Intelligent Information Gathering Using Decision Models

Shlomo Zilberstein & Victor Lesser

CMPSCI Technical Report 96-35
June, 1996

Intelligent Information Gathering Using Decision Models

Shlomo Zilberstein and **Victor Lesser**
Computer Science Department
University of Massachusetts at Amherst

Abstract

This paper describes an architecture for the next generation of information gathering systems. The paper is based on a research proposal whose goal is to exploit the vast amount of information sources available today on the NII including a growing number of digital libraries, independent news agencies, government agencies, as well as human experts providing a variety of services. The large number of information sources and their different levels of accessibility, reliability and associated costs present a complex information gathering coordination problem. We outline the structure and components of an information gathering system that uses an explicit representation of the user's decision model in order to organize its activity. Within this framework, information gathering planning is performed based on its marginal contribution to the user's decision quality.

1 Introduction

The vast amount of information available today on the NII has a great potential to improve the quality of decisions and the productivity of consumers of this information. Currently available information sources include a growing number of digital libraries, independent news agencies, government agencies, as well as human experts providing a variety of services. A rapid expansion of these services is expected over the next 5-10 years. In addition, we anticipate that improved information retrieval (IR) and information extraction (IE) technologies will become available [2, 24]. These technologies will allow a system not only to locate but also to extract necessary information from unstructured textual documents.

The large number of information sources that are currently emerging and their different levels of accessibility, reliability and associated costs present a complex information gathering planning problem that a human decision maker cannot possibly solve without high-level filtering of information. For many information gathering tasks, manual navigation and browsing through all the *relevant* information is no longer effective. The time/quality/cost tradeoffs offered by the collection of information sources and the dynamic nature of the environment lead us to conclude that the user cannot (and should not) serve as the controller of the information gathering process.

The solution outlined in this paper is based on a simple observation that information gathering is normally an intermediate step in a decision making process. We provide the system with an explicit representation of the user's decision model or task so that information gathering activity can be organized on the basis of its marginal contribution to quality of the decision. The resulting system architecture extends the scope of current state-of-the-art information gathering systems by giving an answer to a decision problem rather than collecting the relevant data. Such a service would enhance the capabilities of future digital libraries [7, 9] by taking advantage of two important developments: (1) the rapid improvement in the accuracy of information extraction technology, and (2) the introduction of new standards for structured information exchange between information providers and consumers. For example, we expect that such standards will emerge in the near future for such common documents as product descriptions, resumes, product reviews, and technical reports. When operating under resource constraints (related to cost of communication and database access, limited computational power, and limited amount of time), our system will result in significant performance improvement by intelligent control of information gathering.

From the perspective of a digital library, the proposed architecture is aimed at automating the function of a sophisticated research librarian. This type of librarian is often not only knowledgeable in library science but also may have a technical background relevant to the interests of the organization. In addition to locating relevant documents for their clients, such librarians will in many situations actually distill the desired information from these documents. They will often need to make decisions based on resource concerns such as whether certain

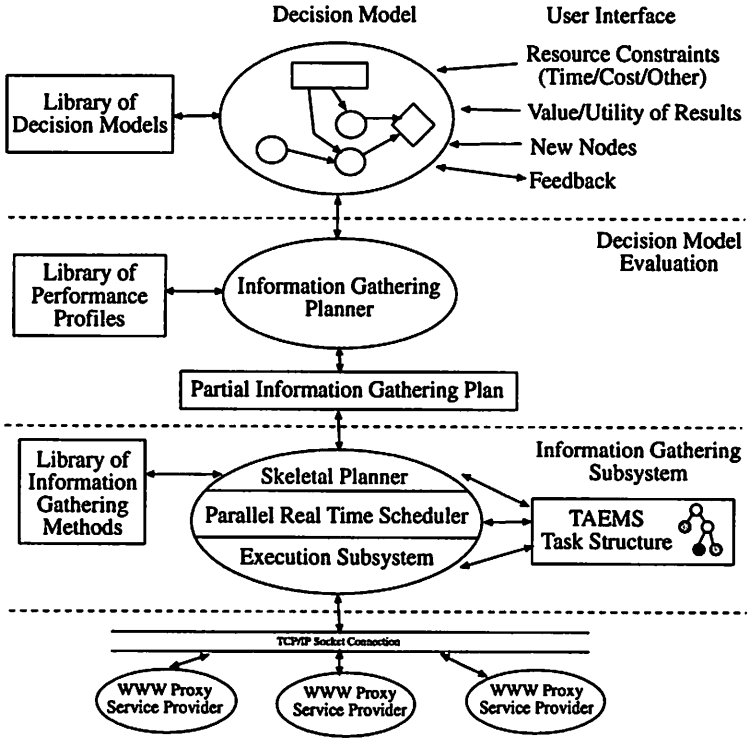

Figure 1: The proposed system is composed of three major components: the user interface, the decision model evaluation subsystem, and the information gathering subsystem.

periodicals are available in-house and if not how long it will take to get them and what they will cost [28]. We see this automation of a sophisticated librarian as a natural step in the evolutionary development of a fully automated digital library. Our approach builds on current document location technology [1, 6, 15, 31, 32] by introducing the value of information and its cost, time and likelihood of being acquired as the driving force behind the decision of when, where and how to locate specific documents.

2 Overview of the System

Our proposed system architecture is based on three primary layers that operate concurrently: the user interface (UI), the decision model evaluation subsystem (DME), and the information gathering subsystem (IG). Starting with the user interface, each layer is engaged in *activation*, *monitoring*, and *negotiation* with the lower layers. Figure 1 shows an outline of the system.

The user interface allows the user to retrieve a decision model from a library, to determine the resource allocation and utility of the task, and to monitor and control the information gathering activity. The decision model evaluation subsystem uses a decision-theoretic approach to construct an information gathering plan for the particular task. The information gathering subsystem selects particular information gathering methods and schedules their parallel execution.

The key question is how could information gathering be guided by the user's decision model. The following situation illustrates our approach to the problem. Consider a user who is seeking information about "Quick 7.5", a new financial management software. Suppose that the user is interested in purchasing the product for personal use. The user's goals are to simplify and improve his capability to track investments, balance checkbooks, and compile tax return data. Before making a decision, the user may gather information regarding the quality of Quick 7.5 and its capabilities as well as the cost of the product. This information may be available in various newsgroups, company catalogs, and on-line magazines. The question is how much time and money is the user willing to commit to this process and how should the information gathering activity be organized as a result.

Figure 2 shows the influence diagram that a user may construct in order to decide whether to purchase Quick

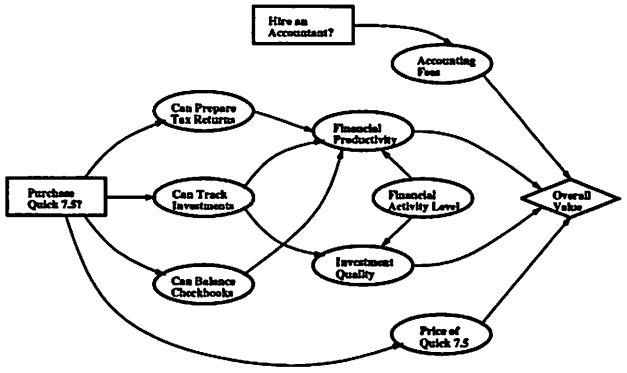

Figure 2: An influence diagram for deciding whether to purchase a new financial management program

7.5. The diagram determines the effect of this decision on the user's capability to perform typical financial tasks such as balancing checkbooks, tracking investments, and compiling tax return data. Together with the level of financial activity, these factors determine the user's financial productivity and investment quality. Finally, an overall utility function determines the expected value of improved productivity minus the price of the product.

Notice that a similar diagrams could be used by a manager who is considering buying 100 copies of Quick 7.5, by an investor who is considering purchasing stocks of the publisher of Quick 7.5, or by a competitor who is considering the release of a similar product. Their different goals and utility function will result in a different information gathering activity. For example, the buyer of 100 copies of the product will be more concerned about uncertainty regarding the price. In such case, more resources should be allocated to the information gathering activity to reduce the uncertainty. The rest of this section describes the main research problems that arise when one tries to optimize information gathering activity based on the user's decision model. The section also summarizes our design goals.

2.1 The Main Research Problems

The growing complexity of the information gathering task and the need to introduce another level of control and filtering of information present several challenging research problems:

1. **Describing the User's Decision Model** Since we want the system to automatically initiate and control information gathering operations based on the user's task, a description of that task must be provided to the system. We suggest using influence diagrams to describe the causal model used by the user in order to select actions [14]. Influence diagrams offer a concise graphical representation for complex decisions and they can be evaluated using efficient existing algorithms [23]. In addition, several techniques for anytime evaluation of belief networks have been developed by Horvitz *et al.* [12] and by Wellman and Liu [30]. We anticipate that a large number of influence diagrams representing "typical" tasks can be constructed and stored in a library for future reuse. We also expect that in many cases minimal modification of existing decision models would be required in order to tailor them to fit particular user's needs.

2. **Describing the Information Environment** The information environment will include a variety of information sources. We assume an open and dynamic environment in which the existence, availability, quality, and cost of access of different information sources is constantly changing. But to be able to operate efficiently in this environment, the system will have to construct a database that characterizes the known information sources and the time/quality/cost tradeoffs that they offer. This database will be constantly revised based on the system's experience and feedback from other systems whose task may be to discover and classify new information sources. In previous work in the area of approximate decision-making we have developed efficient mechanisms to statistically characterize such time/quality tradeoffs using *conditional performance profiles* [33].

3. **Planning Information Gathering Actions** In order to automate the information gathering process, we will construct information gathering plans, similar to the softbot-based interface to the internet proposed by Etzioni and Weld [5]. Our planning approach is based on extending current information-theoretic techniques to derive plans that are most valuable to the user based on the decision model. This approach addresses explicitly the problem of uncertainty regarding the quality, completeness, cost and delay associated with each source of information.

4. **User Participation in the Information Gathering Process** A major advantage of our proposed approach is that it does not rely on the user in initiating and controlling the information gathering process. But the system must *allow* for user participation in this process in the form of modification of the decision model, interpretation of data, and resource

allocation. To facilitate user participation, the current status of the decision model evaluation will be presented to the user in an informative way. Obviously, the user cannot examine every decision that the system is making, but there should be a high level summary that shows what information the system is currently seeking, what quality of responses it is getting, when does the system plan to generate an action and what is the expected utility of that action. The user should be able to intervene at any point by changing the resource constraints, the interpretation of data, or the order of site visit.

5. **Guiding and Monitoring the Information Gathering Activity** The information gathering plan will determine what information is most valuable with respect to the user's decision model and what resources should be committed to this search. This plan will be based on an abstract, probabilistic view of the information gathering environment. This leads to a complex information gathering agent activation and monitoring process that optimizes the chances of getting the necessary information within the resource constraints. The purpose of the monitoring process is to activate agents, monitor their progress, identify and react to failures, and when necessary request additional resources.

2.2 Design Goals

The system architecture that we describe was designed to meet the following goals:

1. **Open System Architecture** While there is a growing effort to standardize the representation of information that is available on the NII, it is clear that a successful system to exploits this information must be flexible enough to handle a large variety of heterogeneous information sources, and different information gathering techniques. The system should be easy to adapt to new types of information sources and information gathering techniques.

2. **Asynchronous, Concurrent Operation** The redundancy in information availability (e.g. there may be dozens of reviews of a new software product), and the possible time pressure under which the user operates makes it advantageous to allow the system to initiate multiple information gathering operations at the same time. In the current design of the system, parallel information gathering activity is initiated and coordinated by a single site. Another possible model that we (and others) have studied involves a distributed search process conducted by multiple sites [4, 20, 21, 22]. To allow for this type of operation, only the IG subsystem would have to be modified. While information requests are being processed, the system should be able to respond asynchronously to various events such as arrival of data and changes in the user's decision model.

3. **Explanation and Justification** To be successful, any automated decision support system must be able to justify its decision in a way that a human user can easily follow. If needed, the user should be able to trace back decisions and the information on which they are based, to force the system to reexamine particular assumptions or information, and to reinterpret the data.

4. **Well-Defined Theoretical Foundation Based on Decision Theory and Information Theory** Another goal is to construct an architecture in which actions are performed based on formal models of cost and utility. The advantage of a normative approach is that the properties of the system can be formally analyzed and that overall performance can be evaluated with respect to the user's subjective utility function. We aim at developing a system that exhibit *resource-bounded optimality*, that is, can maximize decision quality under time pressure and limited computational resources [25].

5. **Exploit the State-of-the-art Information Retrieval and Information Gathering Techniques** Another goal is to use the best available technology in information gathering in order to minimize the degree of user involvement in the intermediate information gathering and interpretation process. Some successful techniques for information retrieval and information extraction have been developed at the University of Massachusetts National Center for Intelligent Information Retrieval [2].

3 The User Interface

Although the user interface (UI) does not represent the focus of this paper, it is an important component that will allow for high level interaction between the user and the DME and IG subsystems. This section summarizes the main functions that will be supported by the UI.

1. **Retrieving and Modifying Decision Models** The user's decision model or task will be represented by an influence diagram. This intuitive representation

has many advantages that are discussed in the next section. The UI will support interactive construction, storage and retrieval, and editing of influence diagrams. We will construct a library of influence diagrams for typical decision models and allow the user to combine them and construct more complex decision models.

2. **Activation and Monitoring of Decision Model Evaluation** The user interface will allow for the activation of decision model evaluation. The user will be able to specify and modify the resource constraints (cost and time) and the subjective time-dependent utility function associated with the task.

3. **Status Display** The UI will give the user high level feedback regarding the status of the current activities. This feedback will include the resources consumed so far, the nodes of the decision model that are being evaluated, the quality of the information gathered so far, the expected quality of the decision based on the available information, and an estimated time for completing the information gathering activity.

4. **Control of Search Parameters** The UI will allow the user for asynchronous control of such search parameters as the overall utility function and the resources allocated to the task. The user will also be able to disable/enable information gathering activities and to examine the raw data that the system gathered and the decisions that the system made.

5. **Active Participation in the Decision Model Evaluation Process** The user will be able to participate in the data interpretation process. This capability is essential since information extraction from textual documents is a hard problem that cannot be fully automated using today's best technology.

6. **Negotiations** The UI will allow the user to negotiate with the subsystems the parameters of the search before and during the evaluation of a decision model. This negotiation process is needed in order to respond to unpredictable problems that may require additional time and cost to gather the necessary information.

4 Decision Model Evaluation

4.1 Representing Decision Tasks with Influence Diagrams

Over the past several years, influence diagrams have become one of the most widely used techniques for reasoning under uncertainty. An influence diagram is an effective method to represent decision tasks. The diagram shows the information about the agent's current state, the decisions that the agent can make, the state that will result from the agent's decision, and the utility of the state. Figure 2 shows an influence diagram that a user may use for deciding whether to purchase a new financial management program. The diagram includes three types of nodes:

1. **Chance nodes** (ovals) represent random variables. Each chance node has an attached conditional probability matrix that determines how the probability distribution of that node depends on the parent nodes. Deterministic relations between nodes can also be represented. The above example includes chance nodes to represent the current level of the user's financial activity, the capability of the program to help in balancing checkbooks, tracking investments, and compiling tax return data, the cost of the product, the user's financial management productivity, etc. Each chance node can take a value from a certain finite domain. For example, the level of financial activity may be characterized as *low*, *medium* or *high* (with a typical profile of number of accounts and number of transactions per month attached to each).

2. **Decision nodes** (rectangles) represent points where the decision-maker has a choice of actions. A single task may involve multiple decisions. The above example includes two decisions: whether to purchase the financial management program and whether to use an accountant for preparing tax returns.

3. **Utility nodes** (diamonds) represent the agent's utility function. In most cases a single utility node is used whose parents are all the nodes describing the outcome state. In the above example a single value node represents the overall utility of the user.

Using the above influence diagram, the user can make the best decision based on the currently available information or gather additional information before a decision is made. For example, the capability of the program to track investments can be assessed using reviews of the product or articles in certain news groups. The

cost of the product may be estimated by extracting the list price from a review or by querying an on-line product catalog.

Influence diagrams offer an effective modeling framework for a diverse array of problems involving reasoning under uncertainty. This effectiveness has many different aspects [27, 29]. First, influence diagrams capture both the structural and qualitative aspects of the decision problem and serve as the framework for an efficient quantitative analysis of the problem. Influence diagrams allow efficient representation and exploitation of the conditional independence in a decision model. Finally, influence diagrams have proven to be an effective tool for not only communicating decision models among decision analysts and decision makers, but also for communicating between the decision maker and the computer.

It is important to emphasize that our approach does not require the use of a complex diagram for every information gathering task. Simple decision tasks can be easily represented by a simple influence diagram. The most obvious case is when the user needs to gather a certain piece of information such as the weather forecast. This task corresponds to an influence diagram with a single chance node representing the weather. The value node may depend on the quality of the weather forecast and on the cost of information gathering. In other words, our system architecture does not exclude the possibility of a simple information gathering task. With the rapid growth in information services, the issue of quality of information versus resource allocation is equally relevant in making simple decisions.

4.2 The Value of Information Gathering

This section describes a method for calculating the value of information gathering (VOIG). Information value theory [13] provides a mathematical foundation to guide this central decision. Previous work in this field has concentrated on the value of *perfect information*. In our problem domain, however, the available information may be inaccurate or incomplete. In this section we show how the basic theory can be extended to handle imperfect information sources, costs of acquiring information and real-time operation. The key question is how to initiate information gathering activities that are most valuable to the user, how to allocate resources to these activities, and how to respond to unpredictable events.

Information gathering activity is aimed at reducing the uncertainty regarding the value of some random variables in the user's decision model and thereby improve the quality of the user's action. For example, the user considering the purchase of Quick 7.5 may be uncertain regarding its capability to track investments[1].

Time/Cost-Dependent Utility Functions

The overall value of the outcome of the user's decision will be represented by a utility function. Since, in general, information gathering will cause a delay in action and will have an associated (retrieval) cost, the overall utility function will be time/cost-dependent. $U(O_i, T, C)$ will represent the utility of a decision outcome O_i with information gathering delay T and information gathering cost C. The utility function will be specified as part of the decision model.

Performance Profiles of Information Gathering

With each node representing a random variable in the influence diagram, N, there will be an associated information gathering performance profile, $Q_N(T, C)$. The performance profile will be a statistical summary of the quality of information offered by the information gathering subsystem when activated with time T and cost C. This dynamic information will be updated periodically by a learning algorithms.

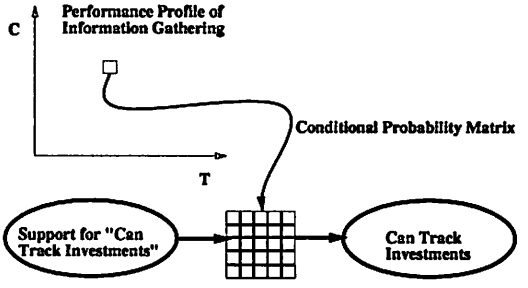

Figure 3: A fragment of the influence diagram showing how evidence is treated. The reliability of the evidence is reflected by the conditional probability table.

To reflect the fact that information gathering will normally return imperfect information, we will use an auxiliary node in the diagram. This node will represent the actual information that was found and will have a probabilistic effect on the value of a random variable in the original decision model. The auxiliary node will affect directly only one node in the diagram. The quality of the information will determine the conditional probability table. The highest quality of 1 will result in deterministic relationship between the nodes and the lowest quality of 0 will result in no causal relationship between the nodes. This approach to imperfect information is demonstrated

[1]This capability can be described by a discrete value: poor, ok, good, or excellent. An initial probability distribution will be attached to each random variable to reflect the user's prior belief. A default distribution will be part of the decision model.

in Figure 3 where an auxiliary node represents the current evidence regarding the capability of Quick 7.5 to track investments. The belief regarding the program's actual capability is affected by this information through a conditional probability table that represents the reliability of the information.

The Value of Information Curve

For any given node, the system will construct the value of information curve by extending the standard theory to incorporate the time/cost/quality information. This extension is described below using the notation of Russell and Wefald [25].

Suppose that the current information available to the agent is K. K corresponds to the current belief regarding the value of different random variables included in the decision model. Let O_i be the possible outcomes of the user's decision. Then the value of the current best decision α is the expected utility of the outcome of the best decision based on the currently available knowledge. This is defined by:

$$EU(\alpha|K) = \max_A \sum_i P(O_i|K, Do(A))U(O_i, 0, 0)$$

Now, consider the situation in which the user can gather information that will provide *some* evidence E_N regarding the variable N included in the decision model. For any given amount of time T and cost C, the user may obtain a different quality of information, $Q_N(T, C)$. The value of the new best action (after the evidence E_N is obtained) will be:

$$EU(\alpha|K, E_N, Q_N(T, C)) =$$

$$\max_A \sum_i P(O_i|K, E_N, Q_N(T, C), Do(A))U(O_i, T, C)$$

But since E_N is a random variable whose value is *currently* unknown, we must average over all possible values E_N^k using the user's *current* information. Hence, the value of gathering information on node N is:

$$V_K(E_N, T, C) = (\sum_k P(E_N = E_N^k|K)$$

$$EU(\alpha|K, E_N = E_N^k, Q_N(T, C))) - EU(\alpha|K)$$

For each node, we can construct a curve that determine the value of information gathering with a particular allocation of (time/cost) resources.

The Value of Information Gathering

Now, for any single node the value of information gathering can be calculated as the global maximum of the above curve.

$$VOIG_K(E_N) = \max_{T,C} V_K(E_N, T, C)$$

A major focus of our work is the design of efficient algorithms to solve this equation and find the value of information gathering for any particular node.

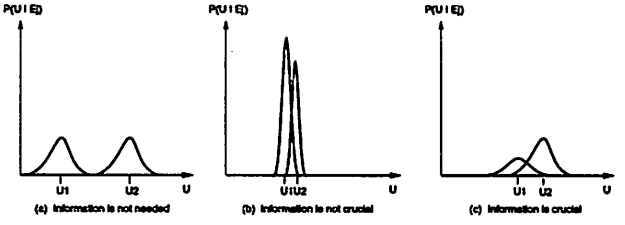

Figure 4: Three generic cases for the value of information

In related work, Russell and Wefald [25] distinguish between three generic cases for the value of perfect information. Figure 4 shows the three cases for a decision problem that include two decisions A_1 and A_2 whose current utility distributions are U_1 and U_2. The figure shows the distribution of utility of each action with respect to E_N. In (a), A_1 is almost certainly superior to A_2, so the information is not needed. In (b), the choice is unclear but because it has little effect on the utility, information gathering is not needed. In (c), the choice is unclear and the information is crucial. By calculating the net value of information gathering (taking into account quality and costs), our system will be able to distinguish between the above cases as well as between less obvious situations.

4.3 Information Gathering Planning

We will develop a decision-theoretic planner for generating information gathering requests. The initial algorithm will be a simple greedy algorithm that will identify the most valuable *single* node for information gathering and generate a request for that node and integrate the new information until the value of information gathering becomes negative.

A natural improvement of this algorithm is to look at a *set* of nodes that have the highest value of information gathering. But the complexity of this process will grow exponentially with the number of nodes. We have started to develop non-myopic approaches to plan sequences of information-gathering actions, with particular attention to managing a tradeoff between information quality and cost. The goal is to construct near-optimal, efficient techniques to construct conditional information gathering plans for a given decision model.

4.4 Negotiation with the Information Gathering Subsystem

A basic information gathering request will be to determine the actual value of a specific node using specific resources (time/cost). The request will specify the expected quality of the information. Negotiation between the DME and the IG subsystems will be necessary when a request cannot be satisfied without either increasing the resources or decreasing the expected quality of information. There are two primary reasons for inconsistencies between a request generated by the DME planner and the capabilities of the IG subsystem. First, the resource allocation and expected quality are derived by the DME planner based on a high-level, abstract view of the information gathering environment while the IG subsystem is using a more detailed representation. Second, the IG subsystem may discover that certain information sources are inaccessible leading to a more limited set of choices.

In order to reduce the level of negotiation between the subsystems, it may be useful for a request to include the highest possible cost (for the given time/quality) and the highest possible time (for the given cost) that would keep the request the most valuable. One way to do that is to look at the time/quality/cost data for the second most valuable node and calculate how much the time (or cost) of the selected node can grow before its information gathering value drops below that of the second best node (and before it becomes negative!). These ranges of time and quality would allow the IG subsystem to deal with certain problems without performing unnecessary negotiation with the DME subsystem.

When negotiations is needed, it will be performed by the IG subsystem informing the DME subsystem what quality of information it can guarantee with what level of resources. This negotiation process is similar to one we have previously developed [8]. The DME subsystem will generate a new request for the same node or will modify the plan and generate an information gathering request for a different node.

5 Information Gathering

The information gathering subsystem (IGS) is responsible for providing the values and certainty factors that are needed by the decision subsystem to evaluate the influence diagram. This process, which is incremental and asynchronous, is initiated by the decision subsystem when it requests the evaluation of a set of decision nodes by a certain time within specified cost and quality constraints. We also want to take into consideration the likelihood of the system achieving these constraints. In order to satisfy this request, the IGS needs to instantiate, schedule, execute and monitor appropriate information gathering activities. As part of this process, the IGS needs to assess whether the requirements laid out by the decision subsystem based on default knowledge about the characteristics of the information gathering activities can be met. If not, there is a negotiation process that occurs between these two subsystems in order to find a new set of requirements for the information gathering activities that can be met. The IGS, through its monitor function, is responsible for updating the decision subsystem with information about the current progress made in accomplishing the requirements, and possibly for rescheduling activities based on unexpected events.

In order for the IGS to be effective, it must be able to dynamically construct information gathering activity plans that match the criterion set forth by the decision subsystem. For instance, in order to meet deadlines, it may be required to schedule activities concurrently. This can involve scheduling concurrent information gathering activities associated with one or more decision nodes. Concurrent scheduling of activities may not always be appropriate because a distributed search process will often incur more costs than a sequential one. In another situation, it may need to construct a tailored version of an information gathering plan that does not gather information from all relevant sources or does not fully analyze the information it gathers in order to meet the desired cost, quality or reliability criteria.

The design of the IGS is predicated on the requirement that it be generic—not tied to a specific application domain, language or problem-solving architecture. This is accomplished by not putting any restrictions on the information gathering activities other than that they can be mapped (described at an abstract level) into a domain-independent framework that we have developed called TAEMS [3]. This mapping has to have sufficient detail so that alternative ways of executing the activities for a specific decision node which trade off cost/quality/duration are appropriately represented. Also, the mapping has to indicate the existence of quantitative characteristics of the activities in terms of expected duration, cost, quality of the decision and the likelihood of these expectations being met, and the existence and character of the relationships among activities. All reasoning that the IGS does in accomplishing its goals is then based on the TAEMS representation.

TAEMS represents computational activity in terms of task structures at multiple levels of abstraction, each

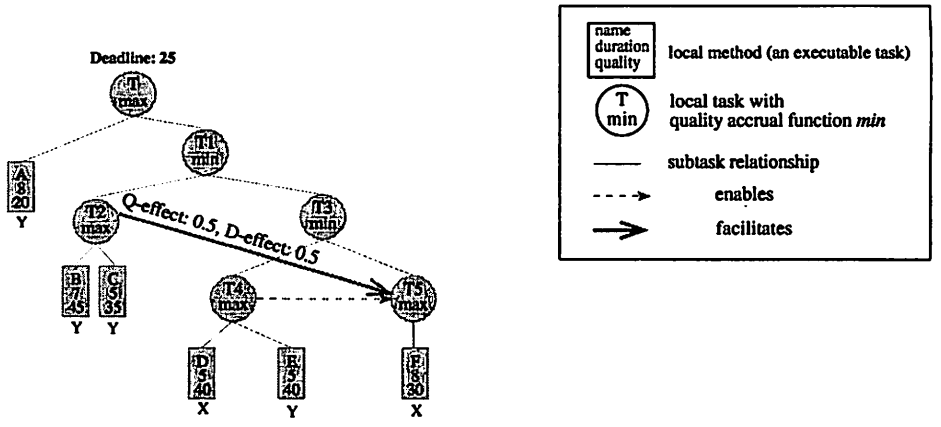

Figure 5: A Simple TAEMS Task Structure: In this example, the max accumulation function is used to indicate that there are alternative methods for accomplishing a task that trade off execution time for result quality : task T2 can be solved in two different ways, one involving method B and the other method C; B takes longer than C (e.g., 7 versus 5) but produces a higher quality result (e.g., 45 versus 35). Thus, in time pressure situations the scheduler can reason, based on the specifics of the task structure, how best to sacrifice expected overall quality of the final result for a higher likelihood of producing a result by the deadline. This task structure also indicates that even though the concurrent execution of task T2 and T5 is possible, this would not be optimal unless there were severe time constraints since T5 would execute much longer without having the result of T2 to facilitate its execution.

with a deadline. From a real-time perspective, the goal in scheduling these activities is to maximize the sum of the quality achieved for each task group before its deadline. A task group consists of a set of tasks related to one another by a subtask relationship that forms an acyclic graph (see Figure 5). Tasks at the leaves of the tree represent executable *methods*, which are the actual instantiated computations or actions the agent will execute to produce some level of quality (in 5, these are shown as boxes). The circles higher up in the tree represent various subtasks involved in the task group, and indicate precisely how quality will accrue depending on what methods are executed and when. The arrows between tasks and/or methods indicate other quantitative task interrelationships where the execution of some method will have a positive or negative effect on the quality or duration of another method. The presence of these interrelationships make this an NP-hard scheduling problem.

Figure 6 shows an example of how an information gathering plan is represented in TAEMS using some of the above relationships[2].

The major research issues that need to be solved in developing the IGS system are the following:

1. how to develop an appropriate interface between the decision support subsystem and IGS to support effective negotiation over achievable quality, cost and deadline criteria;

2. how to make it easy to map information gathering procedures into TAEMS, and how to acquire/learn the quantitative data necessary to represent these procedures fully;

3. how to construct and develop multiprocess schedules for the TAEMS task structures that meet desired time, cost, and quality criteria [17];

4. how to decide how much parallelism in a schedule is cost effective;

5. how to reason about the uncertainty of an overall schedule;

6. how to monitor progress and dynamically replan in situations where expectations will not be met.

[2]For example, in Figure 6 a *favor* relationship exists between the task "Find URL for maker" and "Find URL for competitor." Both tasks involve searching the index of WWW sites. Since this search process can take multiple text strings, either task can accomplish both searches simultaneously by adding the text name of the other. Another relationship, *refine*, that we propose to implement also shows up in this example. This relationship represents a class of task relationships that involve meta-information gathering (i.e., provides information to elaborate the task structure of agents in situations where the task structure below a certain level of abstraction is difficult to predict). In this example, it provides information about load metrics and server accessibility which are used to improve the agent's subjective view of its task structure in terms of its expected distribution of method quality and execution.

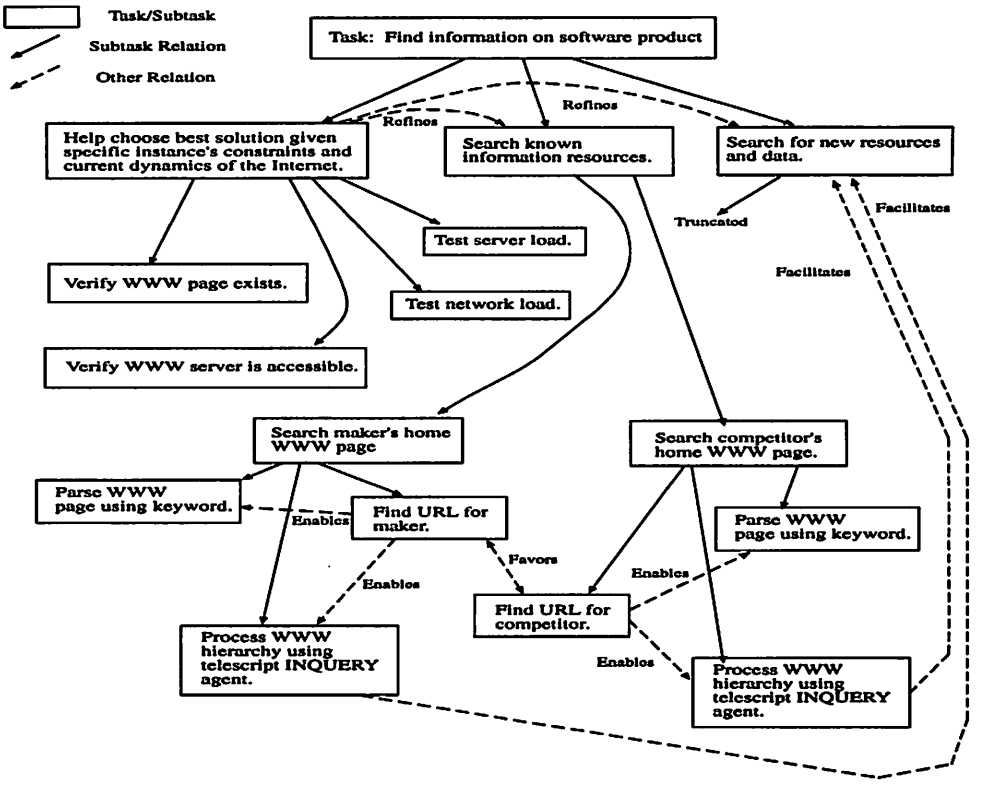

Figure 6: An Example of an Information Gathering Task Structure. This is a fragment of a template for instantiating a task structure for gathering information on the NII about a specific software product based on the knowledge of who makes the product and who are the likely competitors. The high-level task "search for new resources" is not fully described because of space limitations, but it involves such activities as searching through PC software news groups and on-line versions of PC magazines. In order to search unstructured text for appropriate references, we use the INQUERY system which is an advanced information retrieval system. In this task structure, there is the potential for much concurrency involving simultaneously searching each of the known sources and searching for new sources in different data repositories.

To our knowledge, except for the recent work by Knoblock at USC-ISI reported at the AAAI Information Gathering Symposium 1995, there is no similar work on parallel scheduling of information gathering. In comparison to Knoblock's approach, we use a richer representation of information gathering tasks. For example, our approach includes not only resource relationships but also soft coordination relationships like *favor* and *facilitates*. Additionally, our scheduling process takes deadlines and cost factors into consideration in constructing an appropriate plan/schedule.

Acknowledgments

We would like to thank the participants of the Seminar on Information Gathering at UMass for many fruitful discussions. They include James Allan, Jody Daniels, Alan Garvey, Joshua Grass, Eric Hansen, Maram Nagendraprasad, Tuomas Sandholm, Tom Wagner.

References

[1] W. P. Birmingham, E. H. Durfee, T. Mullen, and M. P. Wellman. The Distributed Agent Architecture of the University of Michigan Digital Library. In *AAAI Spring Symposium on Information Gathering in Heterogeneous, Distributed Environments*, Stanford, CA, 1995.

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

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

[4] K. S. Decker, V. R. Lesser, M. V. Nagendra Prasad, and T. Wagner. An Architecture for Multi-Agent Cooperative Information Gathering. *Proceedings of the CIKM'95 Intelligent Information Agent Workshop*, Baltimore, Maryland, 1995

[5] O. Etzioni and D. Weld. A Softbot-Based Interface to the Internet. *Comm. of ACM*, July 1994.

[6] E. Fikes, R. Englemore, A. Farquhar, and W. Pratt. Network-based Information Brokers. In the *AAAI-95 Spring Symposium on Information Gathering from Distributed Heterogeneous Environments*, Stanford, CA, 1995.

[7] E. Fox. Digital Libraries. *IEEE Computer*, 26(11), pp. 79–81, November 1993.

[8] A. Garvey, K. Decker and V. Lesser. A Negotiation-based Interface Between a Real-time Scheduler and a Decision-Maker. *Proceedings of the AAAI Workshop on Models of Conflict Management in Cooperative Problem Solving*, Seattle, WA, July 1994. (Also Computer Science Technical Report 94-08, University of Massachusetts, January 1994.)

[9] H. Gladney, N. Belkin, Z. Ahmed, E. Fox, R. Ashany, and M. Zemankova. Digital Library: Gross Structure and Requirements. *Proceedings of Workshop on On-line Access to Digital Libraries*, 1994.

[10] J. Grass and S. Zilberstein. Anytime Algorithm Development Tools. To appear in M. Pittarelli (Ed.), *SIGART Bulletin Special Issue on Anytime Algorithms and Deliberation Scheduling*, 7(2), April, 1996.

[11] E. A. Hansen and S. Zilberstein. Monitoring the progress of anytime problem solving. To appear in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, 1996.

[12] E. J. Horvitz, H. J. Suermondt and G. F. Cooper. Bounded conditioning: Flexible inference for decision under scarce resources. *Proceedings of the 1989 Workshop on Uncertainty in Artificial Intelligence*, pp. 182–193, Windsor, Ontario, 1989.

[13] R. A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, SSC-2(1):22–26, 1966.

[14] R. A. Howard and J. E. Matheson. Influence Diagrams. In *Principles and applications of decision analysis*, vol. 2, Menlo Park, California: Strategic Decision Group, 1984.

[15] S. B. Huffman. Learning information extraction patterns from examples. In *IJCAI-95 Workshop on New Aproaches to Learning for Natural Language Processing*, August 1995.

[16] V. Lesser, J. Pavlin and E. Durfee. Approximate processing in real-time problem-solving. *AI Magazine* 9(1):49–61, Spring 1988.

[17] Q. Long and V. Lesser. A Heuristic Real-Time Parallel Scheduler Based on Task Strucutures. Technical Report 95-92, University of Massachusetts, Amherst, 1995.

[18] T. Moehlman, V. Lesser and B. Buteau. Decentralized Negotiation: An Approach to the Distributed Planning Problem. *Group Decision and Negotiation*, 1:2, K. Sycara (ed.), pp. 161–192. Norwell, MA: Kluwer Academic Publishers, 1992.

[19] M. V. NagendraPrasad, V. Lesser, S. Lander. Reasoning and Retrieval in Distributed Case Bases. To appear in Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries.

[20] T. Oates, M. Nagendra Prasad, and V. Lesser. Cooperative Information Gathering: A Distributed Problem-Solving Approach. Computer Science Technical Report 94-66, University of Massachusetts, August 1994.

[21] T. Oates, M. Nagendra Prasad, and V. Lesser. Networked Information Retrieval as Distributed Problem Solving. In *Proceedings of CIKM Workshop on Intelligent Information Agents* held in conjunction with the Third International Conference on Information and Knowledge Management (CIKM'94), December 1994.

[22] T. Oates, M. Nagendra Prasad, V. Lesser, and K. S. Decker. A Distributed Problem Solving Approach to Cooperative Information Gathering. AAAI Spring Symposium, Stanford CA, March 1995.

[23] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Los Altos, California: Morgan-Kaufmann, 1988.

[24] E. Riloff and W. Lehnert. Automated Dictionary Construction for Information Extraction from Text. *Proceedings of the ninth IEEE Conference on Artificial Intelligence for Applications*, pp. 93–99, 1993.

[25] S. J. Russell and E. H. Wefald. Principles of metareasoning. *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, R.J. Brachman *et al.* (eds.), San Mateo, California: Morgan Kaufmann, 1989.

[26] T. Sandholm, and V. Lesser. Coalition Formation among Bounded Rational Agents. *14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada.

[27] R. D. Shachter. Evaluating Influence diagrams. *Operation Research* 34(6):871–882, 1986.

[28] D. Steier, S. B. Huffman, and W. C. Hamscher. Meta-information for knolwedge navigation and retrieval: What's in there. Working notes of the *1995 AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, October 1995.

[29] J. A. Tatman and R. D. Shachter. Dynamic Programming and Influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics* 20(2):365–379, 1990.

[30] M. P. Wellman and C.-L. Liu. State-Space Abstraction for Anytime Evaluation of Probabilistic Networks. In *Proc. 10th Conference on Uncertainty in Artificial Intelligence*, pp. 567–574, Seattle, Washington, 1994.

[31] W. Gio. The Roles of Artificial Intelligence in Information Systems. *Journal of Intelligent Information Systems*, Vol. 11, No. 1, pp. 35–56, 1992.

[32] W. Gio. Mediators in the Architecture of Future Information Systems. *IEEE Computer*, pp. 38–49, March 1992.

[33] S. Zilberstein. Operational Rationality through Compilation of Anytime Algorithms. Ph.D. Dissertation, (also Technical Report No. CSD-93-743), Computer Science Division, University of California, Berkeley, 1993.

[34] S. Zilberstein. Optimizing Decision Quality with Contract Algorithms. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1576–1582, Montreal, Canada, 1995.

[35] S. Zilberstein. Resource-Bounded Sensing and Planning in Autonomous Systems. To appear in *Autonomous Robots*, 1996.

[36] S. Zilberstein and S. J. Russell. Optimal Composition of Real-Time Systems. *Artificial Intelligence*, forthcoming, 1996.