain 400-500 times more public content than the "surface Web" [25]. Since the deep Web is largely invisible to current search engines (including academic ones), this wealth of information has not been integrated with the online research tools.

Our system intends to tap into this wealth of domain specific information by focusing initially on the field of Chemistry. We are adapting the Oscar3 tool [25] from the University of Cambridge Chemistry department, that can analyze documents for chemical information including chemical compounds (see the web site of the Chembiogrid project [26]). The capability of performing automated semantic analysis of chemistry papers with Oscar3 enables a range of new tools. For example, one could provide links from the compound names to scientific data associated with this compound which can come from specialized chemistry databases (currently part of Oscar3), PubChem, or from the academic search tools. As another example, one could use the features extracted through Oscar3 to categorize the content of specialized databases, such as PubMed, or the latest information from various chemistry publications venues delivered through syndication feeds (e.g., the tables of contents from the latest issues of a collection of chemistry journals and conferences; see the UBio project [27] for a similar application in the field of Biology).

In the future, we also expect such tools to be extended to other fields, such as Astronomy and Earth Science, as they have developed rich domain specific metadata.

D. Journal and Conference Management

The last category of tools underlying the SRG system, are the Content Management Systems (CMS) used by journals and conferences. Manuscript Central is a journal management system which is a popular choice of many publishers. Likewise, CMT (developed by Microsoft Research) is a popular conference management system. We cannot and should not replace these tools. Instead, we plan to wrap them with Web services and then create new tools which aggregate information from various sources, such as annotation tools and academic search tools, to provide added value to the editors and readers of publication venues. For example, one could download all the papers submitted to a venue and analyze them with CiteSeer-like algorithms to extract front- and back-end metadata, or with tools like Oscar3 to extract domain specific metadata. This metadata could then be fed to a community-building tool which generates a list of referees that are not in conflict of interest with the authors of submitted papers (using methods similar to that used in [28]). Another useful service would be to enable journals build communities of authors—especially in association with "special issues" of papers on a single topic.

In summary, the Semantic Research Grid aims to develop a set of new tools and services that aggregate information from a variety of sources (i.e., "mash-up" tools) and provide added value to communities of researchers. In Section 4, we provide a detailed discussion of the techniques that will be used in developing these tools.

III. DESIGN AND IMPLEMENTATION OF SRG

A. System Design and Architecture

We have followed Web 2.0 design patterns [1] in designing the SRG system. Below, we list these patterns and discuss how they were applied in designing SRG:

Delivering services, not packaged software: SRG is a collection of tools and services that can be accessed over the Web (either through a user interface or programmatically through Web services). It will evolve by introducing new features; still its users won't have to install new versions of the software.

Producing hard-to-recreate data that gets richer as more people use the system: By combining data from a variety of sources, SRG will create added-value data and metadata generated with specific communities in mind. As more people participate in a community, the collection of the data and metadata managed by that community will increase in quantity,

leading to the potential for improved precision of the automated system tools.

Harnessing collective intelligence: Through its integration with the social bookmarking tools, SRG can leverage data and metadata from a large number of researchers. Moreover, the system can handle both individual users and groups of users, and supports sharing and collaboration between group members.

Leveraging the long tail through customer self-service: The term "long tail" here refers to the concept formulated by Anderson [29] that non-hit products can collectively make up a market share that may exceed the relatively few current hits, bestsellers or blockbusters, provided the store or distribution channel is large enough (this business model is leveraged for example by Netflix or Amazon.com)[1]. SRG aims to support research communities, such as the members of a research project, a group interested in a particular chemical compound and so on, by allowing them to create system accounts and to use the community-building tools for their specific usage scenarios.

Software above the level of a single device: Currently, the SRG user interface runs in a browser. However, because of its layered design and the use of J2EE technology (see Section 2.C), system front-ends for other devices, such as PDAs, can be developed at low cost.

In addition to these design patterns, we have followed two general principles: (a) every component is packaged as a service as long as this packaging does not imply an unacceptable performance degradation; b) if a needed capability exists and works well but is insufficient in some fashion, we try not to replace it but rather wrap it as a service so we can interact with its natural interface but easily input and output information through its service interface.

Figure 1 shows the overall architecture of the SRG system. This system consists of three main layers: (a) the *client* layer; (b) the *Web* layer; and (c) the *data* layer. The client layer is made up of Java Server Pages (JSP) which are translated into servlets by an Apache Tomcat J2EE Web container and generate dynamic content for the browser. The client layer communicates with the Web layer over the HTTP protocol through SOAP messages encapsulating WSDL-formatted objects. The Web layer consists of several Web services who handle communication with the existing online tools. The Web layer communicates with the data layer through JDBC connection. Finally, the data layer is composed of several local or remote databases.

B. Key Design Issues

We now discuss several key issues in the design of the SRG system:

Users and Profiles: The SRG system supports individual users and groups of users. Users' personal information and the login information for bookmarking web sites are accessible through the user's profile. More specifically, user's profile contains the system password, email address, full name, login information for annotation web sites (citeulike.org, connotea.org and del.icio.us), and the group membership information. Users can access and modify their profile settings at any time; while logged in users can: (a) Change their system password; (b) Update their profile including the full name, email address and the username and password for the annotation web sites; (c) Make requests to subscribe to any available group.

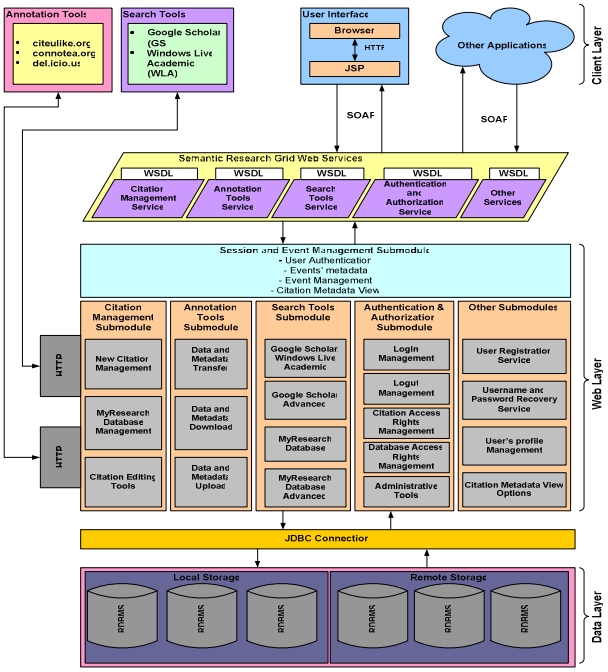

Fig. 1. Semantic Research Grid Architecture.

Group Administration: There are three types of users in the system: Super Administrator (SA), Group Administrator (GA), and (regular) User. There may be more than one SAs; an existing SA can add other SAs to the system. Each group has at least one GA who is appointed by an SA. When a new group is created, the user who requested the group creation becomes GA for this group. Users can make requests to subscribe to any group. GAs confirm/deny the request(s) made by users. Users are allowed to belong to more than one groups.

Access Rights: Users create citations in several ways: (a) using annotation tools (Delicious, CiteULike, Connotea); (b) using search tools (GS, WLA); (c) manually, through "Insert New Citation" interface; (d) using MyResearch Database search tool. During these operations, users have the option of making citations *public* or *private*. Private citations can be accessed only by their owner. Public citations must be associated with at least one group and can be accessed by all users of that group.

For each citation record, there are three types of access rights: *Read* access right, *Write* access right, and *Delete* access right. Users who have Read access for a citation can read that citation. Only users who have Write access for a citation can update that citation. Delete access is required for deleting

[1] The term "long tail" is also used in statistics to describe certain statistical distributions.

citations. These access rights are defined with respect to three kind of users: *Owner* who is the user that initiates the citation metadata creation; *Group* which is the group to which the owner belongs; *Other* users. There is only one owner of a citation record. However, there might be more than one group for a citation. The owner of a citation record can specify the citation rights for all three kinds of users mentioned above.

User Session: Due to the stateless nature of HTTP, a number of alternative mechanisms have been developed for applications that need to maintain a conversational state. The *HTTP session API*, which is a component of the Java Servlet specification, provides a mechanism for web-based applications to maintain a user's state information. This mechanism, which is called *session,* is usually associated with a user and supports the management of the user's state information on the server side. A session is represented by an *HttpSession* object, which stores and provides access to the user specific data. In the SRG system, the user's session is instantiated once a user logs into the system. The session can be later accessed through the JSP pages.

Consistency Model: In a collaborative environment, people work together and share some resources to achieve common goals [7]. In such systems, resources are vulnerable to user mistakes. To provide consistency and to avoid undesired changes in the system, it is necessary to have a mechanism for restoring the system to any previous state. *Versioning tools* for software development, such as *Concurrent Versions System* (CVS) or *Subversion* (SVN), and *Wikis* are well-known examples of collaborative systems that provide such mechanisms.

The SRG system is a collaborative environment that allows multiple users to create, and manage a common set of citations. Data and metadata can be transferred into SRG from different online sources, such as bookmarking web sites, academic search tools, scientific databases, and journal and conference content management systems. Users are allowed to overwrite or modify existing citations; this may lead to various issues. For instance, one user can create an entry for a citation downloaded from Delicious (including tagging metadata). Later, a second user can try to insert into the system the same citation found through a Windows Live Academic search. The second user could choose to overwrite the existing citation, thus causing the tagging information for that citation to be deleted.

To allow such issues to be fixed, we have developed our consistency model based on the concepts of *event* and *dataset*. An event is commonly defined as the act of changing the value of an attribute of some object [30]. Storing all the events about an object, allows users to review and undo these events. In the consistency model of the SRG system, we have adopted the view of an event as a time-stamped action on an object. We distinguish between two types of events: *major* events, and *minor* events. The insertion of a new citation record into a database and the deletion of record from a database are considered major events. Any update or modification of an existing citation record is considered a minor event.

Every event is tied to a particular user. Events are applied (and undone) at the level of granularity of dataset, which is defined as a collection of minor events related to a user. From the moment a user is logged into the system, all minor events are stored in the *session* of this user (described above). A dataset can be created by a user from the available events in the current session. Associated with each citation record, there is an initial set of citation metadata. This initial set of metadata may have come from various sources, such as annotation tools, academic search tools or manual insertion through the user interface. The first dataset will be applied to the initial citation metadata. The citation metadata of a record at a specific moment, is the result of applying one or more ordered datasets to the initial citation metadata

There are two key issues that require attention during the process of creating a dataset: (a) Events belonging to a dataset must be on the same citation, i.e., we do not allow events related to different citations to be in the same dataset; (b) The order of the event time-stamps is important in that the events of a dataset are applied in the order specified by their time-stamps.

In the current implementation, users can choose any set of consecutive events on a citation to form a dataset. Unless the user defines one or more datasets on the collection of events for a particular user session, all the stored events will be lost when the session ends.

C. Current State of the Implementation

The SRG system consists of several modules. Each module has the same layered design consisting of a client layer, a Web layer, and a data layer. We discuss the technologies and software packages used in the implementation of each module:

The c*lient layer* of each module is composed of Java Server Pages (JSP). The JSP pages communicate with the Web layer over HTTP protocol through SOAP messages.

TABLE 1
THE APIs USED IN IMPLEMENTING THE WEB LAYER.

API	Purpose
JDOM	For parsing XML documents
Jakarta Commons HTTP Client	For handling HTTP communication
XPATH	For querying an XML document object
Castor	For XML-to-Java or Java-to-XML binding
JTidy	For parsing HTML documents
Apache Axis	For creating Java Web Services

The *Web layer* is a collection of Web services. The Web services are built using WSDL and SOAP. WSDL is a subset of XML that is used to describe the Web services and their location. SOAP is an XML-based lightweight protocol for exchanging information. The Web service provides methods for communicating with external tools. A number of APIs, summarized in Table 1, are used in the implementation of Web services. Web services are created using Apache Axis. The software modules are deployed in an Apache Tomcat Web container (SRG currently uses Tomcat version 5.0.28).

Web services communicate with the *data layer* using Java Database Connectivity (JDBC). The data layer is composed of several local and remote databases used for storing user specific information, such as the citation records, their access rights, datasets, and so on. Currently, we use MySQL as the Database Management System.

Table 2 lists the software modules of the SRG system that have already been implemented and gives a short description of their functionality. A detailed discussion of these modules may be found in the accompanying technical report [17].

TABLE 2
THE COMPLETED SOFTWARE MODULES OF THE SRG SYSTEM

Module	Function
Citation Management	Allows the manual creation of new citation records and the update of existing records
Citation Metadata View	Allows users to specify the desired level of detail for displaying citation metadata
Annotation Tools	Implements the interface to the annotation tools: Delicious, CiteULike, and Connotea. Allows downloading, uploading and transfer of citation data and metadata.
Search Tools	Implements the interface to the academic search tools: GS, WLA. Provides an interface for searching MyDatabase local or remote databases
Authentication and Authorization	Implements the authentication and authorization functionality
User Registration	Handles user's registration with the system
Username and Password Recovery	Allows users to recover forgotten passwords
User's Profile Management	Allows users to update their profile

IV. FUTURE WORK

Figure 2 gives a high-level view of the various tool categories that will be a part of SRG and the interactions among them.

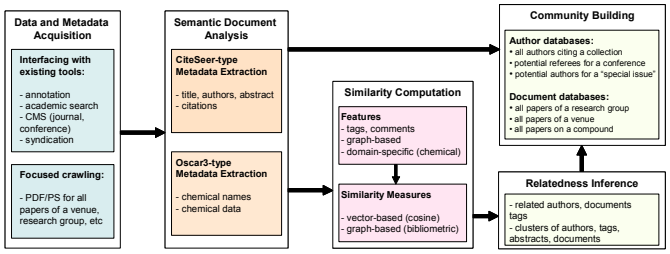

Fig 2. A high-level view of the tools comprising SRG.

For each of the tool categories in Figure 2, we give an overview of the related work, and explain the techniques and algorithms that will be used in developing these new tools.

A. Data and Metadata Acquisition Tools

SRG gathers data and metadata from various sources. As described in Section 3, currently we have implemented wrappers to several annotation tools (Delicious, CiteULike, Connotea) and academic search tools (Google Scholar, Windows Live Academic). Future steps in this direction include: (a) interfacing with the Bibsonomy annotation tool; (b) interfacing with Content Management Systems (Manuscript Central, CMT); and (c) interfacing with RSS feeds from a large collection of chemistry publication venues.

The interfaces to existing systems provide SRG with rich citation and tagging metadata. A desired feature of the system would be a tool that enables harvesting the full text (in PDF, or PS formats) for paper collections defined in various ways, such as the papers of a research group, the papers of a publication venue, and so on. In some cases, this capability could be easily implemented. For example, if a research group maintains a web page with all the publications of the group, then a tool that takes the URL of this page as input could easily download all the papers. In other cases, the problem might be more difficult. Consider for example the task of obtaining the full text of all papers that have appeared in a venue during a specific period of time. The open-access tools, such as CiteSeer, have been shown to contain only a small fraction of such collections in their repositories. For example, Zhuang et al. [31] found that in 2005 the CiteSeer repository contained 25.42% of papers from the International Workshop on the Web and Databases (WebDB) and 26.9% of papers from the Journal of Artificial Intelligence Research (JAIR) for the period 1998-2004. Using *focused crawling* [32] techniques, these authors were able to automatically collect about 81% of papers from these two venues. We intend to apply similar techniques for harvesting all or almost all full-text papers belonging to various other collections (e.g., all papers by a group of authors that have cited a given collection of papers). The focused crawlers could be seeded with the fraction of the desired documents that can be accessed through CiteSeer, Google Scholar, or Windows Live Academic, and then guided by heuristic and machine learning algorithms to direct the crawl toward additional target documents.

B. Semantic Document Analysis

SRG will develop tools that perform two types of semantic analysis on full-text papers: (a) General metadata extraction, which consists in extracting front-end metadata, such as the title, authors, and abstract, and back-end metadata, such as the citations to other papers; (b) Domain specific metadata extraction, which consists in extracting scientific information (e.g., chemical compound names) from the body of a paper.

The first type of analysis could be applied to papers from any domain. A number of algorithms for metadata extraction, most of them related to the CiteSeer system, have been proposed in recent years. These algorithms fall under one of two categories: (i) heuristic algorithms; and (ii) machine learning algorithms. We discuss next a sampling of such algorithms. The paper by Giles et al. [8], which introduced CiteSeer, describes several heuristics for *Automated Citation Indexing*, i.e., the extraction and indexing of citations and their context in the body of the document. A core feature of this method is the ability to *recognize identical citations* (i.e., citations to the same paper) when they have syntactic

differences. A number of techniques for doing so are described in [8]; the basic step of the described techniques is to find for each citation, the maximum number of words that match with a previous citation, normalized by the length of the shorter citation. Two citations are then considered identical if this number exceeds a threshold. Han et al. [33] apply a classifier based on Support Vector Machines (SVM) to categorize each line of a paper's header part into one of 15 basic metadata types (*title*, *author*, *email*, *affiliation*, etc.). Han et al. [34-36] explore a range of machine learning approaches, such as Support Vector Machines (SVM), k-way spectral clustering, and hierarchical Bayes mixture model, for the problem of *author name disambiguation*. All of these methods exploit three features of an author's name: co-author names, paper title words, and journal or proceeding title words. Councill et al. [37] investigated automated extraction of *acknowledgement* information. First, they use regular expressions to identify the sections of a paper containing acknowledgement information. Then, they apply SVMs to classify each line in these sections as "acknowledging" or "non-acknowledging". Finally, they use again regular expression to extract the acknowledged entities (people or organizations) from the "acknowledging" lines. We intend to implement methods similar to the ones just described, to extract metadata from the body of papers.

The second type of metadata extraction will be applied to papers (or, abstracts) in the field of Chemistry and will be based on Oscar3 [34, 38], a tool developed at the University of Cambridge Chemistry department. This tool can extract a variety of chemical features, including chemical names (formulae, acronyms, etc.), chemical data (spectra, boiling point, etc.) and other types of chemistry-specific information.

C. Similarity Computation

The tools in the previous two categories enable the representation of system entities, such as citations, authors, and documents, in terms of high-dimensional feature vectors. For example, for each author, one could build a feature vector based on his co-authors, titles of his papers, his publication venues, etc. Each citation could be represented in terms of the tags and notes used for that citation in the different annotation tools. For each chemical paper, we could build a vector representation based on the names of the chemical compounds it contains, and so on. In addition, to the feature vectors associated with each entity, there exist a variety of networks formed by the different entities, such as author-author co-authorship network, author-author citation network, tag-citation networks in the annotation tool web sites, etc.

Computing *feature vectors* that represent system entities (e.g., TF-IDF vectors), and building structures that represent the *networks* formed by these entities (e.g., disk-based edge lists), will enable us to compute various measures of similarity between entities. Typically, the similarity between feature vectors is expressed in terms of the *cosine* measure, while the similarity between the nodes in a network is expressed in terms of *bibliometric* measures, such as the *bibliographic coupling* between two documents which is defined as the (normalized) number of papers that *cite* the two given documents, or *co-citation coupling* which is defined as the (normalized) number of papers that *are cited by* both documents. We intend to implement algorithms that compute such similarity measures for a variety of entity pairs.

D. Relatedness Inference

Computing similarity measures between various entities enables the automated inference of "relatedness". The "relatedness" problems of interest would typically be formulated in two ways:

1. Given a topic, say a set of feature vectors representing chemistry papers, classify each of a large collection of chemistry papers as being "related" or "not-related" to this topic. In the terminology of Machine Learning, this is a typical *classification* problem, and a variety of classification methods, such as Naive Bayes, or SVM which have been found particularly effective for high-dimensional data, can be readily applied through existing software libraries (e.g., Weka—a Java-based Machine Learning library). A slightly different formulation of this problem would be to find as many as possible documents "related" to a given set of documents. We intend to implement, several greedy, best-first algorithms (similar to those used in "focused crawling") which are guided by the measures of similarity between documents in expanding the set of given documents with related ones.

2. Given a set of entities represented by their feature vectors, partition this set into "clusters" of similar entities. For example, we could specify several sets of chemical feature vectors, each representing the "center" of a different topic. We could then wish to place a large number of RSS syndication feeds from the latest issues of various chemistry journals into these predefined clusters, so that we can keep track of the growth of each topic. *Clustering* is of course a very well studied problem and several algorithms (e.g., k-means) and implementations (e.g., Weka) could be readily integrated with the SRG system.

E. Community Building

The ultimate goal of SRG is to support research communities through a variety of community-building tools developed on top of the metadata-extraction tools and relatedness-finding tools. We note that, due to the rich data and metadata handled by SRG, the space of possible tools and services that one can build is quite large. We will initially support the creation of specialized people databases, such as:

- all authors citing a particular compound;
- a list of potential referees which are not in conflict of interest with the authors of the papers submitted to a journal, or conference;
- a list of potential authors for a "special issue" of a journal;

and document databases, such as:

- all papers of a research group;
- all papers appearing in a specific venue;
- all papers on a set of chemical compounds.

As we have discussed earlier, the users of SRG will be able

to access the system tools either through the user interface, or programmatically through the Web-service interface. The later method will allow them to combine the system tools in novel applications tailored to their needs.

V. CONCLUSION

In this paper we discussed the Semantic Research Grid system, which provides a set of tools and services for supporting scientific research. We described the current state of the development of this system and outlined several direction of future work.

REFERENCES

[1] T. O'Reilly, "What is Web 2.0: Design patterns and business models for the next generation of software," 2005. http://www.oreillynet.com/lpt/a/6228.

[2] CiteULike web site. http://www.citeulike.org

[3] Connotea web site. http://www.connotea.org

[4] Bibsonomy web site. http://www.bibsonomy.org

[5] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson, *Web services platform architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and more* Upper Saddle River, NJ Prentice Hall, 2005.

[6] T. Hey and A. E. Trefethen, "Cyberinfrastructure for e-Science," *Science,* vol. 308, pp. 817-821, 2005.

[7] G. Fox, "Collaboration and Community Grids," in *Proceedings International Symposium on Collaborative Technologies and Systems (CTS 2006)* 2006, pp. 419-428.

[8] C. Lee Giles, K. Bollacker, and S. Lawrence, "CiteSeer: An automatic citation indexing system," in *Proceedings 3rd ACM Conference on Digital Libraries (DL'98)*, Pittsburgh, PA, 1998, pp. 89-98.

[9] J. Giles, "Science in the Web age: start your engines," *Nature,* vol. 438, pp. 554-555, 2005.

[10] G. Fox, "Some comments on CiteULike, Connotea and related tools," Technical Report, Community Grids Lab, Indiana University, 2006. http://grids.ucs.indiana.edu/ptliupages/publications/ToolsEvaluation.doc

[11] B. Lund, T. Hammond, M. Flack, and T. Hannay, "Social Bookmarking tools (II): A case study - Connotea," *D-Lib Magazine,* vol. 11, 2005.

[12] A. Mathes, "Folksonomies: Cooperative classification and communication through shared metadata," 2004. http://adammathes.com/academic/computer-mediated-communication/ folksonomies.pdf

[13] L. Gordon-Murnane, "Social bookmarking, folksonomies, and Web 2.0 tools," *Searcher,* vol. 14, pp. 26-38, 2006.

[14] R. Sinha, "A cognitive analysis of tagging (or how the lower cognitive cost of tagging makes it popular)." http://www.rashmisinha.com/archives/05_09/tagging-cognitive.html.

[15] C. Shirky, "Ontology is Overrated: Categories, Links, and Tags ". http://www.shirky.com/writings/ontology_overrated.html.

[16] P. Merholtz, "Clay Shirky's Viewpoints are Overrated." http://www.peterme.com/archives/000558.html.

[17] G. C. Fox, A. F. Mustacoglu, A. Topcu, and A. Cami, "SRG: A research grid for Cyberinfrastructure based scientific research," Community Grids Lab, Indiana University 2006.

[18] V. Petricek, C. I. J., H. Han, I. G. Councill, and C. Lee Giles, "A comparison of online computer science citation databases," in *Proceedings 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL05)*, 2005, pp. 438-449.

[19] H. Li, I. G. Councill, W. Lee, and C. Lee Giles, "CiteSeer^X: an architecture and web service design for an academic document search engine," in *Proceedings 15 International Conference on the World Wide Web (WWW06)*, Edinburgh, Scotland, 2006, pp. 883-884.

[20] T. Sadeh, "Google Scholar versus Metasearch Systems," *High Energy Physics Libraries Webzine,* vol. 12, 2006. http://library.cern.ch/HEPLW/12/papers/1/.

[21] P. Jasco, "Google Scholar: the pros and the cons," *Online Information Review,* vol. 29, pp. 208-215, 2005.

[22] P. Jasco, "Windows Live Academic " 2006. http://projects.ics.hawaii.edu/~jacso/gale/windows-live-acad/windows-live-acad.htm.

[23] R. Tennant, "Is Metasearching dead?," *Library Journal,* 2005. http://www.libraryjournal.com/article/CA622685.html.

[24] B. Quint, "Windows Live Academic Search: The Details," 2006. http://www.infotoday.com/newsbreaks/nb060417-2.shtml.

[25] M. K. Bergman, "The Deep Web: Surfacing Hidden Value " *Journal of Electronic Publishing,* vol. 7, 2001. http://www.press.umich.edu/jep/07-01/bergman.html.

[26] Chembiogrid web site. http://www.chembiogrid.org

[27] Oscar3 web site. http://wwmm.ch.cam.ac.uk/wikis/wwmm/index.php/Oscar3

[28] B. Aleman-Meza, M. Nagarajan, C. Ramakrishnan, L. Ding, P. Kolari, A. P. Sheth, I. B. Arpinar, A. Joshi, and T. Finin, "Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection," in *Proc. 15th International Conference on World Wide Web (WWW'06)*, Edinburgh, Scotland, 2006, pp. 407-416.

[29] C. Anderson, *The long tail: why the future of business is selling less of more*: Hyperion, 2006.

[30] D. S. Rosenblum and B. Krishnamurthy, "An event-based model of software configuration management," in *Proceedings 3rd International Workshop on Software Configuration Management*, Trondheim, Norway, 1991.

[31] Z. Zhuang, R. Wagle, and C. L. Giles, "What's there and what's not? Focused crawling for missing documents in digital libraries," in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, Denver, CO, USA, 2005, pp. 301-310.

[32] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused crawling: a new approach to topic-specific Web resource discovery," *Computer Networks,* vol. 31, pp. 1623-1640, 1999.

[33] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox, "Automatic document metadata extraction using support vector machines," in *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, Houston, Texas, 2003, pp. 37-48.

[34] J. A. Townsend, S. E. Adams, C. A. Waudby, V. K. de Souza, J. M. Goodman, and P. Murray-Rust, "Chemical documents: machine understanding and automated information extraction," *Organic & Biomolecular Chemistry,* vol. 2, pp. 3294-3300, 2004.

[35] H. Han, W. Xu, H. Zha, and C. L. Giles, "A hierarchical naive Bayes mixture model for name disambiguation in author citations," in *Proceedings of the 2005 ACM Symposium on Applied Computing*, Santa Fe, New Mexico, 2005, pp. 1065-1069.

[36] H. Han, H. Zha, and C. L. Giles, "Name disambiguation in author citations using a K-way spectral clustering method," in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, Denver, CO, USA, 2005, pp. 334-343.

[37] I. G. Councill, C. L. Giles, H. Han, and E. Manavoglu, "Automatic acknowledgement indexing: expanding the semantics of contribution in the CiteSeer digital library," in *Proceedings 3rd International Conference on Knowledge Capture*, Banff, Alberta, Canada, 2005, pp. 19-26.

[38] P. T. Corbett, P. Murray-Rust, N. E. Day, J. A. Townsend, and H. S. Rzepa, "Chemistry publications in CML," *Abstracts of Papers of the American Chemical Society,* vol. 231, 2006.