Metadata Harvesting for Content-Based Distributed Information Retrieval

FABIO SIMEONI and MURAT YAKICI

email:{fabio.simeoni, murat.yakici}@cis.strath.ac.uk

University of Strathclyde

STEVE NEELY

email:steve.neely@ucd.ie

University College of Dublin

FABIO CRESTANI

email:fabio.crestani@unisi.ch

University of Lugano

February 27, 2007

Abstract

We propose an approach to content-based Distributed Information Retrieval based on the periodic and incremental centralisation of full-content indices of widely dispersed and autonomously managed document sources.

Inspired by the success of the Open Archive Initiative's protocol for *metadata harvesting*, the approach occupies middle ground between content crawling and distributed retrieval. As in crawling, some data moves towards 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 whilst promoting the simplicity, effectiveness, and responsiveness of retrieval. Overall, we argue that the approach retains the good properties of centralised 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 multi-model content-based retrieval of distributed file collections.

1 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 ten years, it has been mainly approached by distributing the retrieval process along with the data: queries have been 'pushed' towards the content and the results of their local execution have been centrally gathered and presented to the user

[1] In the lack of a well established terminology, we use the term *content-based* to characterise 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.

(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 towards 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 optimises 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, minimise network interactions and normalise the partial result rankings produced by potentially diverse models of probabilistic retrieval (cf. [Callan, 2000a, Callan et al., 2004b]). Automatically learned descriptions [Callan and Connell, 2001], cooperative protocols [Gravano et al., 1997], fusion heuristics [Callan et al., 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 non-textual forms of content-based distributed retrieval (e.g. [Nottelmann and Fuhr, 2003]) and begun to explore the potential of cooperative and Grid-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 inter-

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

mittently connected peer engines [Callan et al., 2004a]. Hybrid, double-tiered architectures, in particular, have offered ideal ground for bridging optimisations developed for client/server architectures with the advantages of fully decentralised control (i.e. increased potential for resource pooling, fault tolerance, dynamic self-configuration, and privacy)[Lu and Callan, 2003, Yang and Garcia-Molina, 2001, Lu and Callan, 2005].

Rather independently and over a longer period of time, the Digital Library community has also 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 exchanged within the client/server architecture described above; the Z39.50 protocol [Z39.50 Maintenance Agency, 2003], in particular, has standardised the syntax and semantics of such exchange. Recently, more lightweight, Web-based protocols for distributed retrieval have also been proposed (e.g. [Sanderson, 2003, Simon et al., 2003, Paepcke et al., 2003]).

Over the past five years, however, the DL community has progressively favoured the complementary approach of iteratively and incrementally centralising metadata as a pre-condition to the retrieval of the associated data: metadata has been ‘pulled’ towards the queries in advance of their execution and the retrieval function has remained centralised. Figure 2 shows the data flow in metadata harvesting. In the *off-line 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 stores it in a metadata repository MR. In the *on-line phase*, SP interfaces users and resolves their queries against the metadata in MR.

Standardised de facto by the Protocol for Metadata Harvesting of the Open Archive Initiative (OAI-PMH) [Lagoze et al., 2002b], the strategy mentioned above 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 Digital Library field (e.g. [Lagoze et al., 2002a, Anan et al., 2002, van der Kuil and Feijen, 2004, Joint Information Systems Committee, 2001]). A principled analysis of such success is found in [Simeoni, 2004] and is summarised in the next Section.

1.1 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 [Lynch, 1997, Gatenby, 2002]. Bandwidth fluctuations induced by traffic congestions 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, post-processing 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 centralised or *locally* distributed.

From a sociological perspective, the model captures the disparity of strengths and interests which characterise 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, post-processing 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 and 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 necessary requirement (cf. query-based sampling techniques for automatic synthesis of source description [Callan and Connell, 2001]). By centralising 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 decentralisation but it completely eliminates single points of failure. Finally, harvested data is 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.

1.2 Scope and Motivations

In this paper, we investigate the applicability of the harvesting model to content-based retrieval. The motivation is two-fold. Firstly, we hope to expand the scope of Distributed Information Retrieval beyond the assumptions which have bound it so far: as we show below, scale and data ownership may still prove important requirements and yet distributed data need no longer imply distributed retrieval.

Secondly, 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 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 the 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 to be questioned: any Web search engine stands as a witness to the feasibility of moving data towards 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 are also found within more focused communities, with the OAI-PMH in place of crawling as the mechanism for centralising content [de Sompel et al., 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 centralised 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 is neither statically published nor publicly accessible; data which is proprietary, costs money, demands access control, or is 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:

[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

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 Section 2 and Section 3, respectively. In Section 4, 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 Section 5 before drawing some conclusions in Section 6. 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.

2 The Approach

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

2.1 Harvesting Scenarios

Consider first a prototypical harvesting scenario in which a service provider relies on the OAI-PMH to periodically centralise descriptive metadata about 'eprints' – i.e. 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

analysis and thus operations of case normalisation and stemming.

defined and maintained.

At each repository, a dissemination service implements the server side of the OAI-PMH and resolves protocol requests by: (i) 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 (ii) mapping the results expressed in the local metadata model onto instances of a model agreed upon for exchange, say unqualified Dublin Core [DCMI, 2004].

At the service provider, the DC records are normalised and otherwise enhanced; for example, duplicates are removed and subject classification headings are automatically inferred using a third-party web service. Finally, the post-processed metadata is 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 is 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 *off-line phase*, a service provider SP periodically and incrementally gathers metadata M and content statistics I from a number of data providers $DP_1, DP_2, , DP_n$ and persistently stores them in a metadata repository MR and a full-text index FT, respectively. In the *on-line 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 agreed-upon for exchange. At the service provider, the statistical information is extracted and used to update the centralised full-text index, possibly after having been normalised and 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 model 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.

2.2 Requirements

From a conceptual perspective, the extension is relatively straightforward. Its only requirement is for the service provider 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, for 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 et al., 1999]. Overall, service providers can distribute indexing across content sources and maintain their centralised 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:

(i) 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 synchronisation 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.

(ii) accommodate the computational load which is normally associated with the increased size of indexing information over descriptive metadata.

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, data providers 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, Section 4 shows that – under specific deployment assumption and QoS guarantees – low-cost implementations of the proposed extension are certainly possible. It should also be noted that while such 'grassroots' scenarios are well within the remit of current applications of the OAI-PMH protocol – and should thus 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 optimised 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 off-line 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 may also 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 summarisation algorithms (e.g. [Carmel et al., 2001, Lu and Callan, 2002]) – may all grant additional size reductions without excessively compromising the final quality of retrieval.

Admittedly, reducing the *amount* of information exchanged between data and service providers – 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]. It should be noted, 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 data providers 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 normalised at the harvester side; normalisation procedures, in particular, occur off-line 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 centralised one are unavoidably associated with information loss at the harvester side. Notice, however, that it is well within the harvesting philosophy to leave data providers free to choose the optimal trade-off between the consumption of their computational resources – which may be minimised by an aggressive indexing policy – and the visibility of those resources within the federated environment, which may be instead maximised by a relaxed indexing strategy. Put differently, data providers 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.

3 The Protocol

We first summarise 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.

3.1 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 may then use timestamps to temporally scope their

ListRecords request and ListIdentifiers requests. To support *selective* harvesting, servers may organise data in one ore more hierarchies of potentially overlapping datasets; clients may then 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 'tunnelled' 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 metadata; indeed, all servers are 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 lists 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>
```

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

```
      </metadata>

    </record>

    ...

  </ListRecords>

</OAI-PMH>
```

3.2 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 et al., 2003].

In other cases, the exchange semantics has been extended to accommodate additional functionality. For example, protocol extensions have supported inter-components interactions within distributed DL frameworks [Suleman and Fox, 2002], content crawling [Dijk, 2004], authentication, subscription, and notification schemes [Chou et al., 2003], as well as functionality intended to reduce complexity for data and service providers within specific communities of adoption [Simons and 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:

(i) a specialisation of the protocol's data model;

(ii) the definition of a dedicated format for the integrated exchange of descriptive metadata *and* content statistics.

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

(a) resources have at least one digital and text-based physical manifestation ;

(b) a distinguished manifestation of the resource, the *primary manifestation*, satisfies (a) 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, so as to avoid the synchronisation 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 *whilst 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 ad-hoc 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 standardisation 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 specialisation defined above, 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;

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 in Section 3.1 may then be the following:

```
http://www.dp.org/oai?
verb=ListRecords&metadataPrefix=oai_dc&indexPrefix=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, 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 is also *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, whilst a protocol extension denies technical guarantees of

[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 by 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.

backward-compatibility. We believe that the latter offers nonetheless a stronger proof of concept and thus suits best the purposes of this paper. Accordingly, we adopt the proposed protocol extension to test the prototype implementation discussed in the next Section.

4 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 data providers[6]. User queries at the service provider are resolved against a local index asynchronously populated with content statistics which are periodically and incrementally gathered from the data providers. Communication between service and data providers is governed by the protocol proposed in Section 3.

For simplicity, collection management at data providers is modelled 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 data and service providers. The main components at the service provider side are shown in Figure 5 and are briefly described as follows.

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

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-organisational 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 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 (such as PDF or XML vocabularies) may be leveraged towards 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: (i) filtered against a list of stop-words, (ii) normalised into a list of distinct stems, (iii) annotated with their frequency of occurrence, and finally (iv) 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 a `oai_dc` record and a `tf_basic` record. It then serialises the list of such records in XML as shown in Section 3, and finally returns the serialisation to remote clients in a compressed form.

The software stack at the data provider is rooted in the Java platform platform, as shown in Figure 6. The implementation maximises 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

[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`.

by Crawlers, including those accessible via FTP, HTPP/S, 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], an high-performance full text indexing and search system for cross-platform application development.

The architecture at the service provider 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 data providers. Upon receiving some, it decompresses them, deserialises them into indlets, and finally hands the indlets over to an Index Manager component for ingestion into a local persistent index common to all data providers. Asynchronously, the Index Manager exposes the index to two Searcher components which rely on content statistics and descriptive metadata to, respectively, resolve user queries present query results, respectively. The Searchers resolve queries according to a vector space and a language model.

We have 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 have been mapped onto individual files, while randomly selected files have been automatically encoded in PDF to emulate multiplicity of manifestations (SGML manifestations were nonetheless configured as primary). The resulting file collections have then been randomly distributed across ad-hoc storage hierarchies and also partitioned along a temporal dimension, so that we could test the prototype's be-

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

haviour with respect to incremental and periodical harvesting. We have then simulated a sequence of management events at each collection (additions, deletions, etc.) and reflected the same sequence against an index of the centralised 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 in Section 3. 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 minimised.

5 Related Work

The relationships between the proposed approach, distributed retrieval, content crawling, and existing implementations of the harvesting model have been already discussed in Section 1. It is worth emphasising here that client-server retrieval already relies on the harvesting model whenever it centralises content source descriptions for the purposes of selective query distribution [Callan, 2000a, Callan and 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 occurs within a substantially different approach. Source descriptions are course-grained content indices and as such support the selection of content source ass 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 it enables centralised retrieval of remotely distributed content.

Additional synergies between content crawling and metadata harvesting may also be found in [Liu et al., 2002], [Nelson et al., 2005], and [Warner et al., 2006] where the OAI infrastructure of data providers and service providers is leveraged towards improved indexing of Web-accessible content. As referred to in Section 1, 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 have already been mentioned in Section 2.2 and Section 3.2, respectively. 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 et al., 1995] was initially proposed in the mid-nineties as a sophisticated, fully customisable, end-to-end solution for large-scale, distributed, content-based retrieval over the inter-network. 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 Digital Library 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

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

similar to those advocated in this paper.

Our work, however, differs in a number of important ways. First of all, 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 towards 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 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, contextualises, and generalises the good properties of an architectural model which has been previously implemented and yet has to receive widespread acceptance.

6 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 centralised around a periodic and incremental harvest of the indexes produced at each provider. As a result, the latency of the inter-network 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 optimises 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 has already been widely and successfully deployed. We have presented an application as well as a minimal extension of the OAI-PMH protocol in order 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 presenting 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 paper 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 centralised retrieval. While we have provided a body of analytical evidence in justification of the approach, a great deal of experimental evidence is required, particularly in relation to the issue of complexity associated with the infrastructure of data providers.

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.

7 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 et al., 2002] Anan, H., Liu, X., Maly, K., Nelson, M., Zubair, M., French, J. C., Fox, E., and Shivakumar, P. (2002). Preservation and transition of NCSTRL using an OAI-based architecture. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 181–182, New York, NY, USA. ACM Press.

[Bailey, 2005] Bailey, C. W. (2005). *Open Access Bibliography: Liberating Scholarly Literature with E-Prints and Open Access Journals.* Association of Research Libraries (ARL).

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

[Callan, 2000a] Callan, J. (2000a). *Advances in Information Retrieval*, chapter Distributed Information Retrieval, pages 127–150. Kluwer Academic Publishers.

[Callan, 2000b] Callan, J. (2000b). Distributed IR testbed definition: trec123-100-bysourcecallan99. v2a. Technical report, Language Technologies Institute, Carnegie Mellon University.

[Callan et al., 2003] Callan, J., Crestani, F., Nottelmann, H., Pala, P., and Shou, X. M. (2003). Resource selection and data fusion in multimedia distributed digital libraries. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 363–364, New York, NY, USA. ACM Press.

[Callan et al., 2004a] Callan, J., Fuhr, N., and Nejdl, W., editors (2004a). *Proceedings of the SIGIR Workshop on Peer-to-Peer Information Retrieval, 27th Annual International ACM SIGIR Conference, July 29, 2004, Sheffield, UK.*

[Callan and Connell, 2001] Callan, J. P. and Connell, M. E. (2001). Query-based sampling of text databases. *Information Systems*, 19(2):97–130.

[Callan et al., 2004b] Callan, J. P., Crestani, F., and Sanderson, M., editors (2004b). *Distributed Multimedia Information Retrieval, SIGIR 2003 Workshop on Distributed Information Retrieval, Toronto, Canada, August 1, 2003, Revised Selected and Invited Papers*, volume 2924 of *Lecture Notes in Computer Science*. Springer.

[Carmel et al., 2001] Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y. S., and Soffer, A. (2001). Static index pruning for information retrieval systems. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA. ACM Press.

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

[Craswell, 2000] Craswell, N. E. (2000). *Methods for Distributed Information Retrieval.* PhD thesis, Australian National University.

[Crow, 2002] Crow, R. (2002). The case for institutional repositories: A SPARC position paper.

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

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

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

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

[French et al., 1999] French, J. C., Powell, A. L., Callan, J., Viles, C. L., Emmitt, T., Prey, K. J., and Mou, Y. (1999). Comparing the performance of database selection algorithms. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 238–245, New York, NY, USA. ACM Press.

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

[Gravano et al., 1997] Gravano, L., Chang, C.-C. K., Garcia-Molina, H., and Paepcke, A. (1997). STARTS: Stanford proposal for internet meta-searching. In *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 207–218, New York, NY, USA. ACM Press.

[Joint Information Systems Committee, 2001] Joint Information Systems Committee (2001). Information environment: Development strategy 2001-2005. Public Draft.

[Lagoze et al., 2002a] 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., and Kalt, T. (2002a). Core services in the architecture of the national science digital library (nsdl). In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 201–209, New York, NY, USA. ACM Press.

[Lagoze and de Sompel, 2001] Lagoze, C. and de Sompel, H. V. (2001). The open archives initiative: building a low-barrier interoperability framework. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 54–62, New York, NY, USA. ACM Press.

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

[Larson, 2003] Larson, R. R. (2003). Distributed IR for digital libraries. In *Research and Advanced Technology for Digital Libraries, 7th European Conference, ECDL 2003, Trondheim, Norway, August 17-22, 2003, Proceedings*, pages 487–498.

[Liu et al., 2002] Liu, X., Maly, K., Zubair, M., and Nelson, M. L. (2002). Dp9: an oai gateway service for web crawlers. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 283–284, New York, NY, USA. ACM Press.

[Lu and Callan, 2002] Lu, J. and Callan, J. (2002). Pruning long documents for distributed information retrieval. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 332–339, New York, NY, USA. ACM Press.

[Lu and Callan, 2003] Lu, J. and Callan, J. (2003). Content-based retrieval in hybrid peer-to-peer networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 199–206, New York, NY, USA. ACM Press.

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

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

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

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

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

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

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

[Simon et al., 2003] Simon, B., Massart, D., Assche, F. V., Ternier, S., and Duval, E. (2003). Simple query interface specifications. Public Draft.

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

[Suleman and Fox, 2002] Suleman, H. and Fox, E. A. (2002). Designing protocols in support of digital library componentization. In *ECDL '02: Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 568–582, London, UK. Springer-Verlag.

[van der Kuil and Feijen, 2004] van der Kuil, A. and Feijen, M. (2004). The dawning of the Dutch Network of Digital Academic REpositories (DARE): A shared experience. *Ariadne Magazine*, (41).

[Warner et al., 2006] Warner, S., Bekaert, J., Lagoze, C., Liu, X., Payette, S., and Van de Sompel, H. (2006). Pathways: Augmenting interoperability across scholarly repositories. Accepted for the International Journal on Digital Libraries, Special Issue on Digital Libraries and eScience. Currently available in online form at http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0610031.

[Witten et al., 1999] Witten, I. H., Moffat, A., and Bell, T. C. (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers Inc., 2nd edition.

[Yang and Garcia-Molina, 2001] Yang, B. and Garcia-Molina, H. (2001). Comparing hybrid peer-to-peer systems. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 561–570, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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