Cooperative Information Gathering:
A Distributed Problem Solving Approach

Tim Oates, M. V. Nagendra Prasad and Victor R. Lesser[1]
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA.
{oates,nagendra,lesser}@cs.umass.edu
Department of Computer Science
University of Massachusetts

UMass Computer Science Technical Report 94–66-version 2
(11/4/94)

Abstract

With the proliferation of electronically available information, an additional burden has been placed on the implementors of information gathering (IG) systems. The set of data that represents the best response to a query may be the aggregation of data acquired from distributed, heterogeneous information sources. In such environments, we distinguish between two approaches to the problem of information gathering that may be characterized as *distributed processing* and *distributed problem solving*(DPS). The former is characteristic of existing IG systems while the latter is the raison d'être for Multi-Agent Systems (MAS). We examine features of problems that point to the need for a DPS approach, and discuss the benefits of viewing information gathering as distributed problem solving (which subsumes distributed processing). This approach, called Cooperative Information Gathering (CIG), involves concurrent and asynchronous access and composition of associated information spread across a network of information servers by a group of intelligent agents. 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. Finally, we briefly survey current work on distributed and agent-based approaches to Information Gathering.

[1] This material is based upon work supported by the NSF Center for Intelligent Information Retrieval, ONR contract N00014-92-J-1450, ARPA under ONR contract N00014-92-J-1698. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

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 freely available at a multitude of sites to anyone with access to the Internet. Early information retrieval (IR) systems assumed that the 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. 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, it is no longer possible for a user to know which of the possible sources is most relevant to a query or to manually submit a query against more than a very small fraction of those sources. In addition, the heterogeneity and the lack of uniform structure in the information databases rules 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) is an activity involving pro-active acquisition of information, from possibly heterogeneous sources, in response to a complex query that may require the system to possess capabilities like reasoning, representation and inferencing. Traditional IR is a limited sub-case of such information gathering systems. In addition to the complexity of query specification, control of the acquisition process may itself be complex and dynamic in IG systems. On the other hand, queries in an IR system generally map onto static, pre-specified retrieval plans. In this paper, we 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. 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. This paper draws upon a long history of thought in distributed problem solving (DPS) to present a model of this type of cooperative information acquisition.

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. One can easily imagine Telescript servers[36] that receive queries and act as regional planning sites, either further decomposing the search into subregions or sending agents to local corpora for data retrieval. The efficiency and

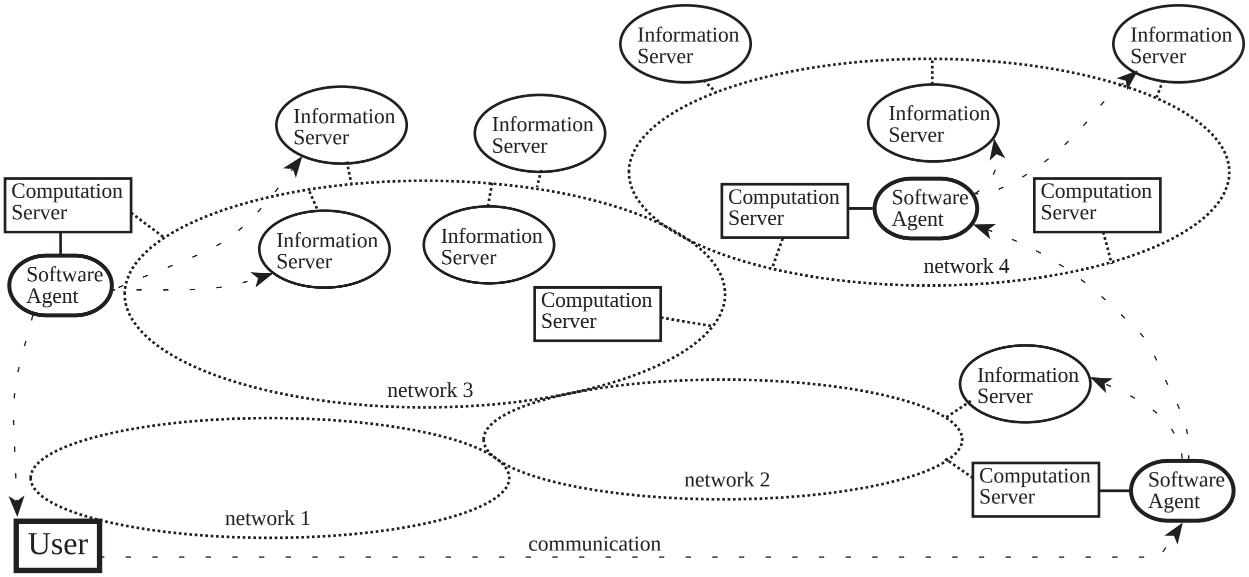


Figure 1: An example milieu for distributed information retrieval

the quality of the local search activity of an agent can be affected by the partial results produced by other agents working in parallel. 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 the parent agents and synthesized into a coherent response to the query.

The complexity of the modern information carrying landscape requires a sophisticated view in which 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 also 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 have an any-time flavor, making partial and incomplete results available to users as the search for information proceeds. That may include 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 may need to employ multiple search methods that allow them to control the tradeoff between completeness, quality, and precision. 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? We follow this section with an attempt to formalize various aspects of the model. 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

We now introduce a dichotomy of distributed information systems based on the decomposability of the acquisition processes being executed concurrently. 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 and we will further dwell on it below. Details of the distributed problem solving model will be expounded 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[13, 22]. 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. 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. [23, 25] 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 uncovered via communication of problem solving activities, such as receiving partial results or meta-information, these interdependencies 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 the implications of those interactions. 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. That 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 that 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, when 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.

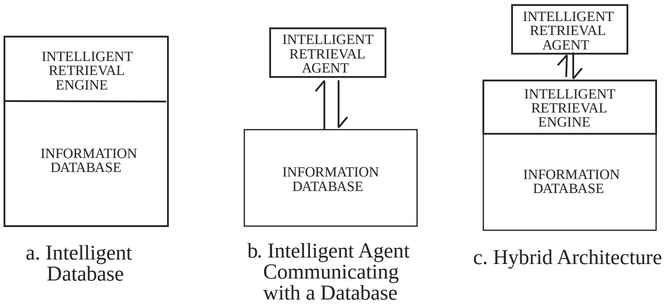
3 Cooperative Information Gathering

Recent trends in information retrieval show an evolution from a relatively syntax-oriented retrieval systems to more semantically guided systems [29, 34] known as Intelligent Information Retrieval (IIR) Systems. These systems are guided by task-level requirements, rather than by just the syntax of the queries, to establish an association and retrieve information from the stored structures to which they have access [29]. In this report we go a step further and propose a model for Cooperative Information Gathering (CIG), where a group of 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 and then discuss the distributed problem solving model of CIG.

3.1 Intelligent Information Retrieval (IIR)

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 like specialization is done 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 [30]: "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 [29, 34] 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. Information servers may actively employ the techniques described in the remainder of Section 3 to extend their scope and thus their ability to respond effectively to user queries. That is, information servers may seek out and build connections with other servers in the network that contain data related to information maintained locally. This view treats information servers as intelligent agents with their own goals, adding to both the richness and 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 the abstraction of the contents in 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. We adopt the hybrid view of IIR (Figure 2c) in our further discussions.


INTELLIGENT RETRIEVAL ENGINE
INFORMATION DATABASE
INTELLIGENT RETRIEVAL AGENT
INFORMATION DATABASE
INTELLIGENT RETRIEVAL AGENT
INTELLIGENT RETRIEVAL ENGINE
INFORMATION DATABASE

a. Intelligent Database

b. Intelligent Agent Communicating with a Database

c. Hybrid Architecture

Figure 2: Conceptual Models for Intelligent Information Retrieval

3.2 CIG as Distributed Problem Solving

The proliferation of network-based information systems motivates the need for distributed information acquisition systems. On the other hand, 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 the 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 the 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 [25]) 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. Further, there are no restrictions on the consistency of the goal structure.

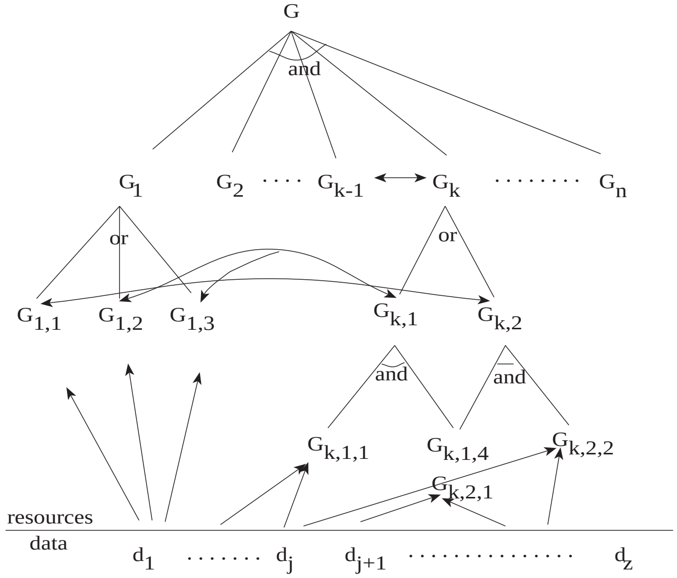


Figure 3: A goal tree: The G's represent goals and the d's represent data produced by queries to information servers. The double headed arrows between goals indicate that the goals are interdependent. The arrows between data and goals indicate that the data is required for that goal's solution.

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[9, 11, 26]. 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 less 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 reformed query can produce results that not only satisfy one goal but also another subgoal). Detection and the use of such goal interrelationships for efficient coordination is a hard problem in complex AI systems[37].

Now consider the case where a goal tree is distributed across multiple agents, none of which may have a complete global view of the goal 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 leads to added complexity in an already complex situation as discussed above. Figure 4 (from [25]) 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.

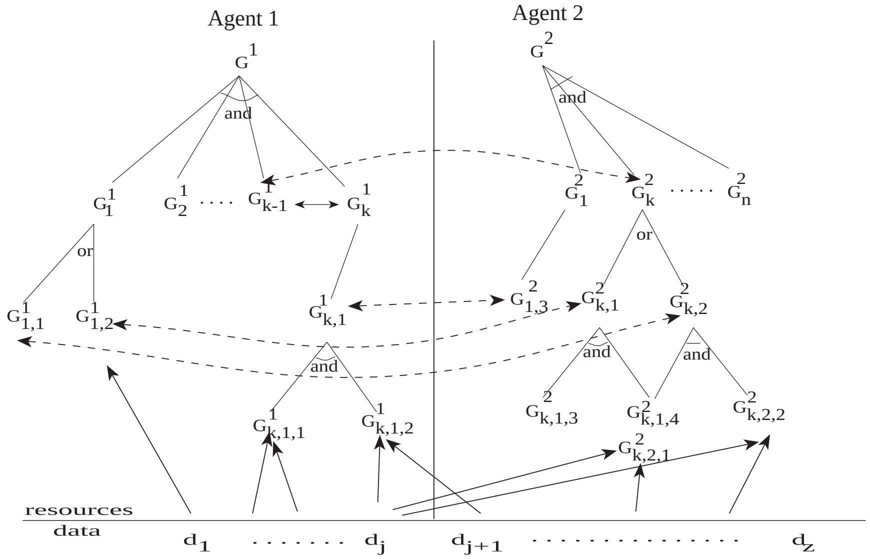


Figure 4: A distributed goal tree: The goal tree of previous figure is distributed with partial replication between two agents. The dotted arrows indicate interdependencies among goals and data in different agents.

Thus in the case of DPS, agents have to make their decisions facing additional uncertainties due to 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 posses certain abilities. Based on their partial views, each of the agents can 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 detection interrelations between various part 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 [22] as taking place in four stages: *problem formulation, focus-of-attention, allocation,* and *achievement* (see Figure 5 which has been slightly modified from [22]). The discussion of goal trees up to this point has treated them as static structures. However, as is clear from Figure 5, 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 5 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 going on asynchronously with different collections of goals. Agents involved in this process concurrently and asynchronously move through the various stages in a dynamic manner until the global goal is sufficiently satisfied given time and resource constraints.

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.

Before we go on to discuss the details of our model, we introduce an example that will be

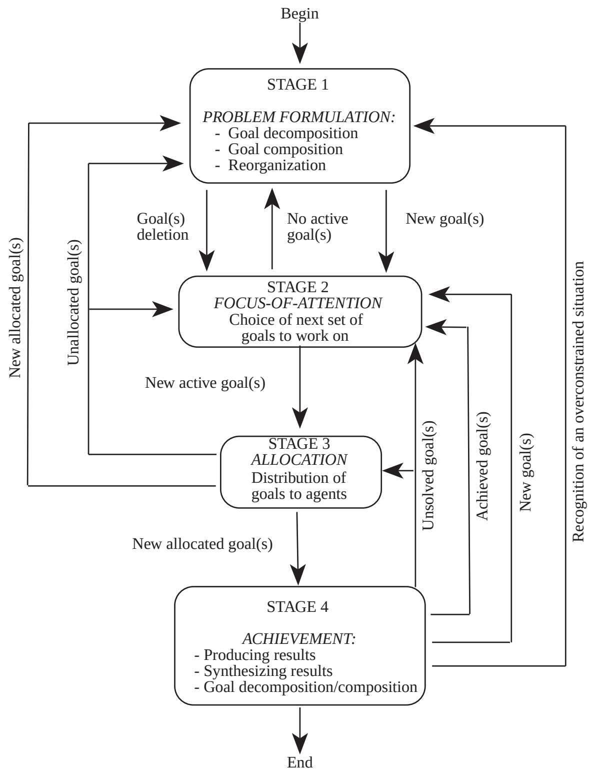


Figure 5: A goal-based view of the stages of Distributed Problem Solving

used subsequently to highlight various concepts of the model. Imagine a user deciding to go on a vacation. She gives a travel planner program a few of her preferences for the vacation. Say she gives her specifications as a vacation for 3 or 4 days around July 20th, preferably in Massachusetts. In addition, either through user specifications or through user modeling, the travel planner knows that this particular user prefers historical sites or nature spots. The travel planner has to plan for at least four different aspects of the vacation — places/sites to visit, weather situation, accommodation and conveyance. So it sends off four agents to deal with the corresponding types of databases. At the lowest level, this process involves gathering data from information repositories like historical text corpora, tourism information databases (possibly object-oriented databases with attributes describing various tourist attractions), weather report servers that could contain structured weather report forms or unstructured satellite imagery, and audio text of weather reports on local radio. Also, towns, parks and hotels could have WWW pages containing information about the entities, including short-term items such as a calendar of events, expected local weather, etc. An agent planning for "fun time" over the week end could access a database of local newspapers and look at the "art and living" sections in them. 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. During the process of query planning and information retrieval, the agents have to interact both among themselves and with the travel planner. For example, severe weather conditions like an expected hurricane on July 20th and 21st might preclude travel on those days. If this information is available to the agent planning conveyance, it can avoid search for any travel conveyance on those days (see Figure 6). Also, if the weather is rainy for a given day, then planning for nature spots may not be a good thing to do. The planner should amend its present plan to concentrate on historical places that mainly involve indoor activities; like visiting the witch museums of Salem.

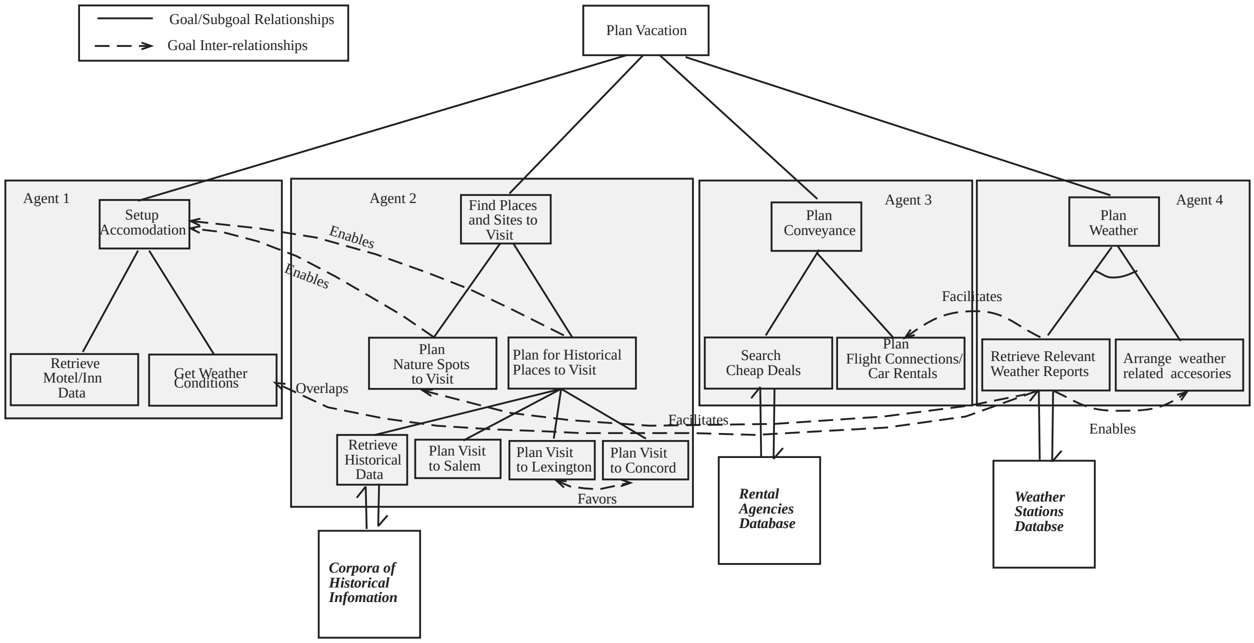


Figure 6: Goal Tree for the Vacation Planning Example

3.2.1 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. We will shortly discuss examples of both. 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"[2] which are representative of the content of the documents in terms of a set of domain primitives.

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

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

[2] See [19, 28] for some of the recent progress on this aspect.

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. [5][8, 7, 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 reducing the amount of computation required[10]. 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 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 6 shows some of the subgoal interrelationships in the travel planning query. The planning of flight connections and car rentals has weather report data as an optional goal specification parameter. Acquisition of weather reports facilitates the planning process for car rentals and flight connections by eliminating or attributing low importance to retrieval of flight reservation and car rental availability on those days when the weather is not conducive to travel. Even though Agent 2 can plan for car rentals and flights without the weather data, if there is no time pressure then it is better off delaying planning for the flight schedule until the availability of weather information. In the meanwhile, it can search for airlines offering cheaper deals. Similarly, acquisition of weather reports facilitates planning for outdoor spots. On the other hand, a plan of the places to visit will enable the accommodation agent to start its work on planning and querying for lodging. The place at which accommodations are to be secured is a required goal specification parameter for the "Setup Accommodation" goal. Note that abstractions of plans are all that is needed for the accommodation agent to start its work. Thus, while Agent 2 is fleshing out the details of the abstract plan of the places to visit, Agent 1 can, in parallel, start its work. Similarly, a favors interrelationship says that once you have made the effort to design a plan to go to Concord, a plan for going to Lexington is obtained by minor modifications to the plan for Concord. An overlaps interrelationship says that the two agents involved may be doing similar work and can hence benefit by sharing their partial results.

Our multi-agent Case Based Reasoning (CBR) system called CBR-TEAM[27] is another example of a sophisticated system exploiting constraints generated by sharing results of a partial search. CBR-TEAM is built on top of a generic negotiated search framework called TEAM[20, 21] and comprises agents that retrieve partial component designs from local case bases and cooperatively assemble them into an overall design for steam condensers. During the process of integration, there may be mismatches at the interfaces of component designs proposed by the agents. This results in a process of negotiation among the agents leading to an exchange of information on detected conflicts. The agents do subsequent rounds of retrieval, but this time with an enhanced view of the requirements of other agents (due to the exchange of conflict data). This process goes on iteratively until requisite number of conflict-free designs are assembled. A conflict-free design represents a consistent cluster of design subcomponents. CBR-TEAM is directly relevant to the information gathering model we propose. Information acquired by an agent can be related to the requirements of information acquisition in another agent. Viewing partial results as information relevant to a query opens up a rich set of possible subproblem interrelationships that may be beneficially exploited. Figure 7 shows an example that highlights the same issues in the document retrieval domain (modified from [12]). For a given query, there may be many sources of relevant information. Product reviews often exist on-line, or may 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 the required review and then retrieve it. 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 Agent1'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 must be consistent. For example, during the process of retrieval of product review discussed above via WWW server, two sites may quote different prices for the product. Product review information at an FTP site accessed via WWW may contain outdated prices, whereas a link to the seller database in a 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.

3.2.2 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 than performing an exhaustive search, one must attempt to locate the "best" response possible given the time and resource constraints. That is, the information gathering process must be *satisficing* along various dimensions like precision, quality, etc[24]. As a motivating

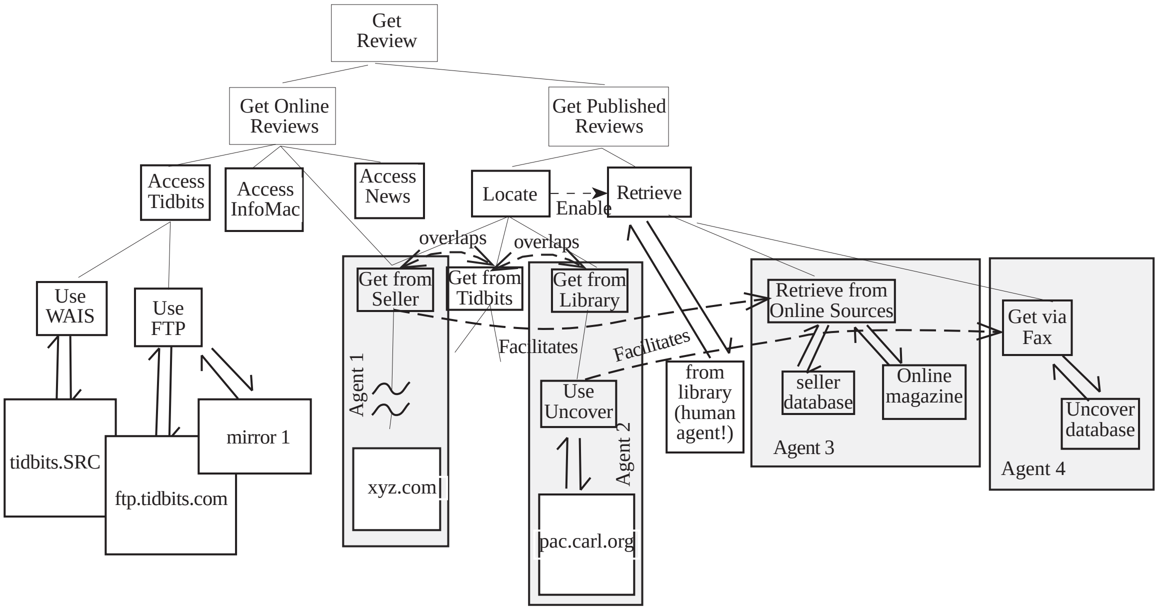


Figure 7: A goal tree for retrieving Macintosh related product reviews

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, 14, 38, 39]. 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.

3.2.3 Redundancy

Redundancy in distributed search, either in the form of replication of data at multiple sites or 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 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 the uncertainty in the retrieved data by providing additional constraints.

3.3 A Formal Model of CIG

In this section we present a formal model of cooperative information gathering. At a very coarse level, CIG can be decomposed into two parts: problem solving and the environment. Problem solving encompasses a number of things that change with each data acquisition problem, including the actual text of the query, the number and organization of the agents used to form a response, as well as the response itself and how the quality of the response is evaluated. The query and how it is handled are largely determined by the user and the information acquisition system being employed. Contrast this with the environment which is typically not controlled in any meaningful way by the user. The environment comprises communication networks, servers for computation and information retrieval, and individual databases (refer to Figure 1). The environment may be highly dynamic, with databases, servers, and entire networks appearing and disappearing over time.

3.3.1 The Environment

To facilitate identification of promising data sources and therefore query planning, various entities in the environment will be labeled with abstract descriptions of the information they contain. Let Ψ be a set of 3-tuples of the form (τ, θ, γ) where each 3-tuple describes the utility of the entity bearing Ψ with respect to a given subject or type of information. τ is a topic of expertise for the entity expressed in the form of a query. Note that τ may be as simple as a single key word or it may be arbitrarily complex, it may be general in scope or very specific. $\theta \in [0,1]$ rates the quality of the information available within the entity with respect to τ, with $\theta = 1$ being the highest quality. We can easily treat θ as a vector that describes various aspects of quality such as completeness, certitude, and precision. γ indicates the amount of information relevant to τ that is available. γ may be expressed as a percentage for entities such as databases, or it may be ordinal indicating LOW, MEDIUM, or HIGH content for entities such as networks. Compare this approach to content abstraction with the one presented in Huhns et. al. [16] where similar information is kept by each user in the form of a set of 4-tuples - $[User1, User2, Keyword, CertaintyFactor]$. Rus and Subramanian [30] acknowledge the importance of data abstraction by viewing the acquisition of partial models of data as a first step in the intelligent information retrieval process that can serve to guide subsequent data acquisition.

Access to resources in the environment is assumed to be neither free nor instantaneous. We define a cost function $\mathcal{C}$ such that $\mathcal{C}(r,l)$ is the cost of accessing resource r from location l. Clearly, the interpretation of $\mathcal{C}$ may vary with the context and may in fact be dynamic, possibly based on a negotiation process between the agent and the information server. For access to networks, $\mathcal{C}$ may be cost per kilobyte of data transmitted, for databases it may be cost per document searched or per document retrieved, for computation servers it may be cost per CPU second used. Similarly, we define a speed of access function $\mathcal{S}$ such that $\mathcal{S}(r,l)$ is the speed with which resource r may be accessed from location l. The interpretation of $\mathcal{S}$ may also vary with context taking on values such as kilobytes per second for communication between networks, documents searched per second for information servers, and cycles per second for computation servers. For both $\mathcal{C}$ and $\mathcal{S}$ it may be the case that the value of the function is unknown. Also, they may return ordinal values such as LOW or HIGH rather than numeric values.

At the highest level, the information retrieval environment is broken down into a set of regions or networks (refer to Figure 1). Let $\mathcal{N} = (\Psi, \kappa, \iota, \delta)$ be a network. Ψ is an abstract description of the information available in $\mathcal{N}$ (as described above). κ is a set of computation servers, ι is a set of the information servers, and δ is a set of the individual databases, all contained within $\mathcal{N}$. Computation servers make it possible for software agents to actually exist or execute within the various networks rather than solely on the machine from which a query is issued. Clearly, searching through data locally will be much less expensive than shipping it across a network and performing the search remotely. We denote an individual computation server $\mathcal{K}$. Information servers contain and execute the programs that store, search, and retrieve data. All access to data is mediated by an information server. Let $\mathcal{I} = (\Psi, \delta)$ be an information server where Ψ is an abstract description of the information available via the server, and δ is the set of the databases to which the server has access. For example, the environment depicted in Figure 1 contains four networks: $\mathcal{N}_1$ – $\mathcal{N}_4$. The network on which the user resides, $\mathcal{N}_1$, provides no resources for information retrieval and would therefore be represented as follows in our notation: $\mathcal{N}_1 = (\emptyset, \emptyset, \emptyset, \emptyset)$. The fourth network in the figure contains two computation servers, three information servers, and some number of individual databases: $\mathcal{N}_4 = (\Psi_4, \{\mathcal{K}_{4,1}, \mathcal{K}_{4,2}\}, \{\mathcal{I}_{4,1}, \mathcal{I}_{4,2}, \mathcal{I}_{4,3}\}, \{\mathcal{D}_{4,1}, \ldots, \mathcal{D}_{4,n}\})$.

At the lowest level, information in the environment is collected into individual databases. Let $\mathcal{D} = (\Psi, \gamma)$ be a database where Ψ is an abstract description of the information available in the database and γ indicates the size of the corpus. Even though a large database may contain more information relevant to a given query than a small one, the costs associated with searching the large database may be prohibitive. Our characterization of a database is purposefully simple and abstract, since agents may interact with a large variety of sources ranging from relational databases to corpora composed solely of text. We assume that IIR agents will have expertise in accessing particular data sources and formats. Our only requirement is that an overview or abstraction of the information contained in each database be available.

3.3.2 Problem Solving

Let $\mathcal{Q} = (\phi, \pi)$ be a query where ϕ is the text of the query and π is a set of parameter/value pairs associated with the query. We do not explicitly specify the syntax or semantics of ϕ due to our reliance on an intelligent information retrieval model. We assume that the query (task with information gathering requirements) is handed to an intelligent system that can decompose the query into $\phi = (\phi_1, \phi_2, \ldots, \phi_n)$, such that the individual ϕ_i can be divided among a set of agents. The parameter set π can include such things as hard or soft bounds on resource usage, the level of involvement desired by the human user in terms of supplying relevance feedback or evaluation of partial solutions, and weightings assigned to such factors as information quality, elapsed time, and cost incurred with regard to evaluating the quality of the final solution.

The problem solving entity responsible for accepting and processing the initial query assigns portions of the decomposed query to one or more software agents. Let $\mathcal{A}_i = (\Delta, \alpha, \phi)$ be an individual agent where Δ represents the agent's current view of the state of problem solving, α is the set of other agents with which $\mathcal{A}_i$ can communicate, along with information about their organizational roles and addresses in the network, and ϕ is the query assigned to the agent. Note that ϕ may be a portion of the decomposed top-level query or it may have been generated as a sub-query at any depth in the goal tree (see Figure 6). Agents have, at all times, both a local and a partial non-local view of the current state of problem solving. As agents communicate, they exchange information about the state of their local problem solving, thereby gaining a more complete global view. The global perspective gained in this manner is likely to be inaccurate, incomplete, and out of date. This can lead to inefficiencies in the search from a global perspective, giving rise to a tradeoff between the amount of inter-agent communication and the level of global coherence and consistency. In order for $\mathcal{A}_1$ to communicate with $\mathcal{A}_2$, it must be the case that $\mathcal{A}_2 \in \alpha_1$. That is, the current agent organization must allow for such communication. The organizational role of $\mathcal{A}_2$ affects the nature of the communication that $\mathcal{A}_1$ will initiate. For example, if $\mathcal{A}_2$ resides above $\mathcal{A}_1$ in a hierarchy, communication may be limited to transmission of partial results for aggregation into a final response to the query. The organization may, however, allow lateral communication between the agents for the purpose of exploiting non-local partial results. Note that we may view the human user who initiated the query as an agent, perhaps $\mathcal{A}_0$. The user may actively take part in the information retrieval process by either searching in parallel with the software agents or by providing relevance feedback on their partial solutions.

3.3.3 An Example

Consider the vacation planning example from Section 3.2 and its goal tree as shown in Figure 6. The vacation planning system is given a top level goal of planning a vacation to Massachusetts, entailing certain information gathering needs. The top level query of the system is roughly $\phi =$ "historical and natural sites of interest in Massachusetts that may be visited from July 20th - 24th." ϕ can be decomposed into ($\phi_1 =$ "accommodations close to historical and natural sites of interest in MA from July 20th - 24th", $\phi_2 =$ "historical and natural sites of interest in MA that may be visited from July 20th - 24th", $\phi_3 =$"means of travel to and within MA from July 20th - 24th", $\phi_4 =$ "weather forecast for MA from July 20th - 24th"). Note that individual ϕ_i may be decomposed even further. For example, $\phi_1 = (\phi_{1,1} =$ "motel/inn data", $\phi_{1,2} =$ "weather forecast for MA"). Since $\mathcal{A}_1$ knows that $\mathcal{A}_4$ is tasked with obtaining weather forecasts (see the explicit description of $\mathcal{A}_1$ below), the *overlaps* interrelationship that exists between their goals (Figure 6) will be discovered. Suppose the user is not pressed for time, but does not want to spend much money on the actual planning process and does not want to be bothered by the system until a complete plan has been composed. The parameter set associated with the query is then $\pi =$ (TIME_LIMIT = *1 hour*, MONETARY_LIMIT = *20 units*, USER_INVOLVEMENT = *none*). The vacation planning system decomposes the top level goal into four sub-goals with the information gathering needs described above, and assigns the subgoals to four software agents ($\mathcal{A}_1$ - $\mathcal{A}_4$). The initial view of problem solving Δ given to all four agents is the same; it includes the top level goals of each agent so that they may reason about the possibility of interactions among sub-goals. Also, the agents may communicate laterally among themselves (α for each agent contains the other three agents) and their partial solutions (solutions to sub-goals) will be collected and merged by the vacation planning system at the top level.

Suppose the information retrieval environment seen from the perspective of $\mathcal{A}_1$ is as follows. $\mathcal{A}_1$ resides on network $\mathcal{N}_0$, which contains no information carrying sites, and can see two other networks, $\mathcal{N}_1$ and $\mathcal{N}_2$. $\mathcal{A}_1$ retrieves the abstract descriptions of the two networks and finds that $\mathcal{N}_1$ contains a small amount of high quality information relevant to the query, whereas $\mathcal{N}_2$ contains a large amount of information that is relevant but of lower quality. Unfortunately, communication costs from $\mathcal{N}_0$ to $\mathcal{N}_1$ are high and there are no computation servers in $\mathcal{N}_1$. Communication costs from $\mathcal{N}_0$ to $\mathcal{N}_2$ are also high but there is an inexpensive computation server in $\mathcal{N}_2$. Since the agent has more time than money to work with, it pays the cost of traveling to $\mathcal{N}_2$ in order to access the large amount of relevant information there. It plans to search through a large amount of inexpensive, lower quality information in hopes of satisfying its needs before resorting to a potentially faster but more expensive search in $\mathcal{N}_1$. Using our notation, this problem solving episode and the two networks can be summarized as follows:

- $\mathcal{Q} = (\{\phi_1, \phi_2, \phi_3, \phi_4\}, \pi)$
  - $\phi_1 =$ "accommodations close to historical and natural sites of interest in MA from July 20th - 24th"
  - $\phi_2 =$ "historical and natural sites of interest in MA that may be visited from July 20th - 24th"
  - $\phi_3 =$ "means of travel to and within MA from July 20th - 24th"
  - $\phi_4 =$ "weather forecast for MA from July 20th - 24th"
  - $\pi =$ (TIME_LIMIT *1 hour*, MONETARY_LIMIT *20 units*, USER_INVOLVEMENT *none*)
- $\mathcal{A}_1 = (\Delta, \alpha, \phi_1)$
  - $\Delta = (\{\mathcal{A}_2, \phi_2\}, \{\mathcal{A}_3, \phi_3\}, \{\mathcal{A}_4, \phi_4\})$
  - $\alpha = (\mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$
- $\mathcal{N}_1 = (\Psi_1, \emptyset, \{\mathcal{I}_1\}, \delta_1)$
  - $\Psi_1 =$ (travel $\wedge$ history $\wedge$ nature $\wedge$ MA, 0.9, low)
- $\mathcal{N}_2 = (\Psi_2, \{\mathcal{K}_2\}, \{\mathcal{I}_2\}, \delta_2)$
  - $\Psi_2 =$ (travel $\wedge$ MA, 0.5, high)
    (history $\wedge$ MA, 0.5, medium)
    (nature $\wedge$ MA, 0.6, high)
- $\mathcal{C}(\mathcal{N}_1, \mathcal{N}_0) =$ HIGH
- $\mathcal{C}(\mathcal{N}_2, \mathcal{N}_0) =$ HIGH
- $\mathcal{C}(\mathcal{K}_2, \mathcal{N}_1) =$ LOW

In this section we presented a model of cooperative information gathering that consisted of two parts, problem solving and the environment. We characterized information carrying environments according to their salient features: abstractions of available data, varying speed and cost of access to resources, and the presence of individual networks, computation and information servers, and databases. We view problem solving as a planning activity that is driven by a top level query which in turn is decomposed and assigned to a set of information seeking agents. Note that query decomposition itself may be a dynamic and evolving process. Consider the case in which some item of information is uncovered during the search that obviates the need for achieving the current goals of one or more agents. The agents may then need to reorganize and redistribute outstanding goals through some sort of negotiation process so that the search may proceed efficiently[22]. Examples of information that may cause reorganization in the travel planning example are the serendipitous discovery of a major concert at Tanglewood in the Berkshires, or forecasts of severe flooding that could make travel impossible for the dates of the vacation. In scenarios that are true to the complexity inherent in today's networked environments, interactions between the environment, the current state of problem solving, and user requirements lead to dynamic, emerging solutions rather than a single exhaustively complete response to a query.

4 Related Work

We now review some of 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 that might be a participant in an information gathering task in a cooperative manner. Note that this review is only representative, and is not intended to be exhaustive.

4.1 Distributed Information Retrieval

As the Internet evolved from a test bed for experimentation in data communication protocols and remote login into a medium for supporting 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 propose 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 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. [16] 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. Besides, when the number of users and the number of topics of interest is large, the propagation of meta-knowledge may easily swamp any given user.

4.2 Intelligent Information Retrieval

Ram and Hunter [29] 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.

Rus and Subramanian[30], 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.

In an IR environment, we want an agent to locate information that is deemed relevant by a user. The notion of relevance is very subjective, leading to research in the area of agents that learn the relevance ranking function of human users. For example, Holte and Drummond[15] describe such an agent, called a learning apprentice, that learns from watching a user browse through an electronic library. The actions taken by a user while browsing, carry with them implicit information about goals and justifications for actions taken. The task of the agent is to infer these goals and justifications in an unobtrusive manner and apply them to locating relevant documents. The approach taken by the authors is to engineer a browsing system such that the actions provided to the user by the system carry with them information that facilitates goal inference. The agent can also infer new features of an item based on existing features of the item and features of items related to it in specific ways. The local processing performed by the learning apprentice consists of goal inference followed by document ranking and presentation. All of this may be augmented with performance feedback from the user.

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 precisely the coordination module. However, note that the delineation between the coordination module and the local processing module may be blurred in many of the existing MAS systems.

4.3 Information Gathering

Following the lead of Oren Etzioni's software robots (softbots), Voorhees[35] 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[17] 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 the 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. Our model is an initial step in this direction.

4.4 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, which allow them to interact in exactly the manner that we require.

Telescript technology, developed by General Magic, Inc. [36], 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 mechanism. 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. Finally, 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.

5 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, it 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[23]. 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 the search quality by providing better solutions in possibly lesser 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 FA/C 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 FA/C 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 discuss implemented aspects of FA/C that have direct relevance to the CIG task. Decker and Lesser[9, 10, 11] 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[26] on coordination in multi-agent planning using favors goal interrelationships and temporal interactions is also relevant here. Garvey and Lesser[14] discuss design-to-time algorithms which basically endow the local problem solver with abilities to deal with real-time considerations and goal inter-dependencies. Such a scheduler is, perforce, satisficing in the solutions it provides and relies on the use of approximate processing techniques. Carver and Lesser[5, 8, 7, 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. 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 process to resolve the deficiencies in the partial data. Other work in MAS, though not directly falling under the umbrella of FA/C approach, could act as enabling technologies for multi-agent based CIG. The contract net[33, 32] 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[31, 40]. 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 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.

6 Acknowledgments

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

[2] M. C. Bowman, P. B. Danzig, U. Manber, and M. F. Schwartz, "Scalable Internet Resource Discovery: Research Problems and Approaches", *Communications of the ACM*, 37(8), 1994, pp 98 - 107, cntd on 114.

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

[4] J. Callan, *Personal Communication*, July 1994.

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

[6] N. Carver and V. Lesser, "A First Step Toward the Formal Analysis of Solution Quality in FA/C Distributed Interpretation Systems", in the *Proceedings of 13th International Distributed Artificial Intelligence Workshop*, Seattle, WA., July 1994.

[7] N. Carver, V. Lesser, and Q. Long, "Resolving Global Inconsistency in Distributed Sensor Interpretation: Modeling Agent Interpretations in DRESUN", in the *Proceedings of 12th International Distributed Artificial Intelligence Workshop*, Hidden Valley, PA., May 1993.

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

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

[10] K. Decker and V. R. Lesser, " Quantitative Modeling of Complex Computational Task Environments", in *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp 217-224, Washington, 1993.

[11] K. Decker and V. Lesser, "Designing a Family of Coordination Algorithms", in the *Proceedings of 13th International Distributed Artificial Intelligence Workshop*, Seattle, WA., July 1994.

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

[13] E. H. Durfee, V. R. Lesser, and D. D. Corkill, "Coherent Cooperation among Communicating Problem Solvers", *IEEE Trans. on Computers*, vol 36, pp 1275-1291, 1987.

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

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

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

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

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

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

[20] 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*, Chambery, France, 1993, pp 438 - 444.

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

[22] B. Laasri, H. Laasri, S. Lander and V. Lesser, "A Generic Model for Intelligent Negotiating Agents", *International Journal of Intelligent and Cooperative Information Systems* 1(2), June 1992, pp 291-317.

[23] V.R. Lesser and D. Corkill., "Functionally accurate, cooperative distributed systems", in *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1), 1981, pp. 81-96.

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

[25] V.R. Lesser., "A retrospective view of FA/C distributed problem solving", *IEEE Systems, Man, and Cybernetics*, 21(6), 1991, pp. 1346-1363.

[26] F. v. Martial, "Multiagent Plan Relationships", in the Proceedings of the Ninth Workshop on Distributed AI, September 1989.

[27] M. V. Nagendraprasad, S. E. Lander, and V. R. Lesser, "Retrieval and Reasoning in Distributed Case Based Systems", January 1994, Unpublished working paper.

[28] 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.

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

[30] D. Rus and D. Subramanian, "Designing Structure Based Information Agents", in *Working Notes of the AAAI Spring Symposium on Software Agents*, 1994, pp. 79-86.

[31] T. W. Sandholm, "An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations", in the *Proceedings of 12th International Distributed Artificial Intelligence Workshop*, Hidden Valley, PA., May 1993.

[32] T. W. Sandholm and V. R. Lesser, "An Exchange Protocol Without Enforcement", in the *Proceedings of 13th International Distributed Artificial Intelligence Workshop*, Seattle, WA., July 1994.

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

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

[35] E. M. Vorhees, "Software Agents for Information Retrieval", in *Working Notes of the AAAI Spring Symposium on Software Agents*, 1994, pp. 126-129.

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

[37] R. Whitehair and V. R. Lesser, "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.

[38] S. Zilberstein and S. J. Russell. "Optimal Composition of Real-Time Systems." *Artificial Intelligence Journal*, forthcoming.

[39] S. Zilberstein. "Meta-Level Control of Approximate Reasoning: A Decision Theoretic Approach." To appear in *Proceedings of the Eighth International Symposium on Methodologies for Intelligent Systems*, Charlotte, North Carolina, October 1994. Will be also published as Lecture Notes in AI, New York, New York: Springer-Verlag, 1994.

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