The Networked Resource Discovery Project: Goals, Design, and Research Efforts

Michael F. Schwartz

CU-CS-387-88

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

ANY OPINIONS, FINDINGS, AND CONCLUSIONS OR RECOMMENDATIONS EXPRESSED IN THIS PUBLICATION ARE THOSE OF THE AUTHOR(S) AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE AGENCIES NAMED IN THE ACKNOWLEDGMENTS SECTION.

The Networked Resource Discovery Project: Goals, Design, and Research Efforts

Michael F. Schwartz

CU-CS-387-88 May 1988

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

Abstract

Computer systems are being used to register an increasing 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 a large scale, administratively decentralized environment. We describe an approach that accesses the distributed collection of repositories that naturally maintain resource information, rather than building a global database to register all resources. This raises two primary issues: preserving the autonomy of the participating organizations (including privacy/security and heterogeneity), and organizing the resource space in a manner suitable to all participants. A key problem is avoiding a strict hierarchical structure for the resource space, since that would limit the flexibility with which users could search for resources. Our approach allows the resource space organization to evolve in accordance with what resources exist, and what types of queries users make. To support this, a set of *agents* organize and search the resource space by constructing links between each other and the repositories of resource information, based on keywords that describe the contents of each repository. 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.

Table of Contents

1. Introduction

Given a diverse collection of resources registered with computer systems in a large scale, administratively decentralized environment, the *Resource Discovery* problem is to allow users to discover the existence of particular resources of interest. 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 1985]. 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.

The problem is also more general than what has been called the *attribute based naming* or *name discovery* problem [CCITT 1984, Schwartz 1987]. These problems are almost universally oriented only towards resources typically associated with computer systems, such as mailboxes and network services. In addition to such 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", "Michael Schwartz' electronic mail name", "TCP implementations for IBM mainframes", "inexpensive lawn mowers", and "movies playing in town tonight".[1]

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], MAP/TOP [Farowich 1986, Kaminski Jr. 1986], and ISDN [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)*[2] 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 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 differences between these approaches can be appreciated by considering the Network Information Center Whois service [Harrenstien, Stahl & Feinler 1985], which allows DARPA Internet users to register their name, address, telephone number, and electronic mail name with a centralized database that can be queried over the Internet. While well motivated, this database is not always helpful because many users are not registered, and those who are often do not update their information when it changes. For example, if an individual leaves one university for a position at another, both universities typically keep track of the employee's telephone and address information for some period of time, since doing so is relevant to their business. In contrast, the Whois service will receive the new information only if the individual remembers to inform SRI of the address change. Hence, it is often more reliable to contact either of these universities in looking for information about the individual than to use the Whois service. Of course, people do not always have enough information about the resources being sought to know the name of an organization that could help them in this fashion, so any mechanism we provide must be capable of functioning with less specific information than this.

[1] 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.

[2] 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).

Throughout this report we derive several ideas from observations about human societies. This does not imply that we intend to utilize artificial intelligence techniques. Rather, the purpose of these observations is to study the nature of successful systems of much larger scale than existing computer systems, and therefore derive ideas about how to organize a large scale resource discovery mechanism.

On a related note, when designing a system for use in a large scale distributed environment, there are a host of societal and technological influences, which we now consider.

1.1. Societal and Technological Influences

Three important societal trends [Naisbitt 1984] are relevant to the goals of this project: the transition from an industrial- to an information-based economy, decentralization, and the move from hierarchy to networking.

The transition to an information-based economy implies an explosion in the amount of information concerning resources. This increases the number of resources that could potentially be discovered through a automated mechanism, but at the same time makes it more difficult for humans to navigate through the information. Put another way, as the world becomes increasingly inundated with information, the structure provided for the information becomes more valuable than almost any particular piece of information. This point is the basis of the tremendously profitable and competitive telephone Yellow Pages business.

While even single administrations can grow large enough that this becomes an issue, the problem takes on new dimensions in a decentralized environment because of the privacy/security concerns of the participants, and because of the difficulty of reaching consensus on how the information should be organized. Since there can be no single administering entity to control the system, any mechanisms provided are more likely to acquire widespread use if they allow organizations to "patch into" the system (the UUCP network model prior to the addition of the "pathalias" protocol [Nowitz & Lesk 1978]) rather than having to register with some central authority (the DARPA Internet model [Stahl et al. 1984]).

The move to networking deserves more comment. Hierarchical organization has historically supported both formal social environments (such as bureaucracies) and technical environments (such as file systems). Yet, 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[3] file name */users/faculty/schwartz/pdp/monte/asynch/init.o* contains (from left to right) information about the file's disk location, creator's role, creator, research project, research subproject, algorithm variant, contents (*init* = "initialization routines"), and file type (*.o* = "object code").[4] 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. This point is underscored by Naisbitt in his popular book *Megatrends*: "In an information economy, rigid hierarchical structures slow down the information flow — just when greater speed and more flexibility are critically needed" [Naisbitt 1984].

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

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

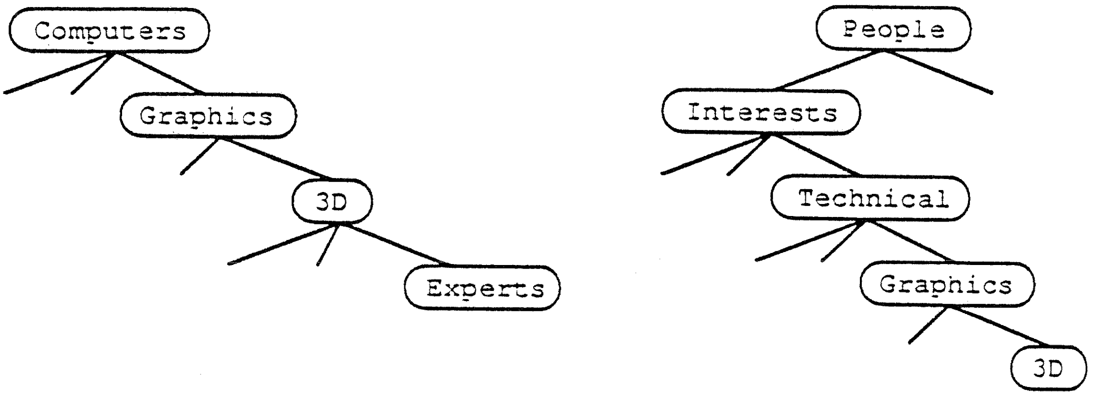


Figure 1: Conflicting Hierarchies

Unfortunately, hierarchy is the only way that scalable computer systems have been built to date [Lampson 1987]. The scalability of a hierarchy derives from its ability to divide management and processing into an arbitrarily large number of units (e.g., allowing disk and CPU requirements to be divided among multiple servers in a distributed file system). Yet, the observations made in the previous paragraphs indicate that no *single, static* hierarchy would suffice.

A final problem concerns overcoming the *inertia* of legal and social constraints on technology. An example is the current legal status of the Regional Bell Operating Companies (RBOCs). In particular, the AT&T Modified Final Judgement divesting AT&T of ownership of the RBOCs prohibits the RBOCs from providing information services. This effectively limits a large potential source of resource information, namely, telephone directories. Over time, we expect these restrictions to diminish, since they unreasonably limit the extent to which new technologies can be deployed. In fact, consideration is currently being given to allow the RBOCs to provide electronic Yellow Pages and videotex services [McPartlin 1988], and they were recently allowed to enter into information transmission and voice storage [McPartlin & Puttre 1988].

1.2. Scope

The resource discovery problem involves many issues, and has very broad potential applicability. It is worthwhile to point out several issues purposefully excluded from the scope of this research. First, we exclude non-research problems that could clearly be solved given enough effort (e.g., problems that would be addressed in a production quality system development). These problems are primarily in the areas of user interface and query format sophistication. Our prototype system will support only simple sets of string names for keywords, rather than more complicated specifications involving regular expressions, relational query languages, etc.

Second, we exclude research problems outside the central focus of the resource discovery problem. For example, accessing resources once they are discovered is outside the scope of the project. (This problem has been addressed before. For example, see [Heimbigner 1985, Litwin & Abdellatif 1986, Schwartz, Zahorjan & Notkin 1987]). Also, we do not intend to incorporate artificial intelligence techniques to decide how to parse the resource descriptions presented by users. Instead, our approach is to let humans perform such functions. The user might choose from among a set of options presented in a menu, to indicate, for example, that graphics is a computer science discipline, and that there are interest groups and magazines concerned with graphics.

Finally, we are not concerned with supporting resource discovery in every conceivable situation. Our techniques focus on large scale distributed environments, and are not well-suited to fine-grained searches, such as searching for suitable software tools or machine instructions in a programming environment. Similarly, we will not support general purpose text retrieval, since doing so requires more sophisticated access controls and bulk data transfer protocols than we consider.

1.3. Organization of this Report

The remainder of this report is organized as follows. In Section 2, we introduce the basic approach, indicating how resources are described, what the resource space looks like, and how the system is organized. Our approach raises two primary areas of focus: preserving the autonomy of the participating organizations, and organizing the resource space. In Section 3, we discuss the issues concerned with preserving autonomy, including heterogeneity and privacy/security. In Section 4 we discuss how the resource space is organized, detailing several protocols that allow the organization to evolve over time without human intervention. In Section 5, we discuss our efforts to demonstrate the validity of the design, including theoretical analysis, measurements of existing systems, and a multi-stage prototype effort. In Section 6, we discuss related work. Finally, in Section 7, we summarize the project.

2. Basic Approach

In this section we describe our basic approach to the resource discovery problem. We begin by considering the problem of how resources are categorized at the user level, and then lay the foundations for what the resource space will look like. Finally we discuss the system organization.

2.1. Resource Categorization and the Vocabulary Problem

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 large and complex systems requiring human typed inputs [Furnas et al. 1987]. Systems based on a standardized set of category descriptors have been successfully deployed (e.g., telephone Yellow Pages categories, ACM Computing Reviews Subject Descriptors, and library system categories), 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. When compact disk and other digital sound technologies entered the marketplace, the worlds of music equipment and information systems became closer, and a resource space reorganization became appropriate (a point 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. This approach is interesting because it is reminiscent of a *participatory democracy*, where individuals decide over time how the world is to be ruled, rather than having an inflexible centrally administered ruling body. 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. We will consider these problems further in Section 4.2.

2.2. Resource Space Structure

The structure of the resource space must somehow allow multiple resource categorizations. Since a single global hierarchy will not suffice, the next obvious approach would be to organize resources into multiple hierarchies. For example, one hierarchy could be concerned with geographical considerations, another with organizational considerations, and a third with functional considerations (i.e., organizing resources according to how they are used). However, constructing such hierarchies would require a large amount of human administrative effort, and the system would still be inflexible once the hierarchies were established. Another potential approach involves dynamic hierarchy reorganization. Lampson's name service [Lampson 1986] provides such a mechanism, but that work applies to a single hierarchy, and focuses on moving and joining individual subtrees, as opposed to reorganizing the basic structure of the naming tree. Banerjee et al. implemented a sophisticated object-oriented database schema reorganization mechanism that would allow more involved reorganizations [Banerjee et al. 1987], but the approach appears to be quite complex, and of questionable scalability.

Human social networks have evolved a structure that utilizes a more direct set of connections between related participants, a mechanism denoted *networking* in the popular jargon. According to Naisbitt, "There are three fundamental reasons why networks have emerged as a critical social form now: (1) the death of traditional structures, (2) the din of information overload, and (3) the failure of hierarchies ... We are all buried in the overload of information being generated and transmitted all around us. With networks to help, we can select and acquire only the information we need as quickly as possible. Networks cut diagonally across the institutions that house information and put people in direct contact with the person or resource they seek" [Naisbitt 1984].

The success of such networks is based on what has been called the "small world" phenomenon [Travers & Milgram 1969]. Consider a graph where nodes are people and edges represent one person's knowing another, as illustrated in Figure 2. 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, Naisbitt states that "Experienced networkers claim they can reach anyone in the world with only six interactions. It has been my experience, however, that I can reach anyone in the United States with only two — three at the very most — exchanges [Naisbitt 1984]." As another example, there is a mathematical game based on the co-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].

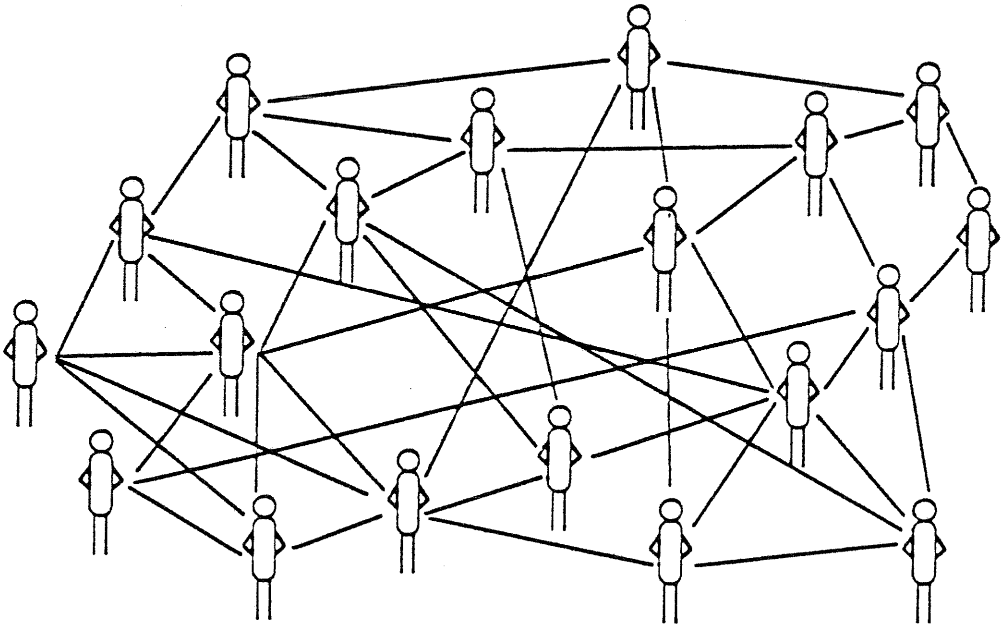


Figure 2: Human Social Network Graph

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 (what would be found in what one might call a *meta-RIR*) do not naturally exist.[5] 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 discovery.

While the network structure shown in Figure 2 is drawn as a general-looking graph, there is actually some notion of hierarchy present in real social networks. Within a collection of individuals there are often subsets of individuals who have more close, specialized relationships (e.g., task forces within large standardization committees). This is not a simple hierarchy, because any person could be a member of multiple hierarchies (representing different interests and roles), and there could be back pointers and cycles in the graph. Individuals can shortcut the hierarchy to find other related individuals in other hierarchies.

To provide a scalable resource discovery mechanism, we will utilize a similarly 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 3. Starting with the simple hierarchy of Figure 3A, 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 3B. 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 3C 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 4.1 we will consider techniques to reduce this redundancy.

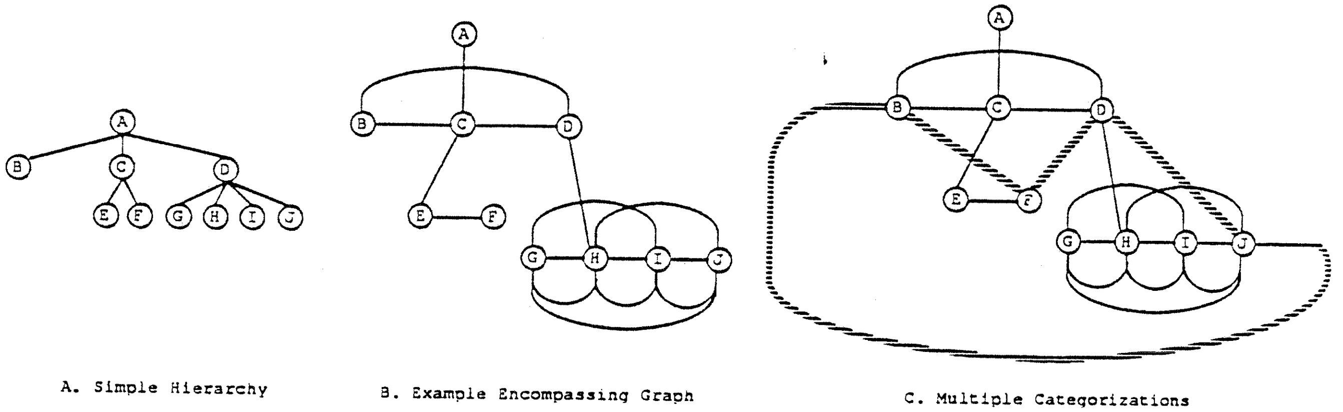


Figure 3: General Graph Construction

Using this general construction, a graph encompassing the hierarchies shown in Figure 1 is illustrated in Figure 4.

[5] 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.

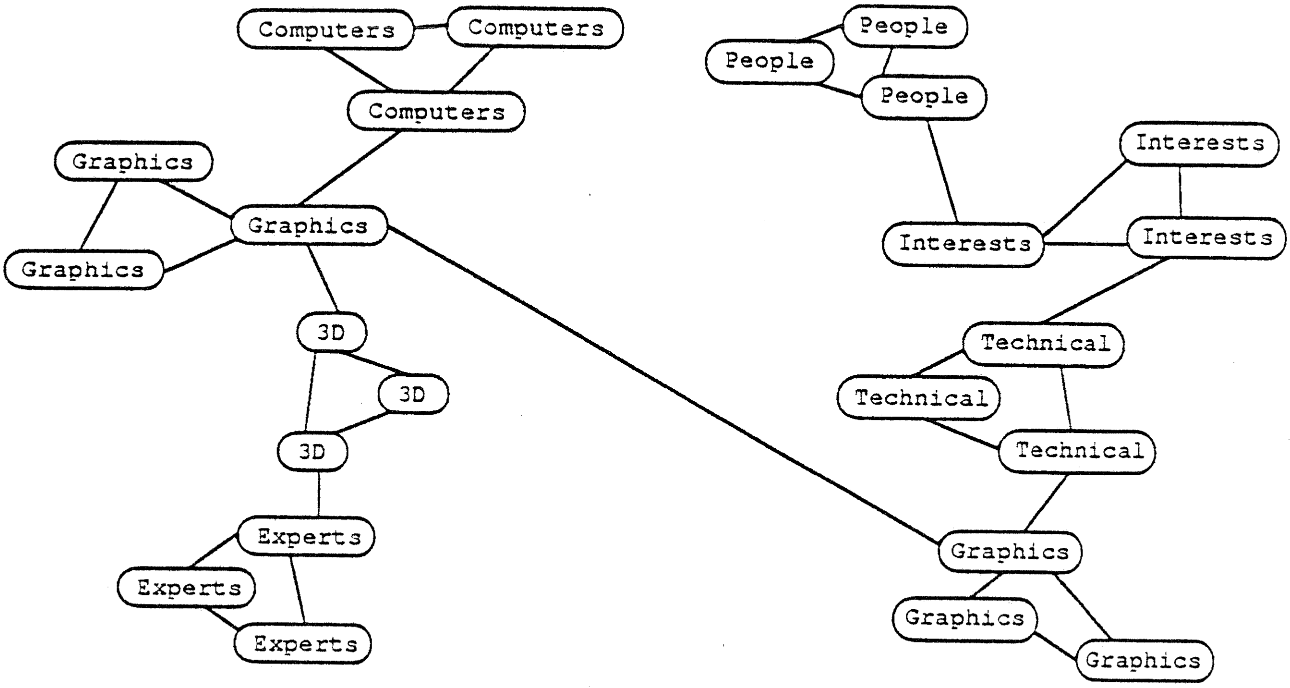


Figure 4: Example Graph Encompassing the Hierarchies of Figure 1

This style of graph organization is essentially a special case of the network database model [Ullman 1982], 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 network topology. For the purposes of the graph organization, we ignore the low-level issues associated with network topology, such as message routing.

2.3. System Organization

To provide a resource discovery mechanism, we introduce three new types of entities: a set of *agents* that dynamically construct links between the RIRs, a set of *brokers* that encapsulate the heterogeneity and access control concerns of the RIRs, and a set of *clients* that initiate resource searches on behalf of users by communicating with agents. This organization is illustrated in Figure 5.

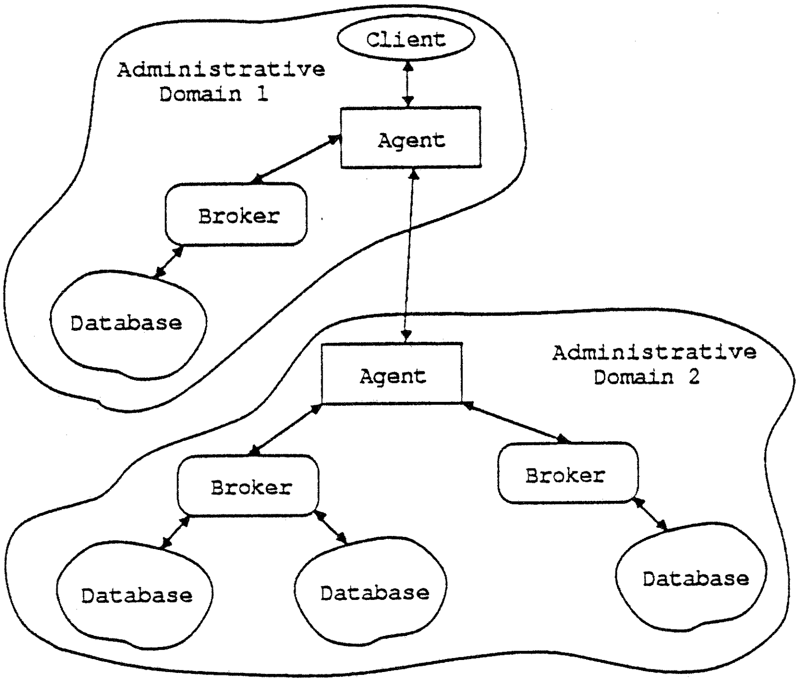


Figure 5: Basic System Organization

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 statistics, 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, searches are routed to other agents that are thought to be "closer" (in the sense of the knows-about relation) to the resources being sought. In other 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.

There are several points to notice about agents and brokers. First, each broker is a piece of special-purpose code that understands the access mechanisms and privacy concerns for a particular RIR, whereas agents are general purpose code 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. (For example, the agents could be administered by a government-monitored agency). 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. Finally, there is an adversarial relationship between brokers and agents — brokers try to restrict information flow, while agents try to increase it.

A useful abstraction for the concrete structure illustrated in Figure 5 is shown in Figure 6. This figure emphasizes the fact that brokers encapsulate RIRs, and agents organize them.

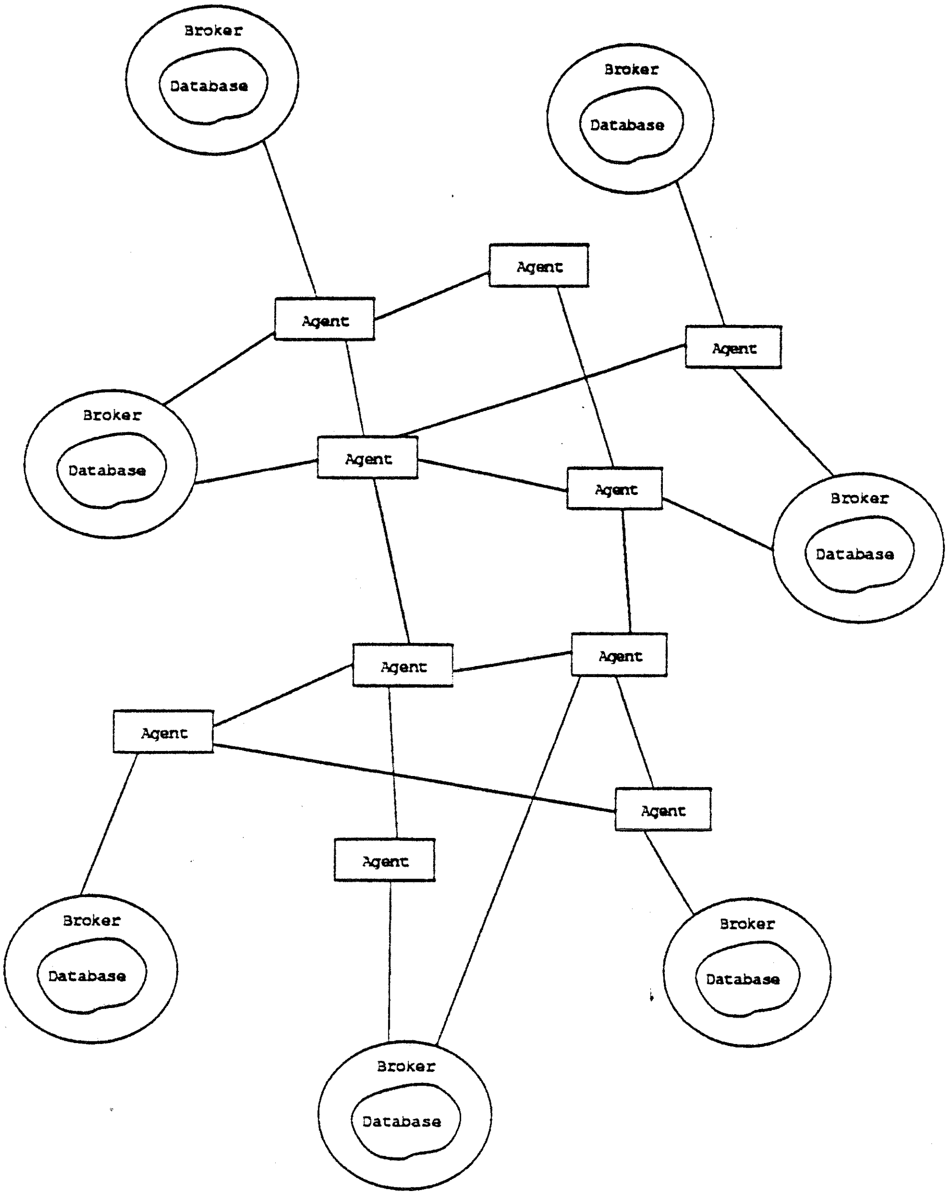


Figure 6: Agents Organizing Encapsulated Resource Information Repositories

In the following two sections we consider in detail the broker mechanisms for preserving autonomy, and the agent protocols for establishing and evolving the resource graph.

3. Preserving Autonomy

An important aspect of this project is preserving the autonomy of the participating systems. This involves two distinct issues: accommodating heterogeneity of the RIRs, and ensuring privacy/security of the constituent organizations.

3.1. Heterogeneity

To accommodate RIR heterogeneity, each broker must be able to translate generic keyword-based resource descriptions into whatever RIR-specific operations are needed, such as formatting a relational query for a local Ingres database. This requirement can be made to scale well in the *heterogeneous dimension* (i.e., may be applied to environments consisting of a large number of different types of systems) by utilizing techniques developed in my recent research into heterogeneous naming [Schwartz, Zahorjan & Notkin 1987], as part of the Heterogeneous Computer System (HCS) Project [Notkin et al. 1988]. In that work I recognized that a key difficulty of heterogeneous naming is the variety of data semantics and access protocols involved in using the information. Based on this observation, I developed a system that separated the management of the global name space from the understanding of the semantics and access protocols of the data in a fashion that minimized the difficulty of adding new naming systems and applications. In the current project, the same separation has been made: agents provide the resource space structure, and brokers are responsible for performing RIR-specific operations. We will use the Heterogeneous Remote Procedure Call mechanism [Bershad et al. 1987] (also developed as part of the HCS Project) to handle communication heterogeneity (e.g., varying network protocols and data representations) between the clients, agents, brokers, and RIRs.

3.2. Privacy/Security

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 (such as employee payroll information) or corporate competitiveness (such as product engineering diagrams). As mentioned briefly earlier, 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 are essentially abstract data types corresponding 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. Similarly, airline information operators will neither confirm nor deny whether a particular passenger boarded a particular airplane. In both examples, humans decide whether to allow a given query, based on privacy/security considerations. It would not be difficult to incorporate such considerations into brokers. Brokers can be built by information providers (or by a third party and inspected by the information providers) to ensure that they are trustworthy. Abstractly, this technique is similar to the "arms length" access control provided by electronic mail and DARPA Internet "anonymous FTP" file transfers, where users outside an administrative domain are allowed access to a limited subset of the information in that domain, through a restricted interface.

Brokers can also perform other privacy-enhancing functions that become necessary because of the automated nature of the information system. For example, even though queries asking what person has a particular telephone number are disallowed, it is possible for someone to program their computer to make queries about the phone numbers of a large number of people in succession, and compare each phone number to the number being sought. To counter such an approach, brokers could refuse to answer queries at a high rate from any individual.[6]

A relatively large amount of resource information can be shared using this simple concept of a broker. 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 would allow more information to flow within than across administrative boundaries. More generally, one could define an *Information Sharing Domain (ISD)* as a grouping within which resource information is freely shared.

[6] We do not intend to implement such functions in the prototype, since it is clear that they can be done. We include this description only to indicate the strength of the model.

An ISD may be smaller than an administrative domain. For example, within a company there may be some groups that do not freely share information with other groups. In this case, there should be ISDs surrounding each of the smaller groups within which information is freely shared, and a larger ISD surrounding them within which less information is shared. Beyond that is the outside world, in which even less (or no) information is shared.

Information may also fall within multiple ISDs. For example, in Figure 7 an employee in a software development group in a company could also be involved with a standards committee that crosses company boundaries. These separate roles each have their own ISD hierarchies. There could be some overlap, but in general different sharing restrictions apply to each ISD.

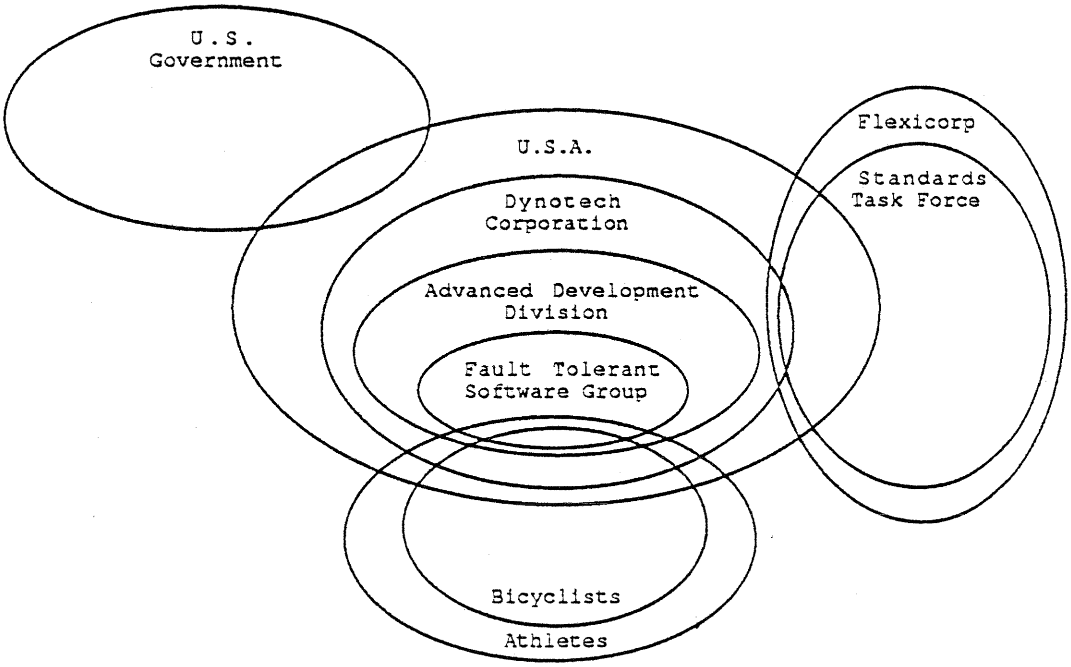


Figure 7: Information Sharing Domains

Extending the model this way clearly supports more general sharing behavior, but this generality comes at a fairly high price. First, the extended model requires an authentication mechanism, which is not required in the simpler model. Second, the extended model would involve more system complexity, e.g., a multiple inheritance hierarchy of objects, each of which performed "privacy filtering" for some ISD, plus a globally meaningful way to name and manipulate (add/delete, etc.) ISDs. It is not yet clear how seriously we will attempt to explore this more sophisticated model.

4. Establishing and Evolving the Resource Graph

The agent protocols are responsible for establishing and evolving the resource graph so that it may be efficiently searched according to multiple organizational schema. In this section we consider these agent protocols in detail.

Human social networks provide an interesting model for establishing and evolving a graph structure. In that domain there are a variety of mechanisms, including logically centralized directories (such as telephone Yellow Pages and Gale Research's Directory of Directories [Ethridge 1983]), administrative agencies (such as government offices), and business gatherings (such as conferences and cocktail parties). Business gatherings are particularly interesting, since they exhibit several properties that can be adapted for use in a networked resource discovery mechanism. First, they operate on a *locality of concern*: any particular gathering is attended by a only very specific group of people sharing some common concerns, such as technical interests at a conference or responsibilities at a business meeting. Second, event announcements are made through a variety of unreliable broadcasts to subsets of the population (e.g., ACM conference announcements mailed to members of the relevant Special Interest Groups). Third, popular concerns are likely to be represented at the gathering (e.g., persons interested in large scale file systems are usually present at any conference on distributed computing system). Fourth, only a small subset of all people sharing a particular concern attend any particular gathering, yet these people can often form the basis for helping each other meet many other people with related concerns. Fifth, when two people who share common concerns meet, they can substantially increase each others' awareness and knowledge by drawing conceptual connections based on both of their sets of experiences. These properties and the corresponding resource graph adaptations are summarized in Table 1. The adaptations are discussed in the following subsections.

Business Gathering Property	Resource Graph Adaptation
Locality of *concern*	Agent Specialization Subgraphs
Popular concerns likely to be represented	More efficient to search using popular keywords/organizational schema
Wide, sparse event announcement	*Sparse Diffusion Multicast* (See Section 4.1)
Few people attend, but form "networking" basis	A few agents could provide an effective basis for resource discovery
People sharing concerns increase each others' awareness	Agents exchange caches at their *specialization level*

Table 1: Business Gathering Properties and Resource Graph Adaptations

4.1. Bootstrapping the Graph Structure

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). While the term "bootstrapping" often connotes operations performed when a system is first started, here bootstrapping happens at various intervals, to update the graph. This "recurring bootstrap" approach is common to many dynamic systems. For example, Eden [Almes et al. 1985], distributed V [Cheriton & Mann 1988], and Sprite [Welch & Ousterhout 1986] all use some form of broadcast or multicast for bootstrapping their object location protocols. They rely on caching to improve performance for objects that move infrequently (the majority of cases), and resort to the broadcast bootstrap upon a cache miss.

In the case of the resource graph, bootstrapping 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. Put another way, for the mechanism to be scalable, knows-about chains of length greater than one must exist.

In designing graph bootstrapping protocols, an important question is what guarantees will be made about the exhaustiveness of searches. Because communication failures may occur and RIRs may be unavailable at any point in time, it is not possible to guarantee perfect service. 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 will 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 new primitive called a *sparse diffusion multicast*. The idea is that we want to pattern bootstrapping after the unreliable broadcast announcements made to subsets of the population who might attend a particular business gathering, but using full-scale broadcast will not scale. We therefore define a sparse diffusion multicast as follows. Given a set of N target agents, a message is sent to a subset of size $k + \log_b(N)$, selected at random, where k is a (tunable) constant that ensures that some minimum set of agents receive the transmission. While not strictly in accord with the laudable Grapevine maxim of scaling by adding more processors of fixed power (rather than more powerful processors) [Birrell et al. 1982], logarithmic growth requirements seem quite reasonable, depending on how b is chosen.

Using this primitive, we can construct a graph bootstrapping protocol wherein agents sparse diffusion multicast the keywords they know about to other agents. As with all probabilistic protocols, the success of this technique will rely 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. Figure 8 illustrates what the graph of Figure 3C might look like based on this probabilistic bootstrapping protocol. In this figure, node I is temporarily isolated, but over time the probabilistic bootstrapping mechanism should allow this node to join the rest of the graph. We consider the details of this bootstrapping protocol further in Section 4.2.

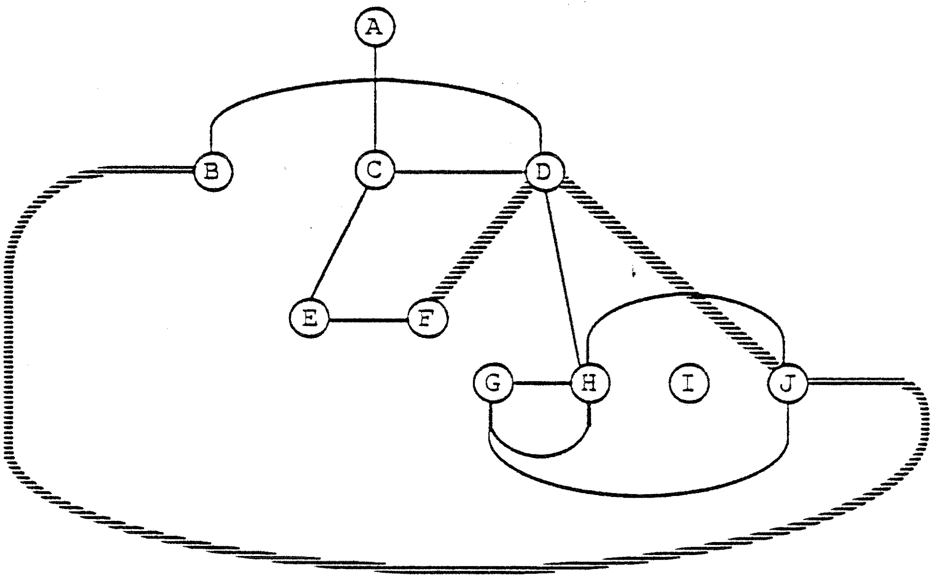


Figure 8: Example Result of Probabilistic Bootstrapping Corresponding to Graph of Figure 3C

If a search is requested at an agent that has no information about the named keywords, the client can ask the user to choose from a menu of keywords that are known by that agent. If this is ineffective, as a last resort the agent can use a sparse diffusion multicast to search for other agents that could help. Users could pay extra to increase the density of the sparse diffusion multicasts, and the fees could go towards supporting the agent network infrastructure.

Notice that sparse diffusion multicasts are qualitatively different from other forms of unreliable multicast that have been introduced in the past, including data link level broadcasts [Metcalfe & Boggs 1976], internet multicasts [Cheriton & Deering 1985], and directed broadcasts [Braden & Postel 1987]: with these other primitives, delivery is unreliable because the broadcasts can exceed node and link capacities, or there could be a node or link failure. In contrast, sparse diffusion multicasts are intended to deliver messages to only a small subset of the nodes, selected at random.

Implementing sparse diffusion multicasts could be difficult, because at any routing point one must ensure that messages traverse the links in proportion to both the sparse diffusion probability and the network connectivity at that point. For example, there are relatively few network lines between the U.S. and Europe, and hence any sparse diffusion multicast spanning hosts in both regions must traverse at least one of these paths. Ensuring this may be difficult without global routing information. One possible solution would be to register agents' existence with an administratively centralized *Agent Registry.* The implementation of this registry can be physically decentralized for the usual reasons of load distribution and availability (as is done with the root registries in the Domain Naming System [Mockapetris 1987].) Utilizing such an administratively centralized registry would not conflict with our goals, since it would not require centralized administration of the autonomous entities "outside of the network" (i.e., the RIRs, brokers, and clients). The Agent Registry is simply a possible means of supporting sparse diffusion multicasts without global routing information, and without requiring each agents to know about every other existing agent. The benefit of centralizing the agent repository is similar to that of providing a logically centralized idle node registry, as analyzed in [Theimer & Lantz 1988].)

4.2. Evolving and Refining the Graph

The graph 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. This keyword cache exchange will cause specialization subgraphs to develop over time.

However, this protocol does not establish knows-about chains of length greater than one (i.e., specialization subgraphs). As mentioned in Section 4.1, such chains must exist for the system to be scalable. Otherwise, when a client initiates a resource discovery search at an agent that does not know about the keyword being requested, that agent would always have to resort to using a sparse diffusion multicast to locate agents that know about the keyword. With chains of length greater than one, the agent could forward the search to another agent more closely concerned with the keyword being sought. We call this the *graph refining problem.* To accomplish this, at randomly selected intervals agents will 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 9. 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 1985].

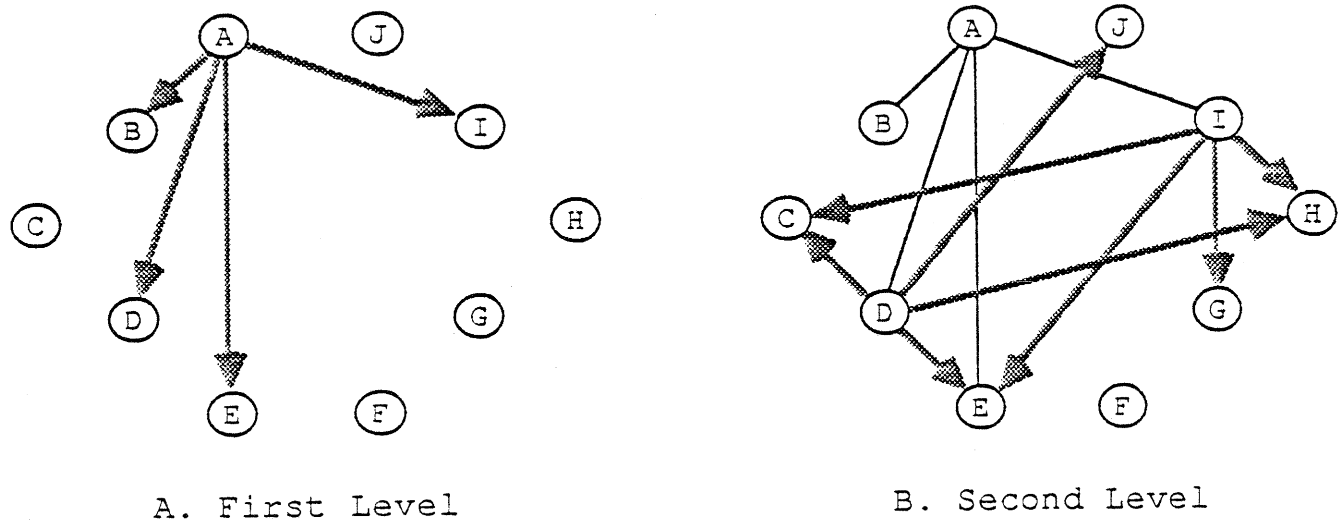
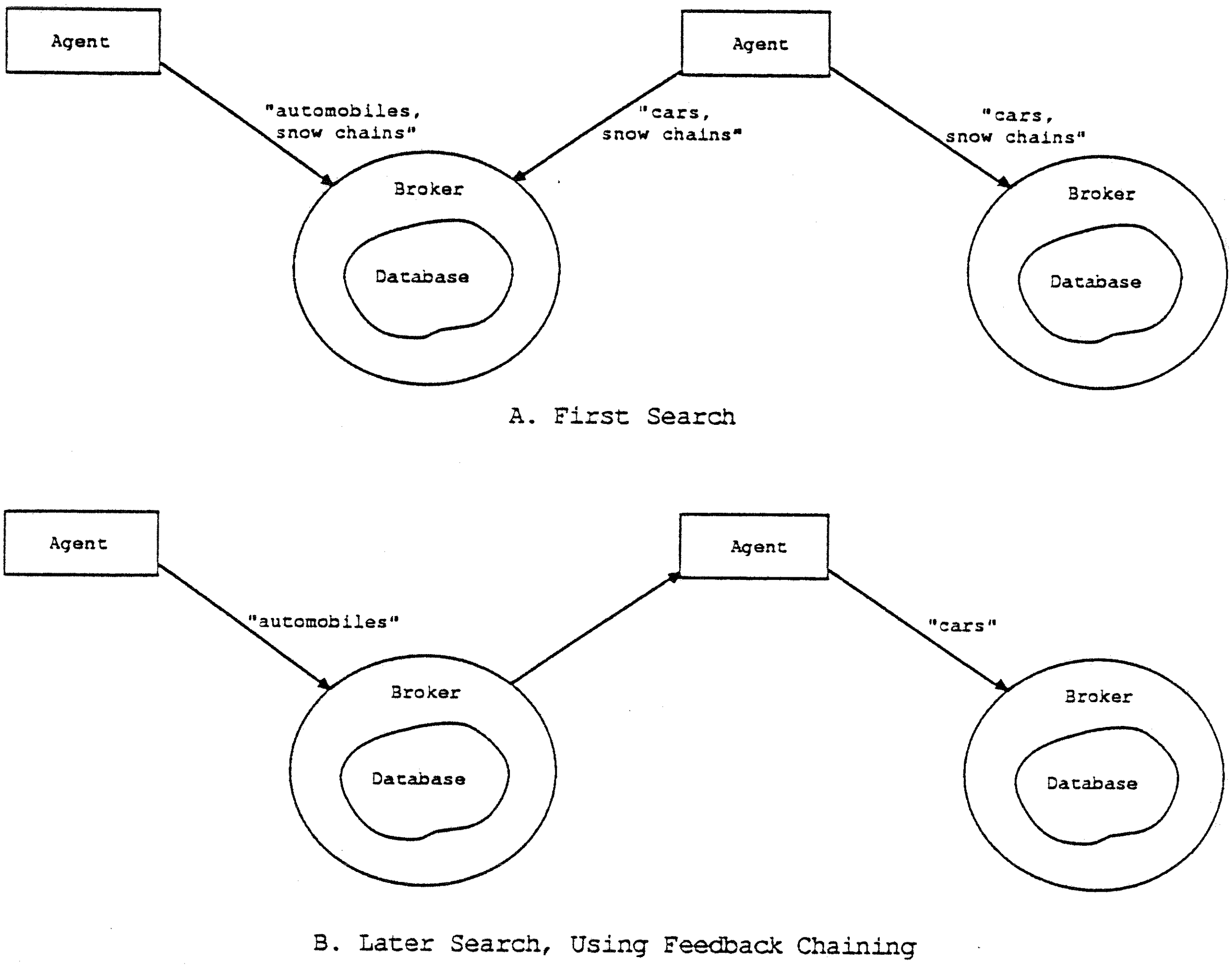


Figure 9: Multiple Hop Chain Construction

Another important issue is the vocabulary problem (see Section 2.1). A simple solution to this problem would be to provide an online thesaurus, so that if a keyword is not recognized, the user could be presented with a choice of related keywords that are present. Because there are few research issues involved in doing this, we will not provide this capability in the initial prototype. However, Mozer introduces another technique that could prove useful for both refining the graph and attacking the vocabulary problem. His system makes use of the internal structure of a database to infer relationships among items, to help overcome incompleteness and imprecision in the database and requests made to the database. He describes the system as follows:

> The user activates a set of descriptors. These descriptors activate a set of documents. The documents in turn activates new descriptors, which will activate some new documents as well as reinforce the activation of already active documents, and so on. Activation continually flows from descriptors to documents and vice-versa. This flow of activation allows descriptors in a query to indirectly suggest other descriptors that may be useful in the document search, and it allows active documents to indirectly suggest other documents [Mozer 1984].

We could provide a related mechanism by having each broker keep track of which agents made queries at that broker, as well as the keywords they used. Then, when other queries are made using these keywords, the broker could return to the requesting agent an indication of the other agents that made similar queries. The agents could then contact each other and form new specialization subgraphs, and could infer that some of their keywords were related. This approach is illustrated in Figure 10. In this figure, the fact that one agent searched for "automobiles, snow chains" and another searched for "cars, snow chains" (part A) could be recorded by the contacted brokers. This information could be used in later searches (part B) to indicate that "automobiles" might be related to "cars" (the vocabulary problem), and to join the two agents into a specialization subgraph based on their common concerns (the graph refining problem).



A. First Search

B. Later Search, Using Feedback Chaining

Figure 10: Feedback Chaining to Refine Graph and Link Related Keywords

A final technique for evolving the graph involves caching *incidentally acquired information.* This notion is similar to a concept introduced by Bates [Bates 1985], and is roughly analogous to how humans find resources by recalling a past experience that, at the time, they did not know would become useful for a future resource search. We can provide such a facility in our system by returning more information in responses that was requested, so that agents along the return path can cache this information. The selection criteria for this additional information is not yet clear, but will probably be some function of the keyword categories shared by the agents that contact each other.

Caching incidentally acquired information has a different goal from most caching mechanisms. Caching is usually oriented towards improving the performance of frequently referenced entities; the cache maintains some type of working set, based on the cache replacement policy. In contrast, incidentally acquired knowledge is cached in case it *might* be useful later. This type of caching could be especially useful if very large caches are available, either in main memory, or possibly in large scale optical storage. In addition, caching incidentally acquired information can support information that is available only on-the-fly, such as AP news broadcasts and electronic bulletin board transmissions.

4.3. Scalability

Resource discovery presumably happens infrequently relatively to accessing resources. Hence, while performance should be reasonable, it is not critical. However, scalability is very important. In this subsection we consider various issues regarding the scalability of the resource discovery mechanisms.

There are several scalability issues that can be solved by imposing some simple limits. For example, if a user specifies a very general query, the response could be quite large, and limiting the response to some fixed maximum size (with a notification that this has been done) is necessary. Similarly, there must be a limit on the amount of information cached at any particular agent.

A more difficult problem arises with the number of keywords in use. Since brokers and clients are free to choose keywords as they please, the number of keywords in use (the vocabulary problem) could become a scalability concern. It is hoped that the cache aging protocols discussed in Section 4.2 will ameliorate this problem. The number of different *types* of resources will probably not be a scalability issue, because of what might be called *cultural commonality*, the tendency for some basic set of resource types to be instantiated in many different communities. For example, consider the categories in current telephone Yellow Pages: the categories in each major metropolitan telephone book overlap to a large extent, because there are a common set of goods and services that most cities support. Of course, this example is biased because the telephone books in the U.S. were originally controlled by a single corporation (pre-divestiture AT&T). Yet, a similar advantage could be obtained in our resource discovery mechanism by installing an initial set of keywords in the agents. This would simplify the user view without constraining the system to stay fixed at the original categories, since users and brokers are free to introduce new categories at any time. How well this level of commonality extends beyond a single society is an open question.

Another scalability issue concerns consistency of the resource information. Our solution to this problem is that agents will treat all graph edges as cached hints [Lampson & Sturgis 1983]. Only edges, and not actual resource information, will be cached. Therefore, all cached information will be detectable as stale upon use.

5. Research Efforts

The preceding sections raised many issues and posed many potential solutions. Clearly, a large amount of work can be put into research and development of these ideas. To limit this level of effort, we will focus on demonstrating the feasibility of the more novel aspects of the system, especially the protocols for establishing and evolving the resource graph.

We will begin with some mathematical analysis of the graph algorithms. To perform this analysis, however, we must propose a model for what typical resource graphs look like. This model will be based on measurements of an existing resource graph structure (internet electronic mail). We will then build a prototype system that will allow us to verify some aspects of the proposed mechanisms, and to gather measurements as a basis for simulations and analytic models that we will use to estimate how well our design would work in a large scale network. Each of these parts of the project are described in the subsections below.

5.1. Mathematical Analysis

To help explore the networked approach to resource discovery, we plan to perform some graph-theoretic analysis about the agent knows-about graph. There are several issues involved. First, we would like to establish an understanding for how the number of resources that can be reached in N hops increases as a function of N. In a tree of uniform branching factor b, this function would grow simply as b^N. Establishing such a relationship for the more general graph structure we have proposed will provide some empirical justification for the "small world" phenomenon that motivated parts of our approach.

We also intend to study the complexity of using parallel searches to locate resources, to help us make scalability predictions. This could provide a more direct measure of the effectiveness of our approach, because a complexity model based on the aforementioned growth function admits a trivial solution wherein an "expanding ring" broadcast is issued, and each time the ring diameter is increased by one, all agents at that distance begin searching. This technique is clearly inappropriate for a large scale network. Instead, we must capture the need for high agent utilization efficiency.

We also intend to analyze the efficiency of the proposed probabilistic bootstrapping algorithms. We will begin by considering the efficiency and robustness of the sparse diffusion multicast mechanism, i.e., how quickly the graph edges are instantiated as a function of the number of iterations of the algorithm. Drezner and Barak consider a related problem, where at each time unit, each node sends a message to a randomly selected node. They showed that it is possible to scatter information to a set of n nodes in $(1 + \ln 2)\log_2 n$ steps with probability approaching 1 for large values of n [Drezner & Barak 1984]. This work indicates that probabilistic information scattering techniques can be effective, but the algorithm they utilize does not take into account any structure in the graph (e.g., specialization subgraphs), which could potentially render a more efficient algorithm (i.e., one in which only a few nodes exchange information at each step).

5.2. Measurements of Internet Electronic Mail Graph

To carry out the analyses of Section 5.1, we need more information about the structure of typical resource graphs: they are certainly more structured than fully general graphs, since they contain nested groupings of related resources (specialization subgraphs). Analysis based on fully general graphs is unlikely to yield useful bounds for our purposes. For example, a recent paper by Megiddo et al. considers the complexity of searching a graph for a fugitive that always knows the location of a set of captors [Megiddo et al. 1988]. It is shown that determining the minimum number of captors needed to guarantee success is linear in the number of nodes for a tree, but the problem is NP-complete for a general graph. While this problem is only marginally related to the resource discovery problem, it is evident that without information about the structure of the graph, determining the complexity of searching the graph can only be bounded very loosely.

By measuring some existing systems, we hope to build a model of realistic knows-about relation graphs, incorporating such features as overlapping knowledge and differential importance of resources. Additionally, this information will be interesting in itself, because it will provide some insight into the societal organization in a large scale environment. The analogous data in human social networks would be very difficult to gather. To our knowledge, studies attempting to characterize societies this way have been small and rather inconclusive [Boissevain 1974, Travers & Milgram 1969].

To carry out these studies, we are beginning to gather data about current electronic mail usage at a collection of institutions that span geographical, functional, and specialization boundaries, i.e., universities and companies working on a variety of research, education, and product development projects around the world. Using data about electronic mail usage is a reasonable place to start because mail names refer to people, which are an example of one type of resource in our model.

To carry out this study, 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 will use a script that collects these lines in UNIX sendmail logs, and then mails the data to us. By comparing the data at these institutions we can determine how much overlap exists between various institutions, as a function of geographical and organizational separation. We can also use this data to compute the diameter of the knows-about relation graph for this sample population. We can also determine the number of different administrative domains represented by these names, to get some feeling for the "reach" of an average administrative domain, i.e., how many other administrative domains are known from an average administrative domain.

Of course, this study presents privacy concerns. While the script never sees the actual contents of messages (since they are not present in sendmail logs), the data we gather can potentially be used for traffic analysis that could provide us with information of a more sensitive nature than that used in this study (e.g., the information might indicate something about a joint product development effort between two companies). Unfortunately, there is little we can do to ensure institutions that we will not exploit their data for purposes other than this study. We nonetheless hope to find enough people who will trust our intentions that we can do a statistically meaningful study.

At this point we have asked approximately fifty institutions (mostly U.S. universities) to participate in this study. While we have not received all of the responses yet, many institutions have agreed to participate. The major problem at institutions that have declined has (predictably) been security concerns, particularly from industrial institutions. The study may be lopsided on the side of universities, since they are typically not quite as concerned with these security issues. Still, we have received permission from quite a few institutions in academia, research industry, and product-development industry. In fact, enough institutions have agreed to participate that we are now studying how best to analyze the voluminous data that we expect. Since many graph computations are too costly to run on the full data we expect to collect, we are currently considering statistical techniques for analyzing the data.

5.3. Staged Prototype Effort

We are currently beginning a prototype effort to experiment with the ideas discussed with this report. The prototype effort is broken into five stages. In the first stage, we will build and experiment with only the agent protocols.

In the second stage, we will construct a system with brokers that access mock RIRs, to test some of the simple privacy concepts discussed in Section 2. These mock RIRs will be 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 this way, including ".plan" files, /etc/passwd files, /usr/lib/aliases files, user files containing hobby member 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. We are currently seeking temporary experimental access to a collection of RIRs (including telephone directories from several long distance, regional, and university networks, and some corporate product line catalogs). Early efforts to gain access to such real systems have (predictably) been mostly unsuccessful, either because of legal restrictions (as in the case of the Regional Bell Operating Companies), security concerns (profitability issues at various companies), privacy concerns (various individuals at universities), or lack of networking capabilities (e.g., local department stores). We hope an industrial development effort can overcome some of these problems, and are currently exploring the possibilities for collaboration on one such effort.

In a possible fifth stage, we may implement the more sophisticated information sharing mechanism discussed in Section 2. For this stage, we would need to find a reasonable "off-the-shelf" authentication service, as well as an object based distributed system with multiple inheritance. (We are currently considering several such systems).

5.4. Prototype Experiments, Measurement, and Modeling

Once we have some of the prototype development complete, we plan to collect statistics to aid our understanding of the agent networking and caching protocols. Using these statistics, we will construct a probabilistic model to help determine system scalability.

There are also several questions to resolve for which experimental deployment of a working system seems the best approach. The first question concerns accuracy of user-supplied information: given information known to be valid at some time in the past, how can the system help to determine more recent information. For example, in trying to locate a colleague whose whereabouts are only known from five years ago, can we find enough forwarding pointers to track that person down? A related question is how well the system will deal with ambiguity, such as two users with similar interest profiles and the same last name.

Another question concerns search strategies. How does one determine that a search is complete? How much effort should be put into individual resource searches?

Finally, a longer-term question concerns cross-cultural applicability: how well will the notion of "cultural commonality" (introduced in Section 4.3) apply to an international network? Answering this question would be difficult in a research prototype, since it requires very large scale system use.

6. Related Work

In this section we consider a wide variety of techniques and systems related to the problem of resource discovery, and to our particular approach. While many of these mechanisms help users to navigate through their respective resource spaces, they are problematic for use in the environment we envision for two reasons. First, they provide no computationally tractable way of searching extremely large, distributed spaces. Second, they require centralized administration (even if the data are physically distributed), preventing the individual subsystems from evolving autonomously; any information derived from outside sources would be reregistered in such a system. This is important because a model based solely on reregistration will suffer from consistency problems at two levels: information about individual resources can become out-of-date if the resources change, and information about the existence of resources can become out-of-date if new resources are added or old ones are removed.

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 centralized file systems, such as the UNIX file system [Thompson & Ritchie 1978]. These systems typically provide facilities to browse through the file name space by reading the entire contents of a single directory, plus tools on top of this mechanism to allow regular expression based pattern matching, interactive file name completion, and exhaustive recursive searching.

Analogous facilities have been proposed for distributed systems, although it is clearly impractical to support exhaustive searches in any sizable distributed system. A well-known distributed directory browsing mechanism is the proposed X.400 directory service [IFIP 1983]. Other directory services include UNIX finger [Thompson & Ritchie 1978], the Network Information Center Whois service [Harrenstien, Stahl & Feinler 1985], and the CSNET name service Whois facility [Solomon, Landweber & Neuhengen 1982]. Each of these services supports queries containing ambiguous strings, responding with the set of all matching names.

Greenspan and Smolensky take an approach more closely related to our work [Greenspan & Smolensky 1983]. They suggest using descriptive means for naming entities in an operating system, by allowing multiple organizational hierarchies, so that each hierarchy can capture particular semantics in its organization, and the system can utilize this knowledge to help users find named entities more easily. However, they do not consider the issues of distribution and scalability.

Electronic Yellow Pages Mechanisms

Several systems have been built to provide administratively centralized global "Yellow Pages" databases of network services. The Sun Yellow Pages [Weiss 1985] registers Sun RPC-based network services. 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 [Peterson 1987]. Their system supports various boolean and relational combinations of attributes, providing a global database model of Yellow Pages for naming network services in an internet environment [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. Common systems include bibliographic database systems (such as INSPEC [Schmittroth 1983]) and online information services (such as CompuServe [CompuServe 1986]). These systems require centralized administration, rather than supporting access to some conglomerate of information sources. In addition, their effectiveness is typically predicated on the degree to which all aspects of document content are recognized and represented in the index. Providing such a precomputed index for the conglomerate environment we envision would be infeasible because of the sheer amount of information that would have to be represented, and the rate the index would have to be updated.

The Telesophy Project at Bell Communications Research is more closely related to the current project. That project has the goal of allowing access to any online information in the world from a user's workstation [Shatz 1987]. In the prototype implementation, information is gathered from a variety of sources and archived locally, unlike our resource discovery design, which accesses information where it natively exists.

There have been several efforts to utilize more sophisticated techniques to increase human effectiveness in using information retrieval system. 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 conveniently in order to scan a document non-linearly [Conklin 1987]. 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 (the vocabulary problem) [Streeter & Lochbaum 1988]. The approach is to group keywords according to meaning in a k-dimensional space (where k is fairly large, like 100), and then extract documents based on a geometrically interpreted distance metric.

Naming in Heterogeneous Systems

In my recent research into heterogeneous naming, I explored the issues involved in providing a global name service for evolving systems composed of a heterogeneous collection of subsystems [Schwartz, Zahorjan & Notkin 1987]. Providing a name service in such an environment is difficult because each of the systems being integrated typically uses its own name service. Two aspects of that work are related to the current work. First, my Heterogeneous Name Service made use of, rather than replaced, name services and associated data already existing in the individual system components. Analogous to the current project's approach, the major advantage of this approach is that it allows the underlying subsystems to evolve independently of the global name service, while still reflecting this evolutionary change to the clients of the global name service. Second, I recognized that a key difficulty of heterogeneous naming is the variety of data semantics and access protocols involved in using the information. Based on this observation, my system separated the management of the global name space from the understanding of the semantics and access protocols of the data in a fashion that makes adding new naming systems and applications as easy as possible. A similar separation has been made in the current project, in the sense that agents provide the resource space structure, and brokers are responsible for performing the RIR-specific operations.

Heterogeneous Database Systems

The database community has been working on integrating heterogeneous systems for several years. The goal is to allow users to read and manipulate data from several independently created/administered databases, each of which has different data formats, access protocols, and manipulation languages. The methods used for accessing these databases vary from multilevel translation (between query languages, data formats, etc.) [Templeton et al. 1986] to meta-query languages that allow the user to name various databases and define relationships between them, for manipulation, privacy, and equivalence dependencies [Litwin & Abdellatif 1986]. These schemes support joining of data in different database schemas, and broadcasting of user intentions over a number of database schemas with varying naming rules for data with similar meanings.

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 [Feldman 1981]) 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.

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 structure. Such a system 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 have designed 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. We are beginning an internet electronic mail usage measurement study to help characterize typical resource graphs to aid in analyzing the scalability of these approaches.

While the graph protocols are the central focus of this research, privacy is also an important issue. For this purpose we have developed a simple model for the early stages of our prototype development. We are currently considering the implications of a more sophisticated model that could extend the applicability of the resource discovery 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 and the vocabulary problem. 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. And last but certainly not least, Tom Gasaway, Carl Koch, and David Wood have been involved in many project design discussions over the past several months, and are currently beginning 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.

[Almes et al. 1985]

G. T. Almes, A. P. Black, E. D. Lazowska and J. D. Noe. The Eden System: A Technical Review. *IEEE Trans. Software Eng.*, SE-11(1), pp. 43-59, Jan. 1985.

[Banerjee et al. 1987]
J. Banerjee, W. Kim, H. Kim and H. F. Korth. Semantics and Implementation of Schema Evolution in Object-Oriented Databases. *Proc. of SIGMOD Int. Conf. on Management of Data*, pp. 311-322, San Francisco, CA, May 1987.

[Bates 1985] M. J. Bates. An Exploratory Paradigm for Online Information Retrieval. In *Intelligent Information Systems for the Information Society*, pp. 91-99, North-Holland, New York, NY, 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.

[Birrell et al. 1982]
A. D. Birrell, R. Levin, R. M. Needham and M. D. Schroeder. Grapevine: An Exercise in Distributed Computing. *Commun. ACM*, 25(4), pp. 260-274, Apr. 1982. Presented at the Eighth ACM Symp. Operating Syst. Prin., Pacific Grove, CA, Dec. 1981.

[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.

[Braden & Postel 1987]
R. Braden and J. Postel. Requirements for Internet Gateways. RFC 1009, USC Information Sci. Institute, June 1987.

[CCITT 1984]
CCITT. *Recommendation X.400, Message Handling Systems: System Model - Service Elements*. CCITT, Study Group 5/VII, Oct. 1984.

[Cheriton & Deering 1985]
D. R. Cheriton and S. E. Deering. Host Groups: A Multicast Extension for Datagram Internetworks. *Proc. Ninth Data Communications Symp.*, Sep. 1985. Appears as Computer Communications Review 15(4), Sep. 1985.

[Cheriton & Mann 1988]
D. R. Cheriton and T. P. Mann. Decentralizing: A Global Naming Service for Efficient Fault-Tolerant Access. To appear, *ACM Trans. Comput. Syst.*, 1988. Available as Tech. Rep. CSL-TR-293, Comput. Sci. Dept., Stanford Univ.

[CompuServe 1986]
CompuServe. *CompuServe Information Service*. CompuServe Inc., Oct. 1986. Sales literature.

[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. *Journal of Telecommunication Networks*, 3(3), pp. 194-209, 1984.

[Drezner & Barak 1984]
Z. Drezner and A. Barak. A Probabilistic Algorithm for Scattering Information in a Multicomputer System. CRL-TR-15-84, Univ. of Michigan, Computing Research Lab., 1984.

[Ethridge 1983]
J. M. Ethridge, editor. *The Directory of Directories*. Information Enterprises, 1983. Gale Research Company, distributers.

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

1986.

[Feldman 1981]
J. A. Feldman. A Connectionist Model of Visual Memory. In G. E. Hinton and J. A. Anderson, editors, *Parallel Models of Associative Memory*, pp. 49-81, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1981.

[Fowler 1985]
R. J. Fowler. Decentralized Object Finding Using Forwarding Addresses. Ph.D. Diss., Tech. Rep. 85-12-1, Comput. Sci. Dept., Univ. Washington, Seattle, WA, Dec. 1985.

[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.

[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. RFC 954, Oct. 1985.

[Heimbigner 1985]
D. Heimbigner. *Diverse: A User Agent for Accessing External Databases.* Dept. Comput. Sci., Univ. Colorado, Boulder, CO, Nov. 1985. Ealier version available as CU-CS-306-85, Nov. 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.

[Lampson & Sturgis 1983]
B. W. Lampson and H. E. Sturgis. Hints for Computer System Design. *Proc. 9th ACM Symp. Operating Syst. Prin.*, pp. 33-48, July 1983.

[Lampson 1986]
B. W. Lampson. Designing a Global Name Service. *Proc. 5th ACM Symp. Principles Distr. Comput.*, pp. 1-10, Aug. 1986.

[Lampson 1987]
B. Lampson. *A Naming Service for a World-Wide Computer Network.* Digital Equipment Corporation, Syst. Research Center, Nov. 1987. Videotape Lecture from the University Video Communications Distinguished Lecture Series on Industry Leaders in Computer Science.

[Litwin & Abdellatif 1986]
W. Litwin and A. Abdellatif. Multidatabase Interoperability. *IEEE Computer Magazine*, 19(12), pp. 10-18, Dec. 1986.

[McPartlin 1988]
J. P. McPartlin. RBOC Plan Meets Resistance. *InformationWEEK*, pp. 20, Feb. 29, 1988.

[McPartlin & Puttre 1988]
J. P. McPartlin and M. Puttre. RBOC Win Doesn't Sway MIS. *InformationWEEK*, pp. 16, Mar. 14, 1988.

[Megiddo et al. 1988]
N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson and C. H. Papadimitriou. The Complexity of Searching a Graph. *J. ACM*, 35(1), pp. 18-44, Jan. 1988.

[Metcalfe & Boggs 1976]
R. M. Metcalfe and D. R. Boggs. Ethernet: Distributed Packet Switching for Local Computer Networks. *Commun. ACM*, 19(7), pp. 395-404, July 1976.

[Mockapetris 1987]
P. Mockapetris. Domain Names - Concepts and Facilities. RFC 1034, USC Information Sci. Institute, Nov. 1987.

[Mozer 1984]
M. C. Mozer. Inductive Information Retrieval Using Parallel Distributed Computation. ICS Report Number 8406, Institute for Cognitive Science, Univ. of California, San Diego, June 1984.

[Naisbitt 1984]
J. Naisbitt. *Megatrends — Ten New Directions Transforming Our Lives.* Warner Books, New York, NY, 1984.

[Notkin et al. 1988]
D. Notkin, A. P. Black, E. D. Lazowska, H. M. Levy, J. Sanislo and J. Zahorjan. Interconnecting Heterogeneous Computer Systems. *Commun. ACM*, 31(3), pp. 258-273, Mar. 1988.

[Nowitz & Lesk 1978]
D. A. Nowitz and M. E. Lesk. *A Dial-Up Network of UNIX Systems.* Bell Laboratories, Murray Hill, NJ, Aug. 1978.

[Peterson 1987]
L. L. Peterson. An Architecture for Naming Resources in a Large Internet. Tech. Rep. 87-7, Dept. Comput. Sci., Univ. Arizona, Tucson, AZ, Feb. 1987.

[Schmittroth 1983]
J. Schmittroth, editor. *Encyclopedia of Information Systems and Services.* Gale Research, 1983.

[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 1987]
M. F. Schwartz. Naming in Large, Heterogeneous Systems. Ph.D. Diss., Tech. Rep. 87-08-01, Comput. Sci. Dept., Univ. Washington, Seattle, WA, Aug. 1987.

[Shatz 1987] B. R. Shatz. Telesophy: A System for Manipulating the Knowledge of a Community. *Proc. Globecom '87*, pp. 1181-1186, 1987.

[Solomon, Landweber & Neuhengen 1982]
M. Solomon, L. H. Landweber and D. Neuhengen. The CSNET Name Server. *Computer Networks*, 6(3), pp. 161-172, July 1982. Presented at the Seventh Berkeley Workshop on Distributed Data Management and Computer Networks, Lawrence Berkeley Laboratory, Pacific Grove, CA, Feb. 1982.

[Stahl et al. 1984]
M. K. Stahl, G. Sherwood, N. Fischer and S. Romano, editors. *DDN Directory.* Network Information Center, SRI Int., June 1984.

[Streeter & Lochbaum 1988]
L. A. Streeter and K. E. Lochbaum. An Expert/Expert-Locating System Based on Automatic Representation of Semantic Structure. *Proc. of the 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.

[Templeton et al. 1986]
M. Templeton, D. Brill, A. Chen, S. Dao and E. Lund. Mermaid - Experiences with Network Operation. *Proc. 2nd IEEE Int. Conf. Data Eng.*, pp. 292-300, Feb. 1986.

[Theimer & Lantz 1988]
M. M. Theimer and K. A. Lantz. Finding Idle Machines in a Workstation-Based Distributed System. To appear, *Proc. 8th IEEE Int. Conf. Distrib. Comput. Syst.*, June 1988.

[Thompson & Ritchie 1978]
K. Thompson and D. M. Ritchie. *UNIX Programmer's Manual.* Bell Laboratories, Murray Hill, NJ, 1978. Seventh Edition.

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

[Ullman 1982]
J. D. Ullman. *Principles of Database Systems.* Computer Science Press, Rockville, MD, 1982. Second Edition.

[Weiss 1985]
P. Weiss. *Yellow Pages Protocol Specification.* Sun Microsystems, Inc., 1985.

[Welch & Ousterhout 1986]
B. Welch and J. Ousterhout. Prefix Tables: A Simple Mechanism for Locating Files in a Distributed System. *Proc. 6th IEEE Int. Conf. Distrib. Comput. Syst.*, pp. 184-189, May 1986.