A Space-Efficient Model for Sharing Personal Knowledge Objects in Peer Communities

Fred S. Annexstein, Kenneth A. Berman,
Kevin Henkener, and Svetlana Strunjaš

Department of ECECS
University of Cincinnati
Cincinnati, OH 45221 USA
fred.annexstein@uc.edu, ken.berman@uc.edu,
henkenkr@ececs.uc.edu, strunjs@email.uc.edu

Abstract. In the future it is likely that peer communities will be routinely established for the purpose of sharing electronic resources and targeted information among groups of peers with common interests. This sharing will be possible by building on existing Internet technologies, such as web applications and services, and peer-to-peer networks. Today many peer-to-peer networks do not work well for sharing large heterogeneous collections among common home users. In this paper we present a model for organizing personal knowledge and a two-fold framework for improving methods of information sharing. Our framework is based on a) the creation of compact, portable, organizational objects through the ad-hoc structuring of knowledge on a personal computer, and b) the aggregation of these organizational objects into a visual catalog that can provide a peer community with a conceptual view of the information resources residing in the community. We call our model the BookQuest framework, and we show how it addresses several key information sharing issues, including space-efficient summarization, naming and locating resources in a peer community, and user interface and visualization issues. Our model introduces a variety of techniques for space-efficient representations, including use of generalized Bloom filters to support relevancy ranked keyword searches, and statistical sampling methods to support similarity comparisons between collections. We describe applications of our model to peer-based, inquiry orientated educational activities.

1. Introduction

In the future it is likely that peer communities will be routinely established for the purpose of sharing electronic resources and targeted information among groups of peers with a common interest (e.g., educators, scientists, art buffs, etc.). This sharing will be possible by building on existing Internet technologies available to common users such as peer-to-peer (p2p) networks, web applications, and web services. In this paper we use the term *personal knowledge* to refer to localized collections of information resources, and includes text and non-text content together with associated metadata (notes, keywords, tags, annotations, etc.). For most users, the bulk of this personal knowledge will reside on their local computers in files of various different types and on the web, either on personal websites or on websites of others. This personal knowledge is likely to be of a higher quality and more relevant to the interest area of a peer community than would otherwise be obtained by general approaches to searching the web.

In order to effectively share and disseminate this personal knowledge among the members of a peer community, information needs to be organized locally and compiled into a compact form. Further, the collective shared personal knowledge of the peers in the network needs to be indexed and cataloged for effective navigation and retrieval. The current state of Internet standards and p2p network technologies is not yet robust enough for effectively sharing narrowly-focused personal knowledge. No standard information organization method is widely used, or is ever likely to exist. What is required is a flexible model for organizing personal knowledge – a model that enables the organization of large, heterogeneous collections of local files, network file shares, websites, and metadata. What is required is a model that enables the creation of compact and concise information organizational objects that can be widely shared across the Internet. Since the content resources referenced in personal knowledge organizations would likely overwhelm traditional servers it is important to establish models for information and organization that can flexibly use services such as naming and location of resources in peer-to-peer networks.

In this paper we describe a model of information that enables personal knowledge sharing, which we call the BookQuest framework. The framework addresses issues of compacting, encapsulating and transporting information organizational objects. We describe the development of a software tool that users can employ that supports the processes of creating these organizational objects storing summaries of resources, along with appropriate naming and location information, separate and independent from the actual resources (i.e., file contents). These encapsulated objects of personal knowledge organization form the core of our framework. We describe the design and implementation of these objects, and an associated software tool for creating and manipulating these objects.

Our central organizing principle is the *space-efficient representation of personal knowledge on a topic.* Topic organization of web-accessible information is a well studied area (cf. Topic Maps [tm]), and we believe this a good design choice since personal knowledge often coincides with topic classification of resources. Through a process of topic-organized structuring (or, as we term, personal knowledge compiling) we begin to address the issue of how personal knowledge can be effectively transported, and thereby retrieved and navigated by others in a peer community. Our algorithms and software implementations address a number of issues associated with the space-efficient means of search and retrieval, naming and locating of resources in heterogeneous environments. We employ a generalized version of a Bloom filter [B70] that is able to rank the relevance of keyword and descriptive terms in relation to particular organized collections. We employ statistical sampling methods based on min-wise hash vectors [BC+00] to facilitate the measure of similarity between organized collections. In support of information retrieval, we use these techniques in combination with techniques for peer location and resource discovery through translations of relative and absolute path addressing.

1.1 Related Work

Recent research has focused on developing varieties of peer-to-peer technologies and peer-to-peer applications, including content distribution and fault-tolerant distributed storage [CKK00, KB+00], distributed file and lookup services [IK+01, RD01, MP02], and anonymous information sharing [CHSW02, HW02]. Many existing systems are focused on broad user bases and often rely on general text-based statistical methods to assist search and retrieval. There are several drawbacks to existing systems which, in their current state, poorly serve peer communities. Peer communities are often focused on well defined topical interests, and such communities share a collective need to exchange electronic resources such as files in diverse heterogeneous types. Widely deployed systems (such as Gnutella [gn]) typically require peers to place files in shared directories, and these applications rely on the use of file names to index and retrieve information. File names alone are not sufficient for users to easily discriminate between query results. Furthermore, discrimination via the repeated downloading and viewing of content files can be prohibitively time-consuming and can overload a p2p network application. Text-based vector space models and massive indexing are not generally viable to address the problems of peer community information sharing. Indexing is quite limited or impossible on non-textual information sources, (e.g., jpegs, mp3s, and mpegs), and the scale of the size of the information and the dynamics of the information base in peer-to-peer network communities is much greater than on the web. Crawling and other automated methods for resource compilation can do very little to provide conceptual organization and classification, which is critical for providing meaningful views of specialized information.

More recently there has been an effort to provide more formal semantics in the context of peer-to-peer environments, with a goal of enhancing methods of search and retrieval for information and other shared resources across a network. Studies concerning the addition of encoded semantic knowledge or metadata in a peer-to-peer environment include [NW+02, Ne02]. Many of these approaches build on RDF standards for metadata [rdf]; however, much work remains on several fundamental challenges relating to extracting and visualizing this metadata. The semantic-based approaches to resource sharing are thus difficult to realize in practice, particularly in the absence of shared ontologies (i.e., encoded knowledge).

The model described in this paper is complementary to developments in semantics-based peer-to-peer. The organizational objects that we propose in the BookQuest framework provides for the capture or extraction of ad-hoc semantics, that is, additional information concerning the meaning and content of resources from a user's personal perspective. These ad-hoc semantics provided by users are likely to be highly useful for automatic metadata extraction methods that work with formalized standard frameworks for semantic metadata, such as those based on RDF schema. In addition, the organizational objects we describe are designed to be sharable in their own right, providing a compact container for information organization, and providing a visualization of file organization and classification. These objects can be shared as any other file resource can be shared, and thereby support the design and sharing of organizational templates. Such templates can be used, for example, by a peer community classroom, to assist inquiry-oriented processes for obtaining better information on a focused topic. Templates can help learners find answers to specific questions in a manner that is superior to simply searching the Internet using a generic search engine. We describe examples of such classroom applications in Section 3.2.

2. Compact Compiling of Personal Knowledge

Our model begins with the premise that users have a unique ability to compile information on topics of their own particular interest. Generally speaking, information resources are readily available to individuals, through the Internet, on LANs, and importantly as files on a user's personal computer. We focus on a compilation process that result in gathering links and other references together and encapsulating them into an organizational object that we call a *mybook-object*. Ideally, this compilation is done with minimal user effort, and created without affecting the contents or the location of information resources. We envision such compilations as consisting of highly disparate and heterogeneous sets of items including word processing documents, read-only documents (such as pdf and ps files), clippings, images and scannings, spreadsheets, computer applications, links to websites, etc. We refer to these collections as topical personal knowledge. Here are a few illustrative examples of such collections,

a) Educational - files such as lesson plans, lecture notes, curricula, and websites for teaching and educational resources such as learning objects.
b) Professional and Work-related – files such as resumes, schedules and calendar applications, spreadsheets, portfolios, and websites for job searching and career building.
c) Family Health – files such as immunization records, medication records, diet and nutrition information, websites for health information.
d) Personal Contacts - files such as address books, phone, fax, email and other contact information, calendar applications, email letters, pictures, websites such as personal websites, dating websites, etc.

The personal knowledge compilation process results in the creation of an organizational object--the mybook-object--that provides a conceptual view of the topical personal knowledge that is independent and physically separate from the contents. The objective of the creation of mybook-objects is to capture or extract topical personal knowledge and create a data structure in a standard format so that is can be used to share views of the associated information collection. Our design of the format incorporates means to access to the content via the Internet, using standard protocols, and provisions for the integration with existing p2p file systems.

In precise terms, the mybook-object is an annotated tree where each node of the tree is associated with four components a) topic name, b) links to files, e.g. local files, files on network shares, or files that are part of a peer-to-peer network, c) links to websites, including a (default) displayed website and d) variety of metadata used to support the efficient search an retrieval of content resources. We have formalized the design of this mybook-object into a recursive XML-based schema definition, see Fig. 2 below. This schema defines the structure of an XML-file we call a mybook file, and we use an .mbk file extension when referring to such files.

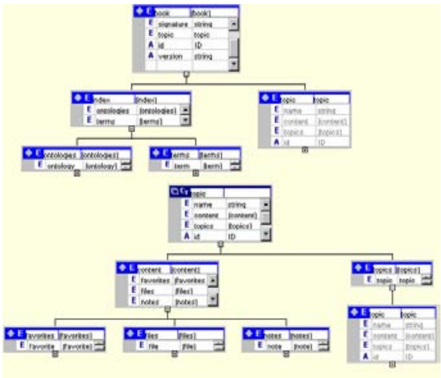


Figure 3. The figure on the left shows the general structure of a mybook object given by an XML Schema definition. This schema defines a file format for a mybook file (.mbk extension). The file stores an annotated tree containing a summary of references for a personal knowledge collection, and contains compact metadata representations used for search and retrieval applications. Refer to the website http://mybook.uc.edu/schema for the current complete schema.

The XML-schema is designed in practice to support space-efficient summaries enabling search, retrieval, and transport functions. The logical design of the schema, however, is based on intuitive concepts reflecting components of traditional book formats. These logical structures simplify many aspects of the system development process though the use of intuitive concepts related to traditional books. The logical components of the schema include the *table of contents*, the *topic pages*, and the *index table*. These logical components aid the software design process and the users of the application. For example, we use the table of contents component to isolate information for visualization purposes, providing to the user, at a glace, an understanding of the structure and topic set of the mybook file. Also, we use common software library components (such as *TreeView* controls) to realize this visualization. Operations on mybook-objects that affect the table of contents include a) add/remove and edit topic names, b) move/reorder topics, and c) open/close topics.

Within the user interface, the operation of opening a topic results in displaying a "page" that renders a visualization of the specific resources/contents associated with that topic. The goal is an organized visual presentation to the user of the set of links to the information content (e.g., file links, network links, web links, p2p links, etc.). Operations on mybook-objects that affect the page contents include a) add/remove favorite websites, b) set/unset default displayed html pages, c) add/remove file and web links, d) renaming links, and e) add/edit text notes.

2.1 Indexing for Relevancy with Generalized Bloom Filters

Our model provides for the creation of a compact indexing data structure. Our approach is based in part on using a generalized version of Bloom filters [B70] which have been shown to provide a space-efficient representation capable of accurately handling membership queries (with small probability of false positive results). Previously, Bloom filters have been used in a number of distributed systems applications, including indexing data in internet caches [FC+00], distributed storage of routing table information [RK02], and indexing keyterms for locating peers in p2p networks [MP01]. Generalizations of standard Bloom filters, include Spectral Bloom Filters [CM03] which extend the data structure to support frequency estimates for multisets, and Bloomier Filters [CK+04] for encoding arbitrary functions with static (small) support sets. In our model we introduce a data structure called a Relevancy Bloom Filter (RelBF) that is optimized to compactly encode relevancy judgments to facilitate ranked responses to keyterm membership queries. In a standard Bloom filter implementation, a collection of hash functions is used to set selected bits in the Bloom filter array. Membership is checked by verifying that each hashed bit from a given key is set. Our RelBF generalizes the standard Bloom filter to store relevance values $0 \leq r \leq 1$ associated with each keyterm. When a keyterm with relevance value *r'* is inserted into the filter, the maximum value is stored at each hashed index position. That is, if *r'* is greater than the value at the hashed position, then the value is updated. If *r'* is less than or equal to the value at the hashed position, then it is left unchanged. A similar *minimal increase* method appears in [CM03]. When testing set membership the minimum relevancy value is used from the set of index positions given by the hash functions. It can be shown that using this scheme, the relevancy value returned is always greater than or equal to the relevancy of the actual term. In addition, the probability that the value returned is strictly greater than the actual value is very small. Using a straightforward calculation, the error probability can be bounded above by value of the expression

$$(1 - (1\text{-}1/m)^{k\, n(r)})^{k}$$

where m is the size of the Bloom filter, k is the number of hash functions, and $n(r)$ is the number of keyterms with a relevancy value greater than r. This expression is asymptotically equivalent to $(1\text{-}\ p)^{k}$, where $p = e^{-k\, n(r)\, /m}$. In our model the relevancy values can be determined by a combination of human judgments and automatic means. The automatic assignment of relevancy values can be supported by the hierarchical structure of the mybook object, i.e., relevancy of key terms can be determined as a function inversely proportional to its depth in the tree hierarchy. The size of the RelBF only minimally affected by our generalized scheme; i.e., the total size of the filter is larger than standard Bloom filter by a (multiplicative) factor that is proportional to the logarithm of the number of different relevancy values used. The problem of supporting dynamic updates is generally more complex (see, [CM03] and [CK+04] for different models), and we are investigating how to best support such operations within a RelBF.

2.2 Use of Semantics and Taxonomies

Relevancy and meaning of keyterms can also be deduced through the use of taxonomies. The index of the mybook object provides the means to incorporate semantic information that is useful in applications such as disambiguating search terms. The mybook schema allows the incorporation of standard and ad-hoc taxonomies (created by the user) for search and classification. We can apply the Bloom filter technique (described in Section 2.1) optimized to sets of terms that are standardized in formal taxonomies. A simple user interface assists users to associate terms to a current topic in the table of contents for a given mybook object. The terms are either categorized in a standard taxonomy or ontology, or free, unclassified terms that are used in an ad-hoc fashion. The users can choose among different named association types. Thus far we have experimented with the inclusion of the WordNet [Fe98] conceptual dictionary to evaluate the usability of standard taxonomies. These taxonomies are also useful with regards to the visual catalog as will be described in Section 3.1. We have also experimented with automatic means of generating term indexes based on the different page-topic context. The operations on mybook-objects that affect index data structure include a) adding topic name keyterms, b) adding descriptive terms from ad-hoc taxonomies c) adding keyterms from standardized taxonomies (dictionaries), d) adding text terms from content files, websites, and free text. Visualization of associated terms for the mybook object is realized in the similar way as the table of content visualization (using TreeView control, mentioned in Section 2).

2.3 File Naming and Handling

When file links are added to the object the software modifies the link data stored using absolute path addresses. When an object is prepared for being shared (or, say published for community use), the resource files are accessed through their absolute path addresses, and the contents are then copied to a default directory or a shared location. During this operation, a new object is created in which the file links are (re)written using relative path addresses to the file contents. The use of relative addressing allows the object, along with the associated file contents, to move freely to other machines and platforms in the peer community, independent of the original base address or file system naming format. In peer community environment in which a p2p file system is supported, such relative addressing methods may be substituted with URL-like naming mechanisms.

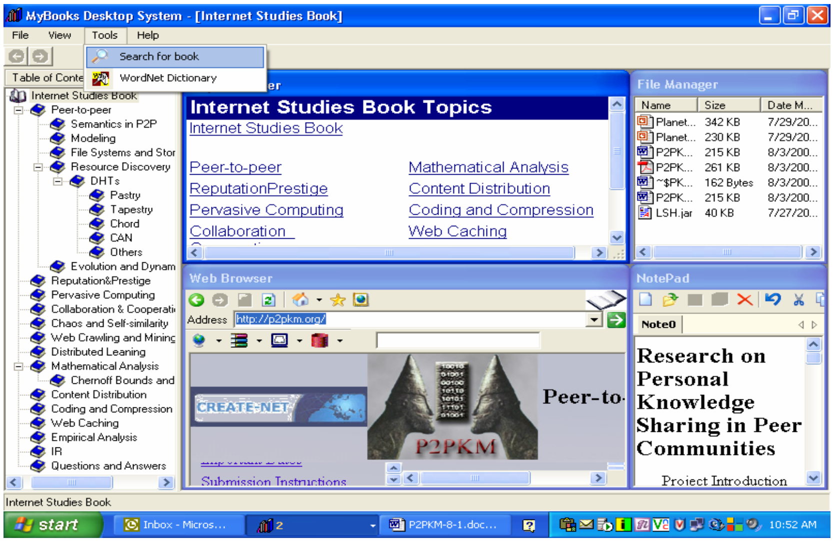
2.4 Preliminary Empirical Results

Our experience with the model has shown dramatic reductions in space requirements while preserving and improving much of the semantic, navigational, and location information. We have seen that the sizes of mybook-objects are significantly smaller than the combined contents to which they refer, by several orders of magnitude. In a recent study, graduate students at the University of Cincinnati created mybook-object files during a research project on topics on their choosing. The assignment was a short term project of just two weeks work. In this study 12 mybook-object files were created with an average size of 50KB each (uncompressed). The size of the topic-annotated trees created averaged about 35 nodes with 2 hyperlinks on average per node. The contents of these books were primarily website collections (favorites) and text research papers created by the students, and so these mybook collections were not created as information repositories. However, the average size of the content collection was about 300KB yielding about a 6-fold increase over the mybook-object size. In another study, one of the authors created a reference mybook-object for a seminar course for his students. In this example, the mybook-object was of size 98KB (uncompressed), and the referenced contents (made exclusively with research resources in pdf, ps, and ppt formats) were of size 14,000KB, a 140-fold increase in size. Other example books that contain media contents such as mp3 and mpg files were found to show several thousand-fold increases in size. In each of these preliminary examples the index stored only small sets of keyterms and therefore did not have significant size impact. The size comparisons did not include measures of the size of the html files from websites referenced.

3. The User Interface - Mybook Desktop Application

We have developed a software tool called the MyBook Desktop (publicly available at http://mybook.uc.edu) which supports users in the creation of organizational mybook-objects. The desktop system renders the visualization of the mybook-objects and provides a user interface for implementing all of the functions for creating and publishing mybook-objects, as discussed in Section 2. The application (currently developed on the Microsoft .NET platform) incorporates methods for handling ‘double-click’ and ‘drag and drop’ events, which demonstrably enhances the processes associated with personal knowledge compilation. The Windows file system and browser functions provide access to the local file systems, access to shared network drives, as well as web accessible resource databases. We have included a notepad editor that facilitates the inclusion of personal annotations in each view.

In our current implementation, the interface provides visualization of mybook topic “pages” through a five panel display; see Figure 2 below. Starting from the upper right of the Figure and moving clockwise, the panels are File Browser Panel, Notepad Editor Panel, Web Browser Panel, Topic Tree View, and the Topics Browser Panel. The image below, Figure 2, is a screenshot of a mybook-object created as part of a test case for information sharing with students – a classroom peer community. The Figure illustrates how a mybook can be designed to facilitate access to reference materials. Students use this reference mybook object as a template, and are encouraged to modify their copies of this book-object to organize and present their own portfolio of research results. We describe more details about this classroom application of the system in Section 3.2.



Name	Size	Date M...
Planet...	342 KB	7/29/20...
Planet...	230 KB	7/29/20...
P2PK...	215 KB	8/3/200...
P2PK...	261 KB	8/3/200...
~$PK...	162 Bytes	8/3/200...
P2PK...	215 KB	8/3/200...
LSH.jar	40 KB	7/27/20...

Research on Personal Knowledge Sharing in Peer Communities

Project Introduction

Figure 2. This figure shows a screenshot of the MyBooks Desktop Software with an open mybook-object containing research papers and resources organized for a Computer Science seminar course on Internet Studies.

3.1 A Visual Catalog for Mybook Objects

The second major goal of the BookQuest framework is facilitating the sharing of personal knowledge. Mybook objects support the creation of a *visual catalog* for a global index created from a repository of mybook files. The visual catalog maintains a catalog index created from joining the individual index data structures associated with each mybook file in the repository. The goal of the visual catalog is to provide an additional layer that can use keyterm relevancy searches to identify a potentially large numbers of relevant mybook files, each representing a summary of resources available from an individual peer. The index data structures based on generalized Bloom filters, by themselves, can not efficiently support similarity measures between mybook objects. Hence, we introduce metadata based on statistical sampling methods that is useful for determining similarity between collections in a space-efficient manner. We employ the idea of using min-wise hashing [BC+00]. This technique represents sets using short vectors of representatives determined by hashing or permutations of the base set. Pairs of sets can be compared for similarity based on the contents of the vectors. We choose our representative sets by relevant topic names and other keyterms. By including these short vectors in our summaries, the similarity between pairs of mybook objects can be computed. In particular, the *Jacaard measure* of set similarity, given by Sim(A,B) = |A B| / |A ∪ B|, can be effectively estimated using min-wise hashing [BC+00].

The visual catalog is designed to use set similarity measure so a collection of mybook-objects, or intuitively, a collection of “subbooks,” can be returned for each user keyword query. This collection of subbooks is returned to the user as a compiled book object using the same data format as a single mybook file, and thereby navigated similarly. The MyBooks Desktop software, as described in the following section, is used to render the compiled results for such queries. Research is currently analyzing automatic methods for selecting and composing such sub-books from large databases of mybook files.

3.2 Sample Applications

In this section we describe two proposed Internet-based applications for which the framework of the BookQuest project can be usefully applied. The two applications we describe are Peer Community Digital Libraries and Peer-based Inquiry-oriented Educational Activities.

The BookQuest framework is suitable for providing peer-based, inquiry-oriented educational activities. Within the educational community, WebQuests [wq] have proven to be popular format for structuring inquiry-oriented activities in which some or all of the information that learners interact with comes from resources on the internet. The instructional goals of WebQuests include knowledge acquisition and integration, as well as extending and refining knowledge. Our BookQuest framework can be used to extend these general educational concepts and methods by providing an enhanced and more flexible environment to educators and learners. Traditionally, Webquests are designed to provide to learners the guidance on how to organize information acquired during certain research tasks. This often takes the form of guiding questions, or directions to complete organizational frameworks. Using the BookQuest framework an educator can create a peer-based classroom providing a flexible environment in which guides to Internet resources are well organized. We call such guides a mybook classroom template. As in WebQuests, the organization of the guides helps learners find much better information on a focused topic; in addition, the guides can be used by learners to help them find answers to specific questions. However, since the BookQuest framework is focused on a more general peer-based information environment it can allow educators greater flexibility in compiling and organizing resources. Since the associated desktop software brings a hybrid environment that provides a web browser, text editor, and link organization tools, BookQuest resources guides can be generated by educators much more quickly than websites, and can be used by learners to readily store and organize their personal research results.

In another related application, we describe how the BookQuest framework is suitable for providing certain digital library capabilities to dynamic peer communities. Traditionally, digital libraries [L97] address the information search problem through the organization of information collections. However, many collections are not easily organized; for example, consider the case of an individual professional, such as an educator, with a personal computer hard drive filled with disparate electronic resources. Recently, there has been development of the concept “personal digital libraries” [Bo03, Bu01] which is aimed at providing flexible digital library capabilities to individual users. We have addressed in this work the related problem of bringing digital library-like capabilities to peer communities. Unlike traditionally digital libraries, our model is applicable to dynamic and potentially transient, organizations of individuals. There are several digital library related issues in which our model addresses for peer communities, namely, a) providing means for individuals to effectively create organized, structured views of information collections, b) providing means to share these organized views, and c) providing means to access information collections offered by these views.

4. Conclusions

Today, the choice options for electronic communication among members of a peer community are many and varied, from email distribution, to website construction, to instant messaging, to Usenet applications, to bulletin boards, to weblogs, etc. These widely available application options are not completely satisfactory when community members want to flexibly share information. For example, websites are limiting due to their expense both in hosting and in required management tasks and email tends to be intrusive and inflexible. The framework of the BookQuest project [bq], as described in this paper, is a proposal for a better way for peer community members to share information, and one that addresses several human and technological challenges not addressed by existing systems. Our framework includes a compact, space-efficient data format standard and software that can be used to share information that resides in diverse formats and diverse organizations. We have provisioned for relevancy search and retrieval within the context of a peer-to-peer network environment. We believe that such a framework is useful, particularly for sharing highly focused or highly technical information, and share it in a manner in which highly accurate search and retrieval methods can be employed. In future work we intend to evaluate the effectiveness, space tradeoffs, and scalability of using the proposed space-efficient data structures, such as the RelBF generalized bloom filter, for applying personal relevance information in p2p information retrieval applications.

5. References

[B70] B.H. Bloom, “Space/time trade-offs in hash coding with allowable errors”, Communications of the ACM, p.422-426, July 1970.

[Bo03] C.L. Borgman, “Personal digital libraries: Creating individual spaces for innovation”, NSF Workshop on Post-Digital Libraries Initiative Directions, June 2003.

[BC+00] A.Z. Broder, M. Charikar, A.M. Frieze, and M. Mitzenmacher,"Min-Wise Independent Permutations", Journal of Computer and System Sciences", 60(3), pp. 630-659.

[Bu01] R.D. Burke, “Salticus: guided crawling for personal digital libraries", ACM/IEEE Joint Conference on Digital Libraries", 88-89, 2001.

[CM03] S. Cohen and Y. Matias, “Spectral Bloom filters”, ACM SIGMOD, 2003.

[CKK00]] Y. Chen, R.H. Katz, and J.D. Kubiatowicz. “Scan: A dynamic, scalable and efficient content distribution network”, in Proceedings of Inter. Conf. on Pervasive Computing, 2000.

[CHSW02] I.Clake, TW. Hong, O.Sanberg, and B.Wiley. “Protecting free expression online with freenet”, IEEE Internet Computing, 6(1):40-49, January-February 2002.

[FC+00] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. “Summary cache: a scalable wide-area Web cache sharing protocol”, IEEE/ACM Transactions on Networking”, 8(3):281-293, 2000.

[Fe98] C. Fellbaum, ed.,"WordNet an Electronic Lexical Database," The MIT Press, 1998.

[HW02] S. Hazel and B.Wiley. “Achord: A variant of the chord lookup service for use in censorship resistant peer-to-peer publishing systems”, in Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), March 2002.

[KB+00] J. Kubiatowicz, D. Bindel, Y Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells and B. Zhao, “OceanStore: An Architecture for Global-scale Persistent Storage”, Proceedings of ACM ASPLOS, 2000.

[L97] M. Lesk, Practical Digital Libraries, Morgan Kaufmann, 1997.

[MP+01] F. Matias Cuenca-Acuna, C. Peery, R. P. Martin, T. D. Nguyen, “PlanetP: Infrastructure Support for P2P Information Sharing”, Technical Report DCS-TR-465, Department of Computer Science, Rutgers University, 2001.

[M01] M. Mitzenmacher, “Compressed bloom filters”, Proceedings of the twentieth annual ACM symposium on Principles of distributed computing, p.144-150,

[Ne02] W. Nejdl, "Semantic Web And Peer-To-Peer Technologies For Distributed Learning Repositories", 17th IFIP World Computer Congress, Intelligent Info. Processing / IIP-2002.

[NW+02] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sinte, A. Naeve, M. Nilsson, M. Palmer, T. Risch, “EDUTELLA: A P2P Networking Infrastructure Based on RDF”, WWW 2002.

[RD01] A. Rowstron and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems”, Middleware, November 2001.

[RK02] S. Rhea and J. Kubiatowicz, “Probabilistic location and routing”, IEEE INFOCOMM, 2002.

[SM+01] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, ACM SIGCOMM 2001

[bq] The BookQuest project homepage, http://mybook.uc.edu
[gn] The Gnutella homepage: www.gnutella.wego.com
[tm] The Topic Maps XTM homepage, www.topicmaps.org
[rdf] The W3C RDF homepage, http://www.w3.org/RDF/
[wq] The WebQuest homepage, www.webquest.org