A Service-Oriented Architecture for Digital Libraries

Yves Petinot[1,2], C. Lee Giles[1,2,3], Vivek Bhatnagar[2,3], Pradeep B. Teregowda[2],
Hui Han[1,3], Isaac Councill[3]

[1]Department of Computer Science
and Engineering
The Pennsylvania State University
111, IST Building
University Park, PA 16802
{petinot,hhan}@cse.psu.edu

[2]eBusiness Research Center
The Pennsylvania State University
401 Business Admin. Building
University Park, PA 16802
{vivekb,pbt105,igc2}@psu.edu

[3]School of Information Sciences
and Technology
The Pennsylvania State University
332, IST Building
University Park, PA 16802
{giles}@ist.psu.edu

ABSTRACT

CiteSeer is currently a very large source of meta-data information on the World Wide Web (WWW). This meta-data is the key material for the Semantic Web. Still, CiteSeer is not yet a Semantic-enabled service and therefore its meta-data, although potentially usable by Semantic Web agents, is not yet reachable using the Semantic Web mechanisms. The complexity of CiteSeer, that is the range of tasks it supports, make the transition to a Semantic-enabled service a non-trivial task. While human users tend to perceive CiteSeer as a single well-integrated service, we believe it is best seen – from a machine perspective – as a collection of services, each service performing a specific task. In this paper we show our approach to enable CiteSeer on the Semantic Web in order to allow the use of its meta-data through the Semantic Web. We first introduce an intuitive Application Programming Interface (API) to the CiteSeer software, then show that an efficient integration of CiteSeer in the Semantic Web can be best achieved by independently integrating the services that comprise it. We believe the effort presented here towards the Semantic-integration of a complex Information Retrieval system could be used as an integration model for arbitrary systems.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: *retrieval models.*
H.3.7 [Digital Libraries]: *dissemination, standards, system issues*.

General Terms: Design, Experimentation, Standardization.

Keywords: Service-Oriented Architecture, CiteSeer-API, CiteSeer, digital libraries, interfaces, services, interoperability, Semantic Web.

1. INTRODUCTION

Although much effort has been put into defining a framework for the Semantic Web, one of the practical questions faced by anyone willing to bring their own service to the Semantic Web is: "ok … so how do I do it?". In essence the picture is simple: most services that are connected to the World Wide Web (WWW) can be mapped to a Web-Service by combining (1) an Application Programming Interface (API) to that service; (2) a standard access protocol layer such as the Standard Object Access Protocol (SOAP) [13]; and (3) a layer describing the service in a standard fashion encoded using the Web Service Description Language (WSDL) [14]. The transition from a Web-Service to a Semantic Web Service then requires expressing the processes that comprise that Web-Service using the Ontology Web Language for Services (OWL-S) [10] and any supporting ontology encoded using the Ontology Web Language (OWL) [9]. This recipe is arguably simple and works well for simple services [15].

Here we present our effort to bring CiteSeer and CiteSeer-like services to the Semantic Web. CiteSeer is currently one of the largest sources of meta-data information on the World Wide Web (WWW). This meta-data is the key material for the Semantic Web. Still, CiteSeer is not yet a Semantic-enabled service and therefore its meta-data, although potentially usable by Semantic Web agents, is not yet reachable using the Semantic Web mechanisms. The complexity of CiteSeer, that is the range of tasks it supports, makes the transition to a Semantic-enabled service a non-trivial task. While human users tend to perceive CiteSeer as a single integrated service, we believe it is best seen – from a machine perspective – as a collection of services, each service performing a specific task. In this paper we show our approach to enable CiteSeer on the Semantic Web in order to allow the use of its meta-data through the Semantic Web. We believe the effort presented here towards the Semantic-integration of a complex Information Retrieval system could be used as an integration model for arbitrary systems.

In section 2 of this paper, we discuss our motivations for bringing CiteSeer to the Semantic Web, and envision potential usage scenarios of CiteSeer through the Semantic Web. In section 3 we analyze related work. In section 4 we introduce CiteSeer-API, our initial approach to enabling CiteSeer on the Semantic Web. CiteSeer-API is an – intuitive – Application Programming Interface (API) to the CiteSeer software. We conclude section 4 by showing that CiteSeer-API, although it allows for the easy – programmatical - integration of CiteSeer in third party applications, does not support the eventual integration of CiteSeer in the Semantic Web. In section 5 we show that an efficient integration of CiteSeer in the Semantic Web can be best achieved by independently integrating the services that comprise it. This leads us to reconsidering the organization of CiteSeer itself, which we discuss along with other future work in section 6.

2. MOTIVATIONS

CiteSeer [2,5,6,7] is an automated service that discovers new academic publications on the Web, downloading, converting and processing them to allow end-users to browse the document

ICSOC'04, November 15-19, 2004, New York, New York, USA.
Copyright 2004 ACM 1-58113-871-7/04/0011...$5.00.

collection following cross-document citations and to identify related publications using citation-based features such as co-citation and active bibliography. CiteSeer also supports functionalities for distributed error correction [6], which allow the user community to correct and update the automatically extracted meta-data items.

CiteSeer has thus far been successful in facilitating access to a large amount of meta-data on the World Wide Web. The integration of CiteSeer in the Semantic Web remains however an unresolved issue. CiteSeer servers have been brought to OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) compliance, so that their meta-data collection can now be accessed by meta-data harvesters [8]. However OAI-PMH does neither provide for an extended support for CiteSeer's specific functionalities, nor does it allow for a direct integration of CiteSeer services into the Semantic Web.

With this in mind we set ourselves to designing and implementing the necessary programming and semantic interfaces necessary to enable full programmatical interoperability with CiteSeer services, and ultimately, to fully integrate CiteSeer services in the Semantic Web. Our specific objectives in doing so are listed below.

- Enable programmatical access to the CiteSeer services, including full text document and citation search and citation-based document discovery
- Enable Interoperability with other distributed and heterogeneous Digital Library and Information Retrieval systems.

As presented in the rest of this paper, we achieve our first objective through the implementation of CiteSeer-API (c.f. section 4), an API to the CiteSeer services. To achieve our second objective we reorganize CiteSeer as a collection of Semantic Web Services (c.f. section 5).

3. RELATED WORK

The work presented here addresses the practical issue of providing a machine accessible interface to a complex Information Retrieval system, while integrating this system on the Semantic Web.

Several efforts currently attempt to provide a standard Digital Library and Information Retrieval platform on the Web. The most active efforts in this domain are certainly those from DSpace Federation [16] and Fedora [17]. Both support the OAI-PMH protocols for meta-data distribution. Although Fedora provides management and access APIs, these systems have limited support for seamless interoperability and seamless integration with heterogeneous systems.

The integration of Web-Services in the Semantic Web is clearly one of the most active research areas in the Semantic Web community. Normalization efforts such as OWL and OWL-S lead the way and many services are already advertised using this technology. Still we are not currently aware of any service with the scale and complexity of CiteSeer that would feature a high level of semantic integration and machine-availability.

4. PHASE 1: AN API FOR CITESEER

In order to enable programmatical access to the CiteSeer services, including full text document and citation search and citation-based document discovery, we developed CiteSeer-API on top of the existing CiteSeer software. Following is a detailed description of the methods supported by CiteSeer-API. A comprehensive reference of CiteSeer-API is available at [1].

4.1. Objects URIs

Three concepts are recurrent inside CiteSeer systems : these are Document, Citation and (Citations) Group. As CiteSeer-API intends to give a programmatical vision of any CiteSeer service, these concepts were mapped into programmatical constructs (XML Schema encoding). In order to enable the access to Document, Citation and Group resources in a distributed environment, these three concepts are mapped to object classes and CiteSeer-API assigns to each instance of theses classes a Unique Resource Identifier (URI). The URI formats associated with each type of resource are presented in Table 1.

Table 1: CiteSeer-API Resource URIs Formats

Resource Type	*URI Format*
Document	http://<server>/document/<doc-id>
Citation	http://<server>/citation/<cite-id>
Group	http://<server>/group/<group-id>

Depending on the specific task to be achieved by the client agents, we find it desirable to support various types of resource identifiers (<doc-id>, <cite-id> and <group-id> in Table 1). To that end, we break down document identifiers into two distinct parts: encoding type and value. The encoding type essentially brings semantics to the value field by identifying which algorithm is used to generate the value field from the actual document. Citation and Group identifiers are constructed using the document identifiers as building blocks. We further discuss the creation of relevant Citation and Group identifiers later on in this section. The format of resource identifiers is summarized in Table 2.

Table 2: CiteSeer-API Resource IDs Formats

ID Type	*ID Format*
<doc-id>	<enc-type>:<val>
<cite-id>	<doc-id1>/<doc-id2>
<group-id>	<doc-id>

In the situation where CiteSeer-API is used to sequentially access the entire document corpus of a CiteSeer service – e.g. to train and test some learning algorithm using part or all of the document corpus and associated meta-data – a simple long integer identifier enables the enumeration of the entire collection. To that end we first introduce a "no-encoding" scheme in which the resource identifier values are the actual internal indexes used by CiteSeer server to uniquely identify each Document, Citation and Group resource. The Document, Citation, and Group internal identifiers are simple long integers in the range [1..N_D], [1..N_C] and [1..N_G] respectively. Note that there exists no relation between these three identifiers. As an example, the actual URI identifying Document number 4999 of PennState's CiteSeer.IST would be: http://citeseer.ist.psu.edu/document/no-encoding:4999.

Alternatively we propose a resource URI scheme that uses digital signatures encoding in order to build system independent resource URIs [3]. For a discussion on this URI encoding scheme please refer to [1].

4.2. Search Methods

The Search methods of CiteSeer-API provide a natural entry point to the system, similar to the web-based search form. CiteSeer-API supports both document and citation full text search, each method returning respectively a list of matching document URIs and citation URIs.

- **findDocumentsByText**: document full text search; equivalent to the web-based document search; the search can be modulated using a specific restriction scheme - document body (default), header or title – and ranking scheme – citation count, date, hub, authority. This method returns a list of matching document URIs along with the documents' scores, titles, and query matching context.
- **findCitationsByText**: citation text search; equivalent to the web-based citation search; the search can be modulated using a specific restriction scheme – full citation text (default), title or authors – and ranking scheme – citation count, date. This method returns a list of matching citation URIs along with the citations' scores and texts.

These resources URIs returned by both methods can be used as handlers for the Object-Access methods and bibliography methods described below in order to access related document/citations, just as through CiteSeer's web interface.

4.3. Object Access Methods

Object access methods return the full meta-data records for a resource given its resource URI.

- **getDocument**: retrieve a Document object; properties of the Document resource include: title, author(s), date of addition, document abstract, URL of original file, URL of cached PDF file, URL of cached PS file, URL of CiteSeer page for this document, associated Group URI if any. Compare with getDocumentAsDC (c.f. below).
- **getCitation**: retrieve a Citation object; properties of the Citation resource include: title, author(s), publication date and associated Group URI.
- **getGroup**: retrieve a Group object; properties of the Group resource include: size and list of Citation URIs.

4.4. Bibliography-Oriented Methods

The following methods are all relative to a specific Document *D* in the collection and allow to identify document related to *D* using one of the four citation-based relationships. Each of the bibliography-oriented methods returns basic information on the Document (or Citation depending on availability) along with their Document (respectively Citation) URIs for access to extended information.

- **getCitations**: get Citations made by *D*, i.e. the list of Citations (as identified by their Citation URIs) that comprise the bibliography of *D*. Upon availability cited documents can be located by determining the associated citation Group URI and the associated Document URI.
- **getCitedBy**: get Documents citing *D*, i.e. the list of Documents (as identified by their Document URIs) that have a citation to *D* in their bibliography. All the Documents listed are themselves available from the CiteSeer service.
- **getCoCitation**: get *D*'s co-citation set, i.e. the list of Citations (as identified by their Citation URIs) made by documents that cite *D*. Upon availability the Document URIs of those documents are also returned.
- **getActiveBibliography**: get D's active bibliography set, i.e. the list of Documents (as identified by their Document URIs) bibliography of which overlaps with *D*'s bibliography. All the Documents listed are themselves available from the CiteSeer service.

Note that these four methods provide the information usually displayed on a document's page through CiteSeer web-interface.

4.5. Miscellaneous Methods

CiteSeer-API supports additional utility methods that are not provided by the traditional web-interface of CiteSeer services.

- **getNewDocumentAdditions**: list most recent additions to the document collection maintained by the CiteSeer service. New documents are listed as Document URIs. The user has the ability to constraint the returned list by size – up to a 1000 documents limit - and oldest addition date. This functionality is intended for agents that need to monitor a CiteSeer collection.
- **getDocumentText**: get full ASCII text of a document. In order to perform autonomous citation indexing, CiteSeer servers convert document from their original electronic format to plain text, this functionality gives access to the full text of a document as converted by the CiteSeer server.
- **getDocumentAsDC**: returns RDF [12] statement describing a document, the statement featuring relevant Dublin Core [4] properties.

4.6. Registration and Administrative Methods

In the perspective of enabling access to CiteSeer-like services on the Semantic Web, the action of registering with the API service is also part of the API.

- **register**: allows agents to register with CiteSeer-API, the authentication key required by each method call is then sent to the specified e-mail address.
- **getUserProperty**: get user property; allow users to get their profile and preferences information.
- **setUserProperty**: set user property; allow users to update their profile and preferences.

4.7. Accessing CiteSeer-API

As illustrated in Figure 1, CiteSeer-API proposes a new interface to CiteSeer servers which is complementary to the regular web-interface and the OAI-PMH interface. The CiteSeer-API service, which is also HTTP based, is advertised through its WSDL description. The WSDL schema was intentionally kept simple to ensure compatibility with most WSDL toolkits and users are expected to generate access stubs based on the current WSDL description.

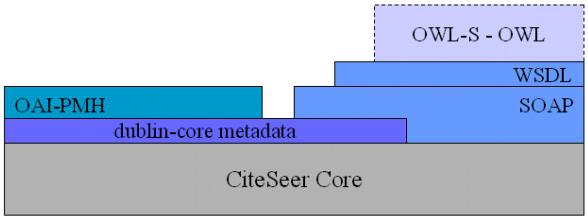

Fig. 1. Protocols Stack for CiteSeer servers

4.8. Semantic Layer

Starting from the WSDL description of CiteSeer-API, the generation of a semantic layer describing CiteSeer-API is straightforward and can be achieved automatically [18]. Now the resulting semantic representation might be satisfying from a developer standpoint, but the concepts involved still limit the machine-usability of such representation. For such a representation to be useful, it is necessary to support it with a semantization of the entire process performed by CiteSeer-like servers – i.e. Autonomous Citation Indexing (ACI) [7]. We currently do not have any solution to reach a satisfying and thorough abstraction of the ACI process. This would especially require a syntactic document ontology for academic documents.

5. PHASE 2: CITESEER AS A COLLECTION OF SEMANTIC SERVICES

In order to overcome the obstacle described in the previous section and to further allow the integration of the CiteSeer services in the Semantic Web, we choose to take an approach different from the one that led to the creation of CiteSeer-API. While CiteSeer-API presents CiteSeer as a monolithic service, here we rethink the organization of CiteSeer as a collection of (Web) services, some of them being specific to the CiteSeer, some of them being common to most Information Retrieval systems. This section presents or motivation for doing so, describes the elementary services comprising CiteSeer, and finally discusses how those services can be taken advantage of by agents on the Semantic Web.

5.1. Motivation

As discussed earlier, CiteSeer-API allows users to programmatically access the CiteSeer service in pretty much the same way a human user would through CiteSeer's regular Web interface [7]. Hence CiteSeer-API merely presents the CiteSeer service as a search engine service coupled with a bibliographical database. Now we can question whether CiteSeer will be used on the Semantic Web in the exact same way it is currently used by human users.

We believe the actual needs of agents on the Semantic Web will be somewhat different. For instance a Semantic Web Agent could simply be interested in downloading a cached document. In that situation it is common to locate a document hosted by CiteSeer through a generic search engine such as Google. Another example is that of a Semantic Web Agent that needs to extract citations from a piece of raw text : here, most of the CiteSeer service is irrelevant to the task, and unless this agent is willing to submit the full document to the CiteSeer for processing, the task cannot be performed. Following this idea it seems that decoupling the functionalities proposed by CiteSeer is the best approach to follow, each elementary functionality being mapped to independent Web Services. Doing so, agents on the Semantic Web will gain access to the most unique functionalities of CiteSeer service, which are automated citation analysis and document interlinking based on citations.

5.2. A Federation of CiteSeer Services

We propose a new organization for the CiteSeer that allows it to offer each of its elementary functionalities as Web Services. Following this approach not only facilitates the integration of those functionalities on the Semantic Web, it also provides support for (1) the distribution of those functionalities across the internet; (2) the discovery of those services by Semantic Web agents; and finally (3) their activation and just-in-time integration by those agents.

The elementary functionalities of CiteSeer can be divided into two categories. The first category is that of the functionalities which are specific to the CiteSeer service and which therefore are the most valuable regarding their integration on Semantic Web. The second category is that of the functionalities currently integrated in the CiteSeer but that are not specific to the application as they are recurrent to many Information Retrieval systems. Figure 2 summarizes our vision of a service-based architecture for CiteSeer. The most recent service descriptions (WSDL and OWL-S) for each service are available at [1].

5.3. Fundamental CiteSeer Services

The following services are specific to CiteSeer and would provide the actual value-added if made available on the Semantic Web. They are those services that enable the processing of citations and the navigation through those citations.

Citation Extraction Service

The Citation Extraction Service provides the autonomous citation parsing functionality. The service is invoked by sending a raw string of text which is expected to contains references (citations) to other documents/entities. In return the service will return a constructs identifying all the documents referenced. The construct specifically provides a URI of the resource if available.

Citation Graph Service

The Citation Graph Service maintains a citation graph. A citation graph is a directed graph where nodes are actual documents/entities and edges represent the citation relationship between documents. The service provides primitives to walk the graph following citations from one node to another, that is by using cites/cited-by relations. The service may also provide functionalities to explore the graph based on extended citation-based relationships such as co-citation or active bibliography.

Utility Services

Aside from the services that give its actual value to CiteSeer, some utility services are of use to the CiteSeer to perform regular operations, and from a general perspective, to manage its document collection. As document collections will become more and more distributed, presenting such functionalities as services is also a step towards the creation of a fully distributed CiteSeer system. We feel the following services are relevant to most Information Retrieval systems and so we encourage the reuse of their definitions in other projects.

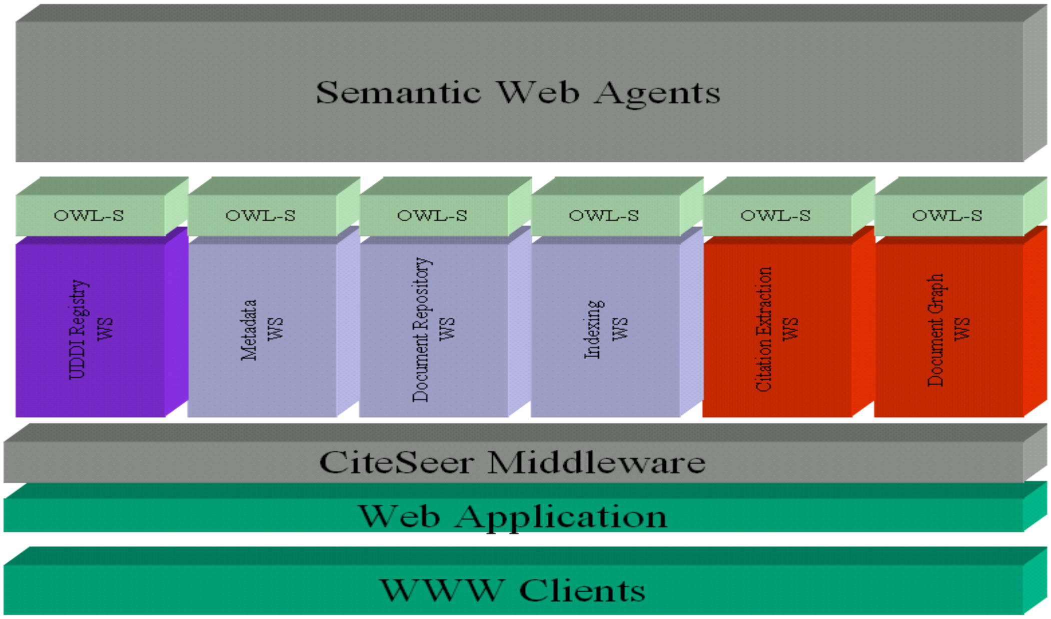

Fig. 2. A service-oriented architecture for CiteSeer. CiteSeer, as an application, is a complex coordination of several elementary services. By identifying these services and reorganizing its architecture around them, we simplify the integration of CiteSeer in the Semantic Web.

Indexing Service

The Indexing Service provides the generic functionality of an inverted file, mapping elementary tokens – e.g. word, date, etc.- to documents/entities. In the context of CiteSeer, two instances of this service are used to independently index documents and their citations [7]. It is interesting to see that, as we extend CiteSeer to deal with additional semantic objects – e.g. acknowledgements – we can take advantage of the service granularity to add a new index service for those objects without affecting existing indexes.

Metadata Service

The Metadata Service associates metadata records to all the objects managed by CiteSeer. Hence we can organize this service as an RDF [12] repository. In the context of CiteSeer and Digital Libraries, the Metadata Service becomes equivalent to the service providing the corresponding OAI-PMH interface [8,11].

Electronic Repository Service

The Electronic Repository Service provides the functionality of a document repository which contains all the documents that are available from the CiteSeer. One important feature of this Service is that it is aware of file duplicates. As CiteSeer progressively evolves towards a federation of such repositories, the service nature of the electronic repository will permit many such services to join in the federation, hence providing an effective solution to concerns such as copyright management or mirroring. We currently consider an implementation of this service based on the SDSC Storage Resource Broker (SRB) [21].

Electronic Conversion Service

The Electronic Conversion Service provides the necessary conversion facility for converting documents from a given electronic format to another. This service is required by the CiteSeer to handle documents in various electronic formats such as PDF and PostScript. Note that by conceiving this functionality as a service, support for new electronic formats in the CiteSeer is immediate, provided the associated conversion service is made available at some location over the Internet.

Duplicate Identification Service

The Duplicate Identification Service provides a functionality for checking the similarity between any two documents. We believe this service is essential to most Information Retrieval systems, especially Digital Library systems, as it allows to check the amount of overlap between any two document contents and therefore allows the identification of duplicates at the content level. Providing this functionality as an independent service allows to use it for other valuable applications such as discovery of alternate URIs for any given resource as identified by one of its URIs or one of its digital signatures.

5.4. Service Registration

The elementary Web Services presented in the previous part can be used by agents - including the CiteSeer application through its middleware (see below) - to perform their intended task. However in order to discover those services a meeting point must exist so that those Web Services can be advertised and discovered. To achieve this we follow the standards of Web Services organization and include in our architecture a UDDI registry [20]. Although we could rely on a third party registry, we choose to host a local registry which will be used by the CiteSeer Middleware itself to manage the elementary services it can resort to. In order to enforce the service interfaces that are acceptable by the system, we make use of UDDI tModels to impose the set of acceptable WSDL service interfaces [20]. We discuss the CiteSeer Middleware in more details in the next section.

5.5. CiteSeer Middleware

While we extracted the elementary services out the monolithic CiteSeer application, an important requirement for us is to continue providing the CiteSeer service as it is known by Web users. To this end we encode the logic of the CiteSeer application as a specific middleware application [19]. Using the Web Service registry presented in the previous section, the CiteSeer Middleware has now access to all the necessary functionalities to achieve its task. The middleware only has to encode the coordination between those elementary services in order to provide support for the original CiteSeer service. In turn, the CiteSeer middleware is available through the CiteSeer Web application which plays the role of the presentation tier.

6. FUTURE WORK

The work presented here focuses on the integration of a single, isolated, CiteSeer server into the Semantic Web. In order to effectively deal with the issues of scalability and copyright management, Digital Library systems progressively evolve towards strongly distributed systems. The requirement for distribution is especially strong for the Document Repository service which would allow to federate several document repositories as a single CiteSeer system, in line with what is currently proposed by DSpace [16]. We also wish to improve on the distributed character of some of the services presented here, especially the Citation Graph service. Doing so would allow to seamlessly combine citation graphs of independent document collection, hence converging towards a CiteSeer that applies to all publications domains, while independent graphs will remain independently maintained for their domain specific collections [11].

7. CONCLUSION

In this paper we presented our effort towards the integration of CiteSeer into the Semantic Web. We first introduced CiteSeer-API, a programmatical API to CiteSeer services. While CiteSeer-API allows the integration of CiteSeer's search and navigation functionalities in third party applications, it does not permit access to the elementary functionalities of CiteSeer. We believe that those functionalities are indeed what future agents will need from CiteSeer on the Semantic Web. Consequently, instead of proposing a monolithic semantic description of CiteSeer services, we choose to semantically describe the basic services on which CiteSeer relies. Doing so not only simplifies the process of semantically describing the CiteSeer, it also allows Semantic Web Agents to make use of those very services that hold the value-added conveyed by CiteSeer.

The Semantic Web is a valuable opportunity for large Web-Services, as it offers them an opportunity to reflect on their own organization and what their value-added really is. We believe the case presented here of turning CiteSeer into a truly Semantic Service can be used as a model for other services of similar complexity.

We encourage research groups to take advantage of both CiteSeer-API and the CiteSeer Web Services in order to enhance their applications with the CiteSeer dataset and associated functionalities.

8. ACKNOWLEDGMENTS

We acknowledge partial support from NSF and from the eBusiness Research Center at the Pennsylvania State University. We also wish to thank Dr. Steve Lawrence for his contribution to this work.

9. REFERENCES

[1]: CiteSeer-API, http://citeseer.ist.psu.edu/api/

[2]: CiteSeer.IST, http://citeseer.ist.psu.edu/

[3] : Crespo, A.; Garcia-Molina, H.. Archival Storage for Digital Libraries, Third ACM Conference on Digital Libraries. Pittsburgh, PA, USA, June 23-26, 1998

[4]: Dublin Core Metadata Initiative, http://dublincore.org/

[5]: C.L. Giles, K. Bollacker, S. Lawrence, "CiteSeer: An Automatic Citation Indexing System", In *Proceedings of the 3rd ACM Conference on Digital Libraries (DL'98)*, pp 89-98, 1998.

[6]: S. Lawrence, K. Bollacker, C.L. Giles, "Distributed Error Correction", In *Proceedings of the 4th ACM Conference on Digital Libraries*, p. 232, 1999.

[7]: S. Lawrence, K. Bollacker and C.L. Giles, "Indexing and Retrieval of Scientific Literature", In *Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM 99)*, pp 139-146, Kansas City, Missouri, November 2-6, 1999.

[8]: "The Open Archives Initiative Protocol for Metadata Harvesting", http://www.openarchives.org/OAI/openarchivesprotocol.htm.

[9]: OWL Web Ontology Language Reference, http://www.w3.org/TR/2004/REC-owl-ref-20040210/

[10]: OWL-S, http://www.daml.org/services/owl-s/1.0/

[11]: Y. Petinot, P.B. Teregowda, H. Han, C.L. Giles, S. Lawrence, A. Rangaswamy and N. Pal, "eBizSearch: an OAI-Compliant Digital Library for eBusiness", In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL 2003)*, pp 199-209, Houston (TX), May 2003.

[12]: Resource Description Framework, http://www.w3.org/RDF/

[13]: Simple Object Access Protocol, http://www.w3.org/TR/soap/

[14]: Web Service Description Language, http://www.w3.org/TR/wsdl

[15] A. Ankolekar et al., "DAML-S: Web Service Description for the Semantic Web," *Proc. 1st Int'l Semantic Web Conf.* (ISWC 02), 2002.

[16]: DSpace Federation, http://www.dspace.org/

[17]: Fedora, http://www.fedora.info/

[18]: M. Paolucci, N. Srinivasan, K. P. Sycara, T. Nishimura, "Towards a Semantic Choreography of Web Services: From WSDL to DAML-S", In Proceedings of the International Conference on Web Services (ICWS 2003), pp 22-26, 2003.

[19]: D. McComb, "Semantics in business systems: the savvy manager's guide: the discipline underlying web-services, business rules, and the semantic web", Morgan Kaufman, 2004.

[20]: UDDI Spec TC, "Using WSDL in a UDDI Registry, Version 1.08", http://www.oasis-open.org/committees/uddi-spec/doc/bp/uddi-spec-tc-bp-using-wsdl-v108-20021110.htm

[21]: Homepage SDSC Storage Resource Broker (SRB), http://www.npaci.edu/DICE/SRB/