Cooperative Information Gathering: A Distributed Problem Solving Approach

Tim Oates, M. V. Nagendra Prasad and Victor R. Lesser
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA.
{oates,nagendra,lesser}@cs.umass.edu

Abstract

We contrast two approaches to the problem of information gathering that may be characterized as *distributed processing* and *distributed problem solving*. The former is characteristic of most existing information gathering systems, while the latter is central to research in multi-agent systems. We examine features of complex information carrying environments and the information gathering task that demonstrate both the utility of viewing information gathering as distributed problem solving and difficulties with viewing it as distributed processing. We propose a new approach to information gathering based on the distributed problem solving paradigm and its attendant body of research in multi-agent systems and distributed artificial intelligence. This approach, called Cooperative Information Gathering, involves concurrent, asynchronous discovery and composition of information spread across a network of information servers. Top level queries drive the creation of partially elaborated information gathering plans, resulting in the employment of multiple semi-autonomous, cooperative agents for the purpose of achieving goals and subgoals within those plans. The system as a whole satisfices, trading off solution quality and search cost while respecting user-imposed deadlines. We also survey current work on distributed and agent-based approaches to information gathering.

1 Introduction

Recent years have seen an explosion in the amount of information available in electronic form, forcing the developers of information acquisition systems to re-evaluate their model of the world. Vast amounts of electronic information are available at a multitude of sites to anyone with access to the Internet. Some of that information is free, some of it is available for a price, and all of it requires time and computational resources to search and access. Early information retrieval (IR) systems assumed that users would supply both a query and corpus (data source) against which the query is to run. Even though the user may have access to multiple corpora, the user, rather than the IR system, is tasked with knowing which one is most likely to contain the correct answer. If the response to a query is inadequate, then either the query may be modified or perhaps another corpus should be investigated, and the entire process is iterated until the user's information need is satisfied. That model is appropriate when the number of corpora available to the user is quite limited. When a user has access to a number of data sources as large as, say, the number of anonymous FTP sites in the Internet (via, for example, Yahoo), it is no longer possible for a user to wade through the sea of potentially relevant documents that are returned to determine whether the query needs to be modified in some subtle way or whether a complete and coherent answer to their query exists somewhere in the current search results. Note that substantial effort is required to both identify and retrieve relevant documents and to construct a meaningful response to the query. The latter task may entail iterating over cycles of query refinement and additional retrieval (for example, because the retrieved data is not highly relevant, contains conflicting information, suggests a related topic that may be useful, etc.). In addition, heterogeneity and lack of uniform structure in information databases, ranging from unstructured text to highly structured relational data, rule out many of the existing approaches to gathering data from diverse sources. Clearly, something more is required of the IR systems.

The problem as described seems amenable to a *cooperative information gathering* approach. Information Gathering (IG) involves pro-active acquisition of information from heterogeneous sources in response to a complex query. Traditional IR is a limited sub-case of such information gathering systems which must be able to reason with and draw inferences from complex representations. In addition to the complexity of query specification, control of the acquisition process may itself be complex and dynamic in IG systems, whereas queries in IR systems generally map onto static, pre-specified retrieval plans. In this paper, we draw upon a long tradition of work in distributed problem solving (DPS) to propose a cooperative agent-based solution for information gathering. In response to a query, multiple semi-autonomous agents can be released to search the distributed "information space" in a cooperative manner for relevant items. A multi-agent approach to information gathering is promising for a variety of reasons [16]:

- One advantage of multi-agent systems is concurrency, which is important in time constrained situations or when the search space is very large, as is the case with networked information gathering. Query plans can often be decomposed into relatively independent sub-plans with few interdependencies. Agents executing sub-plans can function relatively autonomously, but can coordinate with other agents to exploit interdependencies when they exist.

- When a system is dealing with enormous quantities of data, distributed computation at the sites where the data resides may often be more efficient than migrating data to a centralized processing location. Instead of gathering data dispersed across networked information servers at a centralized site and then evolving a coherent response to a query, agents can reside at the data sources and perform distributed coordinated retrieval to prune their data space and send substantially less data to the centralized query system for further processing.

- Agent-based architectures offer modularity, robustness and other advantages of distributed systems. For example, information agents can be constructed and maintained separately to accomodate heterogeneity in access methods, data representations and communication protocols that make it necessary to construct agents with specialized knowledge. Agents can use other agents to provide abstractions of heterogeneous information sources. In addition, passive data sources like databases can be transformed into information providing agents by wrapping them with intelligent interfaces [31], making possible negotiation processes between retrieval agents and intelligent search engines.

Cooperation between agents implies management of interdependencies between their activities so as to integrate and evolve consistent clusters of high quality information from distributed heterogeneous sources. Rather than simply retrieve sets of documents from disparate sources that are relevant to a query, such agents perform a parallel search for information to compose a coherent answer to a user's question. Cooperation is especially important because:

- Users often provide vaguely worded or sparse queries, leading to an explosion in the amount of information that is deemed potentially relevant. Agents that can dynamically exploit relevant information unearthed by other agents can better focus their search processes. Viewing partial results as information relevant to a query opens up a rich set of possible subproblem interrelationships that may be beneficially exploited.

- The amount of data that is relevant to even a precisely worded query may itself be too vast. The agents can exploit cues and hints based on information discovered by other agents at non-local sites to further narrow the set of relevant local data.

Given that the need to efficiently search through networks of information servers is real, the issues involved in using a team of cooperating semi-autonomous agents to search for the desired information are yet to be explored. Large scale networks of distributed information servers with complex interdependent data not only necessitate increased parallelism in search, but also motivate the need for cooperative retrieval and dynamic construction of responses to queries. The domain of such a search consists of multiple wide-area networks that are composed of, among other things, information servers (see Figure 1). In response to a query at a node, following some query planning, agents are dispersed to various regions in the network where they plan their local actions, which may include spawning additional agents to perform certain subtasks. This results in the formation of a search organization for the purpose of satisfying a query [16]. Intelligent servers that receive queries and act as regional planning sites, either further decomposing the search into subregions or sending

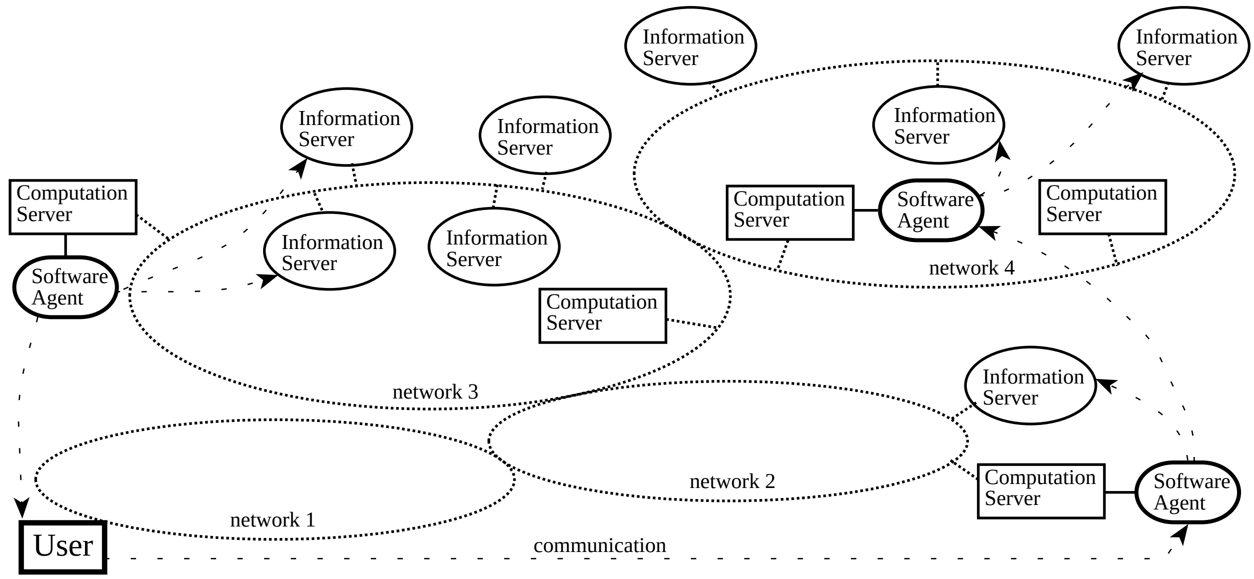

Figure 1: An example milieu for distributed information retrieval

agents to local corpora for data retrieval. The efficiency and the quality of the local search activity of an agent can be affected by the partial results produced by other agents working concurrently. Detecting interactions between decentralized search spaces and exploiting them for improved control in distributed search is the core of the model proposed in this paper for intelligent acquisition of distributed, heterogeneous information. Since the amount of available information is seemingly limitless, yet money, time, and computational resources are not, the agents' search is satisficing; they return the most relevant information available while staying within resource constraints. They must clearly coordinate with each other to maximize coverage, and may need to negotiate with other agents to discover consistent clusters of information. Results of these searches are communicated back to parent agents and are synthesized into a coherent response to the query.

The complexity of the modern information carrying landscape requires a sophisticated view where information is *acquired* rather than simply retrieved; where the process must be dynamic, incremental, and constrained by resource limitations. We present a model of information gathering designed specifically for such complex environments, a model of Cooperative Information Gathering (CIG). In the context of this model, information gathering is a much more complex process than the submission of a well formed query to a single corpus from which a complete response is ultimately obtained (as in the case of classical IR). Our model requires that we take a much more sophisticated view of information servers (entities in the network that mediate access to data). Users cannot be expected to translate their information gathering needs into the native syntax of the myriad of existing IR systems, nor to wait an indeterminate amount of time before some response to a query is produced. Information servers must be able to handle both partial and fuzzy specifications of queries. They must be capable of making partial and incomplete results available to users as the search for information proceeds. For example, by providing meta-level information about the status of their search for information, such as the amount, quality, and completeness of information currently retrieved, or an estimate of the time to completion. To conduct such a search, information servers must be able to control the tradeoff between completeness, quality, and precision of acquired information. This additional sophistication inevitably leads to increased complexity in the interface to the information server. However, there is a concomitant increase in the power and economy afforded to the user.

The purpose of this paper is two-fold: to assess the current state of Information Gathering systems in relation to the distributed processing/problem solving spectrum, and to explore possible synergy between Information Gathering (including Distributed Information Retrieval) systems and existing DPS techniques to enable pushing these systems closer to the distributed problem solving end of the spectrum. Once we have embarked on this journey, it will soon become apparent that existing IG models are left wanting. We begin by looking at the distinction between distributed processing and distributed problem solving in more detail. What features of a problem or problem solving make one of the paradigms a more appropriate model than the other? Specifically, what types of constraints can exist among subproblems and how can they be exploited beneficially from both local and global perspectives? We then present our model of CIG as an initial foray into intelligent information acquisition and discuss the model in some detail using example situations. Distributed Information Retrieval is often viewed as a distributed processing problem. Is that view appropriate? How well does the distributed problem solving view fit, and is there some benefit in taking such an approach? Finally, we review the literature related to distributed and agent-based information acquisition to assess the state of the art in this area and conclude with a discussion of the implications of our model for CIG.

2 Distributed Processing vs. Problem Solving

The task of information gathering in a distributed setting can be viewed in general terms as either distributed processing or distributed problem solving. Each view brings with it a set of conditions or problem features for which it is most appropriate. Distribution implies the decomposition of a problem into a set of subproblems to be solved by multiple processing units such as CPUs or agents. We find it convenient to view the agent as the locus of problem solving activity. Distributed processing is appropriate when subproblems are independent, whereas distributed problem solving is appropriate when subproblems interact and where there is some benefit to be gained both locally and in terms of the global solution from agent communication. This distinction is important in understanding the contribution of this paper. Details of the distributed problem solving model will be discussed in later sections on cooperative information gathering.

Given some computational problem P, the solution is obtained in a distributed manner by first breaking the problem down into n subproblems p_i for $1 \leq i \leq n$, which are then distributed among a set of agents. Each agent performs problem solving locally to arrive at a solution to its own p_i, and the local solutions are combined to arrive at a solution for the original problem P. This process can be viewed as dynamically interwoven phases of problem decomposition, problem solving, and solution synthesis [19, 33]. As stated previously, distributed processing is characterized by complete independence of subproblems. Agents need nothing other than local information to arrive at a subproblem solution of the required quality that can be synthesized with other agent subproblem solutions to arrive at a global solution.

Distributed problem solving, on the other hand, is characterized by the existence of interdependencies between subproblems leading to a need for the agents to cooperate extensively during problem solving, and by the potential for leaving some subproblems unsolved. For example, it may be impossible to solve p_j without first solving p_i, or knowing the solution to p_i may simply make it easier to solve p_j, or knowing the solution to p_i may obviate the need to solve p_j. If one agent finds a number of sites that are highly relevant to a query, communicating those sites to other agents may facilitate their search for relevant information. Agents rely on communication to detect and exploit these interdependencies between subproblems. At the start, agents have only partial and incomplete global views of solution requirements and the state of problem solving. In spite of this deficiency in information, agents can arrive at partial and tentative results that may be exchanged by the agents working on subproblems that are interdependent, to reduce the uncertainty that surrounds local problem solving. That is, agents can exploit the interdependencies between subproblems to their benefit. This is the essence of the functionally accurate, cooperative (FA/C) paradigm presented by Lesser et al. [37, 39] as an approach to distributed problem solving. In FA/C systems, the interdependencies among subproblems motivate agents to augment their local information with information about global problem solving activity in order to enhance the efficiency of the ongoing problem solving process. Once these interdependencies are uncovered via communication of problem solving activities, such as receiving partial results or meta-information, they can be exploited in a variety of ways to improve problem solving both locally and globally.

As we have defined distributed processing and distributed problem solving, any given problem may have features of both paradigms and will lie somewhere on the spectrum between them. To place a problem instance on this spectrum we need to characterize the nature of subproblem and/or agent interactions, both in terms of when they occur and their implications. For example, if subproblems interact only at the time of global solution synthesis, then local problem solving is completely independent and we are closer to the distributed processing paradigm. Likewise, it may be that agents interact before problem solving begins, perhaps to communicate some global data, but not during problem solving. This communication step may alter agent behavior, but it does not represent the exploitation of constraints derived from the interdependencies of dynamically generated partial results. A system that uses this approach is the distributed version of INQUERY [3, 4] (to be discussed later), where the set of statistics used to compute globally comparable relevance rankings is obtained by pooling statistics from all corpora that will be searched. After the initial computation and communication, retrieval at the various sites proceeds independently and in isolation. These examples point to the fact that for distributed processing to be considered distributed problem solving, interactions must be based on the *dynamics* of problem solving (such as the current problem being solved and the current state of problem solving activity). Finally, the tightness of the coupling between subproblems affects the placement of a task on our spectrum. If the interdependencies that hold between subproblems are weak, then the problem is more like distributed processing. For example, it may be the case that local processing can proceed almost to completion, but agents must communicate to interpret results. Likewise, strong interdependencies between subproblems are indicative of distributed problem solving.

3 Intelligent Information Retrieval

Recent trends in information retrieval show an evolution from relatively syntax-oriented retrieval systems to more semantically guided systems [26, 43, 51] known as Intelligent Information Retrieval (IIR) systems. These systems are guided by task-level requirements, rather than just the syntax of the queries, to establish an association with and retrieve information from the stored structures to which they have access [43]. In this report we go a step further and propose a model for Cooperative Information Gathering (CIG), where a group of potentially heterogeneous IIR agents are involved in simultaneous access and composition of associated information spread across a network of information servers. Top level queries drive the creation of partially elaborated information gathering plans, resulting in the employment of multiple semi-autonomous, cooperative agents for the purpose of achieving goals and subgoals within those plans. The rest of this section will briefly introduce IIR systems.

Intelligent Information Retrieval involves content-based access of information, where the meaning and not just the syntax of a query is used to guide and control the retrieval process. Abstractions and models of the data environment and user requirements are used to relate the query to the information so as to facilitate a more pertinent and controlled access to a large array of information repositories. For example, consider retrieving data about transportation to a picnic spot. Domain knowledge about types of transportation is used to form a query for retrieval from an information server. A transformation, such as specialization, is performed on the concept "mode of transportation" to get a concept like "rented car" or "bus" or "train". Further transformations may lead to "Hertz Rentals", "Greyhound" and other transportation companies that are used to retrieve relevant data on the availability of reservations. Consider another example from [45]: "find a mechanism that converts a uniform rotary motion into a reciprocation in Atrobelovsky's design encyclopedia." The retrieval mechanism here should have a model of kinematic mechanisms. Simple keyword-based systems cannot handle such queries.

Most of the models for IIR presented in the literature [43, 51] can be conceptually captured by the abstract models shown in Figure 2. Figure 2a shows an intelligent information system where an "inference shell" is wrapped around the data repository. We will apply the term *information server* to this combination of an inference engine and a data repository. The inference shell contains the "knowledge" or "domain models" or "abstractions" of the information and serves as an interface through which queries are filtered and recast to associate the task-level content in a query with the information. Although information servers are typically thought of as passive processes, pressed into service for the purpose of satisfying externally generated queries, they may take a much more active role. That is, information servers may seek out and build connections with other servers in the network that contain data related to the information maintained locally. This view treats information servers as intelligent agents with their own goals, adding to both the richness and the complexity of the environment. Figure 2b shows an intelligent information retrieval agent, carrying user requirements and task-level knowledge, reaching a data repository to extract information from it. The agent formulates a query based on abstractions of the contents of the repository and its own task-level requirements to perform a content-based retrieval. Figure 2c shows a hybrid model where the intelligent agent carries the user's requirements and, possibly, abstract descriptions of the information sources it can access. The retrieval engine contains a more detailed model of its information database as well as mappings between this model and the abstract descriptions in the agents that can access it. Because both the retrieval engine and the retrieval agent are intelligent, they can engage in a dialog to negotiate the nature of their interaction. That can be beneficial when access to the information database, which is mediated by the retrieval engine, is costly. The retrieval agent can employ its model of user requirements to determine how to trade off expected completeness, quality and precision of the results of its query based on the retrieval engine's model of the database (its access methods, available indices, etc.). In addition, the hybrid view has the advantage of separating the user and the task-level requirements from the conceptual model of the data repository. Therefore, we adopt the hybrid view of IIR (Figure 2c) in our further discussions.

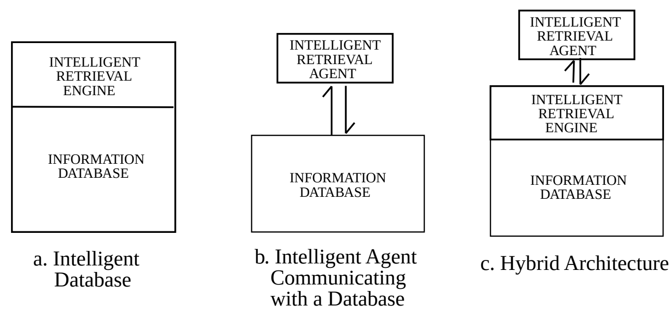

Figure 2: Conceptual Models for Intelligent Information Retrieval

4 CIG as Distributed Problem Solving

The proliferation of network-based information systems motivates the need for distributed information acquisition systems. However, the huge number of available resources makes it impractical for users to specify direct mappings from their needs to the available resources. This necessitates an intelligent retrieval component to IR systems that we argue is best modeled as a search process that is informed by the results of queries to information servers. The nature of these two requirements leads to the need for developing models and technology for Cooperative Information Gathering. Most existing approaches deal with either Distributed Information Retrieval or Intelligent Information Retrieval but there is little that deals with CIG. The central aim of this paper is to provide a model of CIG as a distributed problem solving process and consequently borrow from existing methods in multi-agent systems (MAS) to provide the technology for CIG systems.

Let us start by introducing a conceptual model of DPS as a search problem. Consider a classical AND-OR goal tree as a representation of the search space of a problem-solving system. From an information gathering perspective, we can think of a goal/task node in such a tree as an information query specifying required goal specification parameters and their characteristics, optional goal specification parameters and their characteristics, solution output parameters and their characteristics, and the level of effort (resources and available time) to be invested in producing solutions that meet the requirements. Goals can be related to one another through goal-subgoal relationships and to data and resources via constraining interrelationships. Figure 3 (from [39]) shows an example of such a goal tree. Solutions to high-level sibling goals, like G_{k-1} and G_k, or more distant goals, like $G_{1,1}$ and $G_{k,2}$, can have constraints between them. These interrelationships can be independent of the specific solution(s) to a goal or highly dependent on the exact character of the solution(s). Constraints for goals at a particular level can have implications for achieving goals at both lower and higher levels. Goals may be related through a complex chain of interdependencies. For example, G_1 and G_{k-1} are interdependent through G_k. It is important to note here that the entire goal structure need not have been elaborated before problem solving begins. The structure can be dynamic and can evolve with the agents' emerging composite view of the problem solving process. The elaboration can be top-down, based on the higher-level goals of the agents, or bottom-up, driven by the data, or a combination of both. Furthermore, there are no restrictions on the consistency of the goal structure.

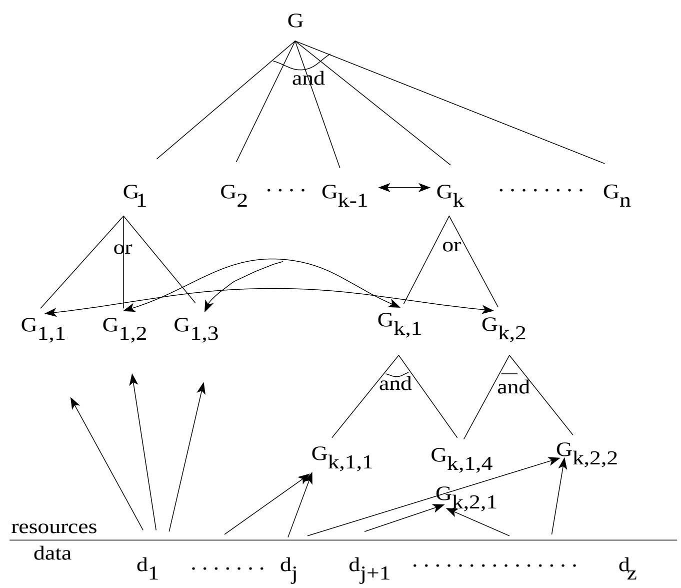

Figure 3: A goal tree: The G's represent goals and the d's represent data produced by queries to information servers. The double headed arrows between goals indicate that the goals are interdependent. The arrows between data and goals indicate that the data is required for that goal's solution.

We further ground this discussion by introducing an example from the Information Gathering domain. Figure 4 shows the goal tree for updating an environmental database by gathering information from relevant on-line sources. Note that we kept the goal tree simple for ease of exposition; actual goal trees are generally much more complex for non-trivial IG tasks. The database maintains up-to-date information about companies and universities involved in environment related work, job information in related areas and any environment-related news briefs. Acquiring company and university information involves locating relevant information and then retrieving and indexing the information appropriately. At the lowest level, this process involves gathering data from information repositories like "Environmental Route Net" and the "Amazing Environmental Organization Webdirectory". In case of unstructured information, there is a need for generating descriptors that map the content of the retrieved material into the semantics of the domain.

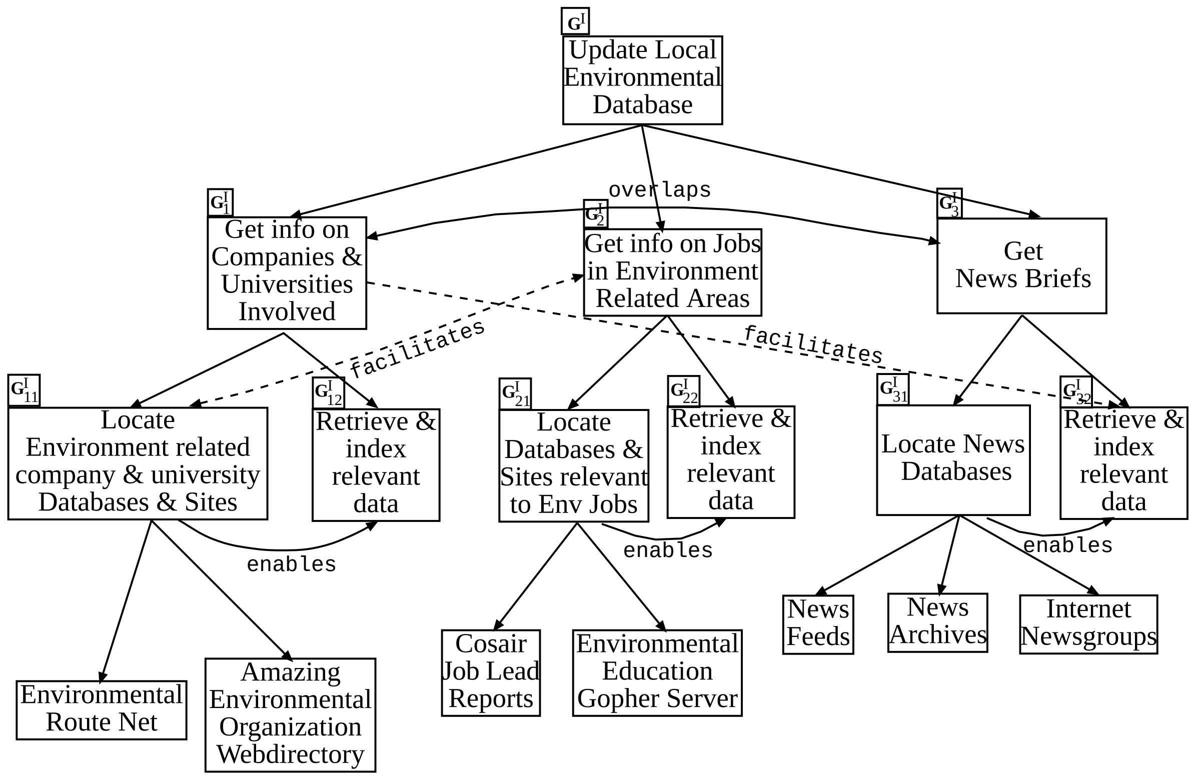

Figure 4: Goal tree for Updating an Environmental Database

In view of this model, one can see the complications involved in performing "efficient" problem solving, even in a single agent scenario. How does the agent detect interrelationships between sibling goals at various levels of the tree so that, for example, solving one goal before another goal can facilitate the later goal's solution quality. The goal interrelationships can be of various types such as facilitates, enables, overlaps, hinders, favors, and so on [11, 14, 40] (these interrelationships will be discussed in detail later). In terms of our goal representation, a facilitates interrelationship implies that the values of a solution output parameter of the facilitating goal can, in some way, determine an optional goal specification parameter of the facilitated goal. The facilitated goal could have pursued its activity without these optional parameters, but having them available will contribute to an improved search during the goal achievement process. The solution or partial result from the facilitating goal provides constraints on the solution of the facilitated goal and consequently make it possible to achieve this goal with fewer resources and/or higher quality. Similarly, an enables interrelationship implies that the enabling goal produces a solution output parameter value that determines a required goal specification parameter of the enabled goal. An overlaps interrelationship exists between two goals that share determinants of some of their solution output parameters. A favors interrelationship implies that a plan for achieving a goal can be used to achieve another favored goal through minor modifications (e.g. changing a query slightly so that the new query can produce results that not only satisfy one goal but also another subgoal). Detecting and the using of such goal interrelationships for efficient coordination is a hard problem in complex AI systems [55].

Figure 4 contains facilitates, enables and overlaps interrelationships between various subgoals. There is a facilitates interrelationship from G_1^I to G_{32}^I because information on the names of companies and universities involved in environment-related activities can provide "key words" for more refined retrieval from a large news-wire text database. An overlaps exists between G_1^I and G_3^I because some of the news briefs provide information on companies and some of the companies maintain a list of all news briefs related to their organization. An overlaps interrelationship says that the two agents involved may be doing similar work and can hence benefit by sharing their partial results. Enables between G_{11}^I to G_{12}^I indicates that an agent has to locate databases related to environmental companies and universities before it can extract appropriate information and update the local environmental database.

Now consider the case where a goal tree is distributed across multiple agents, none of which may have a complete global view of the tree. Each of the agents can model only a part of the global goal structure based on its role in the overall problem solving process. This increases the complexity of the situation discussed above. Figure 5 (from [39]) illustrates an example where the goal tree from Figure 3 is distributed across two agents. Detection of coordination relationships by the agents now becomes more difficult due to their partial view of the goal tree.

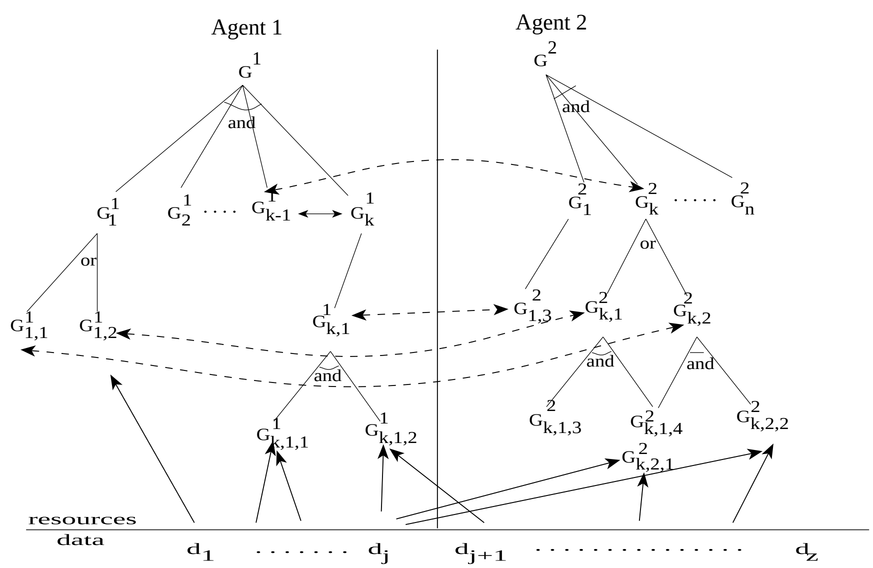

Figure 5: A distributed goal tree: The goal tree of Figure 3 is distributed with partial replication between two agents. The dotted arrows indicate interdependencies among goals and data in different agents.

Continuing with our example in Figure 4, let us distribute the goal tree across three agents such that Agent 1 needs to achieve G_1^I, Agent 2 needs to achieve G_2^I, and Agent 3 is in charge of G_3^I. Just as goals are distributed across agents, the agents may be distributed across the network. Each of the agents may travel to different information carrying sites that are relevant to the completion of their particular goals. Such a multi-agent system possesses a number of desirable qualities such as concurrency, robustness, and separation of concerns that make distribution attractive, especially when the amount of information to be accessed is large and time is a limited resource (which is true of today's Internet and user characteristics, but these aspects will soon become so acute that any viable system has to take them into consideration). However, distribution also gives rise to the need for effective management of interrelationships between activities distributed across agents, so as to derive maximum benefit from multi-agent architectures. Cooperation among agents is crucial. Agent 1 could search for information on companies and universities involved in environmental activities with complete disregard for Agent 3's search for relevant news briefs. However, without prior information on relevant companies, Agent 3 may be forced to perform an exhaustive search through its corpora. On the other hand, if Agent 1 finds company information and shares it with Agent 3, then Agent 3 can perform a much more focused and efficient search (perhaps through the use of a company name index). In other words, cooperation between the agents is required to utilize the facilitation relationship between goal G_1^I and goal G_{32}^I and thus achieve improved performance. Other questions that arise when the goal tree is distributed include the following. What kind of protocols are needed to detect the existence of interrelationships? How are inconsistencies between the redundant data of Agent 1 and Agent 3 (with an overlaps interrelationship between their goals) resolved?

Thus in the case of DPS, agents have to make their decisions facing additional uncertainties due to a lack of complete information about the unfolding problem-solving process. The global goals to be achieved have certain utility measures like quality of solution and time (to complete a set of tasks to achieve the goal). Agents have to coordinate their contributions to the problem-solving process so as to maximize the global utility that they are able to achieve. We assume that the agents in such a system possess certain abilities. Based on its partial views, each of the agents is endowed with the ability to predict global implications (at least approximately) of doing a certain task (or achieving a certain sub-goal) at a particular time. This problem is complicated by the existence of non-local effects (like facilitates and enables), that embody interactions between non-local parts of the goal tree (that can belong to different agents). Thus, augmenting local partial views by information from other agents leads to more informed local decisions by an agent. This brings us to another important ability of the agent — the ability to communicate with other agents for the purpose of detecting interrelations between various parts of the goal structure. These abilities critically affect, both qualitatively and quantitatively, aspects of local planning decisions for problem solving control.

The process of distributed problem solving is described in [33] as taking place in four stages: *problem formulation, focus-of-attention, allocation,* and *achievement* (see Figure 6 which has been slightly modified from [33]). The discussion of goal trees up to this point has treated them as static structures. However, as is clear from Figure 6, goal structures are very dynamic entities that evolve and change as problem solving proceeds. The problem formulation stage involves identification of the set of goals or tasks required to solve a given problem. Problem formulation can be a top-down decomposition of the original problem into a set of subproblems, a bottom-up process that composes supergoals in a data-driven manner, or a reorganization to choose an alternative set of goals/tasks in response to a failure. Different agents and sets of agents may employ different types of problem formulation in parallel. In addition, an agent or set of agents may enter this stage more than once, employing different types of problem formulation each time. When resources are limited or constraints exist among goals, there is a need to determine which goals to work on next. In the focus-of-attention stage, a subset of the goals from the initial problem-solving structure is chosen so that resources may be devoted to their achievement. In the allocation stage, the active goals chosen during focus-of-attention are assigned to one or more agents. Finally, during the achievement stage agents attempt to achieve goals for which they are responsible, and then synthesize a global solution from their local solutions and the solutions obtained by other agents. Note that Figure 6 does not show a single sequential path from problem formulation to focus-of-attention to allocation to achievement. Rather, it may be the case that an earlier stage needs to be revisited in order for problem solving to continue. For example, if an agent has more than one allocated goal it may focus attention locally to decide on an appropriate order. Also, an agent may not be able to directly achieve its assigned goals and therefore needs to further decompose or compose them via problem formulation. It should be clear from the preceding discussion that each stage may involve a single agent, such as focusing attention locally, or it may be distributed over a set of agents, as when agents negotiate over goal allocation. In addition, within an agent or set of agents the various stages may be occurring asynchronously with different collections of goals. Agents involved in this process move through the various stages in a dynamic manner, concurrently and asynchronously, until the global goal is sufficiently satisfied given time and resource constraints.

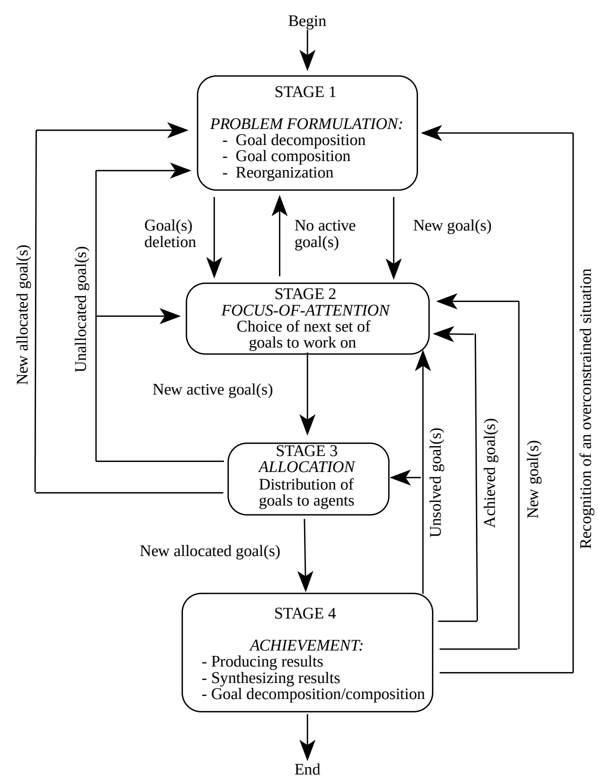

Figure 6: A goal-based view of the stages of Distributed Problem Solving

CIG can be viewed within the framework of DPS as discussed above. In response to a query, one or more agents are released onto the network, each responsible for one or more corpora. Each agent treats its information seeking process as a cooperative planning activity. The global solution is the response to the query, and it is a composition of the information retrieved and transformed appropriately by domain knowledge in the agents. Problem decomposition involves assigning subgoals to agents. The subgoals assigned to each agent involve seeking information relevant to global goals of the retrieval process. There may be interrelationships between the subgoals assigned to the agents and this may necessitate sharing partial results of their search to enhance the efficiency of the overall retrieval process. Subproblem composition involves combining the returned information into a coherent response to the original query. It is also possible to use the DPS framework when the agents involved in information gathering are self-interested. In that case, agents enter into contracts to solve information gathering goals.

4.1 Subproblem Interaction

One of the primary reasons for needing a distributed planning approach to CIG is the existence of interactions between various subproblems and subgoals[1] during the search process. The acquisition process at one information server can be affected by the acquisition at another server at a different site. These effects can be at various levels, either through the high-level semantics of the problem-solving domains or more directly at the level of the content of the information acquired. One of the underlying assumptions of this model is the availability of mappings from the information retrieved into the semantics of the domains involved. For structured data (like relational tables) this may not be a difficult task. However, for unstructured data like ascii documents, we assume the availability of means to generate "descriptors" which are representative of the content of the documents in terms of a set of domain primitives [2].

Various kinds of goal interrelationships like facilitates, enables, overlaps and subsumes that exist between subproblems can be exploited in a variety of ways. For example, the uncertainty that may arise from incomplete local information can be reduced through detection and subsequent exploitation of overlaps and subsumes interrelationships. However, this process involves providing the agent with a more complete global view, and that entails communication costs. Hence an agent should communicate only relevant portions of its local view of the problem solving process to help form a more coherent view of the emerging global problem solving process in other agents. Partial solutions and meta-information received from other agents may lend support to a local solution or may point to an inconsistency in an agent's local processing. Carver et al. [8] [7, 5, 6] address the problem of resolving uncertainty in the sensor interpretation domain. When subproblems overlap, communication among agents may reduce the amount of redundant work performed and therefore reduce the time required to achieve a global solution. Also, it may be the case that a solution or partial solution generated by one agent may facilitate (i.e. serve to focus or constrain) the problem solving of another agent and thereby reduce the amount of computation required [12]. The problem solving process of one agent in some way assists another agent in its problem solving, perhaps by making the other agent more certain of its local solution or by restricting the space of potential solutions that must be considered. The end effect is a "better" or higher quality global solution. The constraints arising out of goal/solution interrelationships may also play a crucial role in exploiting parallelism among the agents. For example, an agent with a facilitates interrelationship from another agent can simultaneously develop a plan with the understanding that when the relevant results are received, it may

[1] We use the terms goals, subproblems and tasks interchangably. The idea is that a goal represents an intention to solve a particular subproblem or task.

[2] See [34, 42, 49, 50] for some of the recent progress on this aspect.

need to iteratively repair or modify its partially developed plan. Alternatively, the agent could perform some other task while awaiting the receipt of information.

Figure 7 shows an example that highlights these issues in the document retrieval domain (modified from [13]). For a given query, there may be many sources of relevant information. Product reviews often exist on-line, or may be obtained from publishers for a fee in paper or electronic format. Relevant reviews may be found on-line in the review section of the TidBits newsletter, in the Info-Mac archives, or in discussions about the product in Usenet news groups. The query may be satisfied by dispatching agents to locate and retrieve the required review. Each agent may employ different access methods (such as WAIS, ftp, http, telnet, etc.), and the access methods may have recourse to the same information at a variety of physical locations (such as the main TidBits archive ftp.tidbits.com, or its various mirrors). Interrelationships exist between some of the goals of the agents involved. Locating a paper review "enables" its retrieval, i.e. paper reviews may be obtained by first finding a citation, and then either finding the actual article or obtaining it from the publisher. Finding a citation via Uncover "facilitates" the goal of getting the article faxed to the user. An overlaps interrelationship exists between Agent 1's "Get from Seller" goal and Agent 2's "Use Uncover" goal. This is due to the fact that once an agent accessing the seller's archive finds a particular citation, Agent 2 can avoid the search for that same citation at the Uncover database. Another source of information that is not exploited in the example above is the increasingly popular World Wide Web (WWW). We can think of the web of citations or the web of hyper-text links associated with a document as a web of consistency constraints. That is, documents linked in this way may contain related information and that information should be consistent, but often it is not. For example, during the process of retrieval of product reviews discussed above, two sites may quote different prices for the product. Product review information at an FTP site accessed via the WWW may contain outdated prices, whereas a link to the seller database in an html document may in fact contain the latest prices. When inconsistencies are uncovered, agents need to work to resolve the associated uncertainty so that a cluster of consistent documents containing "correct" information is presented to the user. In this case, the agents may choose to use the seller database information to override other sources.

Our multi-agent Case Based Reasoning system based on Negotiated Retrieval is another example of a sophisticated system exploiting constraints generated by sharing results of a partial search [41]. More specifically, a response to a query involves assembling related pieces of information from different case bases to form a composite case. The agents have to co-operatively retrieve mutually acceptable responses while negotiating compromises to resolve conflicts. Each agent retrieves subcases from its local case base and all agents together assemble a mutually acceptable overall case from these subcases to produce a response to the user's information needs. Information acquired by an agent can be related to the requirements of information acquisition in another agent. Thus, agents augment their local views through selective and timely communication of partial results among themselves.

4.2 Satisficing

Although the amount of information available on the Internet is seemingly boundless, the resources available to search that information typically are not. For any given query, rather

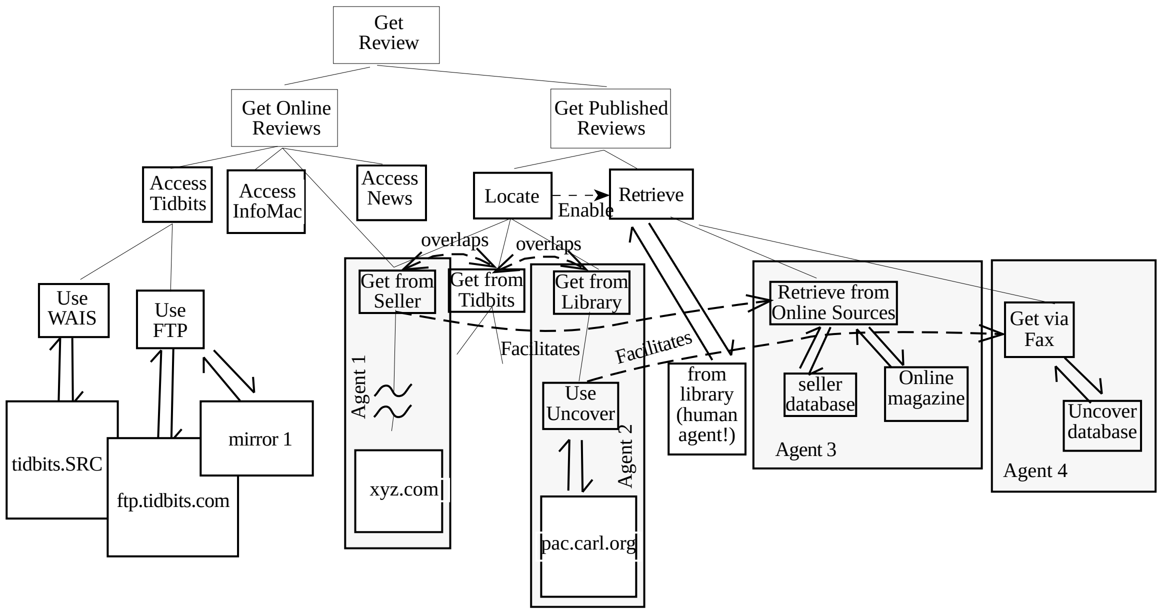

Figure 7: A goal tree for retrieving Macintosh-related product reviews

than performing an exhaustive search, one must attempt to locate the "best" response possible given time and resource constraints. That is, the information gathering process must be *satisficing* along various dimensions like precision, quality, etc [38]. As a motivating example, consider Figure 1 above. A user has access to multiple networks, each of which contains its own data sources, information servers, and servers devoted to computation. The links to each of the networks may have different bandwidths, reliability, and costs of usage. For any given local network, as well as from a global perspective, some data sources may be more relevant than others for the query at hand. Again, the cost and speed of access to the individual data sources and network resources may vary. It may be the case that local users are given preferential treatment, and costs or retrieval time may be lowered by sending an agent to a specific network rather than performing remote access. Finally, individual communication lines, networks, information sources, and servers may be subject to intermittent failures or may not even be operational at the time that the query is submitted, and their cost structure could be dynamic and based on fluctuating market demand.

What are the implications of performing information gathering in such a complicated, unpredictable environment with limited resources? First, management of resources must be an integral part of the process. Simply charging ahead blindly, stopping the search when resources have been exhausted is likely to lead to very poor results. Planned activity under time and resource constrained situations is a hard problem needing sophisticated, knowledge-intensive techniques [1, 22, 56, 57]. Many questions need to be addressed about the efficient usage of resources. Which regions and which information sources are most promising (and should therefore be explored first)? How should the trade-off between speed and cost of communication and utility of data at the various sites be managed? How many agents should be allocated to each region? What amount of parallel effort should be expended to reduce the impact of single point failures in the network? When, where, and how should partial results be integrated? Should the integration take place at a global, centralized site or in a hierarchical fashion beginning with regional sites? The unpredictability of the environment implies that a complete, centrally generated plan may often lead to failure. Some amount of planning must take place regionally in order to deal with unexpected failures or to opportunistically focus the efforts of agents; to effectively deal with the dynamics in each region.

By explicitly representing and reasoning about resource constraints, we bolster our confidence that a "good" response to the query will be obtained relative to the amount of effort expended. If communication is expensive and slow, we may access nearby data with low expected quality first, rather than trying distant data sources of higher quality that may require more time than is available. When more time is allocated to the search process, the scope of the search can be broadened to include higher quality sources, while retaining some amount of effort on inexpensive low quality sources. In this way we are assured that some response will be attained and, if our planning is reasonable, we will include some amount of very high quality information in the response.

4.3 Redundancy

Redundancy in distributed search – overlapping information that pertains to different aspects of a query; replication of data at multiple sites; the possibility of deriving the same conclusion from different sets of data – raises a host of issues. Advantages of redundancy include increased robustness of the system in environments with failure-prone components and increased flexibility in responses. Redundancy can play a role in the reduction of uncertainty when dealing with erroneous or incomplete information. On the other hand, redundancy has the disadvantage of increased resource usage and possibly increased total processing times. For example, the Internet may contain "mirror" sites for certain data repositories or it may contain redundant data from different sources for the same task. When information is available for a fee from multiple providers, an information marketplace develops. In that case, accessing an information server for the same data at different times may lead to different costs due to market influences on pricing. Data from different sources may be of different quality or may be differently organized. A particular task could possibly do with low quality data that perhaps could be locally acquired. Thus, recognizing the role of redundant data and computation could be important for exploiting the possibilities that such redundancy offers in a CIG system. Redundancy could be permitted if the control costs outweigh the benefits of avoiding it. Alternatively, if we are dealing with faulty systems or poor quality data, redundancy could help resolve uncertainty in the retrieved data by providing additional constraints.

5 Related Work

We now review some existing work in the Information Gathering literature. We divide the work into two kinds: Distributed Information Retrieval approaches, which rely on relatively knowledge-poor techniques to acquire information from distributed sources, and Information Gathering approaches, which rely on knowledge-rich, content-oriented techniques. We also review some of the work in Intelligent Information Retrieval. Its relevance arises from the fact that each of the nodes in a CIG network may be an IIR system. Throughout the presentation, our primary interests are in determining the nature of the local processing performed by the agent or an intelligent information source and how they might participate in an information gathering task in a cooperative manner. Note that this review is only representative, and is not intended to be exhaustive.

5.1 Distributed Information Retrieval

As the Internet evolved from a test bed for experimentation in data communication protocols and remote login into a medium for collaborative data-sharing, the need for research on approaches to resource discovery on the net became obvious. Bowman et al. [2] give a good review of the problems and approaches involved in the task of "scalable Internet resource discovery". Information, which can possibly be incomplete and inconsistent, needs to be gathered from diverse and heterogeneous sources. Bowman et al. suggest exploiting the semantics of specific resource discovery applications based on data typing. The authors discuss ways to deal with huge loads on the Internet using methods like data caching, server replication and self-instrumentation. In order to assist users in dealing with enormous volumes of data, they propose content-based searching algorithms and specialized servers for dealing with particular user communities. However, the kind of domain specificity they exploit in their content-based search algorithms and resource discovery engines is weak domain knowledge like file types, gross syntactic and structural features of documents, or keyword based attribute extraction. IG, however, is potentially very knowledge intensive.

Distribution of the INQUERY system [3, 4], on the other hand, is concerned with performance in searching distributed text corpora. The approach to Distributed IR currently planned for the INQUERY system (as with most other IR systems attacking the problem of distribution) clearly falls under the distributed processing rubric. The response to a query in a distributed environment is generated by transmitting the query to INQUERY systems local to the individual information carrying sites. Each INQUERY system finds relevant documents in the local database, and the simple union of all documents found serves as the response to the query. The local systems work in total isolation with only local data. None of the processing performed at any of the individual sites has any impact on the processing at any other site. The subproblems, finding relevant documents at a single site, do not interact. There is a single exchange of information prior to the start of problem solving. The statistics used to compute a document's relevance are based on the composition of the database in which the document resides. Therefore, relevance rankings of documents from different databases are not directly comparable. To overcome this problem, each system transmits its statistics to a centralized location that creates an aggregate, normalized set of statistics used by all of the local systems for relevance ranking for the current query only. The subproblem solutions, sets of documents, are combined in a simple unidirectional synthesis step. While these interactions indicate that the approach taken by INQUERY is not a pure instance of distributed processing, the type, weakness, and timing of the interactions indicates that it does not stray far from that side of the spectrum.

Huhns et al. [24] present a method for learning and updating the relevance of corpora to individual topics of interest via meta-knowledge. Meta-knowledge is kept (learned and updated) by each user on the network and is a 4-tuple of the form [$User1$, $User2$, $Keyword$, $CertaintyFactor$]. For example, [$Smith, Jones, compilers, 0.8$] says that user Smith has high confidence that user Jones can supply articles on compilers that Smith will find interesting (relevant). A query of the form "find all articles related to compilers" will include queries to Jones and other users as indicated by Smith's meta-knowledge. The returned articles will be ordered according to the certainty factors of the associated users. In addition to text, meta-knowledge can be returned in response to a query. For example, $[Jones, Doe, compilers, 0.5]$ may be supplied as a response. User Smith can then combine the certainty factors to arrive at something like $[Smith, Doe, compilers, 0.4]$. Another way that meta-knowledge is propagated is by receipt of a query. If Doe receives a query from Smith about compilers, Doe may rightfully assume that Smith will soon become knowledgeable on the topic. Therefore, Doe may assume $[Doe, Smith, compilers, 0.1]$. The certainty factor is low since Doe does not know if Smith's knowledge will be interesting. The primary item of interest in this article is the direct way in which the authors represent and deal with uncertainty about the relevance of information as a function of its source. The level of uncertainty is used to order the search process (best first) and is updated as the data sources themselves change. However, information gathering can exploit more than just relevance knowledge.

Knoblock and Levy [30] seek to improve the efficiency of information retrieval from large numbers of distributed databases by taking advantage of information that is only available at run-time. Given that access to information sources has an associated cost, either in time or money, it is important to identify relevant sources and to retrieve as little information as possible to satisfy a query. Most approaches to this problem depend on information that is statically available from the query. However, there are a number of reasons why it may be difficult to prune the number of candidate information sources at compile-time. For example, although agents' domain models may describe properties of classes of objects in the domain, they usually do not contain information about specific individuals in the domain. That is, there is no way to reason about how query execution should proceed for possible instantiations of query variables. However, such information may be obtained at run-time (via additional queries) as variable bindings are established. The authors claim that obtaining and using information available at run-time can reduce the cost of information retrieval in distributed environments, and they present an algorithm for extending query planning algorithms to do so. This approach acknowledges the importance of partial results in focusing the search for information that is central to the CIG model. However, the information retrieval process, including the use of partial results, occurs within a single centralized query planner.

5.2 Intelligent Information Retrieval

Ram and Hunter [43] take the view that content-based IIR, as opposed to syntax-based IR, requires inference, leading to a combinatorial explosion of potential inferences. Since computational resources are limited, some method of controlling inference must be employed. The authors treat information acquisition as a planful activity driven by specific desires to retrieve or infer knowledge or information. Those specific desires, termed *knowledge goals*, serve to restrict the space of possible inferences in a dynamic manner that is dependent on what is already known by the system and what it is trying to learn. KG's represent information needs, and as such focus information gathering in both a top-down and bottom-up manner. Gaps or inconsistencies in the system's knowledge may lead to new KG's and thus new information gathering needs. This last point makes it apparent that IIR systems must be able to reason about their problem solving processes and partial solutions if they are to be able to characterize desirable knowledge (formulate KG's), such as the need to fill gaps or account for conflicts. Two systems serve as examples of the authors' theories — the AQUA story understanding system and the IVY differential diagnosis system. AQUA employs KG's in an iterative manner to first explore the text and then fine tune its understanding via the detection and resolution of anomalies, the construction of a causal explanation for events in the story, etc. Likewise, IVY uses KG's to incrementally refine its ability to diagnose structured descriptions of lung tumor pathology images. Feedback from human experts drives KG generation in an attempt to explain and rectify failures based retrospectively on information contained in past cases and actively on as yet unseen cases. Note that this approach is very similar in spirit to the goal-based view of distributed problem solving depicted in Figure 6.

Rus and Subramanian [45], discuss the idea of domain-oriented information capture and access using knowledge intensive modules. Given an electronic data environment, the task is to capture the data by acquiring partial models of it and to access the data as guided by the models. The construction of information agents from structure detectors and navigators is described. Structure detectors (sensors) decide if a set of data has a specified abstract property. Once a desired property is located, navigators (effectors) decompose the data into more detailed units. Navigators use discerned structure to drive their search. It is envisioned that libraries of structure detectors and navigators can be created to facilitate the construction of special purpose information retrieval agents. An example agent (the BibAgent) is described that searches the Internet for technical reports in response to a query. The structure detector uses the Unix "ls" command to locate potentially relevant directories. The navigator then selects specific directories to explore further. It can exploit knowledge of bibliographic data files (.bbl and .bib files) to retrieve complete bibliographic references. The agent incrementally builds a roadmap of the Internet indexed by queries.

Fikes et al. [20] have embarked on a project to develop network-based *information brokers* that can access information from a multitude of diverse information sources on the Internet. Information sources are autonomous, exhibit heterogeneity in access methods and content, contain structured and unstructured data, may return incomplete or irrelevant information, and are subject to change over time. Based on these observations, an argument is made that effective search requires domain-specific information brokers. The information broker architecture contains the following components. Domain and source models describe the broker's domain of expertise and the contents of the information sources that it can access. A "formulator" module helps users form queries, and comprises a product description browser, a query-by-reformulation assistant, and an alternatives advisor. There are also modules for planning, executing and presenting the results of queries. Although centralized and described in terms of modules, this architecture is very similar to distributed agent-based architectures described in Section 5.3.

Rissland and Daniels [44] outline an approach to IR that makes use of more knowledge-intensive methods developed within the framework of Case-Based Reasoning (CBR). CBR systems excel at retrieval of highly relevant cases and symbolic reasoning about problem cases. However, they typically have small case bases. IR systems, on the other hand, often have access to very large databases, but typically use knowledge-poor (e.g. statistical) techniques for relevance assessment. Rissland and Daniels built a hybrid system that uses a standard CBR analysis to find cases within a (traditionally small) case base relevant to a user's query. Based on the set of relevant cases obtained, a modified version of INQUERY's relevance feedback mechanism selects and weights terms for inclusion in a query that is sent to a larger text corpus. The authors demonstrated that the quality of documents retrieved by the hybrid system is better than those returned by standard IR methods alone. The results clearly demonstrate that traditional IR methods can benefit from the addition of more knowledge-intensive techniques involving symbolic reasoning.

Kirk et al. describe the Information Manifold (IM), an IIR system whose architecture is based on a rich domain model that allows the semantic content and the physical properties of information sources to be described and queried [28]. The IM's representation language is based on Horn rules. The authors show that within their language, it is possible to efficiently and completely determine the set of relevant information sources for a given query. The language also allows relational databases to be modeled, and can express queries with negations and descriptions of relations between information sources. The IM comprises a WWW client with a Mosaic-like interface and a knowledge base for organizing and querying information sources. The two components are tightly integrated so that the user can easily switch between hypertext browsing, information space browsing (via descriptions of the contents of information sources), and integration of information about new sources into the knowledge base.

IIR bears the same relationship to CIG as AI bears to MAS. IIR deals with the local processing capabilities and their amplification through the use of intelligent information acquisition techniques. Each IIR system, along with its coordination module, can form a cooperating IG system. The CIG model presented in this paper deals with the coordination module. However, note that the delineation between the coordination module and the local processing module may be blurred in many existing MAS systems.

5.3 Information Gathering

Following the lead of Oren Etzioni's software robots (softbots), Voorhees [52] describes an information gathering system composed of corpusbots and userbots. Each corpusbot serves as the system's model of a single collection of documents (corpus). The corpusbot contains all corpus dependent parameters, controls access to the corpus, and provides topic designators that abstract and summarize the corpus. Each userbot serves as the system's model of a single user. The userbot keeps the user's system preferences (such as the appropriate recall vs. precision tradeoff), a list of topics of interest or expertise for the user, and a dynamic list of known corpusbots and userbots. The userbot also contains a set of scripts that are arbitrary, parameterized programs for data access. Scripts are tagged with keywords that indicate their function and can be searched and retrieved by other userbots. A user with a question about corporate tax law can locate a userbot known to be an expert in tax matters and can search for a script in that userbot tagged with the word "corporate." Distributing a query over several agents may be accomplished by dividing the list of known corpusbots and userbots among the agents. A userbot can interact with other userbots and corpusbots to help guide its search.

Knoblock and Arens [29] discuss an architecture for information gathering agents that is the closest in spirit to our approach. Each of the agents contains a detailed domain model, the models of the information sources available to it, and the relationships between domain models and the contents of information sources. On receiving an information request, an agent identifies the appropriate information sources, reformulates the query using a series of transformation operators, generates an access plan to retrieve the data and sends requests for retrieval to other agents. The agents can improve their performance by caching frequently retrieved data or expensive data. When they are not processing queries they can also gather information that aids future retrievals; for example, learning about the contents of information sources and building abstract descriptions of them to aid in query reformulation.

However, in both of these works, there is no notion of exploiting the dependencies between agents working on different aspects of an information acquisition task. Cooperating to enhance efficiency of a resource-limited information acquisition process or negotiating to dynamically resolve conflicts and inconsistencies in the acquired data, leading to further search or retrieval, may be important aspects of IG systems in the future. Only recently, Decker et al. [16], Decker and Sycara [17], Birmingham et al. [18], Davies and Edwards [10], Foner [21], Kuokka and Harada [32], and Karakoulas and Ferguson [27] have started making some early but promising forays in this direction.

The MACRON architecture [16] is being designed as an instantiation of the principles of CIG. It incorporates capabilities to exploit subproblem interdependencies, manage the uncertainty inherent in multi-agent search, intelligently trade-off solution quality for resource limitations, and either exploit or avoid redundancy as needed. MACRON consists of an overall organizational architecture and three types of autonomous agents:

- DECAF reasoning agents, consisting of several subcomponents like a planner, coordination module, real-time scheduler and an execution monitor. The planner instantiates the set of tasks to be achieved for the particular information gathering activity and the scheduler produces a schedule of execution for these tasks. The coordination module manages the interdependencies between the tasks of different agents and posts constraints to the local scheduler in order to exploit such interdependencies. The function of the execution monitor is apparent from its name. Any deviations from the expected execution time line leads to feedback to the planner and scheduler to either re-plan or reschedule as needed.

- Low level network retrieval agents interface with information repositories and retrieve information as requested by higher level DECAF agents.

- User interface agents communicate with the user and pass on his or her requirements to the appropriate DECAF agents.

From an organization point of view, the agents in MACRON form a matrix organization [16] where the interface agents transform a query and pass it on to functional units that comprise sets of DECAF agents that specialize in dealing with particular types of information resources. Agents in a functional unit in turn schedule low level retrieval tasks and delegate them to the retrieval agents. Implementation of the MACRON system is in its infancy (circa March 1996) but it is hoped that it will serve as a test-bed for validating many of the ideas proposed in the CIG model.

Decker and Sycara [17] present a system that bears similarities to MACRON. User queries are delegated to specialized task agents that in turn could communicate with information gathering agents as a part of their task execution. An information gathering agent has a planner that instantiates task structures, a scheduler and an execution monitor. Each of these agents can retrieve information from a number of databases and compose it to form or update a local database that is more appropriate for answering queries by the task agents. The agents communicate using KQML messages. Future plans for the system include additional high level coordination among the task planning agents, making the system more faithful to the CIG model proposed in this paper.

The University of Michigan Digital Libraries project [18] uses a distributed agent architecture that is populated by user interface agents, mediator agents and collection interface agents. User interface agents provide a gateway between users and other agents. In addition, they publish profiles of the users for other agents to exploit. Mediator agents perform a variety of functions like delegating the query to appropriate collection interface agents, monitoring the progress of a query, allocating resources and coordinating agent activities. Collection agents provide communication wrappers for information repositories and publish their conspectus, which are descriptions of their contents and capabilities.

Davies and Edwards [10] are developing an agent-based approach to knowledge discovery in distributed databases. Although their problem domain involves mining highly structured data, as opposed to unstructured text, all of the issues that we address within the CIG model will appear in their domain. In their architecture, individual agents have access to local data and may communicate with other agents to share acquired knowledge, either to focus the search of other agents or to synthesize a globally coherent theory. The user interacts with the system via a user interface, which may in turn communicate with a supervisory agent that coordinates the activities of the discovery agents. The user can affect the search in a variety of ways, including directing agents to new data sources, altering high level discovery goals, and critiquing acquired knowledge. One goal of the project is to use existing relevant technology; for example, KQML, KIF, Ontolingua, Agent Oriented Programming, and Inductive Logic Programming algorithms (to perform the actual knowledge discovery).

The large number of proposed agent-based approaches to information gathering has spawned interest in mechanisms for helping agents (or, more generally, information providers and consumers) with similar information needs to find each other. Foner [21] proposes mechanisms whereby agents can organize themselves into "clumps" based on the similarity of their information needs. Agents communicate with other agents in their proximity, exchanging information about their own needs and the needs of their neighbors. Given some measure of similarity between needs, agents can perform a type of gradient ascent by iteratively joining clumps whose information needs are most similar to their own and communicating with their new neighbors. Kuokka and Harada [32] take a more centralized approach in which information producers and consumers advertise their needs to an intelligent "matchmaking" service. Their approach is thought to be appropriate for domains in which the needs of both producers and consumers of information change rapidly, because all parties can continuously issue modifications that will be made available immediately to interested parties. Similarly, Karakoulas and Ferguson [27] propose market-based approaches to joining information producers and consumers. Their System of Information Gathering Market Based Agents (SIGMA) relies on market dynamics to ensure robust behavior of collections of agents in the face of changes in the available information and the needs of users.

While it is the case that these projects are still in their infancy, they hold out promise as models of CIG-type information systems. These systems view information gathering as a dynamic process where partial results of a search drive further activity and coordinate agent activities in order to beneficially exploit interdependencies between their tasks.

5.4 Enabling Technology

The motivation for the CIG approach presented in this paper is simply the welter of information carrying sites, data formats, and access methods that are currently available. That is, the problems that we address are quite real and are in need of solutions today. While environments such as the one depicted in Figure 1 that motivate our work already exist, that is not enough. CIG assumes the existence of intelligent semi-autonomous agents as well as support within the environment for the operation of such agents. One exciting aspect of the CIG approach is the availability of technology that facilitates the development of both intelligent agents and supporting structures within the environment, allowing them to interact in exactly the manner we require. This section describes existing technology that is relevant to bringing the CIG vision into existence.

For the CIG approach to be successful, it must be possible for agents to adapt to changing demands from both the environment and the user. No single coordination mechanism or organizational structure will suffice in environments as complex as those for which CIG is envisioned. Generic mechanisms for the elaboration of task structures, the formation of organizational structures, and the coordination of multiple agents are crucial. All of these functions must be available dynamically, as the state of the environment and the state of problem solving changes over time. Decker and Lesser's work on the TAEMS architecture [13] and Generalized Partial Global Planning [11] are relevant here.

Telescript technology, developed by General Magic, Inc. [54], provides the tools required to build an intelligent agent-based foundation for a global electronic marketplace. Telescript abandons the traditional remote procedure calling (RPC) model of client/server interaction for the remote programming (RP) approach. Agents, collections of data and procedures, can actually execute on remote machines as complete "programs", allowing them to exist and operate regardless of the state of the user and machine from which they originated. The Telescript world comprises a number of electronic *places* that correspond to individuals or organizations, known as the place's *authority*, in the physical world. One or more Telescript agents can exist in each place, typically for the purpose of conducting some transaction related to the place itself. For example, there may be a `PUBLIC LIBRARY` place occupied by a `LIBRARIAN` agent and one or more additional agents whose authorities are high school students researching term papers. Both places and agents are written in the Telescript programming language. Agents travel from place to place by obtaining a *ticket* that describes and constrains their trip and then executing the `GO` instruction. Agents occupying the same place can interact by executing the `MEET` instruction and presenting a *petition* that describes the nature of the desired meeting. Agents can communicate with other agents not in the current place by obtaining a *connection* (as in the RPC model). Security is addressed in this environment via three different mechanisms. First, the Telescript language is interpreted, denying agents direct access to local computational resources. All agent actions are mediated by the Telescript engine. Second, an agent's authority and *identity* are obtained and validated from the agent's *telename* via cryptographic mechanisms. Lastly, all agents have *permits* that limit their capabilities, such as the places they may visit, the amount of time they may exist, and the amount of money (as measured by *teleclicks*) they may spend.

The Telescript vision clearly provides a path to filling in the missing pieces of our CIG model. The Telescript language makes it possible to construct software agents that travel from place to place, from network to network (Figure 1), in search of information relevant to a query. The fact that intelligent software agents interact in various information bearing electronic places with the proprietors of those places, which are themselves software agents, fits well with the conceptual model in Figure 2c that we adopted. The intelligent retrieval engine in that figure may simply be a Telescript agent that mediates access to some corpus. Resource bounds for agents are made explicit within Telescript, facilitating the use of satisficing search.

The Java programming language is another technology that may be very important in the implementation of future CIG systems [25]. Java allows applets, such as intelligent information retrieval agents, to be transferred between machines on the Internet, much as Telescript agents travel between electronic places. Although Java was not designed explicitly to provide the infrastructure for an electronic marketplace (as Telescript was) its inherent flexibility and increasingly widespread acceptance make it an attractive candidate for the implementation of CIG systems. Every Java-aware browser on the Internet (of which there are literally millions) is a potential intelligent retrieval agent or intelligent search engine given the right Java applet.

One key aspect of the CIG paradigm is the construction of a complete and coherent answer to a query based on data gathered from a variety of sources, including unstructured text. For an agent to reason about the content of a document and how it fits into the evolving response to a query, the agent must be able to identify portions of the text that are relevant to the query. It is insufficient to simply know that the document as a whole is relevant. Recent work in information extraction is providing the tools that make the identification of relevant items of information within a document possible [9]. Information extraction systems are typically rule-based systems that convert unstructured documents into a case-frame representation. For example, a document on terrorism might have slots for the names of the perpetrators and their targets, the instrument used, the location of the attack, etc. Given an explicit representation of the relevant information in a document, an information retrieval agent can reason about its place in the overall response to a query.

As should be clear from earlier sections of this paper, research in DAI has produced an extensive body of work aimed directly at issues such as guiding the distributed search process of multiple agents. Overall, the CIG picture becomes fairly clear: there is a crying need for technology that addresses information acquisition in complex, distributed environments; products such as Telescript can provide the foundation for intelligent semi-autonomous software agents; and DPS research provides the mechanisms for successfully guiding and controlling the activities of multiple, distributed agents to efficiently manage the complexity involved in complex information gathering systems.

6 Implications and Conclusion

Information Gathering, whether centralized or as it is being handled by newer systems in a distributed setting, has traditionally been a one-shot process: a query is formulated, relevant corpora are identified and interrogated, and the sum of the individual responses is presented as the result of the query. Unfortunately, the complexity of today's networked environments limits the scope of such a model. Among the contributing factors to this complexity are heterogeneity in both hardware and software, uncertainty arising from single point failures, varying costs of access to both network transport and information itself, and the tremendous number of sites carrying potentially useful data. Relevant information in this environment is hard won, and cannot simply be "retrieved" as if from some amorphous distributed encyclopedia with a complete and accurate index. In the previous sections, we attempted to convince the reader that distributed information acquisition tasks characterized by complex, heterogeneous and unstructured data environments can instead be viewed as a distributed problem-solving task within the FA/C paradigm. The benefits of such a view not only stem from the fact that it provides a comprehensive conceptual model for the myriad of methods being proposed for IIR, but also from the fact that the view provides a direct map from the wealth of existing methods in MAS to the IG domain. These methods have evolved over more than a decade, since the time the FA/C paradigm was first proposed [37]. Below, we discuss various techniques and systems from MAS that may have direct bearing on CIG viewed as a DPS task. These methods were originally proposed in contexts different from information gathering, and most of them were developed as techniques to study, understand and exploit various aspects of the FA/C paradigm.

At the risk of being repetitive, we will first summarize the highlights of the FA/C paradigm along with their relevance to the IR task. Complex distributed search spaces are characterized by various soft and hard constraining *goal/task interrelationships*. The ability to exploit these interrelationships to avoid negative interactions and take advantage of positive interactions can enhance search quality by providing better solutions in less time. In a CIG task, potentially useful constraints may exist between different pieces of information, either via content or as a function of problem solving activity. The discovery and exploitation of such constraints is necessarily a dynamic and incremental process that occurs during problem-solving and entails communication of partial results among agents in a timely and selective manner to augment each agent's local view with a more global view. Given the incomplete nature of the local views of the individual agents, another important aspect of the FA/C paradigm is the explicit recognition of the role of solution and control uncertainty. Coupled with the fact that resources and time for conducting a search are limited in real-life problems, this leads to the notion of *satisficing search*. The environment in an information acquisition task is characterized by the fact that the supply of available data is almost limitless, whereas time, money and computational resources are not. Rather than being able to develop an exhaustively complete and accurate response to a query, intermediate results from disparate sources must be pieced together to form consistent clusters of information that can be incrementally refined to form a more accurate solution depending on the extent of available resources and time. Another aspect of the FA/C paradigm is the explicit recognition and exploitation or avoidance of *redundancy*, leading to increased robustness or decreased resource demands depending on the context and the structure of the domain.

We now briefly review a few implemented aspects of the FA/C paradigm that have direct relevance to the CIG task. Decker and Lesser [11, 12, 14] provide detailed studies of quantitative trade-offs involved in explicit recognition and exploitation of task interrelationships for use in multi-agent coordination. Von Martial's work [40] on coordination in multi-agent planning, which uses favors goal interrelationships and temporal interactions, is also relevant here. Garvey and Lesser [22] discuss design-to-time algorithms which basically endow the local problem solver with abilities to deal with real-time considerations and goal interdependencies. Such a scheduler is, perforce, satisficing in the solutions it provides and relies on the use of approximate processing techniques. Carver and Lesser [8, 7, 5, 6] present RESUN and its distributed derivative DRESUN as architectures that explicitly recognize and resolve uncertainties associated with the partial, evolving solutions in the interpretation domain. Interpretation is viewed as an incremental process of resolving sources of uncertainty (SOUs) through directed and intentional accrual of evidence. For example, uncertainty may arise because current evidence only paritially matches an expected model, confirming evidence has not yet been established, evidence in support of conflicting hypotheses exists, etc. From among a number of SOUs at a given time step, the next SOU is selected and pursued, which involves acting to resolve the uncertainty represented by this SOU. Each action may in turn result in the instantiation of further SOUs. This cycle is repetitively performed until the termination criteria are achieved. This seamless integration of data-driven bottom-up and goal-driven top-down processes opens up a huge set of opportunities for information acquisition systems. Information on hand can in turn serve to instantiate and actively direct further retrieval to resolve the deficiencies in the partial data. Other work in MAS, though not directly falling under the umbrella of the FA/C approach, could act as enabling technologies for multi-agent based CIG. The contract net [48, 47] is a top-down work allocation scheme among agent sets, where an agent wanting to delegate or contract out a piece of work for some reason announces the work to the agent set. The agents with capabilities to accomplish it respond with a bid, and the announcing agent allocates the work to the agent with the best bid. The contract net framework can be used to enforce a problem-dependent organization among a set of DPS agents. Along another direction is the work on selfish agents [46, 58]. Unlike the agents discussed previously, a selfish agent places self-interest above any "global" requirements and cooperates to the extent of serving its own interests. In a market economy of information servers [53] as suppliers and "free-lancing" agents as consumers, the selfishness assumption may become essential because these agents may not have been engineered from a single source.

In closing, we hope that this paper encourages IR system designers to take a radically new view of information gathering as a distributed problem solving activity. While there are intelligent agent-based systems in existing literature, the distinguishing feature of our proposal is *cooperative retrieval*, whereby the agents explicitly communicate with each other to control the distributed information acquisition process through detection and exploitation of interrelationships between the goal structures in various agents. We also suggest that existing methods in MAS can serve to leverage future implementations of IG systems based on this view.

7 Acknowledgments

This material is based upon work supported by the following organizations: the NSF Center for Intelligent Information Retrieval, the Department of Commerce and the Library of Congress under Cooperative Agreement #EEC-9209623; the Department of the Navy, Office of the Chief of Naval Research, under Grant No. N00014-95-1-1198; the National Science Foundation under Grant No. IRI-9523419. The content of this paper does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

We thank Norm Carver for his contributions to the creation of Figure 1 and Keith Decker for the example on Mac information retrieval adopted from his thesis. We would also like to thank J. Callan, N. Carver, K. Decker, Qiegang Long, Dan Neiman and T. Sandholm for their insightful discussions and comments on this work.

References

[1] Boddy, M., and Dean, T. Solving Time-Dependent Planning Problems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI., August, 1989.

[2] Bowman, M. C.; Danzig, P. B.; Manber, U.; and Schwartz, M. F. Scalable Internet Resource Discovery: Research Problems and Approaches. In *Communications of the ACM*, 37(8), pp. 98–107, 1994.

[3] Callan, J. P.; Croft, W. B.; and Harding, S. M. The INQUERY Retrieval System. In *Proceedings of the 3rd International Conference on Database and Expert Systems Applications*, pp 78–83, 1992.

[4] Callan, J. P.; Lu, Zhihong; and Croft, W. B. Searching Distributed Collections with Inference Networks. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 21–28, Seattle, Wa., July 1995.

[5] Carver, N., and Lesser, V. R. The DRESUN Testbed for Research in FA/C Distributed Situation Assessment: Extensions to the Model of External Evidence. In *Proceedings of the International Conference on Multiagent Systems*, San Francisco, CA, July, 1995.

[6] Carver, N., and Lesser, V. R. A First Step Toward the Formal Analysis of Solution Quality in FA/C Distributed Interpretation Systems. In *Proceedings of 13th International Distributed Artificial Intelligence Workshop*, Seattle, WA., July 1994. Also available as Technical Report 94-37 (ftp://ftp.cs.umass.edu/pub/lesser/carver-94-37.ps), Department of Computer Science, University of Massachusetts, Amherst.

[7] Carver, N.; Cvetanovic, Z.; and Lesser, V. R. Sophisticated Cooperation in FA/C Distributed Problem Solving Systems. In *Proceedings of AAAI-91*, pp. 191–198, 1991.

[8] Carver, N., and Lesser, V. R. A New Framework for Sensor Interpretation: Planning to Resolve Sources of Uncertainty. In *Proceedings of AAAI-91*, pp. 724-731, 1991.

[9] Cowie, J., and Lehnert, W. Information Extraction. In *Communications of the Association of Computing Machinery*, Vol. 39 (1), pp. 80–91, 1996.

[10] Davies, Winton H. E., and Edwards, Pete. Agent-Based Knowledge Discovery. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments.* Stanford University, Stanford, CA, March, 1995.

[11] Decker, K. S., and Lesser, V. R. Generalizing the Partial Global Planning Algorithm. In *The International Journal of Intelligent and Cooperative Information Systems,* 1(2), pp. 319–346, June 1992.

[12] Decker, K. S., and Lesser, V. R. Quantitative Modeling of Complex Computational Task Environments. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp 217–224, Washington, 1993.

[13] Decker, K. S., and Lesser, V. R. Quantitative modeling of complex environments. In *The International Journal of Intelligent Systems in Accounting, Finance, and Management*, 2(4), Special issue on Mathematical and Computational Models of Organizations: Models and Characteristics of Agent Behavior, December 1993, pp 215–234.

[14] Decker, K. S., and Lesser, V. R. Designing a Family of Coordination Algorithms. In *Proceedings of the First International Conference on Multi-Agent Systems*, San Francisco, CA, July 1995.

[15] Decker, K. S. *Environment Centered Analysis and Design of Coordination Mechanisms*, Ph.D. Thesis, Department of Computer Science, University of Massachusetts, Amherst, 1994.

[16] Decker, K. S.; Lesser, V. R.; Nagendraprasad, M. V.; and Wagner, T. MACRON: An Architecture for Multi-agent Cooperative Information Gathering. In *Proccedings of the CIKM Workshop on Intelligent Information Agents*, Baltimore, Maryland, December, 1995.

[17] Decker, K. S., and Sycara, Katia. Designing Reusable Behaviors for Information Agents, Unpublished Report, 1996.

[18] Durfee, E. H.; Kiskis, D.; and Birmingham, W. P. The Agent Architecture of the University of Michigan Digital Library. In *The IEE/BCS Journal of Software Engineering special issue on Developing Agent Based Systems*, 1997.

[19] Durfee, E. H.; Lesser, V. R.; and Corkill, D. D. Coherent Cooperation among Communicating Problem Solvers. In *IEEE Transactions on Computers*, vol 36, pp. 1275–1291, 1987.

[20] Fikes, R.; Englemore, R.; Farquhar, A.; and Pratt, W. Network-Based Information Brokers. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford University, Stanford, CA, March, 1995.

[21] Foner, L. N. Clustering and Information Sharing in an Ecology of Cooperating Agents. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford University, Stanford, CA, March, 1995.

[22] Garvey A., and Lesser, V. R. Design-to-time Real-Time Scheduling. In *IEEE Transactions on Systems, Man, and Cybernetics: Special Issue on Scheduling, Planning, and Control*, 23(6), 1993.

[23] Holte, R. C., and Drummond, C. A Learning Apprentice for Browsing. In *Working Notes of the AAAI Spring Symposium on Software Agents*, 1994, pp. 37–42.

[24] Huhns, M.; Mukhopadhyay, U.; Stephens, L. M.; and Bonnell, R. DAI for Document Retrieval: The MINDS Project. In *Distributed Artificial Intelligence* ed. by M. Huhns, Pittman Publishing/Morgan Kauffmann Pub., pp. 249–284.

[25] The Java home page - http://java.sun.com

[26] Jawadi, R.S., and Su, S.Y.W. Incorporating Flexible and Expressive Rule Control in a Graph-based Transaction Framework. In *Proceedings of the 4th International Conference on Database Systems For Advanced Applications*, pp. 350–357, Singapore, April, 1995.

[27] Karakoulas, G. J., and Ferguson, I. A. A Computational Market for Information Filtering in Multi-Dimensional Spaces. In *Working Notes of the AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, 1995.

[28] Kirk, T.; Levy, Alon Y.; Sagiv, Y.; and Srivastava, D. The Information Manifold. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford University, Stanford, CA, March, 1995.

[29] Knoblock, C. A., and Arens, Y. An Architecture for Information Retrieval Agents. In *Working Notes of the AAAI Spring Symposium on Software Agents*, pp. 49–56, 1994.

[30] Knoblock, Craig A., and Levy, Alon Y. Exploiting Run-Time Information for Efficient Processing of Queries. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford University, Stanford, CA, March, 1995.

[31] Knoblock, C. A.; Arens, Y.; and Hsu, C. Cooperating agents for information retrieval. In *Proceedings of the 2nd International Conference on Cooperating Information Systems*, 1994.

[32] Kuokka, D., and Harada, L. Supporting Information Retrieval via Matchmaking. In *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford University, Stanford, CA, March, 1995.

[33] Laasri, B.; Laasri, H.; Lander, S. E.; and Lesser, V. R. A Generic Model for Intelligent Negotiating Agents. In *The International Journal of Intelligent and Cooperative Information Systems* 1(2), pp. 291–317, June, 1992.

[34] Lehnert, W.; Cardie,. C.; Fisher, D.; McCarthy, J.; Riloff, E.; and Soderland. S. University of Massachusetts: Description of the CIRCUS System as Used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pp 282–288, 1992.

[35] Lander, S. E., and Lesser, V. R. Understanding the Role of Negotiation in Distributed Search Among Heterogeneous Agents. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 438–444, Chambery, France, 1993.

[36] Lander, S. E. *Distributed Search in Heterogeneous and Reusable Multi-Agent Systems*, Ph.D. Thesis, Department of Computer Science, University of Massachusetts, Amherst, 1993.

[37] Lesser, V. R., and Corkill, D. D. Functionally Accurate, Cooperative Distributed Systems. In *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1), pp. 81–96, 1981.

[38] Lesser, V. R.; Pavlin, J.; and Durfee, E. H. Approximate processing in real-time problem solving. In *AI Magazine*, vol 9, no. 1, 1988, pp 49–61.

[39] Lesser, V. R. A retrospective view of FA/C distributed problem solving. In *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6), pp. 1346–1363, 1991.

[40] Martial, F. v. Multiagent Plan Relationships. In *Proceedings of the Ninth Workshop on Distributed AI*, September, 1989.

[41] Nagendraprasad, M. V.; Lander, S. E.; and Lesser, V. R. Retrieval and Reasoning in Distributed Case Based Systems. To appear in *The Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries.* Also as UMASS CS Technical Report 95-27, 1995.

[42] Ram, A. Question-driven Understanding: An integrated theory of story understanding, memory and learning. Ph.D. Thesis, RR#710, Yale University, New Haven, CT, 1989.

[43] Ram, A., and Hunter, L. A Goal-based Approach to Intelligent Information Retrieval. In *Proceedings of Eighth International Conference on Machine Learning*, Chicago, IL, 1991.

[44] Rissland, Edwina L., and Daniels, Jody J. Using CBR to Drive IR. In the *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.

[45] Rus, D., and Subramanian, D. Designing Structure Based Information Agents. In *Working Notes of the AAAI Spring Symposium on Software Agents*, pp. 79–86, 1994.

[46] Sandholm, T. W. An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 256–262, Washington DC, 1993.

[47] Sandholm, T. W., and Lesser, V. R. Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework. In *Proccedings of the First International Conference on Multiagent Systems*, pp. 328–335, San Fransisco, CA, 1995.

[48] Smith, R. G. The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver. In *IEEE Transactions on Computers*, C-29(12), pp. 1104–1113, 1980.

[49] Soderland, S.; Fisher, D.; Aseltine, J.; and Lehnert, W. CRYSTAL: Inducing a Conceptual Dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1314–1319, 1995.

[50] Soderland, S.; Fisher, D.; Aseltine, J.; and Lehnert, W. Issues in Inductive Learning of Domain-Specific Text Extraction Rules. In *Proceedings of the Workshop on New Approaches to Learning for Natural Language Processing at the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[51] *Software Agents*, Working Notes of the AAAI Spring Symposium, 1994.

[52] Vorhees, E. M. Software Agents for Information Retrieval. In *Working Notes of the AAAI Spring Symposium on Software Agents*, 1994, pp. 126–129.

[53] Wellman, M. P. A market-oriented programming environment and its application to distributed multicommodity flow problems. In *Journal of Artificial Intelligence Research* 1:1-23, 1993.

[54] White, J. E. Telescript Technology: The Foundation for Electronic Marketplace. General Magic White paper, 1994.

[55] Whitehair, R., and Lesser, V. R. A Framework for the Analysis of Sophisticated Control in Interpretation Systems. Technical Report Number 93-53, Dept.of Computer Science, University of Massachusetts, Amherst, 1993.

[56] Zilberstein, S. Meta-Level Control of Approximate Reasoning: A Decision Theoretic Approach. In *Proceedings of the Eighth International Symposium on Methodologies for Intelligent Systems*, pp. 114–123, Charlotte, North Carolina, October 1994.

[57] Zilberstein, S., and Lesser, V. R. Intelligent Information Gathering Using Decision Models. Technical Report 96-35, Computer Science Department, University of Massachusetts, 1996.

[58] Zlotkin, G., and Rosenschein, J. S. Incomplete Information and Deception in Multi-Agent Negotiation. In the *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 225–231, Sydney, Australia, August 1991.