The Networked Resource Discovery Project[1]

Michael F. Schwartz
October 1988

Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309-0430
(303) 492-3902

Abstract

Large scale computer networks provide access to a bewilderingly large number and variety of resources, including retail products, network services, and people in various capacities. We consider the problem of allowing users to *discover the existence* of such resources in an administratively decentralized environment. We describe an approach for a system that accesses the distributed collection of repositories that naturally maintain resource information, rather than building a global database to register all resources. A key problem is organizing the resource space in a manner suitable to all participants. Rather than imposing an inflexible hierarchical organization, our approach allows the resource space organization to evolve in accordance with what resources exist and what types of queries users make. Concretely, a set of *agents* organize and search the resource space by constructing links between the repositories of resource information based on keywords that describe the contents of each repository, and the semantics of the resources being sought. The links form a general graph, with a flexible set of hierarchies embedded within the graph to provide some measure of scalability. The graph structure evolves over time through the use of cache aging protocols. Additional scalability is targeted through the use of probabilistic graph protocols. A prototype implementation and a measurement study are under way.

[1] This material is based upon work supported in part by the National Science Foundation under Cooperative Agreement DCR-84200944, and by a grant from AT&T Bell Laboratories.

1. Introduction

The past decade has witnessed a tremendous increase in the number and capabilities of interconnected computer systems. With these increases have come corresponding increases in computer-supported resources, including a variety of electronic mail services, information retrieval services, and marketing services. While many efforts have been directed towards the basic interconnection and management of such facilities, much less emphasis has been placed on allowing human users to navigate through a large collection of the resources available through them. This problem is not solved by name services, which typically only allow clients to locate an instance of some resource *given its name*. Most systems provide very little support for resource discovery, relying primarily on users' knowing the names for resources *a priori,* from information obtained outside of the system [Fowler 1986]. For example, people typically obtain electronic mail names verbally or through electronic mail messages, and learn of the existence of network services through documentation or bulletin board postings.

In addition to these computer system resources, we consider a wide variety of physical resources, such as retail products registered in corporate inventory databases, and people sharing particular interests registered in special interest group membership lists. Hence, we would like to be able to support a wide variety of resource searches, such as "a nearby laser printer", "the electronic mail names of graphics experts in New York City", "TCP implementations for IBM mainframes", "inexpensive lawn mowers", and "movies playing in town tonight".[2]

Directory services employed by current network providers (including those providing telephone, electronic mail, and online information services) do not by themselves solve this problem, largely because they are fragmented: one must access a variety of different mechanisms and databases to cover this wide variety of resources. Additionally, these mechanisms are human labor intensive, and often impose constraints that the user may not wish to see, such as the geographical divisions in telephone directories. When searching for a particular resource, users must usually try a number of categories, following a chain of suggestions from various sources. If the resource is sufficiently obscure or specialized, the search typically fails. This problem will only become more pronounced as the various networks become more integrated through ongoing international standardization efforts such as the ISO protocols [DesJardins & Foley 1984] and MAP/TOP [Farowich 1986, Kaminski Jr. 1986]. The problem will become still more acute as networking grows to encompass vastly increased numbers of users through the introduction of ISDN technology in people's homes [Gawrys et al. 1986].

One could build a global database that registers all resources, but the conversion from the current collection of *Resource Information Repositories (RIRs)*[3] that hold such information would be quite expensive, and keeping the database up-to-date would be probably be impossible. More importantly, such a global database would

[2] These examples are intended to demonstrate the general applicability of the problem. Whether we will be able to support searches as sophisticated as these is not yet clear.

[3] RIRs can be any form of information that can be utilized in the resource discovery process. While one typically would think of them as databases, they could also be information derived by active processing (e.g., comparing fields of particular files).

require centralized administration, and many organizations would be unwilling to relinquish control over their information. In essence, the problem with building such a *reregistered* database is that the information more naturally belongs in the distributed set of repositories where people own and maintain the information. Therefore, our approach is to access the information where it naturally resides.

The remainder of this paper is organized as follows. In Section 2, we describe our approach to organizing the resource space. In Section 3, we describe the system architecture that supports this organization. In Section 4, we discuss the protocols that establish and evolve the resource space organization without human intervention. In Section 5, we discuss the status of several efforts to demonstrate the validity of the design, including measurements of existing systems and a prototype implementation. In Section 6, we discuss related work. Finally, in Section 7, we summarize the project.

2. Resource Space Organization

Organizing the resource space in a manner suitable to all participants is a difficult prospect. Part of the difficulty is what has been called the *vocabulary problem*, the tendency for people to use a large number of different terms for any concept, requiring support for many alternative access words in systems requiring human inputs [Furnas et al. 1987]. Systems based on a standardized set of category descriptors have been successfully deployed (e.g., telephone Yellow Pages categories and ACM Computing Reviews Subject Descriptors), but none suffice to describe the breadth of resources that could potentially be of interest to users of a resource discovery mechanism. More importantly, standardized categorizations do not allow the world to evolve rapidly and gracefully. As an example of such evolution, when compact disk and other digital sound technologies entered the marketplace, the worlds of music equipment and information systems became closer. At this time, a resource space reorganization became appropriate. This point is demonstrated by how various music and computer trade magazines began advertising more products in each others' respective domains at that time.

What we would like is that at any instant, the most popular organizational schema will be the most efficient in which to search for resources, without restricting more specialized schema from coexisting. We want to allow the resource space organization to evolve over time automatically, in accordance with what resources exist, and what types of queries users make.

In addition to the vocabulary problem, one must consider how the keywords that describe resources are arranged and used. The obvious approach is to arrange all keywords into a hierarchy, as done in many file systems. However, a single, strict hierarchy does not adequately support real world organizational needs. As a hierarchy grows, its organization often becomes convoluted and inconsistent, because users are forced to encode a variety of different information into a single hierarchy. For example, the UNIX[4] file name */users/faculty/schwartz/pdp/monte/asynch/init.o* contains (from left to right) information about the file's disk

[4] UNIX is a trademark of AT&T Bell Laboratories.

location, creator's role, creator, research project, research subproject, algorithm variant, contents (*init* = "initialization routines"), and file type (*.o* = "object code").[5] Reorganizing such a hierarchy is time consuming and not easily accommodated, once a user base has been established.

More importantly, a strict hierarchy is inflexible. For example, in searching for people having technical expertise in three dimensional graphics shading algorithms, one person might prefer to organize the world as "/Computers/Graphics/3D/Experts", while another might prefer "/People/Interests/Technical/Graphics/3D", as illustrated in Figure 1. Moreover, as the world evolves, the resource space organization must change. Requiring global agreement for such changes would slow the process tremendously.

Figure 1: Conflicting Hierarchies

Human social networks have evolved a structure that utilizes a more direct set of connections between participants. Rather than forming contacts with each other based on a simple hierarchical mechanism (such as a bureaucracy), people often establish more direct *networks* by contacting knowledgeable intermediaries who can quickly refer them to other relevant persons, cutting across bureaucratic boundaries. For example, by contacting a professor or business person, someone interested in high-speed networking technology can quickly meet other people who share this interest. These other people can, in turn, introduce the person to others who perhaps more closely share his/her particular interests. At the same time, the newcomer can be instrumental in pointing out individuals who share other interests with the persons he/she meets.

The success of such networks is based on what has been called the "small world" phenomenon. Consider a graph where nodes are people and edges represent one person's knowing another. It has been observed that the diameter of such a graph (i.e., the maximum number of edges in the shortest path between any two nodes) is surprisingly small, even in an enormous setting. For example, there is a mathematical game based on the co-

[5] This example is a modified version of one given in [Greenspan & Smolensky 1983].

author relationship with the prolific mathematician Paul Erdos: One's *Erdos number* is defined to be 1 if that person has written a paper with Erdos, 2 if they have written a paper with someone who has written a paper with Erdos, etc. Based on this definition, the highest Erdos number known to be possessed by a person is 7 [Hoffman 1987]. This small world observation has also been the subject of various small-scale sociological studies. For example, see [Travers & Milgram 1969] and the chapter on Network Interaction and Structure in [Boissevain 1974].

One can imagine an analogous property for resource networks (where diameter refers to this *knows-about* relation, rather than physical network connectivity). Given this property, there should be many paths to discovering information about any particular resource, and resource discovery could, in theory, be very efficient. The difficulty, of course, is finding a path to discovering a resource, given little or no *a priori* knowledge about the global knows-about relation topology. Given only a collection of RIRs, it would not be possible to determine a path in general, since relationships between RIRs do not naturally exist.[6] However, by augmenting the system with some entities that actively develop a useful knows-about relation graph, we can provide a system that does exactly this. We call this the *networked* approach to resource space organization.

To make this resource space organization appropriate for use in large scale systems, we utilize a flexible form of hierarchy, called *specialization subgraphs*. Using this construction, a resource could be a member of multiple hierarchies (representing different organizational schema), and there could be back pointers and cycles in the graph. For example, one specialization subgraph could link databases containing information about automobile parts, while another subgraph could link databases according to geographic boundaries. The automobile parts subgraph could, in turn, have one subgraph organized according to function (engine parts, tires, etc.), another subgraph organized according to manufacturer, etc.

The graph construction is abstractly illustrated in Figure 2. Starting with the simple hierarchy of Figure 2A, a general graph that embeds this hierarchy can be constructed by linking nodes at the same nesting level together, and placing pointers between one or more of these nodes and their parent node, as illustrated in Figure 2B. In this way, it is possible to reach related resources by traversing a chain of pointers between the nodes at the same *specialization level* (e.g., nodes **G**, **H**, **I**, and **J**). Figure 2C indicates how a second set of edges (shown with different shading) might link related resources according to a different organizational scheme. For the sake of simplicity we have not shown many links between related nodes. In a real system we expect the links to be considerably more complex. Also, these figures contain a large amount of link redundancy. In Section 4 we consider techniques to reduce this redundancy.

This style of graph organization is essentially a special case of the network database model, where the network has more structure (i.e., specialization subgraphs) and the graph structure is built and evolved without the need for human intervention. Note that the graph edges in the figures concern resource categorization, not

[6] For example, if one asks the John Deere Company where to buy the most cost effective tractor, their database is not likely to mention International Harvester.

Figure 2: General Graph Construction

network topology. For the purposes of the graph organization, we ignore the low-level issues associated with network topology, such as message routing.

3. System Architecture

To provide a resource discovery mechanism, we introduce three new types of entities: *agents* that dynamically construct links between the RIRs, *brokers* that encapsulate the heterogeneity and access control concerns of the RIRs, and *clients* that initiate resource searches on behalf of users by communicating with agents. This architecture is illustrated in Figure 3. When an RIR enters the network, the broker associated with it announces to any agent the set of resource description keywords about which that RIR can answer queries. These keywords could concern product-specific information (such as movie names or replacement part sizes) as well as generic information (such as prices or geographic location of retail outlets that stock the product). Over time, brokers can detect which announced keywords are most useful, as well as which keywords that were not announced commonly occur in queries. Based on these measurements, the broker's administrators can introduce and remove keywords (up to a predetermined maximum number of keywords per resource type).

A client initiates a resource search by contacting any agent. If the agent does not know about the specified keywords, it must try to find some other agent that does, possibly following a multiple hop chain of agents. The agents examine requests and decide, based on the named keywords, how best to route searches. In some cases, agents initiate transactions with brokers to access online information maintained by various organizations around the network (e.g., a telephone directory or a company's product line description) to discover what resources exist at these organizations. In other cases, searches are routed to other agents that know more about the resources being sought. In Section 4 we discuss mechanisms for deciding how to route searches.

Figure 3: Basic System Architecture

Since resource information is distributed among autonomous systems, sharing the information poses privacy and security problems. It should not be possible to use the system to obtain information that would violate individual privacy or corporate competitiveness. Our approach to this problem involves encapsulating each autonomous RIR by a broker responsible for accessing the RIR and deciding exactly what information may be released to the outside world. Brokers correspond to human operators that currently allow the general public limited access to many existing databases. For example, telephone directory service operators are responsible for deciding to refuse queries asking what person has a particular telephone number, but they will answer many other queries. Brokers can be built by information providers (or by a third party and inspected by the information providers) to ensure that they are trustworthy.

A relatively large amount of resource information can be shared this way. For example, corporations would likely be willing to release product line information (other than sales figures, etc.), since that is essentially a form of advertising. Many individuals would also participate, just as they are currently registered in telephone directories. However, this model does not capture other interesting sharing relationships. A more sophisticated model that allows more information to flow within than across administrative boundaries is considered in [Schwartz 1988].

There are several points to notice about agents and brokers. First, each broker is a piece of special-purpose software that understands the access mechanisms and privacy concerns for a particular RIR, whereas agents are general purpose software that can be implemented as a single set of protocols and distributed/ported to all parts of the environment. Second, brokers are developed and administered by RIR providers, with the incentive to *advertise* their resources, whereas agents must, for the sake of fairness of resource access, be unbiased. Third, the agent/broker division provides *separation of concerns*. Agents are concerned with the agent/category topology; clients and brokers can contact any agent (e.g., by broadcasting on a local network), without concern for how the resource discovery process will progress.

We now consider how agents manage the resource graph.

4. Establishing and Evolving the Resource Graph

Establishing the resource graph edges is a bootstrapping problem. Once some set of edges exists, searches can usually proceed by following the edges according to their labels (i.e., the keywords). Bootstrapping the resource graph involves providing a means for agents to discover the existence of other agents that know about particular resource keywords. Clearly, any solution based on centralized search processing, fully replicated information, or full scale broadcast would not work in a large scale implementation. Instead, each agent should be capable of finding RIRs that hold resource information for some keywords, and should be able to route searches to more appropriate agents for other keywords.

Because communication failures may occur and RIRs may be unavailable at any point in time, it is not possible to guarantee exhaustive searches. This fact, in combination with the scalability problems presented by attempting exhaustive searches, makes it clear that the system should only guarantee reasonably thorough searches. We carry this observation one step further, utilizing a suite of probabilistic protocols to establish and search the graph. By doing so, we hope to gain a measure of scalability beyond what could be achieved by deterministic protocols.

With this motivation in mind, we introduce a primitive called a *sparse diffusion multicast*, defined as follows. Given a set of N target agents, a message is sent to a subset of size $k + \log(N)$, selected at random, where k is a (tunable) constant that ensures that some minimum set of agents receive the transmission. Using this primitive, we construct a graph bootstrapping protocol where agents sparse diffusion multicast the keywords they know about to other agents. As with all probabilistic protocols, the success of this technique relies on its repeated application. By tuning the extent and frequency of sparse diffusion multicasts, we can hopefully construct a system that is scalable yet effective at supporting resource searches: over time the graph will approach a steady state, where most of the edges needed for efficient resource searches are in place. This mechanism will also reduce some of the unnecessary redundancy shown in Figure 2C.

If a search is requested at an agent that has no information about the named keywords and no pointers to other agents that know about the named keywords, a local broadcast can be issued, to contact any nearby agents that may have received a sparse diffusion multicast about related keywords. If this fails, the agent can use a sparse diffusion multicast to search for other agents that could help, possibly increasing the "density" of the sparse diffusion multicasts. Finally, if all of these tactics fail, the client can ask the user to choose from a menu of keywords that are known by that agent. The expectation is that over time, agents will acquire increasing amounts of useful information, reducing the need for these more expensive techniques.

The bootstrapping protocols provide a basis for establishing a set of graph edges. However, some of these edges will be infrequently used, and over time should be replaced by other, more appropriate edges. We utilize a set of caching protocols for this purpose. When an agent receives a sparse diffusion multicast from some other agent, it caches the announced keywords, setting cache timeouts for each keyword in proportion to the amount of information already cached for that keyword. The agent then responds to the multicasting agent with its own set of keywords, so that agent may cache them. In addition, at randomly selected intervals agents sparse diffusion multicast the set of keywords that are in the transitive closure of the keywords they know about. This protocol is illustrated in Figure 4. In part A of this figure, node **A** issues a sparse diffusion multicast, which reaches four other nodes. Later (and not necessarily all at once), each of these nodes will issue sparse diffusion multicasts, reaching many of the other nodes with high probability. Of course, if a query is later made that traverses a multiple hop chain constructed by this protocol, it makes sense to collapse the path for future use, using protocols such as those analyzed by Fowler [Fowler 1986].

Figure 4: Multiple Hop Chain Construction

The keyword cache exchange protocols cause specialization subgraphs to develop, so that over time, a graph such as the one illustrated in Figure 5 develops. In this figure we have redrawn brokers to surround the databases, to emphasize the fact that brokers encapsulate RIRs, and agents organize them. We show several brokers linked by agents based on their keywords. We have shown some links darker than others, indicating the relative strengths of association between the brokers. These edge weights can be drawn from various metrics, such as the number of keywords in common. The lighter shaded edge could, in fact, point to a higher level of a specialization subgraph, based on the fact that only more heavily weighted keywords are in common with the adjacent nodes.

Figure 5: Example Graph Organization

The techniques considered so far are essentially syntactic in nature: RIRs and searches are characterized by simple collections of keywords. However, it would be more effective to exploit the semantics of the resources being sought and the context within which resource discovery takes place. As an example of exploiting semantics, knowing that research personnel are located primarily at universities and a few industrial research laboratories could tremendously narrow the set of brokers that need to be contacted. As an example of exploiting the context within which resource discovery takes place, knowing that the user who initiated the search is more concerned with geographic proximity than price for small appliances could eliminate searching RIRs from distant companies.

These techniques can be used to reduce the level of effort the system must expend during searches, and can also reduce the amount of irrelevant information with which users are confronted. This is important, as demonstrated by experiences with information retrieval systems that must typically sacrifice the ability to recall all relevant information in favor of suppressing large amounts of extraneous information [Salton 1986]. Unfortunately, semantically-intelligent techniques would probably not scale well for use in large environments, since they require special-purpose software for each type of resource and each user. As a compromise, we use primarily syntactic techniques in the agents, aided by semantically-intelligent techniques in a few judicious circumstances. In particular, we are building a system that exploits semantics for only a few often sought resources (such as finding users' electronic mail names), but also provides "hooks" for allowing clients to define their own semantically-intelligent procedures for resources of special interest to them. One particularly important type of semantic information that can be incorporated into agents is location: in some cases resources can be sought from anywhere (e.g., looking for an inexpensive source of tractors, it may be cheaper to buy one from out of state and have it shipped). In other cases, location does matter (e.g., looking for a nearby printer, or a pizza delivery service). Such semantics could be incorporated into agents to limit the scope of searches.

5. Project Status

We are currently pursuing a five stage prototype implementation of the ideas discussed in this paper. In the first stage (which is currently under way), we are implementing the agent protocols, and beginning a set of measurement experiments with these protocols. In the second stage, we will construct a system with brokers that access mock RIRs, filled with information about fictitious resources. This will allow us to concentrate on the central themes of the research, ignoring less relevant details of real-world RIRs. In the third stage, we will access a collection of UNIX file systems, treating them as a variety of different RIRs. There is a rich set of resource information available from these file systems, including users' "plan" files, files mapping user names to account names, files listing electronic mail aliases, files containing interest membership lists (e.g., departmental sporting and technical interest lists), etc. In the forth stage, we hope to begin accessing a collection of real RIRs that span geographic, organizational, and functional boundaries. To do this, we are currently exploring the possibilities for collaboration on an industrial product development effort. In a possible fifth stage, we may implement a more sophisticated information sharing mechanism, as discussed in [Schwartz 1988]. We will use the Heterogeneous Remote Procedure Call facility developed at the University of Washington [Bershad et al. 1987] to interconnect the various system components, so that we can accommodate heterogeneous systems.

In addition to the prototype implementation, we are beginning to gather data about current electronic mail usage at a collection of universities and companies working on a variety of research, education, and product development projects around the world, to help parameterize a theoretical model of our system. We will collect log information from sendmail, the Berkeley UNIX mail agent [Allman 1985]. This involves monitoring the "From:" and "To:" lines of electronic mail on a temporary basis at some representative institutions, to detect who is communicating with whom. For this purpose we use a script that collects these lines in sendmail logs, and then mails the data to us. Using this data, we can compute the diameter of this sample resource graph, as well as a number of other, more detailed graph characteristics. Approximately twenty institutions have agreed to participate in the study, and another fifteen are currently considering doing so. We are considering statistical techniques for analyzing the voluminous data that we expect, since many graph computations are too costly to run on the full complement of data we expect to collect.

6. Related Work

We believe that our approach to resource discovery is strengthened by its connections with a large number of other research problems. In this section we consider a sample of some areas of related work. A more detailed survey is available in [Schwartz 1988].

Name Service and Directory Browsing Mechanisms

Directory browsing mechanisms constitute some of the earliest instances of online support for resource discovery. The most familiar example of such systems are the directory systems of file systems. Well-known distributed directory browsing mechanisms include the directory service components of the proposed X.400 mail standard [IFIP 1983], the Network Information Center Whois service [Harrenstien, Stahl & Feinler 1985], and the proposed X.500 directory service [CCITT 1987]. Each of these services supports queries containing ambiguous strings, responding with the set of all matching names. Peterson et al. provide a more sophisticated mechanism that does not require all resources to be nameable at any point in time, using a collection of tools that support a bottom-up construction of the naming network. Their system supports various boolean and relational combinations of attributes, providing an administratively centralized Yellow Pages for naming network services in an internet [Bowman, Peterson & Rao 1988].

Information Retrieval Services

Information retrieval services support text retrieval based on a set of descriptive keys, such as the author and title of a document. Example systems include bibliographic database systems (such as INSPEC) and online information services (such as CompuServe). These systems require centralized administration, rather than supporting access to some decentralized collection of information sources. There have been several efforts to utilize more sophisticated techniques to increase human effectiveness in using information retrieval systems. Hypertext systems provide complex cross-references between parts of a document, as well as sophisticated user interfaces capable of allowing users to traverse links and stack up sessions, in support of scanning a document non-linearly [Conklin 1987]. The HELGON system allows users to interactively search through information by providing the user with examples of the next level of the naming tree, and by supporting query reformulation through several different specification techniques [Fischer & Nieper-Lemke 1988]. Streeter and Lochbaum describe a system based on a technique called latent semantic analysis, oriented towards representing terms, documents and queries in a manner that accommodates the fact that there are many words that refer to the same concept [Streeter & Lochbaum 1988]. Gordon describes an approach more closely related to our approach, in which the organization of an information retrieval system evolves over time using a set of probabilistic "genetic" algorithms [Gordon 1988].

Connectionist Computing

The idea of agents establishing graph edges in accordance with what resources exist is reminiscent of the "learning" notion introduced by connectionist computing ("neural network") researchers: both cases involve an interconnection graph whose edges are somehow established over time through some feedback process with the real world. However, the goals and techniques used in neural networks are quite different from ours. The goal of connectionist computing is to provide a computing model suitable for building applications that deal with complex aspects of the real world (such as pattern recognition) without the need for the complex programming required by the standard von Neumann architectures. The techniques used typically involve many simple processors, each of which contains a very small amount of information that when combined with the other processors' information can lead to useful function [Tank & Hopfield 1987].

Routing in Communication Networks

The way searches traverse resource discovery agents is somewhat similar to the problem of routing messages through a communication network. However, there are several important differences. First, the addressing scheme in computer networks is relatively simple, as compared with the ambiguous and evolving categorization scheme we consider. Second, routing involves seeking a reasonably direct path to one node. Resource discovery involves seeking a reasonably direct set of paths to find many instances of a specified type of resource. Finally, our approach involves a probabilistic bootstrapping mechanism of a nature that, to our knowledge, is not found in network routing algorithms. Some recent routing research comes closer to our approach, in that messages are routed between nodes according to a mechanism somewhat like the "knows-about" relation we consider [Tsuchiya 1987].

7. Summary

The goal of the Networked Resource Discovery Project is to explore a set of mechanisms that could provide an administratively decentralized means for users to navigate through an enormous resource space without imposing an inflexible hierarchical naming structure on that space. The resource discovery problem is not solved by name services, which typically only allow clients to locate an instance of some resource *given its name*. A resource discovery mechanism has wide applicability, but poses some difficult technical problems. Our approach involves a set of agents that dynamically construct and evolve links in a general graph structure between related repositories of resource information in a manner that corresponds to system usage patterns. Because constructing and evolving the graph links is potentially expensive, we utilize a suite of probabilistic protocols for establishing and searching the graph structure. This structure was motivated in part by a collection of observations about the organization of human social networks. Based on these observations we have introduced several concepts, the most important of which are agent specialization subgraphs, which manifest the notion of locality of concern; sparse diffusion multicasts, which support wide, sparse announcements; and cache exchange and aging protocols that help evolve and refine the graph structure. This approach treats resources primarily as syntactic entities described by keywords, because of the problems of scale inherent in trying to exploit semantics during resource searches. Yet, we believe that exploiting the semantics of the resources being sought and the context within which resource discovery takes place can significantly reduce the level of effort expended by the system and the amount of extraneous information presented to the user. As a compromise, we use primarily syntactic techniques, aided by semantic techniques in a few judicious circumstances. We are currently pursuing a prototype implementation of these ideas, as well as a measurement study to help parameterize a theoretical model of the system.

Acknowledgements

Many people have helped shape the ideas presented here. David Cheriton provided early inspiration by pointing out the "small world" phenomenon in the context of naming systems. Edward Lazowska, Evi Nemeth, David Notkin, Gary Nutt, and John Zahorjan provided early feedback about the project goals. Bob Allen, Randy Katz, Robert Kraut, and Tom Landauer indicated several references related to the "small world" phenomenon. Bruce Schatz and Rick Schlichting offered several suggestions concerning the agent protocols. Lou Ceci and Paul Smolensky pointed out issues related to connectionist models of computing. Jim Driscoll provided suggestions about approaches for analyzing the graph algorithms and the data to be gathered by the Internet electronic mail graph study. Dennis Heimbigner made several suggestions about the prototype effort, as well as the desired user view of the system. Tom Gasaway, Darren Kalmbach, Carl Koch, and David Wood participated in many project design discussions, and are currently involved in the measurement study and prototype implementation.

8. References

[Allman 1985]
E. Allman. Sendmail - An Internetwork Mail Router. *UNIX Programmer's Manual, 4.2BSD*, 2C, Comput. Sci. Division, EECS, Univ. of California, Berkeley, June 1985.

[Bershad et al. 1987]
B. N. Bershad, D. T. Ching, E. D. Lazowska, J. Sanislo and M. Schwartz. A Remote Procedure Call Facility for Interconnecting Heterogeneous Computer Systems. *IEEE Trans. Software Eng.*, SE-13(8), pp. 880-894, Aug. 1987. To be reprinted in *Distributed Processing: Concepts and Structures*, A.L. Ananda and B. Srinivasan, IEEE Computer Society Press.

[Boissevain 1974]
J. Boissevain. *Friends of Friends - Networks, Manipulators, and Coalitions.* Oxford, Blackwell, 1974.

[Bowman, Peterson & Rao 1988]
M. Bowman, L. L. Peterson and H. Rao. Univers: A Name Server for the Next Generation Internet. Tech. Rep. 88-17, Dept. Comput. Sci., Univ. Arizona, Tucson, AZ, Mar. 1988.

[CCITT 1987]
CCITT. The Directory - Overview of Concepts, Models and Services. ISO DIS 9594-1, CCITT, Gloucester, England, Nov. 1987. Draft Recommendation X.500. ISO/CCITT directory convergence document #1. Version 7.

[Conklin 1987]
J. Conklin. Hypertext: A Survey and Introduction. *IEEE Computer Magazine*, 20(9), pp. 17-41, Sep. 1987.

[DesJardins & Foley 1984]
R. DesJardins and J. S. Foley. Open Systems Interconnection: A Review and Status Report. *J. Telecommunication Networks*, 3(3), pp. 194-209, 1984.

[Farowich 1986]
S. A. Farowich. Communicating in the Technical Office. *IEEE Spectrum*, 23(4), pp. 63-67, Apr. 1986.

[Fischer & Nieper-Lemke 1988]
G. Fischer and H. Nieper-Lemke. HELGON: Extending the Retrieval by Reformulation Paradigm. Tech. Rep., Dept. Comput. Sci., Univ. Colorado, Boulder, CO, Sep. 1988. Submitted for publication.

[Fowler 1986]
R. Fowler. The Complexity of Using Forwarding Addresses for Decentralized Object Finding. *Proc. 5th ACM Symp. Principles Distr. Comput.*, pp. 108-120, Aug. 1986.

[Furnas et al. 1987]
G. W. Furnas, T. K. Landauer, L. M. Gomez and S. T. Dumais. The Vocabulary Problem in Human-System Communication. *Commun. ACM*, 30(11), pp. 964-971, Nov. 1987.

[Gawrys et al. 1986]
G. Gawrys, P. Marino, G. Ryva and H. Shulman. ISDN: Integrated Network/Premises Solutions for Customer Needs. *Proc. IEEE Int. Conf. Commun.*, pp. 2-6, June 1986.

[Gordon 1988]
M. Gordon. Probabilistic and Genetic Algorithms in Document Retrieval. *Commun. ACM*, 31(10), pp. 1208-1218, Oct. 1988.

[Greenspan & Smolensky 1983]
S. Greenspan and P. Smolensky. DESCRIBE: Environments for Specifying Commands and Retrieving Information by Elaboration. In *User Centered System Design, Part II: Collected Papers from the UCSD HMI Project*, Institute for Cognitive Science, Univ. of California, San Diego, Dec. 1983.

[Harrenstien, Stahl & Feinler 1985]
K. Harrenstien, M. Stahl and E. Feinler. NICName/Whois. Req. For Com. 954, Oct. 1985.

[Hoffman 1987]
P. Hoffman. The Man Who Loves Only Numbers. *Atlantic Monthly Magazine*, 260(5), pp. 60-74, Nov. 1987.

[IFIP 1983] IFIP. *Naming and Directory Services for Message Handling Systems.* IFIP WG 6.5, July 1983. Working paper, Version 4.

[Kaminski Jr. 1986]
M. A. Kaminski Jr. Protocols for Communicating in the Factory. *IEEE Spectrum*, 23(4), pp. 56-62, Apr. 1986.

[Salton 1986]
G. Salton. Another Look at Automatic Text-Retrieval Systems. *Commun. ACM*, 29(7), pp. 648-656, July 1986.

[Schwartz, Zahorjan & Notkin 1987]
M. F. Schwartz, J. Zahorjan and D. Notkin. A Name Service for Evolving, Heterogeneous Systems. *Proc. 11th ACM Symp. Operating Syst. Prin.*, pp. 52-62, Nov. 1987.

[Schwartz 1988]
M. F. Schwartz. The Networked Resource Discovery Project: Goals, Design, and Research Efforts. Tech. Rep. CU-CS-387-88, Dept. Comput. Sci., Univ. Colorado, Boulder, CO, May 1988.

[Streeter & Lochbaum 1988]
L. A. Streeter and K. E. Lochbaum. An Expert/Expert-Locating System Based on Automatic Representation of Semantic Structure. *Proc. Fourth Conf. on Artificial Intelligence Applications*, pp. 345-350, San Diego, CA, Mar. 1988.

[Tank & Hopfield 1987]
D. W. Tank and J. J. Hopfield. Collective Computation in Neuronlike Circuits. *Scientific American*, 257(6), pp. 104-114, Dec. 1987.

[Travers & Milgram 1969]
J. Travers and S. Milgram. An Experimental Study of the Small World Problem. *Sociomety*, 32(4), pp. 425-443, 1969.

[Tsuchiya 1987]
P. F. Tsuchiya. Landmark Routing: Architecture, Algorithms, and Issues. Tech. Rep. MTR-87W-00174, Mitre Corp., Sep. 1987.