Integration of Collaborative Information Systems in Web 2.0

Ahmet E. Topcu[1,2] , Ahmet Fatih Mustacoglu[1,2], Geoffrey Fox[1,2] , Aurel Cami[3]
[1]Community Grids Lab, Indiana University, Bloomington, IN, 47404, USA
[2]Department of Computer Science, Indiana University
[3]Department of Biomedical Informatics, University of Pittsburgh
{ atopcu, amustaco, gcf }@cs.indiana.edu, camiau@cbmi.pitt.edu

Abstract

We describe a new integration model that uses tools and services for supporting Web 2.0. This integration model defines a structure for a missing feature of Web 2.0. The model integrates a number of existing online tools having a common data model and aims to develop added-value community-building integrated environments. We discuss the overall design, architecture and the components of the integration model, and provide a roadmap of the future work of this model in the Web 2.0 domain.

KEYWORDS: Integration, Web 2.0, information systems, collaboration, academic search.

1. Introduction

The evaluation of the Web shows that people want to access information easily, store them in a personal way, and share them with the others. There are numerous tools and services built in recent years in different categories having Web 2.0 capability. Examples include Social Bookmarking Tools (YouTube, del.icio.us, Flickr,), Blogs (blogger.com, Google Blog), Social Networking Tools (MySpace, LinkedIn) and other related tools. New tools and services are built and open to the Web community continuously. New blogs and data are published every second. The users of these tools have the opportunity to use different tools and decide the best ones in their perspective. Users don't need to know about the version of the tools and services [1]. However, having many tools in similar areas is a problem. If a user wants to use some other tools, how can the user move the data from the previous tool to the new tool? What if the user decides to use similar tools in the same environment and compare information at the same time? In other words, users should have a flexible environment to use multiple tools at the same time. In the current Web 2.0 domain, it is not easy to say that which tools and services are the best because of the large number of existing tools and the continuous development of new tools.

A possible solution may be to define an architecture defining a model for integration to combine similar tools and use multiple services to user community to solve this problem. The web technologies such as RSS (Really Simple Syndication)[2], ATOM[3] AJAX (Asynchronous JavaScript and XML)[4], microformats[5], and REST (Representational State Transfer)[6] provide flexible Web-accessible data and services for Web 2.0 applications. However, although the current systems are for the most part good, they are independent of each other. Huge amount of data distributed over different tools and services exists in the Web A large fraction of this data is duplicated. What is needed is an integration model that would bridge the different tools and services. In the 90s the software and system releases were not frequent. Now, people don't careen to know about version of the software and systems. That is not really needed because today's tools provide services that always improve [1]. There are many tools in Web 2.0 but we are not sure which tools will improve and will be embraced by the web communities. So, in this rapid development cycle one tool might have an advantage to the other tool and vice versa. For example, the annotation tools for scholarly papers are currently detached from the capabilities provided by other research tools.

One of the features of Web 2.0 is the focus on the people. The platform is motivated by questioning how people should interact with each other and easily share data in the Web. The resulting tools are easy to use, and allow people to put information and download them easily. However, there is no such a mechanism to combine them and have richer data or metadata integrated services For example, one metadata captured from one resource may be needed to be stored, shared and uploaded to other tools. This can be achieved using Web services, or Web 2.0 technologies defined earlier such as AJAX and REST. This model is created using native tools and wrappers around them without re-building the tools. Also local capabilities for example local search capabilities can be added and embedded in different client models, such as gadgets. So, the model has the capability to upload information to the tools and download information from them. The model should also provide sharing of logging of users.

In this paper, we describe an integration model and its components for Web 2.0 using web-accessible data and services. This model is motivated by the above concerns to provide flexible mechanism to integrate similar Web 2.0 tools which have similar data model.

The rest of this paper is organized as follows. Section 2 gives an overview of the existing online tools in the Web search domain. They are integrated in the Semantic Research Grid (SRG) prototype model. Section 3 describes the architecture and components of the integration model which is defined for the Web 2.0 platform. Section 4 provides an overview of SRG and its modules. Section 5 presents summary of this integration model.

2. Overview of the Web Search Tools

In our integration model, which will be discussed in Section 3, we use some of the major open-access academic search tools. These tools are discussed next.

2.1 CiteSeer

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 [7]. 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 CiteSeerX [8].

2.2 Google Scholar

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 [9, 10]); (ii) coverage (GS does not provide full coverage of the articles from the publishers that seem to be covered [9, 10]); (iii) accuracy (its metadata extraction algorithms are not very precise, leading to duplicate records, unreliable citation counts, etc. [10]).

2.3 Windows Live Academic

Windows Live Academic (WLA) is one of the online academic search tools like Google Scholar (GS). The service is, as of now, in a beta version. WLA doesn't use citation count as a factor in the determination of relevance. So it does not yet provide *citation indexing* unlike GS and CiteSeer. The initial version of this tool, has been shown to suffer from the same issues of *coverage* and *accuracy* discussed above for GS [11].

3. Integration Model

There is no precise definition of integration in the literature. It is not a property of a single tool but it should have relationship with other tools in the environment [12]. Integration can be categorized into five kinds: (i) *platform*, related with framework services; (ii) *presentation,* concerned with user interaction; (iii) *data,* using information in the tools; (iv*) control*, mechanism for tool communication and interoperation;(v) *process,* related to roles of tools in the systems [12, 13]. The aim of integration is to transform multiple tools into one useful and flexible environment for building communities and to provide multi-functional services to the users. We aim to build such a flexible mechanism by using an integration model on top of Web 2.0 technologies.

The model should have the following capabilities: (i) Tagging and linking of people through uploading and downloading of information; (ii) Sharing information; (iii) Supporting scientific research community; (iii) Integrating the new tools as they are generated in a specific area; (iv) Providing a dynamic environment in which the user can benefit from the capabilities of different tools; (v) Allowing rich content.

The integration model itself doesn't build new tools. It uses the existing tools. One of the application areas is in academic search. In the following section we will define the integration model of similar tools. The key feature is to reuse the tools so that there is no need to rewrite a new tool for specific domain. So, the proposed model should be easier to link together all relating information.

Interoperability for integration is to decide how much work needs to be done for getting data from one platform and use it in other system. Successful integration can be done with respect to interoperability if a system requires having little work to reach data or metadata used in the tools. We define a model that community building systems consist of mechanism to collect information stored in "central" location that offers input/output services. These services should be complete with WSDL (Web Service Definition Language) interfaces to provide wrapper services [14]. These systems should also provide mechanism to have simple internet-scale programming approach such as asynchronous JavaScript and JavaScript Object Notification (JSON), gadgets to make integration powerful and flexible for different systems.

Figure 1 shows the overall architecture of our proposed integration model. This system consists of six components: (a) *Tools*, external web tools to provide services to clients; (b) *Integration Manager*, have information service and provide communication between tools, client, and responsible for integration operation in the system; (c) *Filter*, operate two-way data filtering; (d) *Permission Handler*, checks existing Digital Entity(DE)s permission or build a new permission token for new DEs; (e) *Data Manager*, provides a mechanism to extract data from a repository and insert data into a repository; and (f) *Storage*, maintains user data and permissions in the database.

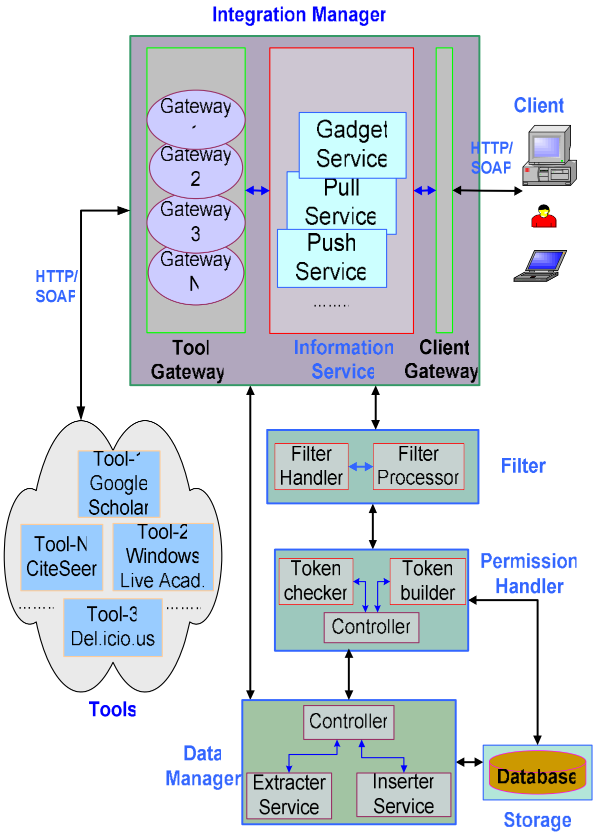

Figure 1. Architecture of the Integration Model

We will explain the key component of the *Integration Manager*. It has two gateways and one core component called *Information Service*. *Tool Gateway* provides a channel between external web tools and *Information Services* such as request a search query or getting response from tools. It provides extensibility for integration system.

Client Gateway provides a mechanism for communication between client and *Information Services*. It gets actions such as search query from client and passes it to the information service to trigger required *Information Service* subcomponents. Another scenario is to pass resultant data coming from *Information Service* to the *Client*.

3.1 Integration Manager

Figure 2 shows the *Integration Manager* and two services of the *Information Services* which are *Pull Service* and *Presentation Service.* It shows also interaction of them with gateways and clients. Pull Service basically interacts with the tools using HTTP or SOAP over HTTP using WSDL through tool gateway to handle client request coming from client gateways. The data communication with tools can be any other HTTP bases services having simple XML message formats or REST style web services. Resultant data which may be in any format such as embedded HTML, RSS feed or any other object. These data coming from tools send to the *Information Handler* again through *Tool Gateway*. *Information Handler* processes the incoming objects in order to extract data or metadata. *Information Handler* use different methods *Gateway* such as heuristics methods to extract data coming from *Tool.* In a heuristic method, data is parsed and extracted for building metadata. *Information Handler* provides extracted data required to build new metadata. Metadata builder builds metadata elements in an XML format. This should be defined as common data format used in this integration systems. Each integration systems should define elements of metadata in order to have successful integration model. We could name this metadata object as Digital Entity (DE). *Presentation Service* provides an interface to display DEs whether coming from web tools or from local integrated systems. Clients interact with these services to do some certain operations such as filtering DEs or insertion to storage or uploading to some other tools. *Presentation Service* has two major components: (a) Simple presentations shows DEs as RSS-style objects; (b) Detailed presentations, shows more metadata elements for each selected DEs in the simple display menu. User can also have option to use different *Information Services* such as event-base[15], and search services in the *Presentation Service*. We have explained the *Pull Services* and interaction with gateways and presentation service in this section. *Push, Gadgets,* and *Local Search Services* are also components of the *Information Service*. These services can be used dynamically and if needed other services can be added to *Information Services* which provides flexibility in *Integration Manager.*

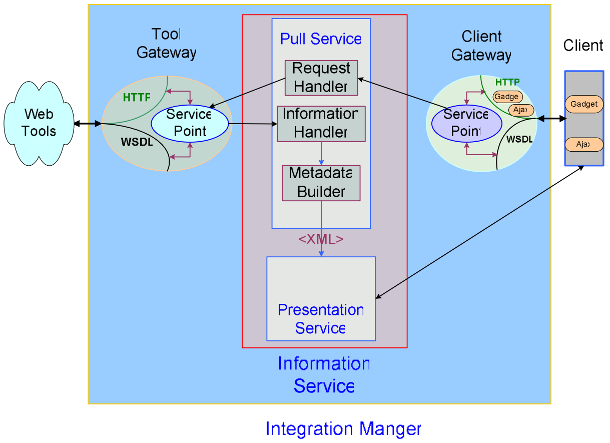

Figure 2. Integration Model and interaction with clients and tools

3.2 Filter

Filter provides two-way capability for reducing number input DEs after using selection operations. Input DEs may come from local search result or as a result of pull or push service operation defined in the *Integration Manager*. Figure 3 shows two-way filter operation.

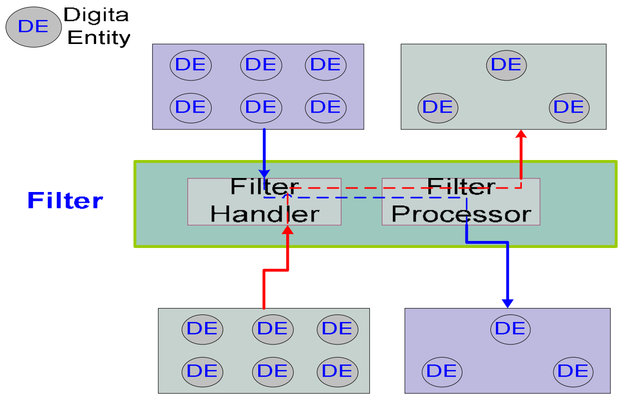

Figure 3. Filter Operation

3.3 Permission Handler

Current Web 2.0 tools don't have a defined clear security model. The restricted access to the resources should be defined and used to protect user community data. Otherwise, without having security model scientific communities suffer from lack of security while using Web 2.0 tools. The model also should still allow using systems without any fine-grained security model. Semantic Research Grid (SRG) project which will be overviewed in section 4 defined a security model using access control matrix and roles [16].Users have ability to define permissions such as *Read, Write* to grant/deny to DEs in the system. This security model can be adopted into the *Integration Model*. *Permission Handler* checks each DE to make sure that user has privilege to access DE. If a user needs to store new DE in the system, the user builds a new permission token for each DE. So, each DE will be protected from other users. A user also can build a security permission tokens for other users for the same DEs. So, users both protect their data and share them with other user.

3.4 Data Manager

This service communicates with the storage through JDBC connection. Controller takes actions to decide whether they are data insertion or extraction operations. *Inserter Service* does insertion operations of the DEs and their permission. *Extraction Service* is responsible for getting DEs and their permissions from *Storage.*

3.5 Storage

All the community building data metadata should be backend by storage.

4. Prototype Model: Semantic Research Grid (SRG) overview

We have applied our proposed integration model to our prototype system called Semantic Research Grid (SRG) described in detail in [16]. The SRG system provides a collaborative environment and it has been built based on the event-based model as explained in detail in [15]The SRG system uses Web 2.0 technologies in its core services and provides extra capabilities to major existing annotation tools and search tools (Delicious[17], Connotea[18], Google Scholar and Windows Live Academic etc.). Tagging and rating are the most common capabilities in most of the Web 2.0 tools, and the SRG system allows its users to annotate/tag the Digital Entities and general URIs in a flexible fashion. Users of the system are also allowed to read, to modify, to update, or to delete a DE based on their access rights. Users of the SRG system have ability to share their DEs with other users or groups in the system by providing the necessary access rights. Our SRG system consists of the following modules: (A) Session and Event Management; (B) Digital Entity Management; (C) Annotation Tools; (D) Search Tools; (E) Authentication and Authorization; (F) Other. A detailed description of the implementation of these modules may be found in [16].The prototype of the SRG system can be accessed from the project demo website [19]

5. Conclusion

In this paper we have shown how one can integrate existing Web 2.0 community and collaboration tools which have a common data model through web services and technologies such as AJAX and REST. This integration model can be used to support different environments where communities can take advantage of the tools in Web 2.0 integrated environments.

References

[1] T. O. R. John Musser, and O'Reilly Radar Team "Web 2.0 Principles and Best Practices," 0-596-52769-1, November 2006.

[2] W. David "RSS 2.0 Specification," http://cyber.law.harvard.edu/rss/rss.html

[3] M. Nottingham. and. R. Sayre (Eds), "The Atom Syndication Format." http://www.ietf.org/rfc/rfc4287

[4] J. J. Garrett, "Ajax: A New Approach to Web Applications." http://www.adaptivepath.com/publications/essays/archives/000385.php

[5] Microformats. http://microformats.org/

[6] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, Irvine USA.

[7] 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.

[8] 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.

[9] T. Sadeh, "Google Scholar versus Metasearch Systems," in *High Energy Physics Libraries Webzine.* vol. 12, 2006.

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

[11] P. Jasco, "Windows Live Academic ", 2006.

[12] B. A. N. Ian Thomas, "Definitions of Tool Integration for Environments," 1992.

[13] I. W. Anthony, "Tool integration in software engineering environments," in *Proceedings of the international workshop on environments on Software engineering environments* Chinon, France: Springer-Verlag New York, Inc., 1990.

[14] G. Fox, "Collaboration and Community Grids " in *Collaborative Technologies and Systems CTS 2006* Las Vegas, 2006.

[15] M. Ahmet Fatih, T. Ahmet E., C. Aurel, and F. Geoffrey, "A Novel Event-Based Consistency Model for Supporting Collaborative Cyberinfrastructure Based Scientific Research," in *Collaborative Technologies and Systems CTS 2007* Orlando, 2007.

[16] G Fox, Ahmet Fatih Mustacoglu, Ahmet E. Topcu, Aurel Cami, "SRG: A Digital Document-Enhanced Service Oriented Research Grid," in *Information Reuse and Integration (IEEE IRI-2007)*, Las Vegas, USA, 2007.

[17] Delicious web site. http://del.icio.us

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

[19] Semantic Research Grid Project (SRG) web site. http://gf6.ucs.indiana.edu:58080/SRGrid