Pharos: A Scalable Distributed Architecture for Locating Heterogeneous Information Sources

R. Dolin D. Agrawal A. El Abbadi L. Dillon
Department of Computer Science
University of California
Santa Barbara, CA 93106

rad@cs.ucsb.edu
805-893-4239 (phone), 805-893-8553 (fax)

May 1, 1997

Abstract

Pharos is a scalable distributed architecture for locating heterogeneous information sources. The system incorporates a hierarchical metadata structure into a multi-level retrieval system. Queries are resolved through an iterative decision-making process. The first step retrieves coarse-grain metadata, about all sources, stored on local, massively replicated, high-level *servers. Further steps retrieve more detailed metadata, about a greatly reduced set of sources, stored on remote, sparsely replicated, topic-based* mid-level *servers. We present results of a simulation which indicate the feasibility of the architecture. We describe the structure, distribution, and retrieval of the metadata in Pharos to enable users to locate desirable information sources over the Internet.*

Keywords: resource discovery, distributed systems, scalability, metadata, Internet, document database.

1 Introduction

Finding appropriate sources of information on the Internet increases in difficulty with the increasing number of Internet users, the number of sources providing electronic documents, the number of systems for indexing these documents, and the volume of available data. This is especially the case since an information source can be considered to be any site on the Internet that provides electronic documents such as journal articles, books, hypertext pages, images, files, etc. Prior to digital networks, the number of potential information sources was relatively limited, mainly by locality; hence, a user could discover what was available by browsing individual sources. In a wide-area network such as the Internet, information resides in numerous heterogeneous systems, which are geographically distributed; each such distributed system may provide many information sources. Various systems have been built to provide tools that aid in the discovery of information, e.g., Archie, Veronica, Yahoo. However, users are still easily lost in the vast space of documents that are indexed by these systems. No system-independent location facility currently exists. What was once perhaps a problem of limited choice has become a problem of too much choice. The existence of thousands of newsgroups, World-Wide Web (WWW) sites, FTP archives, and digital libraries renders traditional methods of locating good information sources infeasible.

There has been much work done in the area of automatic text retrieval [32], for example SMART [33], Latent Semantic Indexing (LSI) [3], and others involved with the NIST Text REtrieval Conferences (TREC) [21]. Our work, though, focuses on the selection of *collections* of documents rather than on particular documents themselves. Thus, we are not directly addressing the problem of a user finding a particular document via

author, title, keyword, etc. However, a particular author or title is generally included in one or more information domains, such as subject area, geographical region, time period, image feature, type of business, etc. By focusing on finding collections which are characterized by such domains, we believe that users are likely to find those authors and titles which are relevant to the concepts underlying specialized queries. The reason for such a change in focus is that we believe that locating collections by domains is a more scalable methodology than locating particular authors, titles, etc.

Current WWW indexes, such as AltaVista[11], Lycos [28], and Yahoo [39], which are designed for locating information on the Internet, are limited in several ways. Notably, these systems do not scale to handle increasingly large numbers of search requests due to limited network bandwidth and server power. Furthermore, because they focus on keyword matching, they are usually unable to provide adequate simultaneous *precision* and *recall*[1]. Existing WWW indexing systems utilize either small, hand-made, hierarchical lists, as in Yahoo, or word-matching on huge document-spaces, as in AltaVista. Standard techniques of automatic text retrieval, such as relevance feedback and local/global term weightings, aid in matching query terms to relevant documents; however, these techniques are seldom used for locating sources. Another factor that limits most existing WWW indexes is their inability to share index information. This lack of sharing results in highly duplicated document fetching and indexing. As the number of indexing sites grows, determining which ones contain the most useful information for a given query becomes increasingly difficult. Furthermore, few WWW indexing systems give sources control over how often they are indexed, or provide for automatic updates when information becomes stale. Another problem with many WWW indexes is that they index the actual texts of documents. This approach is not appropriate for indexing catalogs, especially those which list documents which are not available on-line, and it does not extend to non-text documents, such as maps, images, and sound and video files. Information sources that make their documents available only as query results have no mechanism of including their contents within such indexing systems.

Problems of locating sources of information, one form of *resource discovery* [5], are of particular interest to the digital library community [17]. Such emerging systems must deal with storing, locating, cataloging, distributing, and retrieving large numbers of diverse documents in a distributed, heterogeneous environment. Several projects in the NSF/NASA/DARPA Digital Library Initiative Program [6, 30] are therefore investigating these problems. The Alexandria Digital Library (ADL) Project [2] is focusing on indexing spatial information, as well as on the storage, distribution, and retrieval of large (spatial) images. The Illinois Digital Library Project [34] is building a prototype of the *Interspace*, which presents the Internet as a single space of highly interlinked, distributed information. The University of Michigan Digital Library Project [10] is designing an agent-based system; user interface agents, mediation agents, and collection agents cooperate to allow concurrent searching of multiple collections. At Stanford, the *gGlOSS* resource discovery system [19] represents sources by vectors of term frequencies. This work has been extended to the STARTS proposal [18] in which sources extract their own metadata and pass it to intermediaries. Content Routing [35] is a system in which queries get routed to the available servers based on the expected relevance of the server to the query. Harvest [4] automatically indexes documents within a source and distributes these indexes. Other work includes [1, 7, 8, 16, 37].

The predominant feature of the majority of these systems is that they are based on text documents. Furthermore, their collections are often most easily located when they are more homogeneous in nature, such as those found in the TREC data. Another aspect of these systems is that it is unclear to what degree they are scalable in terms of the number of sources, the number of users, and the number of documents. Finally, although many systems provide for the transfer of collection metadata, they do not specify exactly the content and/or format of the metadata, where specific pieces of metadata are to be placed within the network architecture, or the exact metadata transmission methodology involved in the handling of queries and updates. Our goal is to develop a system that scales well in terms of the information gathering with the growing diversity of data, the information dispersion with the growing user base, and the information visualization with the growing data volume [5]. In this paper, we describe the design of such a distributed architecture, Pharos, for locating desirable information sources. For Pharos, scalability is intended to mean 10^5 or more sources. We achieve this scalability by using a hierarchical metadata structure and a highly decentralized metadata distribution, storage, and retrieval mechanism. Furthermore, our approach provides a significant level of autonomy to the individual information sources to distribute their metadata information.

[1] *Precision* is the fraction of returned documents that are relevant to a query, while *recall* is the fraction of relevant documents in a collection that are returned [32].

Pharos does not use a simple linear ranking of all sources as is done in standard vector-based text retrieval techniques. Instead, it provides users greater freedom in the visualization and selection criteria of their final set. Pharos is intended to work with heterogeneity not only *between* collections, but also *within* collections. This diversity includes both data format and data content. Finally, the system described here specifies the metadata structure, the placement of various metadata components within the network architecture, and the synchronization of the metadata transfers involved in queries and updates.

The next section motivates an iterative, multi-level query method to locate information sources, and gives an overview of our approach. Section 3 describes the hierarchical metadata structure needed for such an approach. In Section 4, we evaluate the feasibility of our methodology with a simulation study. We give a more detailed description of the architectural requirements of the system in Section 5. Finally, we conclude with a description of the contributions of this work and our current research directions.

2 Design Overview

2.1 Motivation

We consider a modern Internet environment with a large number of users and information sources, including WWW sites, FTP archives, digital libraries, file systems, etc. Attempting to locate a small set of sources that best fit user queries requires detailed knowledge of the holdings at each source. There are two extreme approaches for acquiring this metadata. The first, which we informally call the *remote approach*, is to query each source dynamically in turn (or possibly in parallel) without storing any local metadata. The second, which we informally call the *local approach*, is for each user to store locally detailed information about the holdings of each source and not to request anything remotely (until a final set of sources has been selected for direct access). Both of these extremes are impractical. The remote approach would take too long even for a single user query, let alone a large number of simultaneous ones. The local approach needs to have sufficient information for any given query; we believe that the amount of metadata required for accurate source comparison is too large to be practical.

Our approach is therefore a hybrid one in which we attempt to balance the location of the metadata's storage with network traffic by using an iterative, multi-level query method. A limited amount of metadata about each source is massively replicated among "high-level" servers. These servers are locally situated within an organization such as a university campus or a corporate network. They receive requests from local users about most or all known sources and reply with only enough metadata to allow for a rough comparison of sources. More detailed metadata is stored in specialized, sparsely replicated "mid-level" servers. These servers receive requests from remote users about a small set of sources and supply greater detail than the high-level servers. All high-level and mid-level servers store information about each source. However, while all high-level servers store the same metadata, each mid-level server (up to replication) stores unique metadata within a sub-area of a single, pre-specified, domain-specific classification tree, or *taxonomy*. For example, we expect separate mid-level servers specializing in subject areas such as history, art, physics; in geographical regions such as North America, Africa, Asia; etc. We call any such sub-area of a particular information domain a *sub-domain*. This design solves the problem of the remote approach by providing a fast, scalable method of receiving any needed metadata. It also solves the storage problem of the local approach by distributing the metadata throughout the network and requiring only minimal storage within a user's local environment. Finally, network traffic is reduced in several ways. Smaller local storage reduces the amount of broadcast-type traffic generated by the sources for update purposes, while the mid-level servers reduce the amount of query traffic to the sources generated by users. Furthermore, having users communicate with local servers for initial source filtering also reduces network traffic.

2.2 Example Query

As an example query, suppose that a student is studying the following topic: "A Comparison of Political Music in 1967: San Francisco, Shanghai, and Cairo." We performed a search through AltaVista, Lycos, Yahoo, and the University of California's Melvyl online general catalog [12]. AltaVista, Lycos, and Yahoo found little of value for this study. Furthermore, the precision was very low, on the order of 1%, requiring excessive viewing of many irrelevant documents in order to locate the rare relevant ones. Melvyl, on the

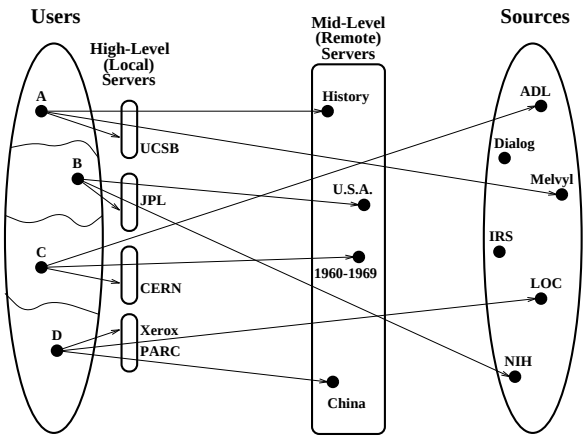

Figure 1: Retrieval Design Overview

other hand, returned 44 items of potential interest, such as "When the music's over: the story of political pop," "Female college students in China," "Rhythm and resistance: explorations in the political uses of popular music," "Qira'at munawi'ah," and several recordings. How do we inform the student that, in this case, Melvyl is a potentially good information source?

In our proposed system, the student attempts to find sources as follows. Initially, at the client-side user interface (UI), she specifies the query in terms of relevant information taxonomies and significant sub-domains within them. The taxonomies involved in this search, for example, include subject, geographical region, and time period. Within each of these general taxonomies, the student is prompted to provide more specific sub-domains that match her query. For the subject domain, the UI attempts to locate the query terms or their synonyms within the taxonomy. For example, the phrase *political music* might be covered by political science, history, and music; sources that excel in these areas are potentially useful for finding information about her query. For the geography taxonomy, the student specifies a longitude/latitude bounding region, or a place-name which is checked against a *gazetteer*[2] to obtain the bounding coordinates.

Once the student has specified the appropriate sub-domains within the three taxonomies, the UI then queries the local high-level server and gives highest weights to those sources that contain relatively more information 1) within the subject taxonomy, in, say, politics, history, or music; 2) within the time period taxonomy, about the 1960's; and 3) within the geography taxonomy, in SW North America, SE Asia, or NE Africa. From the weightings, the student selects some initial set of sources, say the best thousand based on the high-level metadata.[3] Next, the UI queries the relevant seven remote mid-level servers for more detailed metadata about the sources included in the initial set: a political subject server, a history subject server, a music subject server, a 1960-1969 time period server, and the three geographic region servers covering SW North America, SE Asia, and NE Africa. These mid-level servers return their detailed and specialized metadata about the requested sources. For example, the political subject server provides metadata about the percentage of holdings among the initial set of sources particular to political music; the 1960-1969 time period server provides metadata about the percentage of holdings covering 1967; and the SW North America geographic region server provides metadata about the percentage of holdings dealing with the California area. The UI then merges the information returned from the various mid-level servers and presents them to the student. This information can be used to select a small set of sources for further (direct) querying.

2.3 Multi-Level Approach

Our multi-level design overview is depicted in Figure 1. This design assumes an iterative query approach where each source's metadata description is in a standardized format and is broken down into components related to the composition of the holdings with respect to various information taxonomies and metadata

[2] A *gazetteer* is a mapping from symbolic names to geographical coordinates.

[3] Users select sources via a non-textual visualization rather than a list in order to simplify the process.

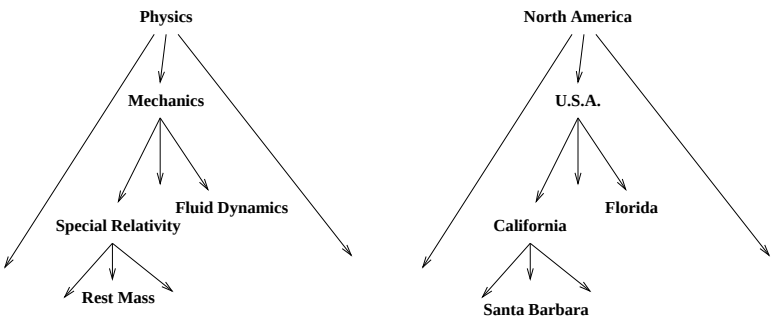

Figure 2: Information Hierarchies

levels. This metadata is propagated and stored at the different servers, such that coarse-grain information resides on massively replicated *high-level* servers and fine-grain information resides on sparsely replicated *mid-level* servers. The high-level metadata should be sufficient to greatly reduce the number of potential sources, say from $\sim 10^6$ to $\sim 10^3$, and continue with the mid-level metadata, down to, say, $\sim 10^1$.

Given such a retrieval system, we need a corresponding metadata distribution system such that the high-level metadata is distributed widely and the mid-level metadata is distributed selectively. The wide distribution mechanism in Pharos is modeled after the distribution of USENET news via NNTP [22, 24]. Unlike the storage mechanism for high-level metadata, mid-level metadata from different taxonomies in Pharos are not all stored together. Each mid-level server stores all the metadata about each source that is related to a unique sub-domain of a particular information taxonomy. These servers are not generally resident within users' local areas, but are sparsely located at remote sites. When a source sends out its mid-level metadata, it sends the different components to the corresponding servers in a point-to-point manner. The distribution of the mid-level metadata in Pharos is modeled after the distribution of indexes in Harvest [4], which supports efficient point-to-point metadata transfer, server replication, caching, and structured querying of servers. Metadata distribution will be further discussed in Section 5.2.

3 Metadata Structure

The metadata must be designed to support a multi-level information system and it should match the intended queries as well as the retrieval system. Therefore, metadata should be grouped according to its informational relationship and designed around a hierarchical structure which lends itself to progressively refined queries. Pharos employs information classifications that fit common query methods, such as a subject-based hierarchy, a geographical hierarchy, and a temporal hierarchy. Once these hierarchies have been defined, we include the top portions in the high-level metadata, and lower portions in the mid-level metadata.

3.1 Information Hierarchies

An *information hierarchy* is a topic-based classification tree, as shown in Figure 2. In order to describe the collection at a source, we quantify the number of documents with content relevant to each part of an information hierarchy. As mentioned above, the initial design of Pharos employs three information hierarchies: subject, geographical, and temporal hierarchies.

The subject hierarchy is modeled after the Library of Congress's *LC Classification* system [26], which has a fixed list of categories. For a given query, terms must be deduced that match these categories. This classification system has several advantages: it is very extensive; it is familiar to most of the library classification community; it is open-ended; and it is used by most academic libraries in the U.S.A. While there are problems with using a fixed list for query terms [14], there are several problems with using vocabularies that are based on the terms found in each document collection. For example, it is difficult to compare sources that do not use the same terms, and we cannot expect different collections to yield the same vocabularies. Moreover, without a static set of terms, it is difficult to construct comparable *hierarchical* structures that can

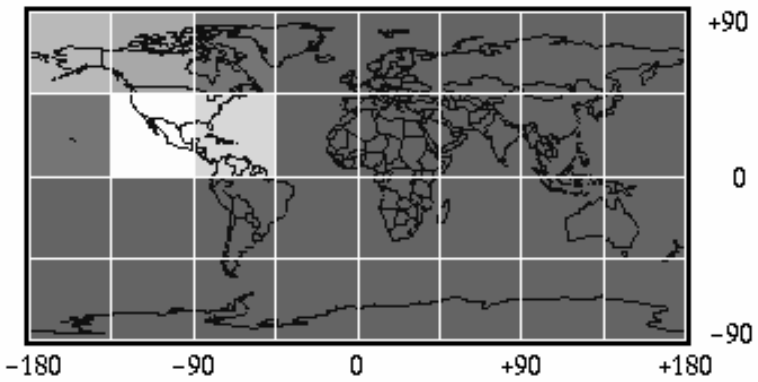

Figure 3: 45° × 45° Grid, 32 Elements

be used to build source metadata within a multi-level approach. The Pharos UI must bridge the gap between the words used by the user and the fixed terms in the subject hierarchy. Techniques such as those described in [9, 14] specifically address the problem of finding the relationships between query terms and potentially different document terms. We plan, therefore, to place users' concepts within appropriate sub-domains in the subject hierarchy, while at the same time fixing the categories so that it is easier to accurately compare sources.

A simple geography information hierarchy is composed by tiling the Earth's surface into progressively smaller, hierarchical, longitude/latitude squares. If the highest level of the hierarchy represents the regions of the Earth's surface defined by 45° × 45° grid lines, as in Figure 3, the result is 32 top-level tiles. In this figure, the actual data density of the ∼421,000 documents of the May, 1996 Alexandria Digital Library (ADL) holdings is shown: the lighter the area in a tile, the more documents it contains. The next level of the hierarchy is formed by sub-dividing each top-level tile by 5° × 5° grid lines yielding 2,592 sub-tiles. Each top-level tile has 81 sub-tiles. This sub-tiling process is continued to create a spatial hierarchy of desired levels of details.

The temporal information hierarchy needs to cover all references to time, including future time. The majority of temporal references, however, deal with the last 300 years. For example, Microsoft's Encarta Encyclopedia 96 allows articles to be looked up by "time". Of the 13,065 temporally referenced articles, 3,731 relate to time before the year 1700; 1,762 relate to the 1700's; 5,487 relate to the 1800's; and 5,641 relate to the 1900's. We therefore prefer to have a higher level of granularity for the time periods nearer to 'now' than those further away on a time-line. By using a modified logarithmic scale, which is approximately linear between the years 1700 and 2000, we can achieve a more even distribution of time-referenced documents (see [12]).

3.2 Information Taxonomies

An information *taxonomy* is defined as a static hierarchy within which documents can be classified, usually according to their content. In addition, a taxonomy is assumed to be a tree structure such that each node in the tree has two attributes: a label and a numerical value. The label denotes a word, term, or phrase from an information hierarchy. The numerical value of each node of a taxonomy is called the *coverage* of the node. It is used to describe a document collection by quantifying the number of documents with content relevant to the meaning of the label of the node. Since most documents cannot be classified completely within a single leaf of a classification tree, we assign portions of a document to different nodes in the tree. Thus for each node in the taxonomy there is a total number of *document equivalents* associated with it. Intuitively, a node's coverage indicates what fraction of a document collection is classifiable within a node's label. In order to calculate this value, we first allocate fractions of each document to nodes in the tree in such a way that not more than 100% of the document is allocated. We allocate as much of a document as we can to nodes closest to the leaves of the tree. In this way, a document is classified as much as possible within the most specific

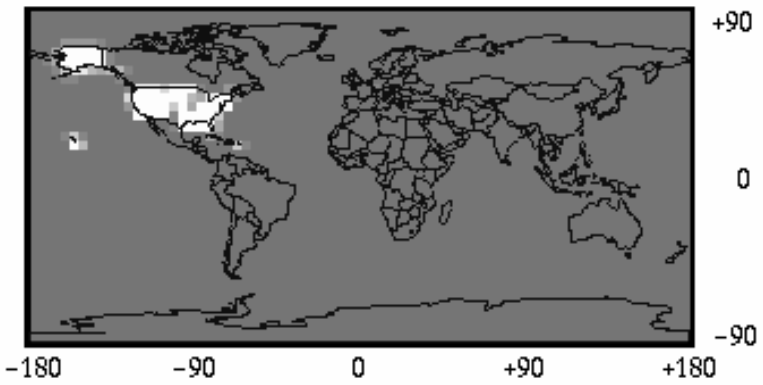

Figure 4: $5° \times 5°$ Grid, 2,592 Elements

sub-domains of the hierarchy. Once all the documents in a collection have been fractionally allocated within the tree, the coverage values can be computed. The children of a parent node have labels which represent information sub-domains which are wholly or partly contained within the sub-domain represented by the label of their parent. By using the notion of coverage, this containment property is captured numerically by defining the coverage at a node to be the sum of the coverages of its children plus the fraction of the collection assigned to it but not to its children. For a leaf node, this is just the fraction of the collection assigned to it. The number of document equivalents associated with a node is its coverage times the total number of documents in the collection.

As an example, consider the geography information hierarchy discussed previously. We create a taxonomy by assigning the $45° \times 45°$ tiles to the depth-1 nodes, the $5° \times 5°$ tiles as depth-2 nodes, etc. Within the ADL collection of $\sim 421,000$ documents shown in Figure 3, the region over Hawaii, defined by the (longitude, latitude) coordinate pair [upper-left$=(-180°, 45°)$, bottom-right$=(-135°, 0°)$], contains $\sim 5,700$ document equivalents. Thus, within the geography taxonomy, this node is assigned a coverage of ~ 0.014. To calculate the coverages of the depth-2 nodes, we break down the depth-1 nodes into $5° \times 5°$ tiles, as shown in Figure 4. This figure shows another view of the same ADL holdings as in Figure 3. The three depth-2 nodes which contain the Hawaiian Islands contain approximately 1500, 2400, and 1100 document equivalents, and so are assigned coverages of ~ 0.0038, ~ 0.0060, and ~ 0.0028, respectively.

Clearly Figure 4 gives a much more precise display of the ADL holdings. However, the 2,592 data elements prohibit that level of detail from being included in the high-level metadata, as discussed in more detail in the next section. On the other hand, Figure 3 does not necessarily include enough detail to be useful. As a case in point, consider a user looking for information about Hawaii. The large-scale region over Hawaii does not appear to contain a sufficient amount of data to make the ADL an attractive source; the average coverage of the 81 depth-2 children is only ~ 0.00017, implying that there are only approximately 220 documents for Hawaii. On closer examination, however, the small-scale map shows that the ADL is potentially a very good source for such a user; there are actually over 5000 documents for that region. With only 5% of the depth-1 node's children containing 97% of its data, a high-level comparison involving the ADL might exclude it and the depth-2 metadata would never be requested. This problem of statistically losing lower level information, and hence potentially valuable sources, can occur within any information taxonomy whenever a small fraction of a node's children contain a vast majority of the node's coverage. Even if we kept track of the variance among a node's descendents, we would not be able to determine which ones held the majority of the information without acquiring more metadata. This situation can be considered a *special collection* within a source's taxonomy. We handle special collections by providing a special metadata attribute.

The geography taxonomy is very different from the subject taxonomy in that it is not a word-based hierarchy, but a spatial hierarchy. To put this in perspective, Microsoft's Encarta Encyclopedia 96 allows articles to be looked up by "place". However, while a query on 'North America' yields 296 articles, a query on 'United States' yields 3,958 articles. These results do not occur under a geography taxonomy which would

place 'United States' within 'North America'; our definition of a taxonomy requires that the spatial nodes covering North America include at least all those articles assigned to the nodes covering the United States.

3.3 Metadata Levels

We briefly discuss here the relevant metadata records that need to be included in the high- and low-level servers. The main constraint in the case of high-level servers is size. The total size T_H of information distributed to the high-level servers is the size S_H of each high-level metadata record describing a given source times the total number N_S of sources described. The overriding constraint is that T_H not grow beyond some maximum value. Using the newsgroup distribution as an example, we assume that each high-level server stores under 10 GB of data. Assuming that we want to be able to handle a large number of sources, say $N_S \leq 10^6$, we derive a value of S_H of roughly 10 KB. A high-level metadata record needs to summarize the coverage information related to various nodes in an information taxonomy. Given these values for N_S and S_H, and the width of the upper levels of the taxonomy trees, we can determine which nodes of the taxonomies should be included in the high-level metadata.[4]

The size of a high-level metadata record is dependent on the number of taxonomies used and the number of nodes that are included from each taxonomy. Although the number of taxonomies a source uses might vary, for a given taxonomy all sources must always include the same nodes in the high-level metadata. High-level metadata refers to 1) any taxonomy-independent source description, and 2) the high-level portions of all taxonomies within which a source's collection has been classified. We determine the number of nodes to include based on our estimated size, S_H, of a high-level record, the number of taxonomies included, N_T, and the size, S_I, of the taxonomy-independent metadata. For simplicity, we assume that each taxonomy's node metadata is the same size, S_N, and that the same number of nodes, $N_H(T)$, will be included from each taxonomy T. Thus we have $S_H = S_I + S_N * N_H(T) * N_T$, or $N_H(T) = (S_H - S_I)/(S_N * N_T)$. Assuming $S_H \approx$ 10KB; $S_I \approx$ 1KB; $S_N \approx$ 100B; and $N_T \leq 4$, we conclude that $N_H(T) \approx 22$. That is, we assume that we can include on the order of 20 to 30 nodes from each taxonomy in our high-level metadata records. We describe the metadata structure in more detail in Appendix A.

Mid-level metadata is not subject to the same space constraints as the high-level metadata for several reasons. First, because mid-level metadata is only sparsely replicated, there are not as many network constraints in distributing it. Second, each mid-level server handles only a specific sub-domain within a single taxonomy, so it needs to store only a very limited subset of all of the mid-level metadata records for any source. Third, because users request this metadata only after they have already gone through a primary filtering using the high-level metadata, mid-level query results contain data for only a relatively small number of sources. The structure of mid-level metadata is similar to the taxonomy-dependent components of the high-level metadata, but covers nodes of greater depth in the information taxonomies.

4 Evaluation

We have performed an initial feasibility study of this model via simulation. Other work compares this model to various network architectures for information discovery [13].

4.1 Simulation Parameters

In order to evaluate the expected performance of Pharos, we have tested it by generating simulated sources and queries and estimating the degree of success of finding the best sources. Such an estimate requires the quantification of a few parameters. When dealing with a standard, single text document database, *precision* usually gives a measure of how many of the returned documents are considered to be relevant to the query [21]. This definition does not extend naturally to the problem of locating sources. A source could be considered relevant if it contains even a single relevant document, resulting in a relevance test that is too broad and unintuitive. Instead, we are interested in a measure that is higher for sources which contain a relatively larger number of relevant documents. For simulation simplicity, we consider measures similar

[4] In the future, we may allow N_S to grow larger by using a single metadata record to represent a hierarchical collection of related sources (e.g., UC source representing the nine UC campus libraries). In order to allow this, the structure of the metadata must be independent of the number of sources described.

to the weights associated with documents based on vector analysis [14, 32], which lead to a ranked list of documents. Assuming that we have a weighting system which will lead to a ranked list of sources, we have several options for defining the precision of the results. We first introduce two such definitions and show how they are problematic, leading us to a final, more intuitive definition. First, we could define source precision of a final set of k sources as the fraction of the k *best* sources; however, this gives too small a value when, for example, the $(k+1)$ best source is substituted for the k^{th} best source in the final set. Second, if we define source precision so as to include the value of such sources, we might use

$$\frac{\sum_{i=1}^{k} i}{\sum_{i=1}^{k} est_i}$$

where est_i is the true rank of the estimated i^{th} best source. The resulting value would be unreasonably small, however, whenever a particularly poor source is selected, even if the other $k-1$ sources are the best ones. Therefore, we define *source precision*, P_S, as follows:

$$P_S \equiv \frac{\sum_{i=1}^{k} 1/est_i}{\sum_{i=1}^{k} 1/i} \tag{1}$$

where est_i is the true rank of the estimated i^{th} best source, and k is the number of sources selected in the final set. Clearly $0.0 < P_S \leq 1.0$. As an example, if we select the best 10 out of 1000 sources, $k = 10$, and the denominator is $1/1+1/2+\cdots+1/10 = 2.93$. Suppose that for a given query, we return a set with the following source rankings: {1,3,4,7,9,14,17,20,27,103}. The numerator becomes $1/1 + 1/3 + 1/4 + \cdots + 1/103 = 2.06$. The source precision is then $2.06/2.93 = 0.70$. The first definition would have yielded 0.50 instead, and the second definition would have yielded 0.27.

The next parameters we need to quantify are the source weightings. A source's *true* weight is different from its *estimated* weight. The former is a function of the actual coverage values and in general is not determinable from a client's machine. The latter is based on the information available through the high-level and mid-level metadata. These weights are complicated by the fact that we cannot assume that all documents in a collection have been classified within each taxonomy. For simulation and evaluation, we compare the rankings based on the true source weights to those obtained from the estimated weights in order to calculate the source precision. We define the weight of a source for a query involving multiple taxonomies, but simplify it here by excluding multiple nodes within any single taxonomy. Queries are then considered to be one node from each of the one or more taxonomies used in the query, limited by the total number t of taxonomies known to Pharos. We currently limit t to 3. Our current weighting is based solely on estimating which sources have the greatest number of relevant documents.[5] A multi-taxonomy query is considered to be a logical AND of the individual nodes within each separate taxonomy. Furthermore, we assume that documents are uniformly distributed among a node's children within a taxonomy. Therefore, we estimate that the total number of documents that are relevant to a query is the product of the total number of documents in the collection times the individual fractions of the collection within each query component. For example, consider a two-taxonomy query with a subject component of political music and a geographical component of San Francisco. If 80% of the documents in a collection match the subject component and 30% of the documents in the collection match the geographical component, then we assume that 24% of the documents in the collection match the query. Therefore, we use the product of the coverages to derive the following *true source weighting* for a given query:

$$S_C \ * \ \prod_{i=1}^{t} \gamma_i \tag{2}$$

[5] Other methods, for example, involve using a conspectus rating or estimates of how well a source covers a subject, region, etc., as well as using any of the taxonomy-independent metadata.

Tax / Query	Number of Queries	Average Precision
1	320	0.76 ± 0.19
2	337	0.63 ± 0.21
3	343	0.58 ± 0.23

Table 1: Precision as a Function of the Number of Taxonomies per Query

where S_C is the size of the collection at the source, t is the number of taxonomies in the query, and γ_i is the coverage at the query node n_i for taxonomy i.

The final parameter we need to define is the *estimated* source weighting. This weight is calculated from the high-level and mid-level metadata. For the taxonomy nodes which define a given query, the corresponding metadata are the high-level and mid-level ancestors of those nodes in the respective taxonomies. If the query node is actually a high-level node, the mid-level 'ancestor' is defined to be the query node itself. We make a slight modification from Equation 2 to account for the fact that the metadata gives a more accurate estimate of nodes which are higher in the taxonomy than those which are lower. The number of nodes which are represented by a high-level (mid-level) node is a function of two factors: 1) Δh, the difference in height between the query node and its high-level (mid-level) ancestor, and 2) N_c, the number of children per node in the taxonomy. For simplicity, we assume here that N_c is a constant for each taxonomy. The number of descendents of the query node's high-level (mid-level) ancestor which are at the query node's height in the tree is then $(N_c)^{\Delta h}$. In order to give more relative weight to query nodes which are higher in their respective trees, we divide the product factor in Equation 2 by $N_c * \Delta h$. This divisor is a compromise between just using the height difference and using the full dilution of $(N_c)^{\Delta h}$. Therefore, the high-level (mid-level) *estimated source weighting* is

$$S_C \ * \ \prod_{i=1}^{t} \left\{ \frac{\gamma_i'}{(N_c * \Delta h_i)} \right\} \tag{3}$$

where S_C is the size of the collection at the source, t is the number of taxonomies in the query, γ_i' is the coverage at the high-level (mid-level) ancestor a_i of the query node n_i for taxonomy i, N_c is the number of children per node in i, and Δh_i is the difference in height between n_i and a_i, or $1/N_c$ if n_i equals a_i. It should be noted that the dilution factor used in this estimate makes no difference for single-taxonomy queries. Equation 3 is a heuristical equation which needs to be compared with other estimated weighting schemes.

4.2 Simulation Results

Using the performance measures developed in Equations 1-3, we ran experiments with 1000 simulated sources. The sources were generated with 3 taxonomies per source, 8 children per node, and a taxonomy depth of 4 (i.e. 4681 nodes per taxonomy). High-level (mid-level) metadata were considered to be the level-1 (level-2) nodes in the taxonomies. Collection sizes ranged from 10^2 to 10^8, with a higher probability[6] given to 10^5. Finally, the coverages of a node's children were chosen such that they summed exactly to the coverage of that node. However, we increased the variability of the children by using a non-uniform probability distribution. Our experiment generated 1000 queries. Each query was randomly chosen to use 1, 2, or 3 taxonomy query components. Furthermore, the actual taxonomy to use for the first, second, or third component was randomly selected as needed, although we disallowed using the same taxonomy more than once per query. Query nodes were chosen uniformly among all nodes in the tree, giving a higher probability that nodes were chosen from among the leaf nodes. By using Equation 3, we selected approximately 100 sources with the high-level metadata, and then from that set and the mid-level version of the same equation, selected a final set of 10 sources. For each query, we used Equation 2 to rank the 1000 sources. Finally, we derived the source precision for each query using Equation 1.

[6] The exponent was chosen as the random variate by averaging two values taken uniformly between 2.0 and 8.0.

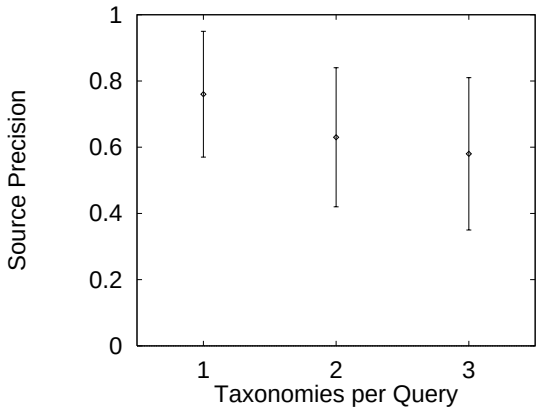

Figure 5: Simulation Precision

Query Depth	Number of Queries	Average Precision
1	0	N/A
2	3	1.00 ± 0.00
3	39	0.84 ± 0.17
4	278	0.75 ± 0.19

Table 2: Precision as a Function of Query Node Depth

The overall average precision was 0.66 ± 0.22. In Table 1 we show the breakdown of the results based on the number of taxonomies per query; the results are shown graphically in Figure 5. In Table 2 we show the breakdown of the results based on the node depth for the single taxonomy queries. We also calculated the average best and worst ranked source in the final selected sets. The former is 1.5; the latter is 324. These numbers indicate that in the vast majority of cases, the source which was assigned the highest or second highest true source weight was in the final set, and that on average the worst source in the final set was within the top 32%.

For proper perspective, precision measurements generally must be compared to their corresponding recall values, which we have not yet defined. While work from TREC [21] generally shows the state-of-the-art of document retrieval to be around 50% precision for 50% recall, these values are for document query results from single collections. We informally define *source recall* as the fraction of total documents in a sub-domain which are contained in the final set of the selected sources. Clearly this is a non-trivial value to determine. One study by the Research Libraries Group of top research libraries in the U.S.A. indicated that the best source can alone provide well over 70% source recall, and that a few sufficient sources can provide a source recall of over 90% [29].

It is difficult to know how these values might be extrapolated from tens of research libraries to tens of thousands of digital information sources; they do, however, help put the simulation results in some perspective. They indicate the potential importance of finding several of the best sources, as our simulations do for most of the final sets. Considering that each source in a final set, once selected, would be directly handed the query, we find these results reasonable. They indicate that a sufficient number of good sources could be located with the Pharos architecture so that the majority of relevant documents could be retrieved. We note that there is a drop of precision as the number of taxonomies per query is increased; while this result is not unintuitive, further work is needed to more definitively explain it. It is also worth noting that the results for 1000 queries are not significantly different than for only 100 queries, including the standard deviations. This indicates that the results are stable, though clearly more experimentation is needed to clarify parameter dependencies. Finally, [12] presents simulation results for the case of collections that are only partially classified.

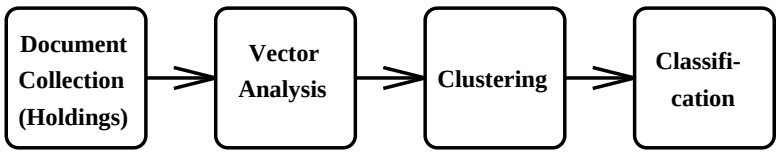

Figure 6: Metadata Extraction

5 Pharos Implementation

We now describe some of the components necessary for the implementation of Pharos. One important aspect of Pharos is the user interface, which is responsible for retrieving the metadata, presenting it to the user, building the Task and User Profiles [36], and aiding in the overall decision-making process [15]. It is this component which allows the user to use both the hierarchical and non-hierarchical metadata in a selection method more general than a strict linear ranking, as mentioned in Appendix A. Although space constraints do not allow us to elaborate these points here, further details may be found in [12]. Other important components of Pharos, briefly discussed below, include metadata extraction and the placement and function of the metadata servers.

5.1 Metadata Extraction

The process of extracting the metadata from a source begins with a document collection, as portrayed in Figure 6. This collection is processed by hand or automatically. For text-based documents, it can be processed by one of several possible automatic text analysis tools, such as LSI [3], SMART [32], or others that are used, for example, in TREC [21]. Many of these systems read through a collection and build a matrix of weighted frequencies of the number of times a given term occurs in each document in the collection. From this matrix, documents can be placed in a multi-dimensional term-vector space. Nodes in a taxonomy can also be placed in this space and then be taken as centroids of document clusters. Depending on the relative closeness of the centroids to the various documents, we can assign the fractional allocation of each document among the nodes in the taxonomy needed to calculate each node's coverage value. High-level automatic subject classification has been shown to be fairly successful, with over a 90% accuracy rate [27]. LSI has been shown to be an effective tool for automated classification [23], and experiments on automatically classifying large document sets within the LC Classification system have yielded correct classifications, under some conditions, of over 80% of newly entered documents [25]. These results indicate that subject-based automatic classification is currently or soon to be sufficiently accurate to characterize sources for comparison purposes, where even order of magnitude estimates can greatly aid in filtering out most irrelevant sources.

While the TREC and related work focuses on subject-based text retrieval, one could use a gazetteer or time-name table to identify geographic or temporal references within text-based documents, or use more sophisticated techniques such as those used in GIPSY [38]. For maps, aerial photographs, and satellite images, documents are often cataloged with spatial extents, and the geographic clustering and classification process is much simpler. Automatically extracting subject information from maps and images, such as identifying vegetation in a map or a particular object in an image, would allow such documents to be automatically classified within, for example, the LC Classification system; such capabilities are ongoing research issues [31].

5.2 Metadata Servers

Once the metadata at each site has been compiled, it needs to be distributed over the network according to the intended storage and retrieval architecture. As previously stated, the high-level metadata needs to be widely distributed and replicated, while the mid-level metadata is very selectively distributed. Because of these storage differences, different distribution protocols are more efficient for the different metadata levels.

All high-level metadata is sent to each high-level server. As briefly mentioned in Section 2.3, the distribution of high-level metadata is similar to the distribution of USENET news via NNTP [22, 24]. NNTP uses a flooding protocol which has proved to be very robust and efficient for widely distributed, massively replicated data. In fact, rather than create a completely new protocol and transport system, we plan to distribute the

high-level metadata in Pharos by using a new newsgroup hierarchy. This can be accomplished, for example, by having each source post their high-level metadata as a news article. Advantages to this approach, among several others, include easy prototyping, wide availability of existing news servers, and possible exploitation of both the hierarchical naming conventions of USENET and the selective distribution mechanisms of NNTP.

Mid-level metadata is not massively replicated, but rather stored on one or perhaps a few mirror sites. As such, a flood protocol is neither necessary nor efficient. Instead, a point-to-point metadata distribution system is more appropriate. Harvest [4] provides a suitable transport mechanism for distributing and storing mid-level metadata in Pharos. Benefits of using Harvest for distributing and storing mid-level metadata include efficient communication, metadata timestamping, and the existence of a replication system, which would allow us to automate the process of mirroring the mid-level servers.

6 Conclusion

Pharos is a distributed architecture for locating diverse sources of information on the Internet. Such architectures must scale well in terms of information gathering with the increasing diversity of data, the dispersal of information among a growing user base, and the visualization of results with the growing data volume [5]. Pharos is designed to scale well in all of these aspects, up to beyond 10^5 sources. The use of a hierarchical metadata structure greatly enhances scalability because it provides for a hierarchical network organization. Our particular hierarchical structure increases scalability additionally because it does not grow as a function of the number of documents in the collection (see [13]). The Pharos distribution scheme also enhances scalability: each user begins with the metadata available at a local high-level server. Moreover, since the mid-level metadata queries are distributed over a number of topic-based servers, there is little or no network bottleneck. Each source determines which taxonomies most appropriately fit its data domain. Diverse information is more easily classified as the number of taxonomies grow, including, for example, image and sound domains. Furthermore, each source determines when to update its own metadata. By collecting several taxonomy-independent factors such as network parameters and the size of a collection, Pharos allows users to broaden their search criteria beyond those allowed by term-matching techniques. We generalize the linear rankings of sources provided in standard vector-based text retrieval techniques, allowing users greater freedom in visualizing and selecting criteria of their preferred sources. Users can use relevance feedback techniques as a result of the iterative query methodology. Pharos is also extensible in that new taxonomies can be introduced as needed, either as part of an accepted Pharos standard such as the LC Classification, or as part of an independent group which defines its own taxonomy within which to classify documents.

An initial evaluation of Pharos, both in terms of simulation and in comparison to other models [13], has shown the architecture to be sufficiently successful to warrant further development. We have started working with data from real research libraries to compare with results from our simulated data. This work will need to be evaluated both by comparison with other systems and with human-based testing. Comparisons with subject taxonomies is more directly related to systems like *gGlOSS* and Content Routing; however, we currently know of no equivalent systems for comparison with the other taxonomies discussed. Another important evaluation method is how effective the system is in leading users to what they consider to be good sources, ones that have relatively more documents that they consider relevant than other sources. A human-based evaluation of our system will depend on empirical results based on asking users to complete specific tasks and measuring both their actual success rates and their own perceived success rates.

References

[1] ACM SIGIR. Workshop on Networked Information Retrieval. http://www.ciir.cs.umass.edu/nir96/, 1996.

[2] D. Andresen, L. Carver, R. Dolin, C. Fischer, J. Frew, M. Goodchild, O. Ibarra, R. B. Kemp, R. Kothuri, M. Larsgaard, B. S. Manjunath, D. Nebert, J. Simpson, , T. R. Smith, A. Wells, T. Yang, and Q. Zheng. The WWW Prototype of the Alexandria Digital Library. In *Proceedings of the International Symposium on Digital Libraries*, Tsukuba, Japan, 1995.

[3] M.W. Berry and S. Dumais. Using linear algebra for intelligent information retrieval. *SIAM Review*, December 1995.

[4] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber, et al. The harvest information discovery and access system. In *Proceedings of the Second International World-Wide Web Conference*, pages 763–771, Chicago, Illinois, October 1994. http://harvest.cs.colorado.edu/.

[5] C.M. Bowman et al. Scalable internet resource discovery: Research problems and approaches. *CACM*, 37(8):98–107, August 1994.

[6] CACM. Digital Libraries (section on Digital Library Initiatives. *Communications of the ACM*, 38(4):57–64, April 1995.

[7] J. Callan, Z. Lu, and B. Croft. Searching distributed collections with inference networks. In *Proc. of the 18th ACM-SIGIR Conference*, pages 21–28, Seattle, Washington, U.S.A., 1995.

[8] A. Chakravarthy and K. Haase. Netserf: Using semantic knowledge to find internet information archives. In *Proc. of the 18th ACM-SIGIR Conference*, pages 4–11, Seattle, Washington, U.S.A., 1995.

[9] H. Chen and K.J. Lynch. Automatic construction of networks of concepts characterizing document databases. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5):885–902, 1992. http://ai.bpa.arizona.edu/papers/ieee91/ieee91.html.

[10] L. Crum. University of michigan digital library project. *CACM*, 38(4), April 1995.

[11] Digital Equipment Corporation. AltaVista. http://www.altavista.digital.com/, 1997.

[12] R. Dolin, D. Agrawal, L. Dillon, and A. El Abbadi. Pharos: A scalable distributed architecture for locating heterogeneous information sources. Technical Report TRCS96-05, Computer Science Department, University of California, Santa Barbara, 1996. http://www.cs.ucsb.edu/TRs/TRCS96-05.html.

[13] R. Dolin, D. Agrawal, and A. El Abbadi. Classifying network architectures for locating information sources. In *Proceedings of the Fifth DASFAA Conference*, Melbourne, Australia, April 1997.

[14] S. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, 23(2):229–236, 1991.

[15] R. Ernst and M. Yovits. Information science as an aid to decision-making. In H. Brinkers, editor, *Decision-Making: Creativity, Judgment [sic], and Systems*. Ohio State University Press, 1972.

[16] D. Flater and Y. Yesha. ALIBI: a novel approach to resource discovery. *Internet Research: Electronic Networking Applications and Policy*, 5(4):17–30, 1995.

[17] E. Fox et al. Digital libraries. *CACM*, 38(4), April 1995.

[18] L. Gravano, K. Chang, H. García-Molina, and A. Paepcke. STARTS: Stanford Protocol Proposal for Internet Retrieval and Search. Stanford Digital Library Working Paper SIDL-WP-1996-0043, Computer Science Department, Stanford University, California, July 1996. http://www-diglib.stanford.edu/cgi-bin/WP/get/SIDL-WP-1996-0043.

[19] L. Gravano and H. García-Molina. Generalizing *GlOSS* to vector-space databases and broker hierarchies. In *Proceedings of the 21st VLDB Conference*, Zurich, Switzerland, 1995.

[20] J. Guyton and M. Schwartz. Locating nearby copies of replicated internet servers. In *ACM-SIGCOMM '95*, number 4 in Computer Communication Review, pages 288–298, October 1995.

[21] D. Harman. Overview of the Fourth Text REtrieval Conference (TREC-4). Gaithersburg, MD, 1996. http://potomac.ncsl.nist.gov/TREC/.

[22] M. Horton. Standard for interchange of USENET messages. RFC850, IETF Network Working Group, 1983.

[23] D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proc. of the 17th ACM-SIGIR Conference*, pages 282–291, 1994.

[24] B. Kantor and P. Lapsley. Network News Transfer Protocol: A proposed standard for the stream-based transmission of news. RFC997, IETF Network Working Group, 1986.

[25] R. Larson. Experiments in automatic library of congress classification. *JASIS*, 43(2):130–148, 1992.

[26] Library of Congress, Washington, D.C. *LC Classification Outline*, fifth edition, 1986.

[27] R. Losee and S. Haas. Sublanguage terms: Dictionaries, usage, and automatic classification. *JASIS*, 46(7):519–529, 1995.

[28] Lycos. Lycos. http://www.lycos.com/, 1996.

[29] P. Mosher. The nature and uses of the RLG verification studies. *C&RL News*, pages 336–338, July 1985.

[30] National Science Foundation, Washington, D.C. *Research on Digital Libraries, Program Guideline NSF 93-141*, September 1993. http://www.nsf.gov/ftp/CISE/program/nsf93141.txt.

[31] J. Richards. *Remote sensing digital image analysis: an introduction*. Springer-Verlag, Berlin, 2nd rev. and enlarged edition, 1993.

[32] G. Salton. *Automatic Text Processing*. Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[33] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, New York, 1983.

[34] B. Schatz. Building the interspace: The illinois digital library project. *CACM*, 38(4), April 1995.

[35] M. Sheldon, A. Duda, R. Weiss, J. O'Toole, and D. Gifford. Content routing for distributed information servers. In *Proceedings of the 4th International Conference on Extending Database Technology*, 1994.

[36] B. Shneiderman. *Designing the User Interface, Strategies for Effective Human-Computer Interaction*. Addison-Wesley, second edition, 1992.

[37] C. Viles and J. French. Dissemination of collection wide information in a distributed information retrieval system. In *Proc. of the 18th ACM-SIGIR Conference*, pages 12–20, Seattle, Washington, U.S.A., 1995.

[38] A. Woodruff and C. Plaunt. Gipsy: Automated geographic indexing of text documents. *JASIS*, 45(9):645–655, 1994.

[39] Yahoo! Inc. Yahoo. http://www.yahoo.com/, 1997.

A Metadata Description

Table 3 shows the portion of the high-level metadata that is independent of the taxonomies. In the `Description` column, we illustrate the corresponding attribute with an example. Other than the source identification information, these attributes can be viewed as quality and cost factors that are presented to the user through the UI (see [12]). The user can then consider a range of factors in determining specific sources relevant to current queries. The first attribute, `HL_Desc_Ver`, is the version of the metadata schema and is used to allow for backward compatibility. The `Src_*` attributes describe basic source identification information. `Siz_Col` describes the size of the collection in MB, pages, and items. These units are used to calculate the fractional assignment of documents in three units. The reason that we use multiple units is that they may yield very different ratios. For example, an image library whose images may range from over 100MB/image to less than 10KB/image will show very different geographic coverages in MB than in items. `Siz_Cat` describes the size of the catalog in MB and items. `Count_Tax` is set to the number of taxonomies used for classification by the source. `Spec` is the list of character strings used to describe special

Attribute Name	Type	Range	Description: *Example*
HL_Desc_Ver	String	-	Metadata descriptor version number: *0.0.7*
Src_ID	String	-	Name of source: *Alexandria Digital Library*
Src_Type	String	-	Type of source: *DLIB*
Src_Loc	String	-	Geographical location of source (if relevant): *Santa Barbara, California*
Src_Sch	String	-	Availability schedule of source (if relevant): *Closed Mondays*
Siz_Col	Real[3]	0.0+	Collection size in MB, pages, and items: *1.843E6,,8.91E5*
Siz_Cat	Real[2]	0.0+	Catalog size in MB and items: *2.172E6,3.892E6*
Count_Tax	Integer	0+	Number of taxonomies used: *3*
Spec	String[M]	-	List of M Special Collections: *Geography:BE16*
Net_Band	Real	0+	Network bandwidth (Kbps): *1.5E3*
Net_Avg_Util	Real	0.0 - 1.0	Avg. network utilization: *0.47*
Net_Avg_Delays	Integer[N]	0+	Avg. network delay (ms) to N beacons[20]: *,200,,521,,,147,*
Net_Avg_Thru	Integer[N]	0+	Avg. network throughput (Kbps) to N beacons[20]: *,117,,,,,246,*
Count_Avg_Hits	Real	0+	Avg. number of accesses (hits/day): *1.237E5*
Pol_Charg	String	-	Charging Policy: *See http://www.alex.ucsb.edu/doc/charging.html*
Pol_Lend	String	-	Lending Policy: *Must be UC affiliate*
Count_Src	Integer	1+	Number of sources: *1*

Table 3: Taxonomy-Independent High-Level Metadata

collections, described earlier. The set of parameters `Net_*` are included for connectivity estimation between the users and the sources [20]. `Count_Avg_Hits` describes the average number of accesses the source receives per day. Additional attributes are available for various library policies relevant to charging, lending, etc. Finally, `Count_Src` denotes the number of sources included in a particular metadata record to allow condensed multi-source records. If this value is greater than one, some of the other values might not be included, such as the network parameters.

Tables 4 and 5 show the taxonomy-dependent part of the high-level metadata. Within the full metadata record for a source, the attribute-values in Table 4 are repeated once for each taxonomy. `Tax_ID` and `Tax_Desc_Ver` identify the taxonomy name and version, respectively. The next attribute, `Cov_Root`, gives the fractional amount of the collection which has been classified within the taxonomy, with respect to MB, pages, and items. The next four attributes give statistical information about the root's children (the depth-1 nodes in the taxonomy). Furthermore, within each taxonomy, the attribute-values of Table 5 are repeated once for each high-level node in the tree.

The geography taxonomy is treated as a special case in order to include factors such as image or map

Attribute Name	Type	Range	Description
Tax_ID	String	-	Name of taxonomy
Tax_Desc_Ver	String	-	Taxonomy descriptor version number
Cov_Root	Real[3]	0.0 - 1.0	Root node's coverage value (MB, pages, items)
Cov_Avg	Real[3]	0.0 - 1.0	Avg. coverage of depth-1 nodes
Cov_SD	Real[3]	0.0 - 1.0	Std. Dev. of coverage of depth-1 nodes
Cov_Min	Real[3]	0.0 - 1.0	Minimum coverage of depth-1 nodes
Cov_Max	Real[3]	0.0 - 1.0	Maximum coverage of depth-1 nodes

Table 4: High-Level Metadata for Each Taxonomy T_i

Attribute Name	Type	Range	Description
Node_ID	String	-	Node's label
Cov_Node	Real[3]	0.0 - 1.0	Node's coverage value (MB, pages, items)
Cov_Avg	Real[3]	0.0 - 1.0	Avg. coverage of children
Cov_SD	Real[3]	0.0 - 1.0	Std. Dev. of coverage of children
Cov_Min	Real[3]	0.0 - 1.0	Minimum coverage of children
Cov_Max	Real[3]	0.0 - 1.0	Maximum coverage of children

Table 5: Metadata for Each High-Level Node in T_i

resolution, which are not relevant to most other taxonomies. This taxonomy is a superset of that in Tables 4 and 5. Additional attributes are included to describe image resolution statistics which summarize all the documents which are associated with the geographical region covered by each node.