Intranet Document Management Systems as Knowledge Ecologies

Mark Ginsburg
Fisher Center, Haas School of Business
University of California, Berkeley
ginsburg@haas.berkeley.edu

Abstract

Document management inside an organization is an interesting socio-technical problem. This paper considers Intranet Document Management as a knowledge ecology and presents a document lifecycle model consisting of five phases: creation, publication, organization, access, and destruction (adapted from [13]).

Specific attention is paid to the knowledge ecology issues faced by document management systems (DMS), such as identifying typical heterogeneous work groups in an enterprise document management system, the implications of inter-user and inter-group coordination (via, for example, annotation) and the consequences of DMS use over time.

We conclude with a discussion of how the basic stages of a document lifecycle can tie together with the socio-technical goal of a scalable DMS to support a lively and dynamic document knowledge ecology.

1 Introduction

Organizations spend significant time and effort to create and organize documents on internal networks. The rapid rise of popularity of the World-Wide Web and the lowering of cost of individual computing and the internetworking of computers means that firms have little technological or financial barrier to creating documents locally, thus making them available to the organizational community. Document management, though, is more than publication and organization — it also includes efficient access and retrieval. Current schemes are hampered by the fact that access is usually an ad-hoc process which lacks coordination between readers as they search, and between readers and the original document authors. Ad-hoc access and retrieval leads, in ecological terms, to an unhealthy situation with artificial barriers between segmented information sources (islands), mismanagement of the authors' expertise, and reader frustration and dissatisfaction with the access experience — in sum, an inferior level of aggregate knowledge transfer from the readership session.

Since documents constitute such an important part of a firm's knowledge assets, a well-designed document management system (DMS) should provide collaboration and coordination mechanisms to provide the readers with an active role in growing the knowledge base and linkage mechanisms between readers and authors, as well as providing a basis for evaluating the system.

To approach the problem of effective DMS design, we must set the stage with essential terminology. A document might have very little context and merely present transaction-oriented *data*. Or, a document might be data given context by the author — *information* taking shape as a message with sender (author(s)) and recipients [8]. Thus, a document might be classified as a systematic knowledge asset [23] (i.e. well-structured, following a template) or it might be quite irregular and unsystematic. In any event, an organization's documents will vary in length, presentational format, semantic content, and quality.

A DMS design must be flexible and extensible with its goal being a healthy ecology which breeds optimal knowledge transfer in the readership session.

The paper is divided into the following sections. Section 2 presents the Intranet Document Lifecycle waterfall model, tracing a document from its creation through the publication process.

Section 3 then goes on to divide the components of a DMS into foreground elements (actors and their range of actions in conjunction with document creation, editing, and readership) and the background elements that are carried out behind the scenes by agents and other automated and semi-automated tools.

The stage is then set for Section 4 which presents an agent framework which is useful to integrate the foreground and background elements introduced in Section 3 and design an efficient DMS.

Finally, Section 5 discusses how we expect DMS architectures to evolve in our socio-technical "knowledge ecology" framework.

2 The Intranet Document Lifecycle

Consider the case of a federalist organization; one with decentralized information technology authorities residing at various business units [29]. Such a firm's Intranet is typically organized with a central Web server, containing general information for the employees plus a navigation system to secondary servers. Thus the central Web server is a *hub* and the subordinate servers, each managed by local business groups, are *spokes*.

Document publication in a federalist organization is shown in Figure 1. Documents are created locally and typically uploaded to one or more spokes. The figure depicts sample departmental spokes such as Fixed Income, Economic Research, and so on taking as an example a financial services firm. The documents, after being lodged in the spokes, are indexed at intervals to be included in a search mechanism. Furthermore, the hub sometimes must be updated to point to new spokes or to point to new resources which might arrive at the spoke(s).

Each of the spokes has a typical audience, depicted by the dashed circles. Such an audience may bookmark preferred spokes. The audiences are defined by a natural match between the members' work functions and the spoke content. However, a unified search mechanism, which integrates the 'hub' and all the spokes, may occasionally bring the seeker to a spoke not commonly visited. The failure to provide an effective search is one of the most common shortcomings in document management praxis. This contributes to the recent finding by Newell et al. that in some multinational firms, the Intranet has the "ironic" outcome of reinforcing "existing functional and national barriers with electronic knowledge silos" [22]. Naturally, our goal is a DMS with the opposite effect. Prior research in Web document management has been rather ad-hoc, with an emphasis on integrating disparate pieces of software [1]. It is more fruitful to divide document management into phases for a more orderly approach to the problem. This suggests an *Intranet Document Lifecycle* perspective which we now discuss in detail to gain a better understanding of the potential problems facing a Document Management System (DMS).

Table 2 shows a waterfall model of the Intranet Document Lifecycle as adapted from [13]. The five stages are: creation, publication, organization, access, and destruction (not all of these stages may be actually implemented in a given organization's Intranet, most notably the destruction phase – many firms have no mechanism to systematically expire out-of-date documents). We now discuss each of these stages in turn.

Authors *create* a document locally and may, as stated previously, follow a template and create a well-structured document or create an ad-hoc, ill-structured one. Presentational metadata may be added, for example HTML tags or word processing formatting tags. Other useful metadata, such as author, keywords, title and so forth (cf. the Dublin Core author metadata standard [20]) may also be added by the author. Lassila [17] points out that metadata can be used to "describe the contents of an individual Web resource, such as a page, an image, or the content of a collection — Web site, directory, and so forth." He goes on to point out that "Search engines could take advantage of metadata . . . to perform more accurate searches." However this assumes an up-front investment to embed accurate metadata into the documents, a tricky proposition especially considering that most organizations have vast amounts of unstructured legacy documents, which lack metadata, on their intranet.

After the document acquires original content and metadata, it is then *published* to one or more of the spoke servers depicted in Figure 1, or, more rarely, the hub server. In the publication process, there may be conversion filters to change from one format to another (e.g. changing a MS-Word document to HTML, or changing MS-Powerpoint slides to GIF images). The converters may introduce inefficiencies [11] by increasing the net direct or indirect costs of document publication.

After a document is published to the target server(s), it must be integrated into an existing *organization*. First of all, a place in the local server's filesystem hierarchy must be selected. This selection may be up to the author's discretion or controlled by publication software (i.e. a mapping between the author's subject area and a priori filesystem organization). Once the document is fit into a certain hierarchical node, hyperlinks to and from the new document must be updated. Furthermore, it may be necessary to update the hub server's structure as well. Since the structure reflects the organization's view of the Intranet 'reality' (a representation of reality from the perspective of the structure creator(s)), it is commonly referred to as an *ontology*. If an organization imposes an ontology on a series of document collections, there is the possibility of vocabulary conflict across business units. As Pejtersen notes [24], there is a significant cost associated with forming classification schemes which cover the organization's various work domains. Furthermore, there may be political difficulties in merging ontologies across many business units — particularly in a federalist organization [25].

Documents which have been published and organized are subsequently eligible for *access*. Note that the *access* stage shown in Table 2 captures the entire user session.

3 The Gestalt of Document Management

At this point, it is useful to distinguish between the *foreground* and *background* processes inherent to document management. The foreground elements are the actors (authors, readers, editors) and their range of action; the background elements are automated and semi-automated system organization and feedback functions. Agent technologies, as well as more traditional data processing techniques, fall into the background category. We will term the combination of foreground and background elements the *gestalt* of the Document Knowledge Ecology.

Table 1 presents the foreground elements of Document Management Systems.

Table 1. The Gestalt of the Document Ecology, Part I: Foreground

Actors	Actions	Description
Author	Create	Document Creation
	Upload	Document to Staging Area
	Update	Edit, Overlay, Delete
Editor	Commit	Staged Document to Server(s)
Reader	Locate	Via Access Interfaces
	Read	Possible Knowledge Transfer
	Extend	Via Document Annotation

As Table 1 shows, the foreground actors are responsible for document creation and certification (approval) which leads to placement on one or more Intranet servers. Also part of the foreground is the everyday process of locating and reading interesting documents. It is also possible to leverage the intelligence of the aggregate readership by permitting annotations on the base documents. In this way, readers may act as secondary authors and create a hybrid object: a base document plus one or more annotations. Originally, a Web browser was also envisioned to be an editor [3] and since then work has continued to enable readers to act as secondary authors [12] [28] [18] [9]. If the DMS has annotation facilities, the readers can leave footprints as a historical record of their access trail — both annotation data (freeform comments) and metadata (supplying context regarding the annotation rationale) [12]. This conforms to Bush's notion that information seekers should be able to make use of *associative trails* [2] — the search might include a primary goal, but the searcher might also have a set of subordinate goals which may be satisfied by the Retrieval set. If sufficient numbers of users communicate their trials via interface changes in the search session, theoretically both workgroups and the organization in general could benefit. This contrasts with the approach taken in the `Amalthaea` system [19] where user information discovery and filtering agents are strictly compartmentalized and there is no inter-user coordination.

However, the background (system) elements are integral to the success of a given DMS. Table 3 lists these elements and demonstrates that attention must be paid to Document Organization, Access, and Maintenance. If the Intranet is small, the servers can be self-managed. However, as the number of servers and range of topics grows, both editorial control and agent-based software solutions may be brought into play to help determine where the documents should be placed.

To access documents, the users must choose a search strategy. For example, a full-text search might be used or a parametric one (where documents belong to nodes in a prespecified tree). Once the search strategy is fixed, the user must supply keyword(s) to the Search Interface and subsequently navigate an array of hyperlinks to base documents (the Retrieval Interface). The lowest layer is the Document Interface, where the user must browse the content to determine if the keyword match was indeed sufficient. The users spend most time iterating between the Retrieval and Document Interfaces in a given search session, with limited data and metadata clues [12] [14].

Document maintenance is another important piece of the background. If we structure the topics as a tree, with headings being nodes and documents being members of various nodes, then we should pay attention to trees which become imbalanced (empty nodes, or over-full nodes). In addition, we have the problem of out-of-date documents, or documents which have stale information (with more current information perhaps resident elsewhere in the archive). Manual deletion is a typical choice of large Intranets, not trusting automatic rules which are dangerous (for example, some documents are meant to have a lifetime of many years, such as corporate strategy statements as opposed to short-lived document types such as memos). However, the failure to clean up effectively leads to an overall decrease in document quality and hence lowers the prospect for effective knowledge transfer when documents are read.

The final phase of the Document Lifecycle is, naturally enough, *destruction*. A failure to systematically purge the archives of stale documents contributes to an overall degradation in document quality. However, it is difficult to determine from a document's content its natural lifespan. Again, annotation facilities are useful here to flag documents which have been supplanted by others or which are simply no longer of interest [14].

3.1 Metadata of Document Management Systems

Documents serve as the container housing the data of a document archive. However, foreground actors' interactions with the data containers also add important metadata improving the overall data content.

If annotation facilities are provided, the annotation event also builds metadata about the base document. For example, in the `Annotate` system [12], the annotation is composed of both freeform notes and integer-valued appraisals. The appraisals may be used as the basis for a social filter in the Search interface. Another user-supplied event is simple search feedback — typically, a system might ask if the user felt the document helped solve the task at hand [19] [15]. All events which are user-dependent will be sporadic, with indefinite but slower growth rates than raw system metrics. A screenshot from the `Annotate` system is shown in Figure 2.

Figure 2 shows that the results from a search may also include icons identifying prior reader annotation activity. The workgroup is identified in the "D" (domain) column and various qualitative ratings are assigned using a Likert Scale. The 1 to 7 Likert Scale is transformed into intuitive happy- and frowny-face icons; a technique borrowed from experiments with poker-playing agents [16].

Metadata can also come from the publication process. The positioning of a document at a certain level of a spoke server hierarchy is important metadata which may be used in document mining operations [10] [30]. In a system which relies on ontologies, the document will add to the membership set of a given node or nodes. If new categories must be created to accommodate the document, the breadth or depth of the ontology will change. This consideration is equally applicable to local (spoke) and central (hub) ontologies.

Ontology maintenance and update can be quite difficult to implement. The next section explores the use of software agents to attack this important subproblem of document management.

4 Using Agent Technology in DMS Design

Software agents can act on behalf of three distinct entities in the organizational DMS setting: the individual, the workgroup, and the organization itself. In the case of the individual, agents can assist in the authoring and access processes. Workgroups, which often have focused tasks at hand [27], can benefit from agents to coordinate their access efforts.

Finally, the organization (the entity which took the time and effort to implement the DMS) can make use of agents in two important ways. Firstly, agents can serve by facilitating *evaluation* of the DMS. Secondly, and more ambitiously, agents can pro-actively react to poor evaluation findings by helping to create or revise *incentive strategies* in those DMS which have collaborative features which rely on individual voluntary contributions.

Table 4 shows a general DMS agent framework, indicating the entity, or level which the agent supports, its name, the DMS stage at which the agent can assist, and an action example of the agent at work [13].

To assist the individual, an *Access Agent* may assist in personalizing an access coordination model (*coordination* in the IR sense). Some users may want to make use of an a prior ontology; others may not trust that representation and go directly to full-text search. Search results, tracked over time, can help modify the agent's coordination strategy recommendation.

A *Notification Agent* is useful in systems which support annotation. When an annotation event occurs, this agent can be configured to either notify or not notify the base document author as desired and be further configured to offer a choice of notification media.

The next three individual-level agents are related: *Search Interface*, *Retrieval Interface*, and *Document Interface* Agents are all instances of Interface Agents. These three interfaces comprise the complete set of user access interfaces in the document search and retrieval process. The Interface Agents monitor ecological conditions based on access history and current activity. Given a set of trigger rules, these agents are trusted to alter the access interface(s) to assist in inter- and intra-interface user navigation.

In the initial Search Interface, global usage metadata may be used to alter the search interface (e.g. showing the most popular keywords to date). Annotation data (if it exists) can act as a social filter [12].

In the subsequent Retrieval Interface, annotations can be mapped to simple icons to reflect opinions and annotation source and reason [12]. Another possibility is to overlay the document's position in the filesystem together with the search engine confidence scores to provide additional clues (this is the focus of Hearst and Chen's Cha-Cha project [5]). In addition, visual techniques can be used: fan-out diagrams [21] [7] or Kohonen map visualizations [4]. Another possibility is to use a nearest-neighbor statistical algorithm, such as Memory Based Reasoning [31] to find current *situations* (keyword(s) and the retrieval list) which are close to prior situations and to show the user prior *actions* taken (documents chosen from the retrieval list).

Finally, the lowest layer of the Search session, the Document Interface, can be altered to show a summary table of annotation to date in one frame while the unadulterated base document is shown in another [12]. This layer may also provide the ability to give feedback on the efficacy of the core search engine [19]. It is also possible to decompose the document into constituent layers, e.g. text and OCR (scanned) images, each with a separate annotation facility [26]. Finally, individual information authors and readers would benefit from *Usage Evaluation* agents which would help collate and analyze an individual's search history.

Agent support should also be provided to workgroups: either groups that band together for an ad-hoc mission or groups which are natural peers based on an organizational structure. Both types of groups need efficient access to documents in the enterprise knowledge base [23]. A *Publicity Agent* is useful to announce the arrival of new documents to other spoke servers which may partially overlap in functional content with the publication target server. This announcement could be processed by a similar Publicity Agent on the other server(s) in the spirit of a 'message of the day'. And, Usage Evaluation Agents would also be useful to Workgroups to gauge the efficiency of peer search. Recent work along these lines is described in [27].

In a continuous, batch process a *Metadata Agent* may convert legacy documents by placing simple metadata wrappers around them indicating basic facts such as format, author, and keywords. These wrappers may be declared in, for example, XML Document Type Definitions. The Metadata Agent may also be used to align current documents to an organizational metadata specification, by filling in sparse or missing data). The simplest organizational agent is the *Validation Agent*: it operates in a Publication Event by checking for valid document data formats and metadata fields and values. The Publication Event may also trigger a more complex set of agents, the Global and Local *Ontology Agents* which are responsible for revising local (spoke) or global (hub) ontology structures. They would necessarily need to be armed with domain-specific vocabulary.

An agent which maps to organizational policy in the case of the DMS annotation feature is the *Authentication Agent*. This agent can permit the annotator and/or the seeker to configure full- (the annotator's real name), semi- (the workgroup to which the annotator belongs), or none (anonymous; no identification of the annotator). Some organizations might opt for full authentication to lend full credibility and responsibility for the notes; others might opt for anonymous commentary out of deference for the contributor's privacy. This choice affects the overall composition of the annotations. For example, prior research suggests that anonymity tends to lead to higher levels of criticism [6].

Hypothetical agents which assist in the overall evaluation and improvement of an implemented DMS are not explored in the literature, yet critical for DMS success. Evaluation is a meta-stage, shadowing the document lifecycle stages. Strategically speaking, the organization should be able to measure the effectiveness of the DMS since the costs are significant (implementation, resources needed to collate and index information, and the opportunity costs of the seeker. The evaluation component thus must be addressed.

For example, an organizational *Usage Evaluation Agent* can track the global history of document access, user feedback (if any) on the search process, annotation history (if any), as well as node memberships in local and global ontologies (if they exist). Conditions which might be cause for alarm can be mapped to agent actions. If annotations are found to be lacking on a given spoke server, notifications can be generated to the appropriate author list and site administrators.

While not exhaustive, the discussion of agent possibilities in this section makes it clear that document management offers a rich coordination problem between authors, readers, editors, and DMS administrators.

The problem of insufficient coordination is the biggest problem facing DMS solutions today. No document knowledge ecology can thrive in the absence of critical coordination paths between the readers and between the readers and authors. There is no need to limit ourselves to the asynchronous HTTP protocol, though, as we supply coordination pathways. For example, if a reader annotates a given author's work, push technologies can alert the author in real-time (as compared to the author only learning about it sometime later via passive Web exploration and discovery that the retrieval interface as changed as shown in Figure 2).

5 Conclusion

The problems posed by constructing effective Document Management solutions in firms with large archives have captured significant research interest of late since they combine Information Retrieval issues and the practical viewpoint of being a common problems faced by today's global, federalist firms. What we are lacking, though, is an understanding of how coordination and the integration of foreground and background elements can help improve the document knowledge ecology. If we can manage the documents, which are a key enterprise resource, more effectively, then by extension we are also managing the authors' expertise more effectively.

We have presented the discrete phases of the Intranet Document, the foreground actors and actions, and the background system elements, such as agent software, which might help design and implement an effective DMS to breed a lively document ecology.

Since a DMS encompasses inextricable technical and social elements, coordination pathways must integrate these themes as well by conforming to the social norms of the implementation site and the technical organization of the internal network. As we have seen, background elements can assist in such key areas as coordination, authentication, evaluation, and incentive strategy.

In the future, we expect agents and other automated pieces of the DMS background to play an ever-increasing role in the Intranet Document Management process, particularly in DMS systems with collaboration and coordination elements. Integration of the background, system elements with the foreground, human elements is an essential prerequisite to the construction of a healthy document knowledge ecology.

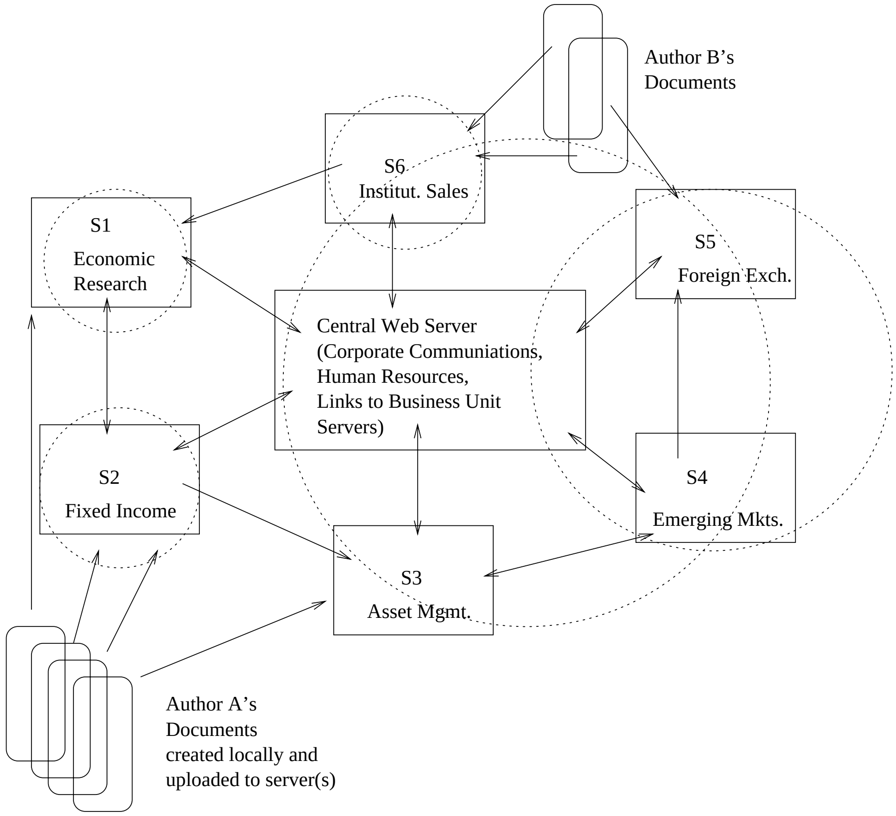

Figure 1. The document publishing process in a Federalist Organization. A central 'hub' server and subordinate 'spokes' controlled by business units; regular readership communities denoted by dashed-line circles. The spoke business units shown are drawn from an actual Financial Services Firm Intranet.

Table 2. Waterfall Model of the Intranet Document Lifecycle. Parenthetical items are optional features.

Step	Creation	Publication	Organization	Access	Destruction
1	Document				
(1a)	Author Meta-data				
2		publication to selected server(s)			
(3)		conversion to semantically richer, poorer, or lateral file formats			
4			placement in local server hierarchy		
(4a)			inclusion into existing local ontology		
(4b)			update existing global ontology		
(4c)			build or update hyperlinks to and from new document		
5				choose search strategy	
6				choose search terms	
7				navigate retrieval list	
8				read selected documents	
(8a)				add annotation data and metadata to document	
(8b)				provide search feedback	
(9)					stale documents deleted or overwritten

Table 3. The Gestalt of the Document Ecology, Part II: Background

Processes	Subtypes	Comments
Document Organization	Manual	Possible for small archives
	Author-driven	Can muddle the structure
	Overseen	Editorial Control, popular for large archives
	Dynamic	Utilize both editors and agent software
Document Access	generic	Standard full-text or parametric search
	customized by individual	tracking individual search history
	customized by group	leveraging static and dynamic workgroup structures
Document Maintenance	manual	leads to document bloat and overall decrease in archive quality
	automatic	difficult given range of document types
	hybrid	coordination via consensus

Table 4. Agent Framework

Level	Agent Name	Stage	Action Example
Organizational[1]	Incentive	Evaluation	Monitors Cooperation and Suggests Incentive Policy Shifts as needed
	Usage Evaluation	Evaluation	Collate, Analyze and Visualize Global Usage History
	Metadata	(batch)	Add Metadata to Legacy Documents; align all documents to common metadata specifications
	Validation	Publication	Check for valid data and metadata when new documents enter the archives
	Local Ontology	Publication	Revise local structures as needed based on existing framework and newly entered document(s)
	Global Ontology	Publication	Revise global structure as needed based on existing framework and newly entered document(s)
Workgroup	Publicity	Post-Authoring	Announce New Contents to Other Servers; announce document(s) to peer workgroup members
	Authentication	Search	Display full-, semi-, or no annotation authentication information in the Retrieval and Document interfaces
	Usage Evaluation	Evaluation	Collate, Analyze and Visualize Group Usage History
Individual	Access	Search	Personalize Pre- Post- or Hybrid Coordination
	Notification	Post-Annotate	send e-mail to document authors upon annotation event
	Search Interface	Search	Coordination Prior Usage to assist in in individual search formulation
	Retrieval Interface	Retrieval	Annotation Events alter detail and aggregate Retrieval Interface
	Document Interface	Document	Annotation Events alter the Document Interface
	Usage Evaluation	Evaluation	Collate, Analyze and Visualize Individual Usage History

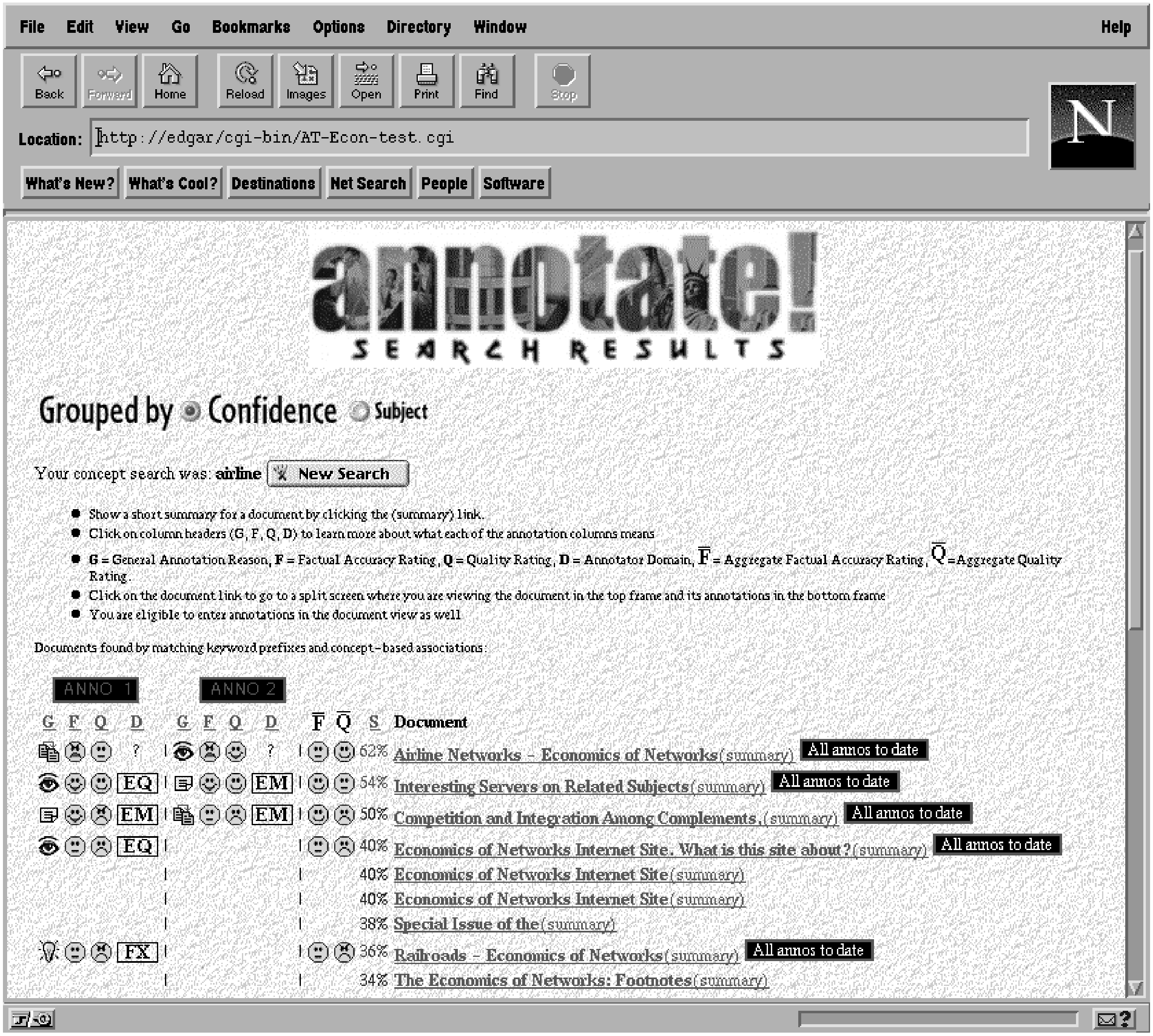

Figure 2. The Retrieval Interface of the Annotate System

References

[1] V. Balasubramanian and A. Bashian. Document management and Web technologies: Alice marries the mad hatter. *Communications of the ACM*, 41(7):107–115, July 1998.

[2] V. Bush. As we may think. *Atlantic Monthly*, 176(1):101–108, July 1945.

[3] R. Caillau. How it really happened. Technical report, IEEE, April 1998. Internet Computing Online.

[4] H. Chen, J. J. Nunamaker, R. Orwig, and O. Titkova. Information visualization for collaborative computing. *IEEE Computer*, 31(8):75–82, August 1998.

[5] M. Chen, J. Hong, J. Lin, and M. Hearst. Cha-cha: A system for organizing intranet search results. In *Proceedings, World Wide Web Conference*. W3C, 1998.

[6] T. Connolly, L. M. Jessup, and J. S. Valacich. Effects of anonymity and evaluative tone on idea generation in computer-mediated groups. *Management Science*, 36(6):689–703, 1990.

[7] J. W. Cooper and R. J. Byrd. OBIWAN — a visual interface for prompted query refinement. In *Digital Documents*, volume 2, pages 277–285. 31st Hawaii International Conference on System Sciences, IEEE, January 1998.

[8] T. H. Davenport and L. Prusak. *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, Boston, MA, 1998.

[9] D. DeRoure, W. Hall, S. Reich, A. Pikrakis, G. Hill, and M. Stairmand. An open architecture for supporting collaboration on the Web. In *Proceedings, Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages –80, Los Alamitos, CA, June 1998. IEEE Computer Society, IEEE Press.

[10] O. Etzioni. The World-Wide Web: Quagmire or gold mine? *Communications of the ACM*, 39(11):65–68, November 1996.

[11] J. Farrell and G. Saloner. Converters, compatibility, and the control of interfaces. *Journal of Industrial Economics*, 40(1):9–35, 1992.

[12] M. Ginsburg. Annotate! a tool for collaborative information retrieval. In *Proceedings, Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 75–80, Los Alamitos, CA, June 1998. IEEE Computer Society, IEEE Press.

[13] M. Ginsburg. An agent framework for intranet document management. *Autonomous Agents and Multi-Agent Systems*, 2(3):271–286, 1999.

[14] M. Ginsburg and A. Kambil. Annotate! a Web-based knowledge management support system for document collections. In *32st Hawaii International Conference on System Sciences*. IEEE, Sony Electronic Press, January 1999.

[15] G. Golovchinsky. What the query told the link: the integration of hypertext and information retrieval. In *Proc. 8th ACM Conference on Hypertext*, New York, NY, 1997. ACM, ACM.

[16] T. Koda and P. Maes. Agents with faces: The effects of personification of agents. In *Proceedings of HCI'96*, London, 1996.

[17] O. Lassila. Web metadata: A matter of semantics. *IEEE Internet Computing*, pages 30–37, July-August 1998.

[18] G. Marchionini and G. Crane. Evaluating hypermedia and learning: Methods and results from the perseus project. *ACM Transactions on Information Systems*, 12(1):5–34, January 1994.

[19] A. Moukas and P. Maes. Amalthaea: An evolving multi-agent information filtering and discovery system for the WWW. *Autonomous Agents and Multi-Agent Systems*, 1:59–88, 1998.

[20] L. D. Murphy. Digital document metadata in organizations: Roles, analytical approaches, and future research directions. In *Digital Documents*, volume 2, pages 267–276. 31st Hawaii International Conference on System Sciences, IEEE, January 1998.

[21] R. Neches, F. Hu, I.-Y. Ko, K.-T. Yao, Q. Zhu, and P. Will. Collaborative information space analysis tools. In *Proceedings, Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Los Alamitos, CA, June 1998. IEEE Computer Society, IEEE Press.

[22] S. Newell, H. Scarbrough, J. Swan, and D. Hislop. Intranets and knowledge management: Complex processes and ironic outcomes. In *32st Hawaii International Conference on System Sciences*. IEEE, Sony Electronic Press, January 1999.

[23] I. Nonaka. A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1):14–37, 1994.

[24] A. M. Pejtersen. Semantic information retrieval. *Communications of the ACM*, 41(4):90–92, April 1998.

[25] C. J. Petrie. The XML files. *IEEE Internet Computing*, pages 4–5, May-June 1998.

[26] T. A. Phelps and R. Wilensky. Toward active, extensible, networked documents: Multivalent architecture and applications. In *Proceedings of the 1st ACM International Conference on Digital Libraries*, pages 100–108, Bethesda, MD, March 1996. ACM.

[27] N. C. Romano Jr., J. F. Nunamaker Jr., D. Roussinov, and H. Chen. Collaborative information retrieval environment: Integration of information retrieval with group support systems. In *Proceedings, 32st Hawaii International Conference on System Sciences*. IEEE, Sony Electronic Press, January 1999.

[28] M. Röscheisen, C. Mogensen, and T. Winograd. Interaction design for shared world-Wide Web annotations. In *CHI 95: Human Factors in Computing Systems*, pages 328–329, New York, May 1995. ACM.

[29] J. W. Ross and J. F. Rockart. Enabling new organizational forms: A changing perspective on infrastructure. *Proceedings of the International Conference on Information Systems (ICIS)*, December 1996.

[30] E. Spertus. ParaSite: Mining structural information on the Web. In *Proceedings, Sixth International World Wide Web Conference*, Santa Clara, CA, 1997. W3C, W3C.

[31] D. I. Waltz. *Natural and Artificial Parallel Computation*, chapter Memory-Based Reasoning. MIT Press, 1990.