CiteSeer-API: Towards Seamless Resource Location and Interlinking 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 Administration 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

We introduce CiteSeer-API, a public API to CiteSeer-like services. CiteSeer-API is SOAP/WSDL based and allows for easy programmatical access to all the specific functionalities offered by CiteSeer services, including full text search of documents and citations and citation-based document discovery. In order to enable operability and interlinking with arbitrary software agents and digital library systems, CiteSeer-API uses digital content signatures to create system-independent handles for the Document, Citation and Group resources of CiteSeer servers. We discuss specific functionalities of CiteSeer-API that take advantage of these handlers in order to enable seamless location of CiteSeer resources. Finally we argue that the digital signature scheme used by CiteSeer-API is well suited for the creation of machine-usable semantic descriptions of digital library services which is the key toward seamless discovery and integration of services such as CiteSeer-API. CiteSeer-API is currently showcased on CiteSeer.IST, the CiteSeer server of the School of Information Science and Technology at the Pennsylvania State University.

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

CiteSeer-API, CiteSeer, digital libraries, interfaces, services, interoperability, interlink, Semantic Web.

CIKM'04, November 8-13, 2004, Washington, DC, USA.

1. INTRODUCTION

Digital Libraries (DL) systems remain strongly proprietary in the way they collect, index, store, and present their document collections. This phenomenon is usually unavoidable as different digital library systems tend to address different content types – e.g. textual content vs. multimedia - and are consequently tuned towards these contents and their specific audiences. Efforts for access normalization, such as that of the Open Archives Initiatiative's Protocol for Metadata Harvesting (OAI-PMH) [20] address the issue of presenting collections in a standard format that allows, if not interoperation of those systems, at least the creation of meta-systems able to virtually aggregate many heterogeneous collections. The interoperation of digital library systems themselves is however not addressed by those efforts.

We introduce CiteSeer-API [3,33], a SOAP/WSDL-based [28,30] API to CiteSeer-like servers [3,4,8,13] that, in addition to enabling programmatical access to CiteSeer functionalities, supports system-independent primitives that allow arbitrary agents and digital library systems to effectively interoperate with CiteSeer-like services. In the context of digital library services we refer to interoperation as the ability for an agent to locate a specific digital resource hosted by a digital library while having no knowledge of the internals of that digital library. By providing such functionalities on top of the CiteSeer server, CiteSeer-API offers many opportunities for digital library systems to localize and interlink with CiteSeer-hosted resources.

The rest of this paper is organized as follows. In section 2 we address the general issue of making digital library services more interoperable and discuss API requirements to allow interoperation between digital library systems. Based on this discussion, Section 3 defines a model for resource handlers that is based on digital signatures and that permits CiteSeer-API to provide system-independent access to CiteSeer-hosted resources. The standard functionalities of CiteSeer-API for search and citation-based document discovery are described in Section 4. Section 5 puts a special emphasis on CiteSeer-API functionalities that enable interoperation of CiteSeer services with arbitrary agents and digital library systems. Section 5 also describes a simple semantic framework that enables the use of CiteSeer-API through the Semantic Web. In section 6 we discuss potential applications of CiteSeer-API for access and visualization, interlinking, mirroring and version control applications. We

conclude in section 7 with related work and a roadmap towards further integration of CiteSeer-API in the Semantic Web.

We encourage research groups to register at [3] and take advantage of the terabyte scale data-set available through CiteSeer-API.

2. INTEROPERABLE DL SYSTEMS

In this section we take a general perspective on digital libraries interoperation. In the context of digital library services we refer to interoperation as the ability for an agent to locate a specific digital resource hosted by a digital library while having no knowledge of the internals of that digital library. In the following we consider the necessary features that a digital library API should provide in order to enable this level of interoperability. First we address the specific case of interoperability between CiteSeer-like services then the issue of interoperation between CiteSeer services and arbitrary agents and digital library systems.

2.1. Overview of DL interoperation

Current digital library systems, including CiteSeer servers, do not allow for easy interoperability. From the perspective of service designers, a very desirable feature for a digital library system is its ability to let other agents – or services - automatically locate a specific resource it hosts. Hence in the case of CiteSeer services, the ability to link to other bibliography-focused services is important in order to provide results that are as complete as possible. CiteSeer currently links to DBLP [9] and HomepageSearch [15] to enhance its author and homepage information respectively. However the linking to these external services is often more informative than precise as the links take the form of queries to those services and there is therefore a limited confidence in the fact that the searched author or homepage is actually listed by those services or, if it is, that it is a valid answer to the user's original query. The weakness of the previous linking approach is that it relies on keyword-querying the services we want to link to. While this certainly allows for rapid and automatic linking, a usual service won't return a single link matching the request but instead a list of the top N matches for that query. Without extra intelligence or the introduction of concept search [19] or semantic search [14], it is unlikely that linking to search engine services can be achieved more efficiently. The issue of linking to digital libraries is however more approachable as such systems effectively hold content, and therefore any agent with sufficient knowledge of the resource it wants to link to should have the ability to do so without having an understanding of the DL internals. In this section we consider the requirements on DL interfaces to enable resource location. We address the specific requirement for interlinking between CiteSeer services and generalize to interlinking between CiteSeer services and arbitrary DL systems.

2.2. Requirements for DL Interoperability

Digital Library systems are geared toward the task of managing collections of digital objects and their associated metadata. Each digital object is meta-tagged [10] in order to provide additional information on its content and its relatedness to other digital objects. Current efforts for normalization of content access [29] and presentation [20] build on top of the metadata layer provided by digital library and information retrieval systems. However metadata-based access to digital libraries reduces the location process to a search-engine query which, as discussed earlier, does not enable efficient interlinking with the digital library resources unless additional logic is provided.

Here we want to provide support for resource linking with a good level of confidence and not by simple query-forwarding. As such we propose an alternative for large-scale resource location and interlinking.

The interoperability of digital library systems relies on their ability to provide APIs for non-human access to their content. Still as outlined above, existing interfacing standards simply shift the human search problematic to an agent search problematic, not simplifying the location of digital objects themselves. Digital library resources are best represented by digital signatures which uniquely identify them in the digital space and which can be computed directly from the original digital resources [7]. Consequently we propose an extension to traditional DL API functionalities where agents are allowed to search the digital repositories using digital signatures. In the rest of this section we further discuss the relevance of using digital signatures as API object handlers and show how CiteSeer services can take advantage of such search features to automatically interlink. We finally describe additional features that enable interlinking from heterogeneous DL systems with CiteSeer services.

2.3. Digital Signatures As Object Handlers

Most digital library systems – including CiteSeer services - tend to assign internal – arbitrary - identifiers to the resources they manage. Although this is a perfectly acceptable practice in a non-distributed environment, it becomes much of a problem when considering the issue of interlinking collections from two or more distributed digital libraries. The internal identifiers usually convey no information on the resources themselves hence preventing immediate cooperation with other DL systems. Note that this issue remains even if digital library systems make use of Document Object Identifiers (DOIs) [34] as handles for their electronic objects: even though each DOI represents an agreed-on/standard object identifier, it has no direct relation to the original document and requires the use of a resolution service to map those "opaque strings" [34] to actual resource locations and/or associated metadata.

One of our goals in designing CiteSeer-API is to enable the interoperation of heterogeneous digital libraries. We believe that - from the API standpoint – resource handles should be features that can be computed directly from the original digital resources and should therefore be implemented as checksums or CRCs of these resources. By using this approach, distributed system can, without any communication being required, compute the same identifier for any given document. Document URIs created from the document checksums will allow heterogeneous DL to locate CiteSeer resources while having limited knowledge of the system itself. Although a fully-fledged discussion would be necessary on this subject alone we can argue that for most digital resources, including those that do not actually have a digital existence, one could determine an acceptable digital signature that unambiguously represents that resource. For example if a digital library manages author resources, a good representation is to use the public PGP key of the associated individual as a resource handler, preferably to the actual author name which would fall in the metadata category.

2.4. Interoperation between CiteSeer services

Independent CiteSeer servers currently do not have the ability to interlink with each other. For instance consider the case of eBizSearch [12] a CiteSeer-like search engine for e-Business publications. Assume a document *A* is indexed by eBizSearch. Document *A* cites a document *B* which is not indexed by eBizSearch but which is indexed by CiteSeer [5]. How can eBizSearch locate document B in CiteSeer's repository with a high level of confidence ? Since eBizSearch holds a document that cites document B, it has a – possibly non-canonical – bibliographical entry for document B. Making use of the citation search functionalities of CiteSeer-API (section 3), eBizSearch can locate a matching citation in CiteSeer's database and access the corresponding document. Although this approach is likely to work as expected, we choose to extend the standard functionalities of CiteSeer-API with a method that takes as its input a bibliographical entry in its raw textual form and returns the matching Document URI only if the document is available from the service. Compared to the keyword based search methods, which would propose alternative matches, this method is designed to locate the exact resource if it is available, or to inform the client agent that it is not available otherwise.

2.5. Interoperation with arbitrary agents

Finally we consider the case of an arbitrary agent wishing to locate a specific resource hosted by a CiteSeer server. In that case the bibliography lookup method outlined in 2.4 may be sufficient in most cases. However one can envision scenarios in which the client agent does not possess any sufficient metadata on the resource it is searching for and simply possesses a digital signature for that resource. In that case, it is desirable to perform a lookup against the CiteSeer service using the resource digital signature. If the resource is available or known to the service, it returns its Document URI. Otherwise the service informs the client that the resource is unknown.

3. CITESEER-API : AN API FOR CITESEER SERVICES

CiteSeer has established an original Web-interface model where bibliographical references of academic publications are mapped to hyperlinks, allowing a given document collection to be browsed by following citations from one document to another. In this section we give a detailed presentation of CiteSeer-API, an API to CiteSeer services that provide programmatical access to these CiteSeer-specific functionalities. Although CiteSeer servers have been brought to OAI-PMH compliance so that their metadata collection can be accessed by metadata harvesters [23]. many of their functionalities cannot be accommodated by OAI-PMH, including full text document and citation search and citation-based document discovery. Our motivations for CiteSeer-API are therefore (1) to provide programmatical access for all the functionalities supported by CiteSeer-like systems; (2) to enable interoperability of CiteSeer services with distributed and heterogeneous DL systems as discussed in section 2. Following is a detailed overview of the functionalities supported by CiteSeer-API. A full reference for these functionalities and registration to this service are available at [3].

3.1. CiteSeer Organization Overview

Three concepts are recurrent inside CiteSeer systems : these are Document, Citation and Group. As CiteSeer-API intends to give a programmatical vision of any CiteSeer service, these concepts were mapped into programmatical constructs (XML Schema encoding). A collection of Documents instances $\{D_i\}$ maintained by a CiteSeer-like service is organized as follows. Each Document instance D_i contains a set of Citation instances $\{C_{ij}\}$ that refer to other documents $\{d_j\}$ that may or may not be part of the collection. Each C_{ij} uniquely identifies a bibliographical entry in D_i's reference section, meaning that a reference C_{kj} - in another Document instance D_k - to the same document d_j is such that $Cij \neq Ckj$. Citation instances $\{C_{ik} : k \in [1..K]\}$ that refer to the same document D_k are however grouped under a single Group instance G_α. Thus the mapping of a Citation instance C_{ik} to the Document instance D_k is practically seen as the mapping of the associated Group instance G_α to the Document instance D_k. In case the referenced document d_k is not part of the collection, the Group instance G_α is not mapped to any Document instance.

3.2. CiteSeer Object URIs

In order to enable the access to Document, Citation and Group resources in a distributed environment, the three concepts discussed above are mapped to object classes and CiteSeer-API assigns to each instance of these classes a Unique Resource Identifier (URI). The URI formats associated with each type of resource are presented in Table 1. Note that the URI formats presented in Table 1 could fit in the OpenURL [35] specification.

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 metadata – 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 #4999 on the CiteSeer.IST server at PennState University 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. In the rest of this section we discuss the creation of such resource URIs and their relevance towards interoperation with arbitrary agents and digital library systems.

3.3. Digital Signature based Identifiers

In order for digital libraries to be able to cooperate, the choice of document indices / document pointers should be made in a way such that two DLs can independently compute the exact same index value for the same document [7].

The CiteSeer software package makes use of the SHA [26] algorithm in order to prevent additions of exact file duplicates in the system. Upon completion of the download phase, a 32 bit string is computed from the binary file and used as a lookup key in CiteSeer's checksums database. If the exact digital resource has been submitted before, there will be an entry associated with that key so that the new submission can be discarded safely.

Here we choose to adopt the SHA algorithm in order to generate the URIs advertised at the API level: this is consistent with the internals of the CiteSeer software and allows for a readily availability of checksums inside CiteSeer-API. Using this checksum-based approach and according to Table 1 and Table 2, the URIs will now take the following form : http://<server>/document/SHA1:<doc-sha1-checksum>.

Citation and Group resources, on the other hand, are by-products of the information extraction process. As such it is unlikely that two heterogeneous systems will generate identical objects from a binary representation point of view. While it makes sense to compute a checksum for a document file and assign it to the corresponding Document resource, Citation resources are parsed out - with more or less accuracy – from the document text and are thus artifacts from the CiteSeer software. Citation resources are therefore objects that are not shared by heterogeneous systems, if only because the parsed data is candidate for community correction. It makes more sense to consider Citation resources under a different approach, that is as directed edges from one document to another. So far our scheme assigns a checksum to each document in the collection. We therefore extend this scheme by assigning to each Citation resource an ordered pair of checksums (C1, C2) that indicate respectively the fact that the document with checksum C1 cites the document with – class representative - checksum C2. Note that this scheme does not assume any particular medium or electronic format and therefore can readily be applied to any digital library system. Of course the document being cited is not always available in the CiteSeer collections, and therefore we create a semantic object called "UnknownDocument" that will serve as a placeholder in these situations. As can be seen in Table 2 our URL formatting supports the use of different checksum algorithms for the source and the sink of the citation relationship, which can be necessary in the situation where heterogeneous digital library systems – e.g. CiteSeer-like service with non-CiteSeer service - need to interoperate.

As explained in 3.1, each Group resource regroups Citations to a given Document. As formulated above, each Citation in this Group is an ordered pair of Document checksums. While the sources of these Citations can be any Document, the sink is a single document for which this Group holds the Citations. Therefore a Group identifier should in effect be no different than the identifier for the Document for which it stands. With this in mind we simply define group-ids to be actual document-ids (Table 2). Note that CiteSeer internals do not follow this approach, the reason for this being that arbitrary index values are used instead of checksums in order to identify documents, citations and groups.

3.4. Problematic of Duplicate Documents

The issue of duplicate documents has a direct impact on the use of a scheme for resource identifiers based on digital signatures. Duplicate documents can take two forms : (1) exact file – or binary representation - duplicates and (2) exact content duplicates. The first case is actually a false problem as exact file duplicates will, by definition, have identical digital signatures. In the second case however we have to deal with the issue of having non-identical files that contain the exact same content. The CiteSeer software deals with this situation by performing a comparison at the sentence level of all the documents it indexes: when the threshold of 99% sentence co-occurrence is reached for a pair of documents, then these are considered exact (content) duplicates. For non-textual content it is arguably also possible to design analogous algorithms that can, based on specific similarity/identity criteria, identify exact content duplicates.

To address this issue, digital library services should maintain, for each document, the list of digital signatures that this document is known to have. By doing so digital libraries can account for the existence of a given document in various digital formats and encodings. A potential application to supporting such feature is that of versioning control (section 6.4).

The implication for the resource URIs presented above is that digital signatures that represent distinct files with identical content should be treated as equivalent when writing or reading these URIs. We aim to extend CiteSeer-API in order to support a protocol that allows cross-DL negotiations in order to identify alternative identifiers for a given content.

The URIs that have been described in this section are used as both return values for the search methods and as input parameters for the resource access methods of CiteSeer-API.

4. CITESEER-API STANDARD METHODS

Following is a detailed description of the methods supported by CiteSeer-API. A comprehensive reference is also available at [3].

4.1. 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.

4.2. Object Access Methods

Object access methods return the full metadata 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 (4.4).
- **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.

4.3. Bibliography-Oriented Methods

The following methods are all relative to a specific Document *D* in the collection and allow to identify documents 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.

4.4. 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 constrain 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 [25] statement describing a document, the statement featuring relevant Dublin Core properties.

4.5. 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.

4.6. Accessing CiteSeer-API

As illustrated in Figure 1, CiteSeer-API offers 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.

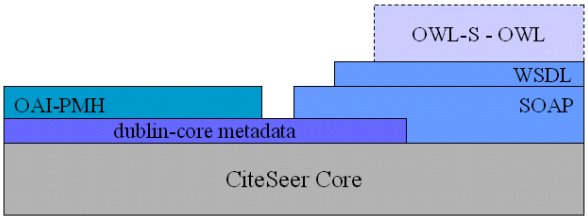

Figure 1: Protocols Stack for CiteSeer servers

5. CITESEER-API SUPPORT FOR INTERLINKING

As motivated in section 2, CiteSeer-API needs additional functionalities than the standard ones for search and retrieval presented in the previous section in order to facilitate the location of its resources and therefore to allow interoperation and interlinking with heterogeneous distributed services. In this section we focus on the API functionalities that allow for dynamic lightweight interlinking of DL systems with CiteSeer services. The interlinking functionalities provided through the API allow to determine document URIs based on either a reference entry for the document or a checksum of the actual document files. We discuss how such functionalities can be taken advantage of by third-party digital library systems and CiteSeer services themselves in order to achieve higher – autonomous – interlinkage between those services.

5.1. Bibliography Lookup Service

We introduce a bibliography lookup method for CiteSeer-API that extends the range of services provided by CiteSeer servers. The bibliography lookup method provides access to CiteSeer's functionality for citation parsing and corresponding paper identification. If so far this functionality has been exclusively used internally by the CiteSeer software to process bibliographical entries, and identify cited papers, we believe that, as Web-services, this is an extremely relevant functionality to be provided by CiteSeer servers since this is the elementary feature on which they rely. By turning the citation analysis into a service we seek to allow online use of CiteSeer's algorithms, hence enabling arbitrary clients to integrate CiteSeer functionalities into their own applications.

The bibliography lookup method is named **lookupBibliography** and works as follows. Client agents send a raw bibliographical entry (simple string) to the service (Figure 2). In response the service returns a flag indicating whether the bibliographical entry is known, and if it is, an RDF statement with Dublin Core properties for the corresponding document (as identified by its Document Resource URI, c.f. section 3.2).

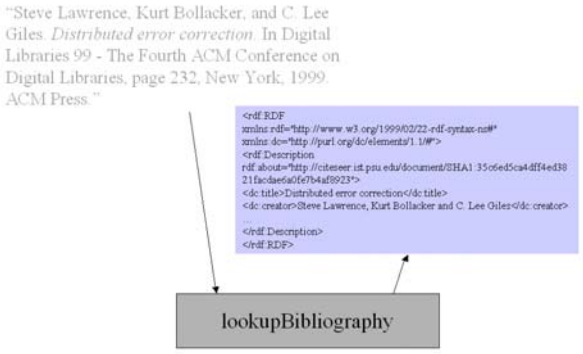

Figure 2: CiteSeer-API's bibliography lookup service

Future versions of this functionality will allow users to specify which citation parsing algorithm they want the CiteSeer service to use. Currently only the standard parsing algorithm used by CiteSeer [16] is available. We are also considering extensions to this functionality that would allow clients to obtain the canonical form of the bibliographical entry they lookup.

5.2. Digital Signature Lookup Service

In line with our attempt to create resource URIs that are system independent (Section 3.3) we introduce a digital signature lookup method for CiteSeer-API that allows for direct interoperation between heterogeneous digital library systems. Using this functionality, an arbitrary agent can locate a digital resource based on its digital signature(s). Based on our previous discussion (Section 2), we consider such functionality a basic block for seamless interoperation and interlinking between digital library systems.

The digital signature lookup method is named **lookupDigitalSignature** and works as follows. Client agents send a digital signature string (encoded as proposed in Table 2) to the service (Figure 3). In response the service returns, similarly to the bibliography lookup method, a flag indicating whether the digital object is known, and if it is, an RDF statement with Dublin Core properties for the corresponding resource (as identified by its Document/Citation Resource URI, c.f. section 3.2).

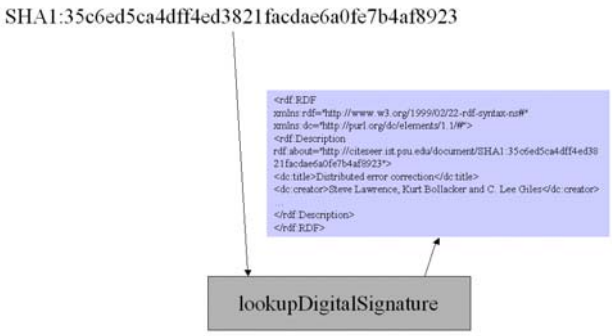

Figure 3: CiteSeer-API's digital signature lookup service

5.3. A Simple Semantic Framework

We finally provide a simple semantic framework for digital objects, digital signatures and citation relationships between digital objects. This framework demontrates the possibility of addressing the semantic description of digital library services using a bottom-up approach, which we believe is the well suited for a "machine friendly " integration of these services. We envision this framework as the corner stone for the introduction of digital library systems into the Semantic Web.

We declare three fundamental OWL classes [21] that allow us to manipulate the concepts discussed in the previous sections.

```
<owl:Class rdf:ID="DigitalEntity"/>
<owl:Class rdf:ID="DigitalSignature"/>
<owl:Class rdf:ID="DigitalSignatureAlgorithm"/>
```

The DigitalSignature class presents two properties for defining which DigitalSignatureAlgorithm is used and the actual value taken for that algorithm.

```
<owl:ObjectProperty
           rdf:ID="forDigitalSignatureAlgorithm">
    <rdfs:domain
       rdf:resource="#DigitalSignature"/>
    <rdfs:range
       rdf:resource="#DigitalSignatureAlgorithm"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="value">
    <rdfs:range rdf:resource=
       "http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain
       rdf:resource="#DigitalSignature"/>
</owl:DatatypeProperty>
```

The relationship between a digital entity and one of its digital signatures is expressed below.

```
<owl:ObjectProperty rdf:ID="hasSignature">
    <rdfs:range
        rdf:resource="#DigitalSignature"/>
    <rdfs:domain
        rdf:resource="#DigitalEntity"/>
</owl:ObjectProperty>
```

Finally the relationship induced by citations is expressed as:

```
<owl:ObjectProperty rdf:ID="references">
    <rdfs:domain
        rdf:resource="#DigitalSignature"/>
    <rdfs:range
        rdf:resource="#DigitalSignature"/>
    <owl:inverseOf
        rdf:resource="#referencedBy"/>
</owl:ObjectProperty>
```

This semantic layer is intended to be further integrated with OWL-S [22] service description in order to enable seamless discovery and activation of CiteSeer-API.

6. USAGE SCENARIOS

In this section we envision the range of applications that can take advantage of the functionalities offered by CiteSeer-API.

6.1. Alternative User Interfaces to CiteSeer

Since its release, many research projects have created alternative interfaces to the CiteSeer.org Web-site in order to gain access to its document and metadata database or to provide alternative visualization of the citation-based relationships it maintains [2,6,11,18,24]. These projects all have in common the fact that wrappers around the traditional Web-interface had to be developed in order to make use of the data available from the CiteSeer server. We believe that CiteSeer-API will simplify such tasks by facilitating the integration of CiteSeer services in third party applications. CiteSeer-API is also a valuable tool towards the fast prototyping of new features for CiteSeer's Web-interface.

6.2. Interlinking Heterogeneous Digital Libraries

With OAI-PMH, heterogeneous DL systems are brought into cooperation via higher-level aggregators that make abstraction of the fundamental incompatibilities between those systems. Using CiteSeer-API and its functionalities for bibliography lookup and digital signature lookup, arbitrary clients can now directly determine a link to a specific resource hosted by a given CiteSeer-service. A direct application to such functionalities is the interlinking of digital library systems. In the specific case of interlinking CiteSeer-servers, each server can, using the CiteSeer-API interface to the server it wishes to interlink to, perform a lookup for each bibliographical entry known to it but for which the actual document is unavailable. Upon availability of the corresponding document, an external link to the document on the hosting server can be dynamically generated and incorporated in the response to a user query, hence overcoming the inherent incompatibility of indexing between any pair of CiteSeer servers. A practical example of this functionality is for eBizSearch - a niche search engine for e-Business publications - [12, 23] to attempt to link to CiteSeer.org [4, 5] for each reference to a pure Computer Science publication. Similarly eBizSearch would attempt to link to SMEALSearch [27] for each reference to a pure Business publication. We currently work on an extension of the Web-interfaces for CiteSeer servers that would take advantage of these functionalities. In the case of interlinking arbitrary heterogeneous digital library systems the bibliography lookup service of CiteSeer-API can be used in a similar fashion to accurately locate resources on any given CiteSeer server. Alternatively, the digital signature lookup functionality of CiteSeer-API can be used to achieve cross-DL compatibility, and allow digital library systems that don't use textual citation information to interlink with CiteSeer servers.

*6.3. **Soft**-Mirroring of Digital Libraries*

A current issue with CiteSeer is that of its expansion and synchronization with its mirrors. CiteSeer is currently mirrored at the School of Information Science and Technology at the Pennsylvania State University, and it is expected that mirrors will be maintained at several other research institutions, raising the issue of dealing with management variations from one mirror to another and with the main CiteSeer server itself. A possible variation from one mirror to another is the use of different software versions of the CiteSeer package that will in essence results in variations in the automatically generated metadata database. Another plausible variation is in the crawling policy of each institution towards the extension of their document collection, which ultimately results in different document collections being maintained at each mirror location. In this context it is desirable to come up with a mirroring policy that preserves the software and/or policy differences between mirrors while maintaining the document collections – i.e. documents

repositories, but not necessarily the metadata databases - synchronized. We call this approach soft-mirroring by opposition to mirroring in the traditional sense where databases are synchronized regardless of the data/metadata distinction. Note however that a special case is the synchronization of user-corrected metadata items. CiteSeer-API enables the use of digital signatures to identify CiteSeer hosted resources, hence based on CiteSeer-API it is possible to compute the difference between any two document collections and to update a slave collection on a document-by-document basis. Whether documents are to be fully reprocessed on the slave mirror or whether pregenerated metadata is acceptable is however dependent on the mirror update policy.

6.4. Versioning Control

The digital signature functionalities introduced in CiteSeer-API could be used in versioning control applications for digital libraries. Certain digital library systems have the capacity of identifying resource duplicates even when these are not exact binary/file duplicates. For instance the CiteSeer software manages document duplicates not only on the basis of identical digital signatures but also using a sentence-based text-similarity approach [17] that allows it to discard alternative version of the same document with content similar beyond 99%. Based on this capacity, classes of equivalent digital signatures can be maintained by digital libraries. The class representative(s) of each class can then be set to be the resource "official" version, which becomes the only version allowed for dissemination. This scheme is extremely relevant in digital library systems that collect unauthenticated materials and where copyright infringement issues must be addressed.

7. RELATED AND FUTURE WORK

The OAI-PMH protocols for metadata harvesting, although providing a standard set of properties for describing digital library resources, do not address the practical issue of digital libraries resource location. SRW, the Search/Retrieve Web Service standard [29], supports functionalities that resemble that of CiteSeer-API however SRW is geared toward standard access to – arbitrary – databases, and therefore does not take into account the specific task of digital libraries which is to manage actual digital entities that are best represented and manipulated using intrinsic digital signatures.

While several Web sites currently provide access to bibliographical databases [1,8,9], we are not aware of any efforts towards enabling machine-based access to these databases. Several closely related efforts currently attempt to establish a standard Digital Library and Information Retrieval platform on the World Wide Web. The most active efforts in this domain are certainly those from DSpace Federation [31] and Fedora [32]. Both support the OAI-PMH protocols for metadata distribution. Although Fedora provides management and access APIs, these systems have limited support for seamless interoperability and seamless integration with heterogeneous systems. With specific functionalities for bibliographical entry lookup, CiteSeer-API lets arbitrary clients and digital library systems locate CiteSeer-hosted resources and interlink with these resources.

To fully leverage the functionalities of CiteSeer-API, it is desirable to bring it into the context of the Semantic Web. CiteSeer-API is described using WSDL. Although this allows for the automatic generation of code stubs to programmatically access the CiteSeer services, the WSDL description does not carry the semantics of the underlying service. In Section 5.3 we presented a simple OWL-based semantic framework that constitute the first step toward the semantic integration of CiteSeer-API's functionalities for cross-DL interoperation. We want to pursue the integration of CiteSeer-API by further developing Web ontologies that further describe digital content, digital signatures, and associated applications. As discussed earlier, such ontologies are potentially a good common basis for natural interaction and interlinking between digital library systems. Finally in order to allow the seamless discovery and integration of CiteSeer-API services on the Semantic Web, the creation of OWL-S [22] service descriptions is necessary. These service descriptions will exploit the OWL ontologies for digital resources that have been discussed in Section 5.3.

8. CONCLUSIONS

We introduced CiteSeer-API, a SOAP/WSDL-based API to CiteSeer-like services. CiteSeer-API was designed not only to allow interactions between CiteSeer-like services but also with other DL systems. In this regard, the choice of resource identifiers that stem from the resources themselves is fundamental to ensure the interoperability of CiteSeer services with arbitrary heterogeneous DL systems. CiteSeer-API uses digital signatures as resource handlers. By doing so the internal complexity of CiteSeer servers is hidden from client agents. While CiteSeer-API turns CiteSeer-like niche search engines into actual Web-services, it still requires developers to have an understanding of the service in order to make use of it. We presented a simple semantic framework that readily allows for the description of CiteSeer-API's functionalities on the Semantic Web. The addition of semantic service description to CiteSeer-API using OWL-S will enable automated agents to discover, register and seamlessly exploit CiteSeer-like services. We encourage research groups to take advantage in their own projects of the functionalities and data available through CiteSeer-API.

9. ACKNOWLEDGEMENTS

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 and Isaac Councill for their contributions to this work.

10. REFERENCES

[1]: ACM Portal, http://portal.acm.org/portal.cfm

[2]: M. Bawa, G.S. Manku, P. Raghavan, "SETS: search enhanced by topic segmentation", in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003), pp 306-313, 2003.

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

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

[5]: CiteSeer, http://www.citeseer.org.

[6]: CiteSeer Relator, http://www.pmbrowser.info/citeseer.php

[7]: Crespo, A.; Garcia-Molina, H.. "Archival Storage for Digital Libraries", in *Proceeding of the 3rd ACM Conference on Digital*

Libraries (DL'98), pp. 69-78, Pittsburgh, PA, USA, June 23-26, 1998.

[8]: The Collection of Computer Science Bibliographies, http://liinwww.ira.uka.de/bibliography/index.html

[9]: DBLP, http://dblp.uni-trier.de/

[10]: Dublin Core Metada Initiative, http://dublincore.org/

[11]: A. Doan, Y. Lu, Y. Lee, and J. Han, "Object Matching for Data Integration: A Profile-Based Approach", in *Proceedings of the IJCAI-03 Workshop on Information Integration on the Web*, pp. 53-58, Acapulco, Mexico, August 9-10, 2003.

[12]: eBizSearch, http://www.ebizsearch.org.

[13]: 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, Pittsburgh, PA, USA, June 23-26, 1998.

[14]: J. Heflin, and J. Hendler, "Searching the Web with SHOE". in *Artificial Intelligence for Web Search. Papers from the AAAI Workshop.* WS-00-01. AAAI Press, Menlo Park, CA, 2000. pp. 35-40

[15]: HomepageSearch, http://hpsearch.uni-trier.de/

[16]: S. Lawrence, K. Bollacker, C.L. Giles, "Distributed Error Correction", in *Proceedings of the 4th ACM Conference on Digital Libraries*, pp. 232, Berkeley, CA, USA, August 11-14, 1999.

[17]: 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.

[18]: Q. Lu, L. Getoor, "Link-based Classification", in *Proceedings of the 20th International Conference of Machine Learning (ICML 2003)*, Washington, DC, USA, pp 496-503, 2003.

[19]: F. Lu, T. Johnsten, V. Raghavan and D. Traylor, "Enhancing Internet Search Engines to Achieve Concept-based Retrieval", in *Proceeding of Inforum'99*, Oakridge, TN, USA, May 1999.

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

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

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

[23]: 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.

[24]: A. Popescul, L.H. Ungar, S. Lawrence, D.M. Pennock, "Statistical Relational Learning for Document Mining", in Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), pp 275-282, 2003.

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

[26]: FIPS 180-1, "Secure Hash Standard", NIST, US Department of Commerce, Washington D.C., Apr. 1995.

[27]: SMEALSearch, http://smealsearch.psu.edu

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

[29]: SRW – Search Retrieve Web Service, http://lcweb.loc.gov/z3950/agency/zing/srw/

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

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

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

[33]: Yves Petinot, C. Lee Giles, Vivek Bhatnagar, Pradeep B. Teregowda, Hui Han, "Enabling Interoperability For Autonomous Digital Libraries : An API To CiteSeer Services" , in *Proceedings of the 4th ACM/IEEE Joint Conference on Digital Libraries (JCDL 2004)*, pp. 372-373, Tucson (AZ), June 2004.

[34]: The Digital Object Identifier System, http://www.doi.org/.

[35]: The OpenURL Framework for Context-Sensitive Services, http://www.niso.org/committees/committee_ax.html