Autonomous Agents and Multi-Agent Systems, 2, 271–286 (1999)

An Agent Framework for Intranet Document Management

MARK GINSBURG
Fisher Center, Haas School of Business, University of California, Berkeley

Abstract. Document management inside an organization is a complex and broadly scoped problem. This paper approaches the technical and social issues of Intranet document management by developing a straightforward document lifecycle model consisting of five phases: creation, publication, organization, access, and destruction. A document management system (DMS) which encompasses these areas should also have an evaluation component so its effectiveness can be measured.

The document lifecycle is visualized as a waterfall model to help explore the discrete phases of an idealized Intranet DMS. The discussion of this model pinpoints where traditional DMS have fallen short, most notably in the areas of user-to-user and user-to-evaluator communication and coordination.

From the document lifecycle, we derive an agent framework to integrate technical and social considerations and guide the design, implementation, and evaluation of a flexible and efficient DMS. The lifecycle model and agent framework are useful to organize both technical and social perspectives in this area.

Keywords: Intranet, document management, document lifecycle, information retrieval

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. Furthermore, 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.

Document management is particularly interesting because it poses an elusive and ill-structured goal which combines technical and social challenges. The first area of uncertainty is the term *document* itself. 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 [1]. Thus, a document might be classified as a systematic knowledge asset [2] (i.e. well-structured, following a template) or it might be quite irregular and unsystematic. In any event, an organization's documents[1] will vary in length, presentational format, semantic content, and quality.

DMS solution attempts fall at the intersection between software engineering (information retrieval systems) and social theories of group dynamics and organizational incentive models. After all, even the most clever document coordination models are moot if nobody elects to use the implemented system. Therefore, a DMS design must be flexible and extensible not just to technical (quantitative) sea-changes, but also to changes in the social environment. This motivates our presentation of an agent framework to help us understand how to position a DMS in this socio-technical environment.

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

In Section 3 we identify and characterize the data and metadata stores which are available to an idealized DMS.

The stage is then set for Section 4 which presents an agent framework which is useful in the design of an efficient DMS which will be nimble in the face of socio-technical change.

Finally, Section 5 presents concluding remarks on the role of agents in Intranet document management.

2. The Intranet document lifecycle

Consider the case of a federalist organization; one with decentralized information technology authorities residing at various business units [3]. 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" [4]. Naturally, our

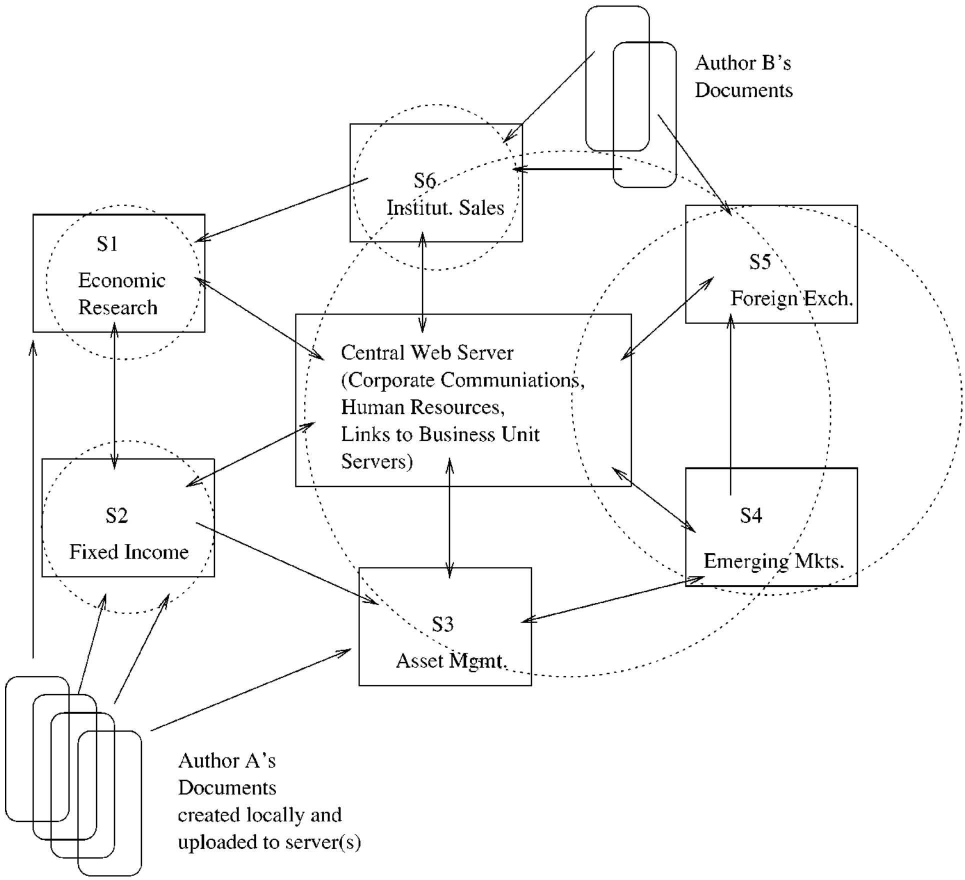

Figure 1. The document publishing process in a federalist organization. A central "hub" server and subordinate "spokes" controlled by business units; regular readership communities denoted by dashed-line circles. The spoke business units shown are drawn from an actual financial services firm Intranet.

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 [5]. 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 1 shows a waterfall model of the Intranet document lifecycle. 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 process-

Table 1. Waterfall model of the Intranet document lifecycle. Parenthetical items are optional features.

Step	Creation	Publication	Organization	Access	Destruction
1	Document Author				
(1a)	Metadata				
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 coor-dination method (pre- post- or hybrid)	
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

ing formatting tags. Other useful metadata, such as author, keywords, title, and so forth (cf. the Dublin Core author metadata standard [6]) may also be added by the author. Lassila [7] 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 [8] 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 [9], 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 [10].

Documents which have been published and organized are subsequently eligible for *access*. Note that the *access* stage shown in Table 1 captures the entire user session.

2.1. Access coordination mechanisms

Users often wish to search on multiple concepts at the same time and hence submit queries of several keywords. There are two major strategies to handle the combination, or coordination of the search terms: pre-coordinate or post-coordinate search.[2]

Many firms elect to collate the contents of the hub and spoke servers into a full-text search mechanism, also known as a *post-coordinate* search. However, a prior ontological structure offers an interesting access alternative: *pre-coordination*. Pre-coordination is defined as fixing the citation order of a subject heading at index time [11]; this assumes significant time and effort on the part of a cataloger before subject term search is possible. The terms follow a standard order, for example `Europe | Labor | Wage`. Post-coordination is so named because the keywords are combined at search time; there is no subject term taxonomy specified a priori. This is a possible weakness, as the user may want a dependency, i.e. a masking of secondary terms if the primary term does not occur. Many full text search engines, such as `Excite`, apply statistical methods to an unordered vector of keywords.[3]

Theoretically, the most effective search is *hybrid*: combining pre- and post-coordination [12] although this is rarely implemented in practice. An example implementation would be to construct a relational database mapping subject term headers from the ontology to the members (i.e. documents) at each node. This would permit a simple SQL interface to query the ontology. To accomplish the hybrid search, it only remains to enable full-text search as a fall-back.

Once the coordination access type has been chosen, 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 [13, 14].

It is also possible to leverage the intelligence of the aggregate readership by permitting annotations on the base documents. Originally, a Web browser was also envisioned to be an editor [15] and since then work has continued to enable readers to act as secondary authors [13, 16, 17, 18]. 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) [13]. This conforms to Bush's notion that information seekers should be able to make use of *associative trails* [19]—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 [20] where user information discovery and filtering agents are strictly compartmentalized and there is no inter-user coordination.

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. The building blocks of a document management system

Both everyday system use and special, sporadic events contribute to data stores which are of potential interest to DMS evaluators.

Table 2 presents and describes common metadata stores which grow as the DMS is used. We now present these metadata stores in more detail. Document *access* via any of the coordination mechanisms (pre-, post- or hybrid as discussed in Section 2.1) will build raw usage metrics. For example, the keywords, hyperlink retrieval list arrays, and document traversals can be captured. Furthermore, the timings at the search, retrieval, and document interface layers can be captured [13]. There is no limit to the growth of the raw metrics and it will grow at a steady rate as users conduct search sessions.

If annotation facilities are provided, the annotation event also builds metadata about the base document. For example, in the `Annotate` system [13], 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 [20, 21]. All events which are user-dependent will be sporadic, with indefinite but slower growth rates than raw system metrics.

The final class of metadata store stems 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 [22, 23]. 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. Over time, the rate of local and global ontology change will decrease and reach an asymptotic value. Generally speaking, the global ontology is faster to reach an asymptote since its function is often mainly to point to spoke table of contents entry point pages.

The question now becomes: how can we make use of the document lifecycle phases and the metadata stores discussed in this section to guide our understanding of how agents can usefully contribute to DMS design and implementation? This is the focus of the next section.

Table 2. Document management system metadata

System metadata	Data source (step in Table 1)	Provided by	Growth	Growth rate
raw usage metrics	everyday system usage [6–8]	seeker/automatic	indefinite	steady
annotation	annotation event [8a]	seeker/voluntary	indefinite	sporadic
search feedback	feedback event [8b]	seeker/voluntary	indefinite	sporadic
local server hierarchy	breadth, depth, node membership [4]	system/automatic	asymptotic	decreasing
local ontology structure	breadth, depth, node membership [4a]	system/automatic	asymptotic	decreasing
global ontology structure	breadth, depth, node membership [4b]	system/automatic	asymptotic	decreasing

4. An agent framework for 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 [24], 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 3 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.

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 priori ontology; others may not trust that representation and go directly to full-text search. Still others may want to reverse the coordination order in a hybrid approach and try full-text search first, then revert to an ontology-search. It should be possible to use actual search results over time to modify the agent's 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* are all instances of interface agents which alter, on a user's behalf, the user interface [25]. 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). Or, as previously mentioned, annotation data (if it exists) can act as a social filter [13].

In the subsequent retrieval interface, annotations can be mapped to simple icons to reflect opinions and annotation source and reason [13]. 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 [26]). In addition, visual techniques can be used: fan-out diagrams [27, 28] or Kohonen map visualizations [29]. Another possibility is to use a nearest-neighbor statistical algorithm, such as memory-based reasoning [30] 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).[5] However, MBR may fail in the DMS domain due to a potentially quite large keyword and document space.

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

Table 3. Agent framework

Level	Agent name	Stage	Action example
Organizational[4]	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 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

unadulterated base document is shown in another [13]. This layer may also provide the ability to give feedback on the efficacy of the core search engine [20]. It is also possible to decompose the document into constituent layers, e.g. text and OCR (scanned) images, each with a separate annotation facility [31]. 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 [2]. 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 [24].

The most abstract entity which the agent may serve is the organization. At this level, agent possibilities range from the very simple (e.g. validation agents) to the quite complex (incentive agents). In general, the more mechanized the process (i.e. the less that social considerations enter the picture), the simpler the agent implementation. The idea of having agents assist the enterprise in enhancing the adoption of a large-scale software system such as a DMS is very appealing. At the same time, agent interactions with the social structure of the firm guarantees complexity.

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. These agents may also be programmed to delete nodes with no members after a certain amount of time.

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 [32]. The important point is that organizational norms and policies, which are social issues, must be mapped to the agent implementation [14].

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 metastage, 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.

In the case of a collaborative DMS which relies on annotations to coordinate associative trails among the users, a hypothetical *incentive agent* has appeal. To understand incentive agents, it is useful to consider the Schelling diagram, which is a compact way to measure free-ridership (viewing others' annotations without contributing oneself) versus cooperation (opting to contribute annotations).

The time and effort of participation is the definition of the opportunity cost. In terms of a Schelling diagram [33] we have Figure 2 as the simplest possibility, where everybody is purely motivated by self-interest and there are no incentives to "join" (in the case of *Annotate* [14] and similar systems, to annotate documents). It follows that an important design goal of an annotation feature should be to minimize the opportunity cost, all other things being equal, or annotating.

In this figure, the incentives facing the marginal chooser are shown contingent on all others' choice. Going from the left (0, or no cooperation) to the right (1, or full cooperation) we consider the marginal choice. The vertical distance from the horizontal axis measures V, the value to the individual [33] for choosing free ridership (FR) or cooperation (C).

At any point on the cooperation spectrum, the marginal chooser always benefits from free-riding. Thus the unraveling forces movement to the left, although socially everyone would be better off at the right [33].

However, the free-rider (FR) and cooperation (C) curves can be altered by lowering the private opportunity cost with positive incentives, such as a reward bestowed by the organization, or negative ones, such as monetary penalties. Typically, positive incentives suffer from decreasing marginal effectiveness as cooperation levels

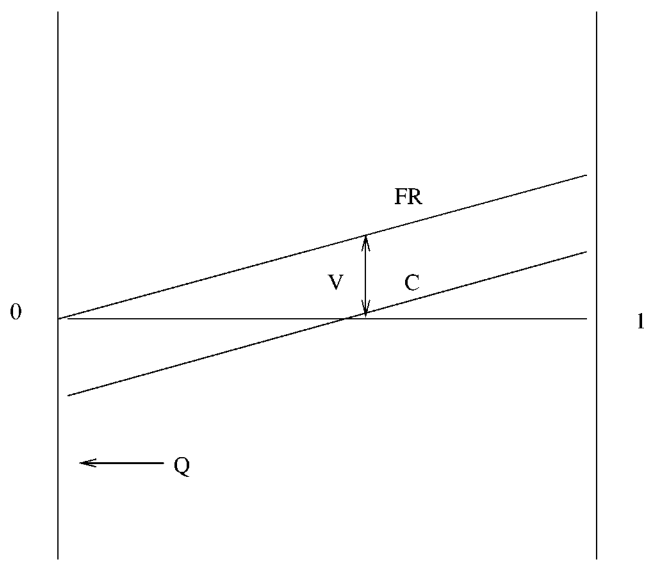

Figure 2. In the absence of incentives, free ridership is preferable to cooperation at all levels of aggregate cooperation.

increase [33]. Margolis also points out that there is a social dimension to participation; i.e. the social value attached by the individual to the choice. Social motivation can influence Schelling diagrams in a complex manner, especially when considering also the private incentive effects. Margolis presents the "NSNX" heuristics: "neither selfish nor exploited" [33]. This states two rules: firstly, that an individual will be more likely to use resources socially if they can be applied effectively compared to what he or she could do with the same resources to advance private interests. Secondly, the individual will be less inclined to spend resources socially if he or she notices that others are not spending resources socially. This implies that if there is a high degree of cooperation, the individual will be inclined by this principle to go along and cooperate as well.

Considering incentives and social effects, we can invert sections of the free ridership and cooperation curves as shown in Figure 3.

There exist two equilibria in Figure 3. The inferior one, $Q-$, is reached when the cooperation level is shifted to the left of the *tipping point*, t. The superior equilibrium, $Q+$, is reached when one moves to the right of t.

A firm without a well thought-out incentive policy will probably find itself at *Q*-point shown in Figure 3. An incentive agent can act autonomously, sensing lack of cooperation in the aggregate or on a per-workgroup basis, and select one of a battery of incentive shifts. Again, the range of possible incentives must conform to organizational policies and norms. For example, the incentive agent, on behalf of senior management, might offer an "Article of the Week" award (this is a "soft" or non-monetary incentive) to authors who generate the most annotation feedback. The agent can track the history of incentive type, target audience, and duration of the incentive. Furthermore, negative incentives can be part of the battery. Non-participants can be threatened with non-monetary or monetary sanctions. It

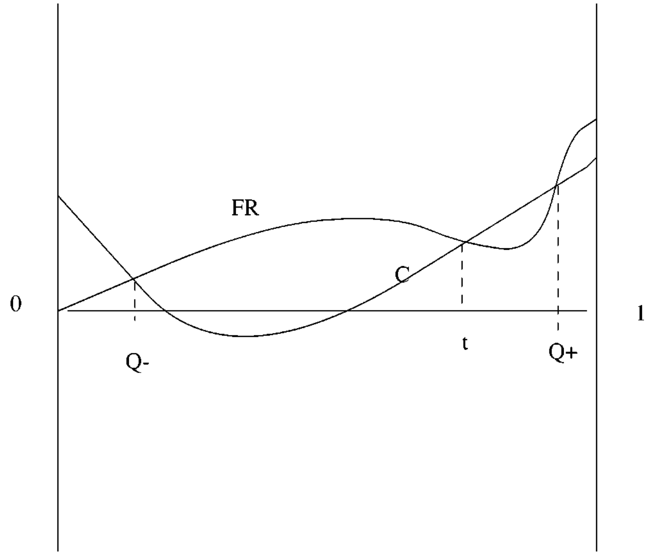

Figure 3. Incentives and social considerations alter the free-ridership and cooperation curves.

is possible to allow incentive agents a free hand in selecting various combinations of negative and positive incentives, within policy guidelines, and track the results over time. The wide-ranging effect of such agents will require a high degree of organizational trust.

While not exhaustive, the discussion of agent possibilities in this section makes it clear that document management offers a rich, multi-agent system (MAS) domain for further study. The typical problems of MAS research such as inter-agent communication and coordination of processes [34] hold true in this domain as well. And, since DMS implementations must take into account subjective factors such as overall user satisfaction with document search and retrieval, there is the extra human dimension to add complexity. Numerous coordination problems arise: for example, in an ontology (in IR terms, *pre-coordinate*) search, how should we design global and local ontology agents to coordinate with the interface agents which the user sees during the access phase of Table 1? For each given social level (individual, workgroup, or enterprise), there are unique agent coordination problems. Mapping agent implementations to organizational policies and norms is an important step since, again, we cannot separate the technical from the social in the document management context. A successful DMS will have coordination pathways between as many stages of the lifecycle model as possible for a given implementation and also provide coordination between the lifecycle stages and the important metastage of DMS evaluation.

5. Conclusion

Both collaborative information retrieval and document management have captured significant research interest of late since they combine elements of classic information retrieval theory and the practical advantage of being common problems faced by today's global, federalist firms. What we are lacking, though, is an understanding of how agent software can help in these tasks: the management of documents, which are a key enterprise resource, and the design of an efficient and effective information access mechanism which ensures that the authors' expertise is not managed inefficiently.

We have presented the discrete phases of the Intranet document, the metadata stores that accrue from the use of a document management system, and the agent classes which might help design and implement an effective DMS to shed light on these important questions.

We note that a DMS must encompass inextricable technical and social elements. Hence, agent implementations must combine 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, agents can assist in such key areas as coordination, authentication, evaluation, and incentive strategy. This application of agent technology is certainly more qualitative than most, since effective access is quite subjective and focused on the information recipient.

Going forward, we expect agents to play an ever-increasing role in the Intranet document management process, particularly in DMS systems with collaboration and coordination elements. The document lifecycle presented here should prove useful in a divide-and-conquer approach to this broadly scoped, common, and strategically important problem.

Notes

1. The following definition of *document* is useful: "A structured amount of information intended for human perception, that can be interchanged as a unit between users and/or systems" [35] (cited in [36]). Furthermore, there is an "intentionality" behind the creation of text which differentiates it from other kinds of data [37].
2. In this section, we are staying within the information retrieval (IR) context where the term *coordination* simply means the combination of the users' concepts at search time and does not relate the notion, from agent research, of coordination between agent activities [38].
3. To clarify the difference between pre- and post-coordination in the IR context, this quote is helpful [39]: "When concepts are combined or coordinated to form complex subjects, such coordination may be carried out by the indexer or by the searcher. The former is referred to as pre-coordinate indexing and the latter as post-coordinate indexing." Note in passing that this is orthogonal to the idea of *controlled vocabulary*. A pre-coordinate ontology does not have to be restricted to a pre-defined vocabulary list (the decision is up to the cataloger(s)). Similarly, it is possible to impose a controlled vocabulary on post-coordinate search.
4. *Organizational* in the management sense of *enterprise* or *firm*, not in the agent sense of organizing the agent community.
5. This is similar to Lashkari *et al.*'s work [40] where a successful agent implementation was demonstrated on top of Eudora, a commercial e-mail program. Memory-based reasoning (MBR) was used to capture user patterns as pairs: the e-mail content and context information represented a *situation* and the user's handling of the message represented an *action*. For example, an *action* in this context might be forwarding the mail to a set of other users, saving it into a particular folder, or deleting it.

References

1. T. H. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What They Know*, Harvard Business School Press, Boston, MA, 1998.
2. I. Nonaka, "A dynamic theory of organizational knowledge creation," *Organization Science*, vol. **5**(1), pp. 14–37, 1994.
3. 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.
4. S. Newell, H. Scarbrough, J. Swan, and D. Hislop, "Intranets and knowledge management: Complex processes and ironic outcomes," *32nd Hawaii International Conference on System Sciences*, January 1999, IEEE, Sony Electronic Press.
5. V. Balasubramanian and A. Bashian, "Document management and Web technologies: Alice marries the mad hatter," *Communications of the ACM*, vol. **41**(7), pp. 107–115, 1998.
6. L. D. Murphy, "Digital document metadata in organizations: Roles, analytical approaches, and future research directions," in *Digital Documents, 31st Hawaii International Conference on System Sciences*, January 1998, vol. 2, pp. 267–276, IEEE.
7. O. Lassila, "Web metadata: A matter of semantics," *IEEE Internet Computing*, pp. 30–37, July-August 1998.
8. J. Farrell and G. Saloner, "Converters, compatibility, and the control of interfaces," *Journal of Industrial Economics*, vol. **40**(1), pp. 9–35, 1992.
9. A. M. Pejtersen, "Semantic information retrieval," *Communications of the ACM*, vol. **41**(4), pp. 90–92, 1998.

10. C. J. Petrie, “The XML files,” *IEEE Internet Computing*, pp. 4–5, May-June 1998.
11. E. Svenonius, “Unanswered questions in the design of controlled vocabularies,” *Journal of the American Society for Information Science*, vol. **37**(5), pp. 331–340, 1986.
12. D. Bodoff and A. Kambil, “Pre-coordination plus post-coordination: The case for partial coordination,” Tech. Rep. IS-97-14, New York University, Information Systems Department, New York, 1997.
13. M. Ginsburg, “Annotate! a tool for collaborative information retrieval,” in *Proceedings, Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Los Alamitos, CA, June 1998, IEEE Computer Society, pp. 75–80, IEEE Press.
14. M. Ginsburg and A. Kambil, “Annotate! a Web-based knowledge management support system for document collections,” in *32nd Hawaii International Conference on System Sciences*, January 1999, IEEE, Sony Electronic Press.
15. R. Caillau, “How it really happened,” Tech. Rep., IEEE, April 1998, Internet Computing Online.
16. M. Röscheisen, C. Mogensen, and T. Winograd, “Interaction design for shared world-Wide Web annotations,” in *CHI 95: Human Factors in Computing Systems*, New York, May 1995, pp. 328–329, ACM.
17. G. Marchionini and G. Crane, “Evaluating hypermedia and learning: Methods and results from the perseus project,” *ACM Transactions on Information Systems*, vol. **12**(1), pp. 5–34, 1994.
18. 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*, Los Alamitos, CA, June 1998, IEEE Computer Society, pp. 90–95, IEEE Press.
19. V. Bush, “As we may think,” *Atlantic Monthly*, vol. **176**(1), pp. 101–108, 1945.
20. A. Moukas and P. Maes, “Amalthaea: An evolving multi-agent information filtering and discovery system for the WWW,” *Autonomous Agents and Multi-Agent Systems*, vol. **1**, pp. 59–88, 1998.
21. 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.
22. O. Etzioni, “The World-Wide Web: Quagmire or gold mine?,” *Communications of the ACM*, vol. **39**(11), pp. 65–68, 1996.
23. E. Spertus, “ParaSite: Mining structural information on the Web,” in *Proceedings, Sixth International World Wide Web Conference*, Santa Clara, CA, 1997, W3C, W3C.
24. 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, 32nd Hawaii International Conference on System Sciences*, January 1999, IEEE, Sony Electronic Press.
25. P. Maes, Modeling adaptive autonomous agents, in *Artificial Life Journal*, MIT Press, 1994.
26. 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.
27. 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.
28. J. W. Cooper and R. J. Byrd, “OBIWAN—a visual interface for prompted query refinement,” in *Digital Documents, 31st Hawaii International Conference on System Sciences*, January 1998, vol. 2, pp. 277–285, IEEE.
29. H. Chen, J. Nunamaker Jr., R. Orwig, and O. Titkova, “Information visualization for collaborative computing,” *IEEE Computer*, vol. **31**(8), pp. 75–82, 1998.
30. D. I. Waltz, Memory-based reasoning, in *Natural and Artificial Parallel Computation*, MIT Press, 1990.
31. 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*, Bethesda, MD, March 1996, pp. 100–108, ACM.
32. T. Connolly, L. M. Jessup, and J. S. Valacich, “Effects of anonymity and evaluative tone on idea generation in computer-mediated groups,” *Management Science*, vol. **36**(6), pp. 689–703, 1990.
33. H. Margolis, Free riding versus cooperation, in *Strategy and Choice*, MIT Press, 1991.

34. N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous Agents and Multi-Agent Systems*, vol. **1**, pp. 7–38, 1998.
35. ISO, "ISO 8613: Information processing—text and office systems—office document architecture (oda) and interchange format," Tech. Rep., International Organization for Standardization, 1989.
36. W. Weitz, "SGML nets: Integrating document and workflow modeling," in *Digital Documents, 31st Hawaii International Conference on System Sciences*, January 1998, vol. 2, pp. 185–194, IEEE.
37. M. Hearst, "Distinguishing between Web data mining and information access," in *Proceedings, Knowledge Discovery*, Newport Beach, CA, 1997, KDD.
38. V. R. Lesser, "Reflections on the nature of multi-agent coordination and its implications for an agent architecture," *Autonomous Agents and Multi-Agent Systems*, vol. **1**, pp. 89–111, 1998.
39. E. J. Hunter and K. G. B. Blakewell, *Cataloguing*, 3rd ed., Library Assocation Publishing, Ltd., London, England, 1991.
40. Y. Lashkari, M. Metral, and P. Maes, "Collaborative interface agents," in *Proceedings of AAAI'94*, Seattle, Washington, August 1994, AAAI.