Metadata Harvesting for Content-Based Distributed Information Retrieval

AQ1 **Fabio Simeoni and Murat Yakici**
University of Strathclyde. E-mail: {fabio.simeoni, murat.yakici}@cis.strath.ac.uk

Steve Neeley
University College of Dublin. E-mail: steve.neely@ucd.ie

Fabio Crestani
University of Lugano. E-mail: fabio.crestani@unisi.ch

We propose an approach to content-based Distributed Information Retrieval based on the periodic and incremental centralization of full-content indices of widely dispersed and autonomously managed document sources. Inspired by the success of the Open Archive Initiative's (OAI) Protocol for *metadata harvesting*, the approach occupies middle ground between content crawling and distributed retrieval. As in crawling, some data move toward the retrieval process, but it is statistics about the content rather than content itself; this grants more efficient use of network resources and wider scope of application. As in distributed retrieval, some processing is distributed along with the data, but it is indexing rather than retrieval; this reduces the costs of content provision while promoting the simplicity, effectiveness, and responsiveness of retrieval. Overall, we argue that the approach retains the good properties of centralized retrieval without renouncing to cost-effective, large-scale resource pooling. We discuss the requirements associated with the approach and identify two strategies to deploy it on top of the OAI infrastructure. In particular, we define a minimal extension of the OAI protocol which supports the coordinated harvesting of full-content indices and descriptive metadata for content resources. Finally, we report on the implementation of a proof-of-concept prototype service for multimodel content-based retrieval of distributed file collections.

Introduction

Our interest is in content-based retrieval of widely dispersed and autonomously managed document sources.[1] This is the central problem of Distributed Information Retrieval (DIR), and over the past 10 years, it has been mainly approached by distributing the retrieval process along with the data: Queries have been "pushed" toward the content, and the results of their local execution have been centrally gathered and presented to the user (cf. Callan, 2000a).

Traditionally, distributed retrieval services have relied on simple client/server architectures in which brokers route queries submitted by local or remote clients toward a number of mutually autonomous and potentially uncooperative retrieval engines. Figure 1 shows how client/server distributed retrieval works. A Search Broker B interfaces Clients C and dispatches their Queries Q to a number of autonomous search engines. S_1, S_2, S_n, each of which executes it against an Index FT_i of some Content C_i before returning Results R_i back to B, which merges them and relays them to C. Optionally, B optimizes query distribution by selecting a subset of the engines based on previously gathered descriptions of their content. Based on summary descriptions of the content served by each engine, advanced techniques of source selection and data fusion have been produced to, respectively, minimize network

Received May 3, 2006; revised November 15, 2006; accepted February 27, 2007

 ● Published online 00 xxxxxx 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20694

[1] In the lack of a well-established terminology, we use the term *content-based* to characterize retrieval processes defined over indices of essentially unstructured documents. Content-based retrieval lies at one end of a spectrum which is otherwise bound by *structure-based* retrieval, where indices are extracted from rigidly structured data. Full-text retrieval and relational database retrieval are by far the most common examples of content-based and structured retrieval, respectively.

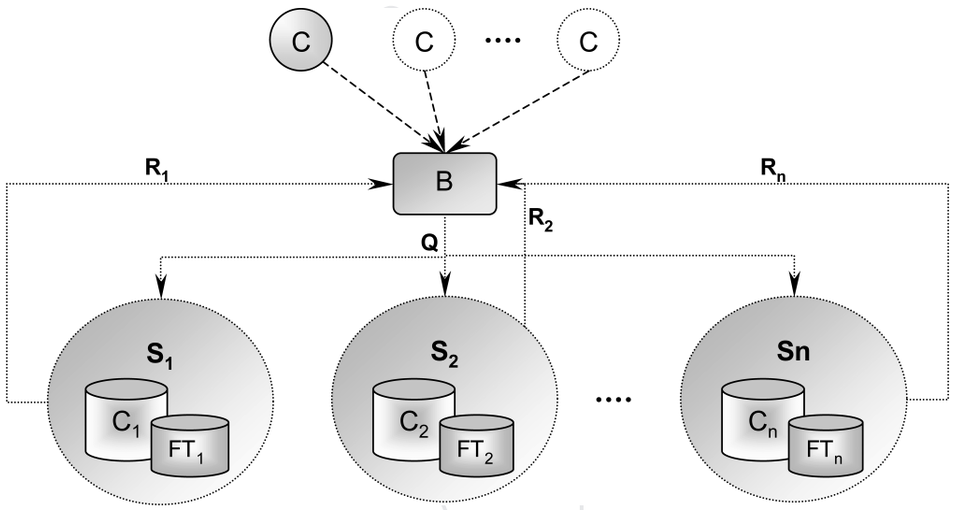

FIG. 1. Client server distributed retrieval. *A* search broker *B* interfaces clients *C* and dispatches their queries *Q* to a number of autonomous search engines S_1, S_2, Ö, S_n, each of which executes it against an index FT_i of some content C_i before returning results R_i back to *B* which merges them and relays them to *C*. Optionally, *B* optimises query distribution by selecting a subset of the engines based on previously gathered descriptions of their content.

interactions and normalize the partial result rankings produced by potentially diverse models of probabilistic retrieval (cf. Callan, 2000a; Callan, Crestani, & Sanderson, 2004b). Automatically learned descriptions (Callan & Connell, 2001), cooperative protocols (Gravano, Chang, Garcia-Molina, & Paepcke, 1997), fusion heuristics (Callan, Crestani, Nottelmann, Pala, & Shou, 2003), and experimental testbeds (e.g., Callan, 2000b; French et al., 1999) have been proposed and extensively tested. Finally, a few research projects have carried out preliminary investigations into nontextual forms of content-based distributed retrieval (e.g., Nottelmann & Fuhr, 2003) and begun to explore the potential of cooperative and gird-enabled retrieval infrastructures.[2] More recently, core techniques developed for client/server retrieval have been repurposed in the more dynamic context of peer-to-peer retrieval, where queries may be posted, routed, and directly executed by any of a number of mutually, but intermittently connected peer engines (Callan, Fuhr, & Nejdl, 2004a), Hybrid, double-tiered architectures, in particular, have offered ideal ground for bridging optimizations developed for client/server architectures with the advantages of fully decentralized control (i.e., increased potential for resource pooling, fault tolerance, dynamic self-configuration, and privacy) (Lu & Callan, 2003, 2005; Yang & Garcia-Molina, 2001).

Rather independently and over a longer period of time, the Digital Library (DL) community also has explored the potential of distributed retrieval in the practice of its information services. Here, retrieval has been mainly interpreted as a deterministic process defined against the explicit structure of descriptive and manually authored metadata records. Nonetheless, queries and results have still been exchanges within the client/server architecture described earlier, the Z39.50 protocol (Z39.50 Maintenance Agency, 2003), in particular, has standardized the syntax and semantics of such exchange. Recently, more lightweight, Web-based protocols for distributed retrieval also have been proposed (e.g., Paepcke, Brandriff, Janee, & Larson, 2003; Sanderson, 2003; Simon, Massart, Assche, Ternier, & Duvall, 2003).

Over the past 5 years, however, the DL community has progressively favored the complementary approach of iteratively and incrementally centralizing metadata as a precondition to the retrieval of the associated data: Metadata have been "pulled" toward the queries in advance of their execution, and the retrieval function has remained centralized. Figure 2 shows the data flow in metadata harvesting. In the *offline phase*, a Service Provider *SP* periodically and incrementally gathers Metadata *M* from a number of Data Providers DP_1, DP_2, & DP_n and persistently stories it in a Metadata Repository *MR*. In the *online phase, SP* interfaces users and resolves their queries against the metadata in *MR*.

Standardized de facto by the Protocol for Metadata. Harvesting of the Open Archive Initiative (OAI-PMH; Lagoze, de Sompel, Nelson, & Warner, 2002b), the strategy mentioned earlier has become known as the *harvesting* model of retrieval over distributed content. The model has proved particularly suitable to meet the technical and sociological requirements of retrieval—and, in fact, of many other metadata-based services—within large-scale Federated Digital Libraries (FDLs), most noticeably those built around Institutional Repositories (Crow, 2002), and the Open Access movement (Bailey, 2005). Among these are the cross-sectoral, nationally scoped initiatives which account for most of current development efforts within the DL field (e.g., Anan et al., 2002; Joint Information Systems Committee, 2001; Lagoze et al., 2002a; van der Kuil and Feijen, 2004). A principled analysis of such success is found in Simeoni (2004) and is summarized in the next section.

[2] See the DILIGENT project at http://www.diligentproject.org

DOI: 10.1002/asi

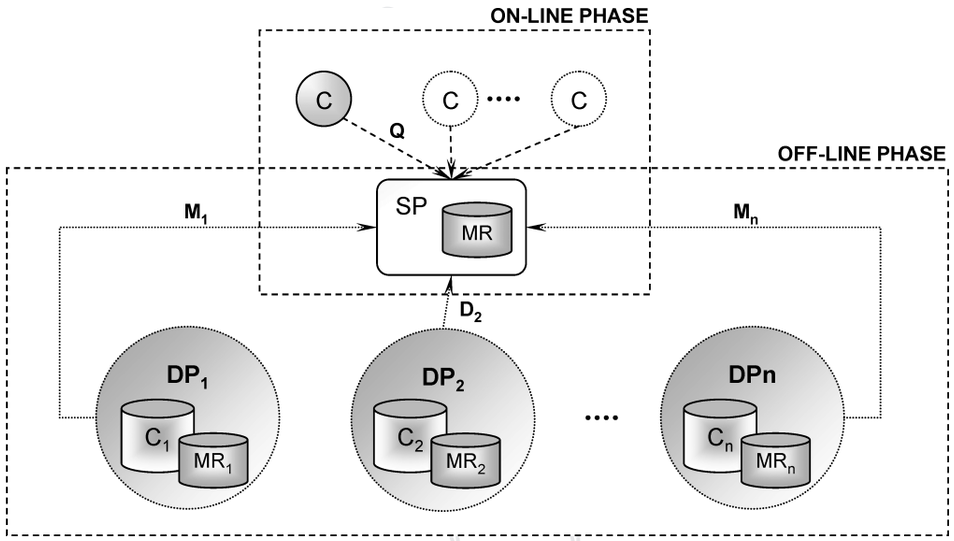

FIG. 2. Data flow in metadata harvesting. (a) *off-line phase*: a service provider *SP* periodically and incrementally gathers metadata *M* from a number of data providers DP_1, DP_2, Ö, DP_n and persistently stores it in a metadata repository *MR*; (b) *on-line phase*: *SP* interfaces users and resolves their queries against the metadata in *MR*.

The Harvesting Model

From a technical perspective, the harvesting model eliminates the wide-area network as a real-time observable of service provision and, with it, a major obstacle to its medium and medium-large scalability (Gatenby, 2002; Lynch, 1997). Bandwidth fluctuations induced by traffic congestion and latency-inducing factors associated with slow, unavailable, or particularly distant data sources have no impact on the consistency, reliability, responsiveness, and even effectiveness of service provision. Similarly, postprocessing of results—whether distributed or centrally performed—need no longer to occur in real time with respect to query submission. More generally, retrieval may regain the simplicity, generality, and Quality of Service (QoS) guarantees which are normally associated with local computations, be them centralized or *locally* distributed.

From a sociological perspective, the model captures the disparity of strengths and interests which characterize FDLs; in particular, it clearly distinguishes the roles, responsibilities, and costs of *service providers* from those of *data providers*. Data providers may give broad visibility to their data without having to face the complexity of full-service provision (e.g., query language support/wrapping, postprocessing of results, query load, etc.); comparatively, dissemination of metadata is a simple task and one which offers more resilience across different services and communities. Service providers also benefits from simplified participation since the scope and usefulness of their services may scale beyond previously experienced bounds.

Overall, harvesting offers a two-phase view of service design, which separates communication from implementation and contribution to service provision from service provision itself. By doing so, the model lowers the barrier to interoperability without compromising service efficiency or effectiveness (Lagoze & de Sompel, 2001).

Of course, these benefits come at a price, and while harvesting encourages participation, it still relies on the will to disseminate some data; in distributed retrieval, in contrast, cooperation may be harder to achieve, but it is not always a requirement (cf. query-based sampling techniques for automatic synthesis of source description; Callan & Connell, 2001). By centralizing retrieval, harvesting also limits the potential for cost-effective resource pooling, which may be required to achieve massive scale. Similarly, harvesting does not exhibit the fault tolerance which is associated with fully distributed processes; peer-to-peer retrieval, on the other hand, faces the challenges of decentralization, but it completely eliminates single points of failure. Finally, harvested data are copied data, and under the assumption of update, it is bound to be in a temporary state of staleness; some applications may tolerate no delays in change propagation, no matter how short they may configured to be. Outside the scope of these constraints, however, the harvesting model is an increasingly common infrastructural assumption for metadata-based retrieval of distributed data.

Scope and Motivations

In this article, we investigate the applicability of the harvesting model to content-based retrieval. The motivation is twofold. First, we hope to expand the scope of DIR beyond the assumptions which have bound it so far: As we show later, scale and data ownership may still prove important requirements, and yet, distributed data need no longer imply distributed retrieval.

Second, we aim to extend the benefits of the harvesting model within the same domains which to date have successfully, but only partially, adopted it. Currently, the model may support keyword-based retrieval against the content of the harvested metadata, but the full content remains opaque to

DOI: 10.1002/asi

federated services. A reconciliation of harvesting with content-based retrieval would guarantee homogeneous scope and QoS across both metadata-based and content-based services. Using the OAI-PMH for this purpose, in particular, would immediately leverage a widely deployed infrastructure of tools and data providers.

Under a generic interpretation, of course, the applicability of harvesting to content-based retrieval need not be questioned: Any Web search engine stands as a witness to the feasibility of moving data toward the retrieval process. If anything, popular engines prove that given the concentration of sufficient resources, the bounds of local scalability may be remarkably stretched. Recently, attempts to harvest content for dissemination and preservation purposes also are found within more focused communities, with the OAI-PMH in place of crawling as the mechanism for centralizing content (de Sompel, Nelson, Lagoze, & Warner, 2004; Dijk, 2004).

Here, however, we focus on a stricter, but more advantageous, interpretation of harvesting in which retrieval remains predicated on the sole movement of metadata. Of course, we now give to metadata the technical meaning which it normally assumes in Information Retrieval, and thus focus on automatically generated content statistics rather than on manually authored, descriptive records only. In particular, we assume that the primary content remains distributed and that a full-content index of the union of the distributed sources is centralized instead.

By doing so, we aim to promote efficiency, for we avoid the costs of fine-grained and large-sized content transfers over the network; in particular, we expect to make better use of shared bandwidth and to reduce load at both data and service providers. We also aim to promote scope, for the approach may offer visibility to data which are neither statically published nor publicly accessible; data which are proprietary, cost money, demand access control, or are simply dynamically served may still be safely disseminated.

Overall, we shift the assumption of distribution from the retrieval process to the indexing process, and thus explore the existence of middle ground between distributed retrieval and content crawling.[3] In doing so, we are guided by the following research questions: Can we distribute and incrementally execute the full-content indexing process? And from a more practical perspective: Can we leverage the OAI infrastructure for the purpose?

We address these questions in the next two sections. Then we show that they admit at least one positive answer by describing a prototype implementation based on an extension of the OAI-PMH. We discuss related work in the following section before drawing some conclusions. Although the content type is orthogonal to our approach, we henceforth concentrate on text in reflection of the state of the art in the field.

[3] Here and in the following, we use the term "indexing" broadly, as any form of content processing which yields input for the retrieval process. In particular, we intend it to subsume automated content analysis and thus operations of case normalisation and stemming.

The Approach

We use an example to clarify the approach and identify the requirements it raises at both ends of the data-exchange scenario.

Harvesting Scenarios

Consider first a prototypical harvesting scenario in which a service provider relies on the OAI-PMH to periodically centralize descriptive metadata about "eprints"—that is, published and in-progress research documents—from a federation of Institutional Repositories.

Independently from dissemination agreements, the repositories maintain their metadata in local databases and use it routinely to offer local services to their users, including a retrieval service based on fielded queries. Some repositories also maintain full-text indices on their file systems and use them to complement the retrieval service with keyword-based queries. Models and languages for source description, indexing, and retrieval are locally defined and maintained.

At each repository, a dissemination service implements the server side of the OAI-PMH and resolves protocol requests by: (a) executing a fixed range of system-level queries against the metadata database (e.g., find all records which have been updated since a given date) and (b) mapping the results expressed in the local metadata model onto instances of a model agreed upon for exchange, for example, unqualified Dublin Core (DCMI, 2004).

At the service provider, the DC records are normalized and otherwise enhanced; for example, duplicates are removed and subject classification headings are automatically inferred using a third-party Web service. Finally, the postprocessed metadata are added to the input of a Web-accessible, interactive retrieval service. Like some of its counterparts at the data providers, the retrieval service accepts both fielded and keyword-based queries, but it executes both types of query against the harvested DC records.

We propose an extension of the previous scenario in which the descriptive metadata exposed by repositories are augmented with automatically generated content statistics, such as frequency of term occurrences within and across documents. Figure 3 shows the data flow in full-text index harvesting. In the *offline phase*, a Service Provider *SP* periodically and incrementally gathers Metadata *M* and Content Statistics *I* from a number of data providers, DP_1, DP_2, DP_n, and persistently stores them in a Metadata Repository *MR* and a Full-Text Index *FT*, respectively. In the *online phase*, *SP* interfaces users, resolves their queries against the statistics in *FT*, and uses the metadata in *MR* to present the results.

Like descriptive metadata, statistical information obeys the constraint of an exchange model implicitly or explicitly identified by harvesting requests. To obtain such information, repositories interrogate existing or dedicated full-text indices, rather than databases, but they still map results onto the model agreedupon for exchange. At the *SP*, the statistical information is extracted and used to update the centralized full-text index, possibly after having been normalized and

DOI: 10.1002/asi

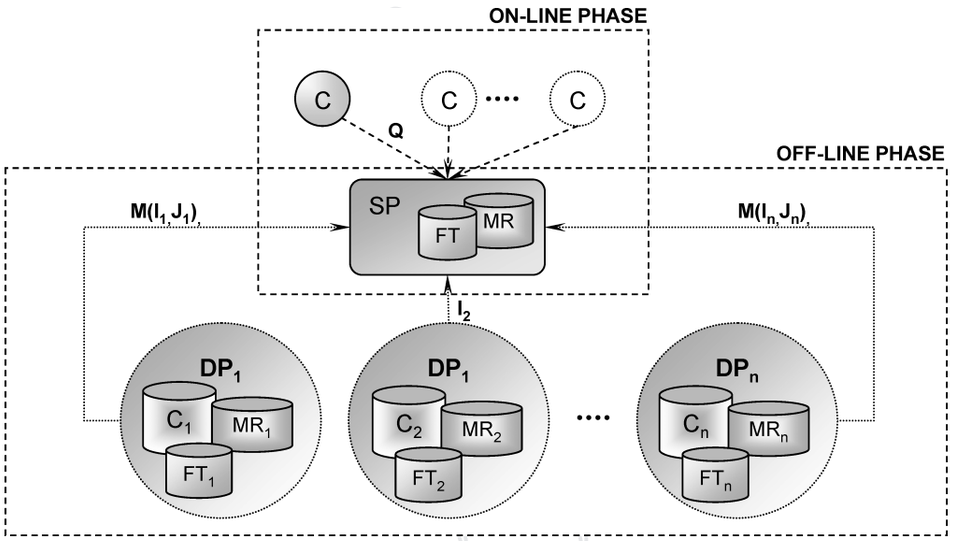

FIG. 3. Data flow in full-text index harvesting. (a) *off-line phase*: a service provider *SP* periodically and incrementally gathers metadata *M* and content statistics *I* from a number of data providers DP_1, DP_2, Ö, DP_n and persistently stores them in a metadata repository *MR* and a full-text index *FT*, respectively; (b) *on-line phase*: *SP* interfaces users, resolves their queries against the statistics in *FT*, and uses the metadata in *MR* to present the results.

enhanced to reflect current index statistics and local indexing requirements, respectively. The index and the descriptive metadata are then used to, respectively, satisfy full-text queries and to support the presentation of results. Since the approach separates indexing and retrieval processes, (subsets of) the same content statistics may be used to concurrently support multiple models of retrieval. For instance, the same central index may be used to test the effectiveness of a vector space model and a language model against a given distributed collection.

Requirements

From a conceptual perspective, the extension is relatively straightforward. Its only requirement is for the *SP* to rely on a model of indexing which allows *modular* representation of content over space and time. More formally:

> (*Modular Indexing*) Let M be an indexing model, C a content source, and C_0 and C_1 two snapshots of C at time t_0 and t_1, respectively. If I_0 and I_1 are the M indices of C_0 and C_1, then M is *modular* if the difference $\Delta C = C_1 - C_0$ implies a difference $\Delta I = I_1 - I_0$ such that ΔI is computable from I_0 and ΔC only.

In the context of the proposed approach, C is the union of the content sources indexed by a harvester and M the model employed for the indexing. Interpreted along a spatial dimension, ΔC reflects the inclusion of an additional content source; modularity then guarantees the distributivity of the indexing process across two or more independently maintained content sources. Interpreted along a temporal dimension, ΔC reflects a change in one of the existing content sources; modularity then guarantees the incremental nature of the indexing process against each content source. In both cases, modularity of indexing relies on content properties which can be measured over document-grained increments. Most indexing models satisfy this requirement because they either rely on term-related properties which pertain to individual documents—such as in-document term number, frequency, and location—or else pertain to groups of documents and yet may still be progressively derived, such as inverse document frequency (Witten, Moffat, & Bell, 1999). Overall, *SP*s can distribute indexing across content sources and maintain their centralized index over time as sources change or new sources are identified.

From a pragmatic perspective, on the other hand, the enriched semantics of the exchanged data may inject additional development complexity and resource consumption into the standard harvesting scenario. Most noticeably, it relies on the availability of collection-management environments which:

- offer integrated management of descriptive metadata and full-text indices. In many cases, this may be accomplished within the boundary of a single technology; most full-text retrieval engines, for example, store content statistics and descriptive metadata within a single index structure. In other cases—normally when complex metadata structures are maintained and used independently from content-based retrieval services—the approach may require the synchronization of collection-management procedures (e.g., identification, insertion, modification, removal) across different technologies, from general-purposes relational databases with standard interfaces to full-text indexing engines with proprietary APIs. AQ2
- accommodate the computational load which is normally associated with the increased size of indexing information over descriptive metadata.

DOI: 10.1002/asi

Clearly, issues of data integration and size concern both ends of the exchange scenario. On an absolute scale, problems may seem more acute at the client side of the protocol, but the harvesting philosophy indicates that the server side is where adoption and scalability may be more obviously at stake. After all, *DP*s must now sustain the cost of generating, maintaining, and exposing full-text indices within their resource allocation policies; whenever such costs may not be directly justified in terms of local requirements, accommodating the novel dissemination requirements may prove difficult. In these cases, cost estimates will vary from case to case and only deployment experience may indicate what level of tool support may help to reduce complexity; for example, the prototype section (discussed later) shows that—under specific deployment assumption and QoS guarantees—low-cost implementations of the proposed extension are certainly possible. Note that while such "grassroots" scenarios are well within the remit of current applications of the OAI-PMH protocol—and thus should be accounted for by any of its extensions—they are normally outside the scope of DIR approaches, where the availability of a local search engine is a basic requirement on content sources. Under these assumptions, there is no reason to associate our proposal with increased integration costs.

As to the issue of size, we expect compression to play an important role at both ends of the protocol. Lossless compression techniques based on optimized representation structures are the first obvious choice, be it for the persistent storage of indices, their in-memory management, or their transfer on the wire. Transport-level compression, in particular, is already within the standard OAI-PMH exchange semantics, albeit it has been seldom used so far. In addition, the inherent offline nature of the harvesting process suggests that compression ratios may be pushed further than they tend to be when decompression is a real-time observable of service provision.

Lossy compression techniques also may be conveniently used to complement lossless approaches. Well-known algorithms in Information Retrieval—ranging from standard case folding, stop-word removal, and stemming algorithms to static index pruning and document summarization algorithms (e.g., Carmel et al., 2001, Lu & Callan, 2002)—may all grant additional size reductions without excessively compromising the final quality of retrieval.

Admittedly, reducing the *amount* of information exchanged between *DP*s *SP*s—rather than its size only—may reintroduce the problems of semantic interoperability, which have proved to complicate the distribution of retrieval in the past. In Z39.50 parlance, for example, variations in stop-word removal and stemming algorithms across "targets" (i.e., servers) have been previously associated with lack of retrieval consistency at "origins" (i.e., clients) (Lynch, 1997). Note, however, that semantic variations are now limited to indexing and *do not otherwise impact on the consistency granted by a single model of retrieval*. Furthermore, variations in indexing policies across *DP*s must be related to the indexing policy employed at the harvester side (i.e., the policy which ultimately determines the inclusion or exclusion of content from query results). These may well differ, but provided that the former are *less* aggressive than the latter, they can be normalized at the harvester side; normalization procedures, in particular, occur offline with respect to query submission and may thus be as sophisticated as they need to be. Remote indexing policies, which are instead *more* aggressive than the centralized one, are unavoidably associated with information loss at the harvester side. Notice, however, that it is well within the harvesting philosophy to leave *DP*s free to choose the optimal tradeoff between the consumption of their computational resources—which may be minimized by an aggressive indexing policy—and the visibility of those resources within the federated environment, which instead may be maximized by a relaxed indexing strategy. Put differently, *DP*s have full control on the impact that their local indexing policies may have on the dissemination of their resources.

One final, pragmatic question concerns the suitability of the OAI-PMH to support the extended exchange semantics. We dedicate the next section to a possible answer.

The Protocol

We first summarize the main features of the OAI-PMH and then assess two strategies to deploy the extended exchange semantics on top of the existing OAI infrastructure.

OAI-PMH

At its heart, the OAI-PMH is a client-server protocol for the selective exchange of self-describing data (Lagoze et al., 2002b).

As shown in Figure 4, six types of requests are available to clients: three *auxiliary* requests to discover capabilities of servers (`Identify`, `ListMetadataFormats`, `ListSets`) and three *primary* requests to solicit data from servers in accordance with their capabilities (`GetRecord`, `ListRecords`, `ListIdentifiers`). To support *incremental* harvesting, servers associate their data with timestamp information and then maintain it with a granularity of days or seconds; clients then may use timestamps to temporally scope their ListRecords request and ListIdentifiers requests. To support *selective* harvesting, servers may organize data in one are more hierarchies of potentially overlapping datasets; clients then may specify a dataset to spatially scope their ListRecords requests and ListIdentifiers requests. Simple session-management mechanisms support large data transfers in the face of transaction failures. For ease of deployment, the overall semantics of exchange—including error semantics—is "tunneled" within HTTP's while XML provides syntax and high-level semantics for response payloads. Infrastructural issues of authentication, load balancing, and compression are outside the protocol's semantics and must be resolved within a broader scope (e.g., at the HTTP level).

The exact semantics of the exchanged data is formally undefined, but by design, it is expected to fall within the domain of content (DC) metadata; indeed, all servers are

DOI: 10.1002/asi

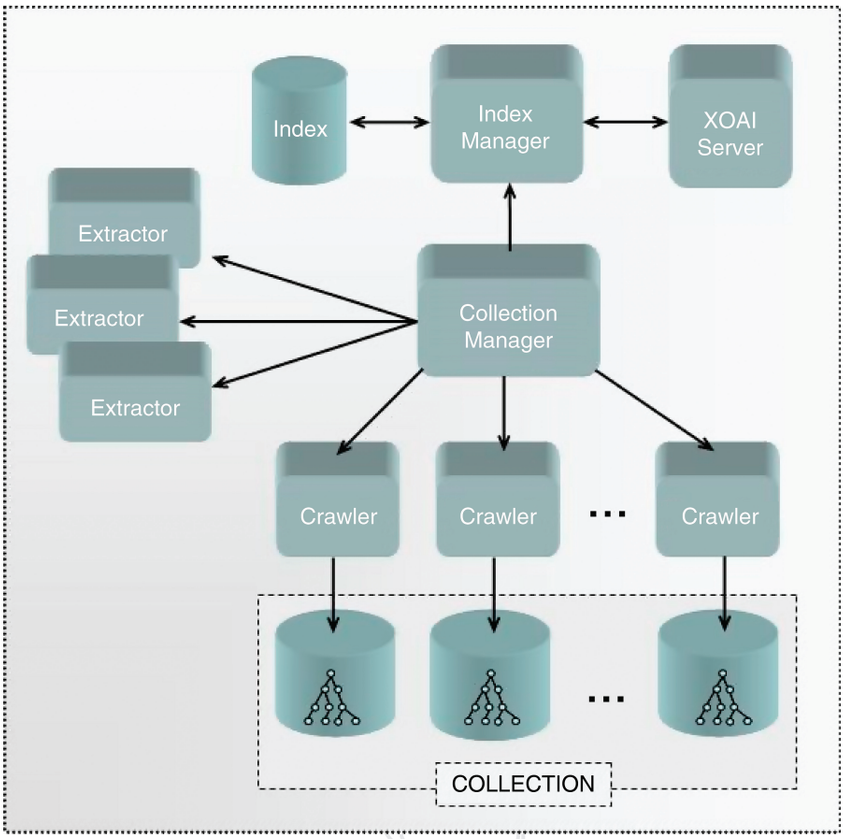

FIG. 4. Data Provider Implementation Architecture.

required to produce DC metadata on request. In particular, an exchange model associates servers with repositories of *resources* and resources with one or more metadata descriptions, or *records*; the latter form the basic unit of exchange. The model says little about resources (e.g., degree of abstraction, content semantics, location, identification, accessibility, persistence, etc.), but it offers a layered model of metadata in which records are format-specific instantiations of fully abstract resource descriptions, or *items*; items support the association of multiple metadata descriptions with a single resource (e.g., context-dependent or task-dependent annotations). The identification of items and formats is explicit; the protocol suggests an implementation scheme for item identifiers (e.g., `oai:dp:hep-th/9901001`) and defines an extensible list of format identifiers (e.g., oai_dc for the required DC). Individual records are instead implicitly identified by their format and the item they instantiate; they are nonetheless explicitly associated with datestamps and thus may change independently from their items. As an example of OAI-PMH data exchange, the following HTTP GET request:

```
http://www.dp.org/oai?
verb=ListRecords&MetadataPrefix=oai_dc&
  from=2005-01-01
```

asks a server available at `http://www.dp.org/oai` to return all the DC records which have changed since the start of the year. The following is a sample response:[4]

```
<OAI-PMH>
  <responseDate>2005-01-01T19:20:30Z</responseDate>
  <request verb="ListRecords" from="2005-01-01"
    metadataPrefix="oai_dc" >http://www.dp.org/OAI
  </request>
  <ListRecords>
    ...
    <record>
      <header>
        <identifier>oai:dp:hep-th/9901001</identifier>
        <datestamp>2005-02-18</datestamp>
      </header>
      <metadata>
        <dc>
          <title>Opera Minora</title>
          <creator>Cornelius Tacitus</creator>.
            <identifier>http://www.dp.org/res/
              9901001.html</identifier>
...
        </dc>
      </metadata>
    </record>
    ...
  </ListRecords>
</OAI-PMH>
```

[4] For clarity, namespace information is omitted in this and the following examples.

DOI: 10.1002/asi

Design Strategies

The increasing popularity of the OAI-PMH has generated some interest in using the protocol beyond its original design assumptions.

Building on the generality of the data model, original use has sometimes been predicated on creative instantiations of the modelling primitives. As resources have been mapped onto usage logs, thesaurus terms, registry entries, and even users, the protocol has shown its suitability for generic distributed state maintenance (de Sompel, Young, & Hickey, 2003).

In other cases, the exchange semantics has been extended to accommodate additional functionality. For example, protocol extensions have supported intercomponents interactions within distributed DL frameworks (Suleman & Fox, 2002), content crawling (Dijk, 2004), authentication, subscription, and notification schemes (Chou, Kuo, Ho, & Lee, 2003) as well as functionality intended to reduce complexity for *DP*s and *Sp*s within specific communities of adoption (Simons & Bird, 2003).

Both design routes are available for our protocol; in particular, we could conceive it as either an *application* or an *extension* of the OAI-PMH. The first solution may be simply predicated on:

- a specialization of the protocol's data model;
- the definition of a dedicated format for the integrated exchange of descriptive metadata *and* content statistics.

The data model specialization would simply introduce constraints on the notion of resource, which are required by the assumption of full-text indexing. Namely:

- Resources have at least one digital and text-based physical manifestation;
- a distinguished manifestation of the resource, the *primary manifestation*, satisfies the first constraint and is designated to represent the content of the resource for harvesting purposes.

The dedicated format would instead bind descriptive metadata and content statistics of primary manifestations to individual request/response interactions to avoid the synchronization problems which may arise if each form was harvested independently from the other.

Overall, an application of the protocol drafted along these lines is appealing, as it proves the concept *while requiring no change to the protocol and its deployment infrastructure*. While it may immediately serve the needs of specific communities, however, its design is rather adhoc and requires the definition of dedicated formats for each variation in the shape of descriptive metadata and/or content statistics. This induces a "combinatorial" approach to standardization which may unnecessarily compromise interoperability across communities of adoption.

To illustrate the full potential of the approach, we concentrate instead on the definition of a more modular exchange mechanism which may gracefully accommodate arbitrary forms of descriptive metadata and content statistics. Specifically, we retain the data model specialization defined earlier as well as the binding of metadata and content statistics within individual request/response interactions; however, we now identify each form of data independently from the other and thus assume that a record includes both a metadata part and an index part. In particular, we expect requests to specify a format for the metadata part and a format for the index part.

This leads to a protocol extension defined by:

1. the addition of an auxiliary request `ListIndexFormats` with associated response format;
2. the addition of an optional parameter `indexPrefix` to primary requests; and
3. the addition of an optional `index` child to `record` elements contained in responses to primary requests.

`ListIndexFormats` allows the discovery of the index formats supported by servers, and is thus a straightforward extension of `ListMetadataFormats` to the index part of records. Similarly, `indexPrefix` specifies the format of the index part of records and thus mirrors `metadataPrefix` and its associated error semantics. Finally, index elements contain the `index` part of records and follow the standard `metadata` elements within responses.

The extension of the sample request/response pair shown earlier may then be the following:

```
http://www.dp.org/oai?
  verb=ListRecords&metadataPrefix=oai_dc&index
    Prefix=tf_basic &from=2005-01-01
<OAI-PMH>
  ...
<ListRecords>
  ...
  <record>
    ...
    <metadata>
      <dc>... </dc>
    </metadata>
    <index>
      <terms>
        ...
        <term name="opera" freq="26">
        <term name="minora" freq="36">
        ...
      </terms>
    </index>
  </record>
  ...
 </ListRecords>
</OAI-PMH>
```

Here, `tf_basic` is the identifier of a simple format which captures the name and frequency of occurrence of the terms chosen to represent primary manifestations (possibly after stemming and stop-word removal). The underlying model serves the purpose of a proof of concept, but supports most of the indexing models which may be employed at the client side. Variations are of course possible; for example, a format which captures only term names and document lengths would decrease resource consumption and still support simple models of boolean retrieval. On the other hand,

DOI: 10.1002/asi

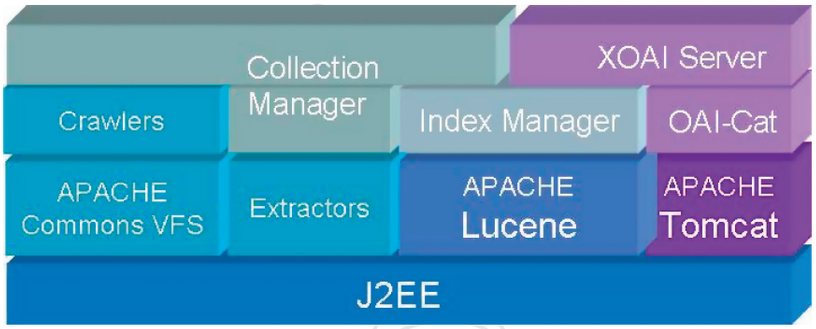

FIG. 5. Data Provider Implementation Architecture.

a model which includes positional information for each term occurrence would increase resource consumption, but also support proximity searches at the client side.

Overall, we believe that implementing the proposed extension does not layer excessive complexity over existing clients and servers. Conceptually, the extension also is *backwards-compatible* for the optionality of its features within requests, and responses need not be observed by standard client implementations. The latter, in particular, would simply omit optional parameters, ignore the existence of new requests, and always process responses which are structurally identical to those produced by standard server implementations.

Unfortunately, technical reasons inject more disturbance within the protocol infrastructure than conceptually necessary. In particular, the carefully controlled extensibility associated with the OAI namespace requires the modified semantics of the `record` element—and in fact all elements within protocol responses which recursively depend on it—to be defined within a new namespace.[5] Accordingly, namespace-aware clients would necessarily break upon receiving responses from extended server implementations. Ultimately, this forces standard and extended server implementations to live (and be maintained) side by side at two different network locations.

In conclusion, both design solutions have advantages and disadvantages: A protocol application lacks in generality while a protocol extension denies technical guarantees of backward-compatibility. We believe that the latter nonetheless offers a stronger proof of concept and thus suits best the purposes of this article. Accordingly, we adopt the proposed protocol extension to test the prototype implementation discussed in the next section.

The Prototype

As a proof-of-concept implementation of the approach, we have built a prototype service for full-text searching of remote content collections held at one or more *DP*s.[6] User queries at the *SP* are resolved against a local index asynchronously populated with content statistics which are periodically and incrementally gathered from the *DP*s. Communication between *SP*s and *DP*s is governed by the protocol proposed earlier.

For simplicity, collection management at *DP*s is modeled as a dedicated and entirely automated activity: Content resources are inferred from Web-accessible files and described by mechanically derivable properties (from URIs to, when possible, titles and authors). In a production environment, this model may not grant high-quality descriptive metadata across all resources, but it makes our prototype self-contained and thus suits the purpose of an easily deployable demonstration.

The architecture of the prototype may be illustrated along the divide between *DP*s and *SP*s. The main components at the *SP* side are shown in Figure 5 and are briefly described as follows.

A Collection Manager component is configured to infer a collection from one or more storage hierarchies. The hierarchies are physical parts of the collection, but their semantics is not predefined; they may reflect a logical partition of the collection, a storage allocation strategy, or simply the presence of intra-organizational boundaries. In particular, a hierarchy may reside on local or remote storage, provided that there exists a base URL which can be extended with the path that connects the root of the hierarchy to a file below it to yield the URL of the file. At the leaves of the hierarchy, the Collection Manager interprets homonymous files as manifestations of the same resource and infers primary manifestations on the basis of a configurable ordering of the supported file formats. For example, Portable Document Format (PDF) manifestations may be preferred over HTML ones if they are or become available at some point in time.

The Content Manager allocates a Crawler component to each storage hierarchy with the intention to periodically monitor additions, modifications, and deletions of primary manifestations in the hierarchy. Upon observing a collection-management event, the Crawler reports it to the Collection Manager, which reflects it onto a persistent index of the collection through the

[5] Of course, this reflects an assumption that namespaces are owned and that ownership extends to element semantics rather than element names alone. There are many who do not subscribe to this view and consider third-party extensions of namespaces an acceptable practice, especially when the extended element semantics is, as in our case, fully backward-compatible.

[6] The prototype is currently available for demonstration and download at `http://www.ilab.cis.strath.ac.uk/ft-oai`

DOI: 10.1002/asi

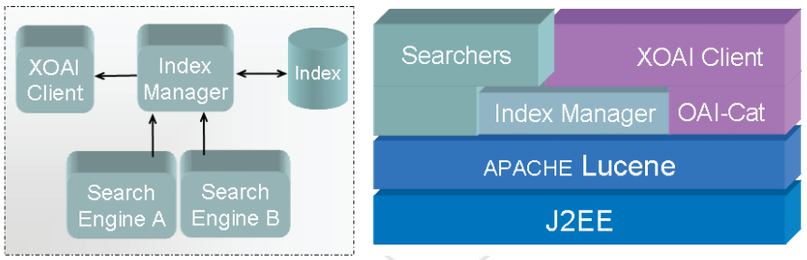

FIG. 6. Data Provider software stack.

mediation of an Index Manager component. For a new resource or a modification of a resource, in particular, the Collection Manager delegates to a format-specific Extractor component the task of deriving the full-text content and metadata from the resource's primary manifestation. Collectively, we refer to the Extractor's output as the *indlet* of the resource. Clearly, metadata properties may vary in nature, quantity, and quality across Extractors, with an expectation that structured data formats (e.g., PDF or XML vocabularies) may be leveraged toward better metadata generation.

The Collection Manager enriches the indlet of the resource with its URL, date of last modification, and other format-independent, system-level resource properties, and finally submits it to the Index Manager. Here, the full content of the resource is subjected to a configurable process of lexical analysis, during which the individual terms which comprise it are: (a) filtered against a list of stop-words, (b) normalized into a list of distinct stems, (c) annotated with their frequency of occurrence, and finally (d) persisted in the collection index along with the associated metadata.

Asynchronously, the Index Manager exposes indlets to a server-side implementation of the extended OAI protocol. From each indlet which matches the scope of the client requests, the extended OAI server extracts an `oai_dc` record and a `tf_basic` record. It then serializes the list of such records in XML as shown earlier, and finally returns the serialization to remote clients in a compressed form.

The software stack at the *DP* is rooted in the Java platform, as shown in Figure 6. The implementation maximizes reuse by leveraging three projects of the Apache Software Foundation and a project of the Online Computer Library Centre. OCLC's OAI-Cat[7] is a mature server-side and client-side implementation of the latest version of the standard OAI protocol. The flexibility of its design—particularly the abstraction over the back end and the modularity of its components—has greatly simplified the implementation of the extended OAI server and its interaction with the Index Manager. As a servlet-based Web application, OAI-Cat[8] runs within a dedicated run-time, and Apache Tomcat has provided here the obvious instantiation. Apache's Commons VFS[9] has instead provided the abstraction over local and remote file systems required by Crawlers, including those accessible via FTP, HTPP/S, and WebDav as well as those embedded within compressed files. Finally, and most importantly, the Index Manager offers high-level access to a selection of the functionality provided by Apache's Lucene,[10] a high-performance full-text indexing and search system for cross-platform application development.

The architecture at the *SP* is comparatively simpler, and the software stack exhibits a smaller number of dependencies, as shown in Figure 7. A configurable client-side implementation of the extended OAI Client component periodically requests new `oai_dc` and `tf_basic` records from one or more *DP*s. Upon receiving some, it decompresses them, deserializes them into indlets, and finally hands the indlets over to an Index Manager component for ingestion into a local persistent index common to all *DP*s. Asynchronously, the Index Manager exposes the index to two Searcher components, which rely on content statistics and descriptive metadata to resolve user queries and present query results, respectively. The Searchers resolve queries according to a vector space and a language model. AQ12

We tested the prototype against a distribution of the three collections in the Aquaint TREC corpus across two institutions located in different countries. The SGML documents in each collection were mapped onto individual files while randomly selected files were automatically encoded in PDF to emulate multiplicity of manifestations (SGML manifestations were nonetheless configured as primary.) The resulting file collections then were randomly distributed across ad hoc storage hierarchies and also partitioned along a temporal dimension so that we could test the prototype's behavior with respect to incremental and periodical harvesting. We then simulated a sequence of management events at each collection (e.g., additions, deletions, etc.) and reflected the same sequence against an index of the centralized union of all collections. The small differences observed over time between the index of the global collection and the index built from harvesting each collection have given us confidence in the soundness of the protocol proposed earlier. As importantly, the implementation has confirmed that the additional development complexity associated with the protocol extension concentrates on back-end interactions and that under specific development assumptions, it can be minimized.

[7] See `http://www.oclc.org/research/software/oai/cat.htm`

[8] See `http://tomcat.apache.org`

[9] See `http://jakarta.apache.org/commons/vfs`

[10] See `http://lucene.apache.org`

DOI: 10.1002/asi

Related Work

The relationships between the proposed approach, distributed retrieval, content crawling, and existing implementations of the harvesting model already have been discussed earlier. Note that client-server retrieval already relies on the harvesting model whenever it centralizes content source descriptions for the purposes of selective query distribution (Callan, 2000a, Callan & Connell, 2001, Craswell, 2000). The use of the Z39.50 protocol as an infrastructural medium for the exchange of content source descriptions is explored in Larson (2003); however, these applications of the model occur within a substantially different approach. Source descriptions are course-grained content indices and, as such, support the selection of content source as the run-time targets of query distribution; full-content indices are fine-grained content descriptions and, as such, support local query execution. In the first case, harvesting is ancillary to distributed retrieval; in the second case, it enables centralized retrieval of remotely distributed content.

Additional synergies between content crawling and metadata harvesting also may be found in Liu, Maly, Zubair, & Nelson, (2002), Nelson, de Sompel, Liu, Harrison, & McFarland, (2005), and Warner et al. (2006), where the OAI infrastructure of *DP*s and *SP*s is leveraged toward improved indexing of Web-accessible content. As referred to earlier, the direct use of the OAI-PMH for content crawling is addressed in de Sompel et al. (2004) and in Dijk (2004).[11] The relevance of Information Retrieval techniques, primarily those related to both lossy and lossless compression, and the relationship with other extensions of the OAI-PMH already have been mentioned earlier. Here, we concentrate on what, to the best of our knowledge, is the only work which directly shares some of our motivations.

The Harvest system (Bowman, Danzig, Hardy, Manber, & Schwartz, 1995) was initially proposed in the mid-1990s as a sophisticated, fully customizable, end-to-end solution for large-scale, distributed, content-based retrieval over the internetwork. Its open-source implementation has attracted some attention and, to some extent, survives to these days. Harvesting is a central component of the system's architecture, and its contribution to the development of the OAI-PMH has been repeatedly (if somewhat superficially) acknowledged in the DL community. Unlike the OAI-PMH, however, the system poses no conceptual constraints on the semantics of the harvested data, which may range from manually authored, descriptive metadata to automatically computed statistics specific to the type of the processed resources. Text-based formats, in particular, are processed along lines similar to those advocated in this article.

Our work, however, differs in a number of important ways. First, it frames the approach in an evolved infrastructural context, where later developments—particularly XML and the role-based model of OAI-PMH itself—are leveraged toward a more general exchange mechanism than what may be found buried within a closed system. In particular, we operate in a context in which interoperability is predicated on protocol-based solutions rather than on end-to-end implementations. Further, Harvest focuses on the indexing of type-specific content summaries for text resources, which represents just one of many possible applications of the proposed approach; our prototype, for example, follows a standard indexing paradigm. Overall, our work motivates, contextualizes, and generalizes the good properties of an architectural model which has been previously implemented and has yet to receive widespread acceptance.

Conclusions

A topological separation between the processes of indexing and retrieval suits DIR systems in which content is widely distributed and autonomously managed by a scalable number of heterogeneously resourced providers. Indexing is conceptually distributed along with the content and remains the only responsibility of providers; located elsewhere on the network, retrieval is centralized around a periodic and incremental harvest of the indices produced at each provider. As a result, the latency of the internetwork is an observable of harvesting alone while retrieval may interface its users with the good properties normally associated with local processes. Furthermore, the distribution of indexing optimizes the use of shared bandwidth, respects local access control policies, and promotes cost-effectiveness of both content provision and resource pooling within the overlay network.

Outside the scope of content-based DIR, the OAI infrastructure for harvesting descriptive metadata in support of structured retrieval already has been widely and successfully deployed. We have presented an application as well as a minimal extension of the OAI-PMH protocol to show how the infrastructure of harvesting—not only its motivations—may be leveraged for full-text retrieval. While we believe that uncontrolled extensions of a well-established protocol do not normally reflect good practices—not even backwards-compatible ones, such as the one we propose—we have preferred to explore the rich design space of an optimal solution as well as present a more pragmatically viable alternative. By doing so, we hope to induce the OAI community to engage in a debate over the merits of extending the protocol and, ideally, to plan for greater extensibility in future revisions of the protocol.

The work presented in this article addresses architectural and infrastructural issues for content-based DIR. Accordingly, issues of evaluation may only relate to the consumption of system resources rather than to the models and algorithms which normally impact on retrieval *effectiveness*. Indeed, one of the main motivations underlying the approach is to deliver guarantees of effectiveness and consistency, which are normally associated with centralized retrieval. While we have provided a body of analytical evidence in justification of the

[11] See also Sitemaps, a recent Google initiative which uses OAI as one of the optimization mechanisms for crawling (https://www.google.com/webmasters/sitemaps/docs/en_GB/about.html).

DOI: 10.1002/asi

approach, a great deal of experimental evidence is required, particularly in relation to the issue of complexity associated with the infrastructure of *DP*s.

Finally, we have tested and demonstrated the approach in a prototype service for multimodel retrieval of distributed and potentially unmanaged file collections. Implementing the prototype has increased our confidence in the analytic conclusions, but it is clear that real-world experience on a much larger scale is required before the viability of the approach may be safely concluded. In this sense, our hope is that this work may raise sufficient interest within communities of practice to solicit additional implementations of the approach.

Acknowledgments

This research was partially supported by the EC in the context of the DIgital Library Infrastructure on Grid ENabled Technology (DILIGENT) project. More information on the DILIGENT project can be found at http://www.diligentproject.org

References

Anan, H., Liu, X., Maly, K., Nelson, M., Zubair, M., French, J.C., Fox, E., & Shivakumar, P. (2002). Preservation and transition of NCSTRAL using an OAI-based architecture. In Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 181–182). New York: ACM Press.

Bailey, C.W. (2005). Open access bibliography: Liberating scholarly literature with e-prints and open access journals. Association of Research Libraries (ARL).

Bowman, C.M., Danzig, P.B., Hardy, D.R., Manber, U., & Schwartz, M.F. (1995). The Harvest information discovery and access system. Computer Networks and ISDN Systems, 28(1–2), 119–125.

Callan, J. (2000a). Advances in Information Retrieval, chapter Distributed Information Retrieval (pp. 127–150). London, Kluwer.

Callan, J. (2000b). Distributed IR testbed definition: trecl23-100-bysourcecallan99. v2a. Technical Rep. Carnegie Mellon University, Language Technologies Institute.

Callan, J., Crestani, F., Nottelmann, H., Pala, P., & Shou, X.M. (2003). Resource selection and data fusion in multimedia distributed digital libraries. In Proceedings of the 26th annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 363–364). New York: ACM Press.

Callan, J., Fuhr, N., & Nejdl, W. (Eds.). (2004a, July). Proceedings of the SIGIR Workshop on Peer-to-Peer Information Retrieval, 27th annual International ACM SIGIR Conference, Sheffield, United Kingdom.

Callan, J.P., & Connell, M.E. (2001). Query-based sampling of text database. Information Systems, 19(2), 97–130.

Callan, J.P., Crestani, F., & Sanderson, M. (Eds.). (2004b, August). Distributed Multimedia Information Retrieval, SIGIR 2003 Workshop on Distributed Information Retrieval, Toronto, Canada, Revised Selected and Invited Paper. Vol. 2924 of Lecture Notes in Computer Science. Springer.

Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y.S., & Soffer, A. (2001). Static index pruning for information retrieval systems. In Proceedings of the 24th annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 43–50), New York: ACM Press.

Chou, C.-C., Kuo, P.-X., Ho, J.-M., & Lee, D. (2003). Union catalog using extended oai-pmh. Public Draft.

Craswell, N.E. (2000). Methods for Distributed Information Retrieval. Unpublished doctoral dissertation, Australian National University.

Crow, R. (2000). The case for institutional repositories: A SPARC position paper.

DCMI. (2004). The Dublin Core Metadata Initiative, Dublin core metadata element set, Version 1.1: Reference description. Public Draft.

de Sompel, H.V., Nelson, M.L., Lagoze, C., & Warner, S. (2004). Resource harvesting within the oai-pmh framework. D-Lib Magazine, 10(12).

de Sompel, H.V., Young, J.A., & Hickey, T.B. (2003). Using the oai-pmh . . . differently. D-Lib Magazine, 9(7/8).

Dijk, E. (2004). Sharing grey literature by using OA-x. Public Draft.

French, J.C., Powell, A.L., Callan, J., Viles, C.L., Emmitt, T., Prey, K.J., & Mou, Y. (1999). Comparing the performance of database selection algorithms. In Proceedings of the 22nd annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 238–245). New York: ACM Press.

Gatenby, J. (2002). Aiming at quality and combined: Blending physical and virtual union catalogues. Online Information Review, 26(5), 326–334.

Gravano, L., Chang, C.-C.K., Garcia-Molina, H., & Paepcke, A. (1997). STARTS: Stanford proposal for internet meta-searching. In Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data (pp. 207–218). New York: ACM Press.

Joint Information Systems Committee. (2001). Information environment: Development strategy 2001–2005. Public Draft.

Lagoze, C., Arms, W., Gan, S., Hillmann, D., Ingram, C., Krafft, D., Marisa, R., Phipps, J., Saylor, J., Terrizzi, C., Hoehn, W., Millman, D., Allan, J., Guzman-Lara, S., & Kalt, T. (2002a). Core services in the architecture of the national science digital library (nsdl). In Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 201–209). New York: ACM Press.

Lagoze, C., & de Sompel, H.V. (2001). The open archives initiative: Building a low-barrier interoperability framework. In Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 54–62). New York: ACM Press.

Lagoze, C., de Sompel, H.V., Nelson, M., & Warner, S. (2002b). The open archives initiative protocol for metadata harvesting (2.0). Public Draft.

Larson, R.R. (2003, August). Distributed IR for digital libraries. In Research and Advanced Technology for Digital Libraries, Proceedings of the 7th European Conference, Trondheim, Norway (pp. 487–498).

Liu, X., Maly, K., Zubair, M., & Nelson, M.L. (2002). Dp9: an oai gateway service for web crawlers. In Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 283–284). New York: ACM Press.

Lu, J., & Callan, J. (2002). Pruning long documents for distributed information retrieval. In Proceedings of the 11th International Conference on Information and Knowledge Management (pp. 332–339). New York: ACM Press.

Lu, J., & Callan, J. (2003). Content-based retrieval in hybrid peer-to-peer networks. In Proceedings of the 12th International Conference on Information and Knowledge Management (pp. 199–206). New York: ACM Press.

Lu, J., & Callan, J. (2005). Federated search of text based digital libraries in hierarchical peer-to-peer networks. In ECIR, pages 52–66.

Lynch, C.A. (1997). Building the infrastructure of resource sharing: Union catalogs, distributed search, and cross-database linkage. Library Trends, 45(3).

Nelson, M.L., de Sompel, H.V., Liu, X., Harrison, T.L., & McFarland, N. (2005). mod_oai: An apache module for metadata harvesting. Public Draft.

Nottelmann, H., & Fuhr, N. (2003). The mind architecture for heterogeneous multimedia federated digital libraries. In Distributed Multimedia Information Retrieval, pages 112–125.

Paepcke, A., Brandriff, R., Janee, G., & Larson, R. (2003). Search middleware and the simple digital library interoperability protocol. D-Lib Magazine, 6(3).

Sanderson, R. (2003). Srw: Search/retrieve webservice. Public Draft.

Simeoni, F. (2004). Servicing the federation; The case for metadata harvesting. In ECDL, pages 389–399.

Simon, B., Massart, D., Assche, F.V., Ternier, S., & Duval, E. (2003). Simple query interface specifications. Public Draft.

Simons, G., & Bird, S. (2003). The open language archives community: An infrastructure for distributed archiving of language resources.

DOI: 10.1002/asi

Suleman, H., & Fox, E.A. (2002). Designing protocols in support of digital library componentization. In Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries (pp. 568–582). London: Springer-Verlag.

van der Kuil, A., & Feijen, M. (2004). The dawning of the Dutch Network of Digital Academic Repositories (DARE): A shared experience. Ariadne Magazine, 41.

Warner, S., Bekaert, J., Lagoze, C., Liu, X., Payette, S., & Van de Sompel, H. (in press). Pathways: Augmenting interoperability across scholarly repositories. International Journal on Digital Libraries. Currently available in online form at http://www.citebase.org/abstract?id=oai:ar Xiv .org:cs/0610031 AQ11

Witten, I.H., Moffat, A., & Bell, T.C. (1999). Managing gigabytes: Compressing and indexing documents and images (2nd ed.). San Francisco: Kaufmann.

Yang, B., & Garcia-Molina, H. (2001). Comparing hybrid peer-to-peer systems. In Proceedings of the 27th International Conference on Very Large Data Bases (pp. 561–570). San Francisco: Kaufmann.

Z39.50 Maintenance Agency. (2003). Information retrieval (z39.50): Application service definition and protocol specification. AQ10

DOI: 10.1002/asi

Author Queries

AQ1: Please provide complete addresses for each author.
AQ2: Please spell out "APIs."
AQ3: Please provide the location of the publiser (ARL).
AQ4: Is there a report number?
AQ5: Please provide the location of the publisher (Springer).
AQ6: Is there a draft number? Is the draft available online or in print? If so, please give the appropriate information.
AQ7: Was this ever published? If so, please update the information.
AQ8: Please list the book/journal(?) title in full, and provide the Editor(s), publisher information, and/or a volume number, if applicable.
AQ9: Please list the Editor(s) and publisher information, if applicable.
AQ10: Please list the publishing informatioin.
AQ11: Is there an update?
AQ12: Please provide Art for figure 7.