TIBOR: A Resource-bounded Information Foraging Agent for Visual Analytics

Dingxiang Liu, Jayasri Vaidyanath and Anita Raja
Department of Software and Information Systems
University of North Carolina at Charlotte,
Charlotte, NC 28223
{dliu, jvaidyan, anraja} @uncc.edu

Abstract

Visual Analytics is the science of applying reasoning and analysis techniques to large, complex real-world data for problem solving using visualizations. Real world knowledge gathering and investigative tasks are very complex because the problem-solving context is constantly evolving, and the data may be incomplete, unreliable and/or conflicting. We describe a mixed-initiative reasoning agent that will assist investigative analysts to choose from and reason about enormous databases of text, imagery, video and webcast. This agent leverages an AI blackboard system and resource-bounded control mechanisms to support hypothesis tracking and validation in a highly uncertain environment. Interactive visualizations will enable analysts to gather and sift large amounts of evidence and to collaborate with and, where necessary, to control the agent.

Motivation and Overview

An analyst's tasks are very complex because the data are constantly evolving, and typically the data are incomplete, unreliable and/or conflicting nature. The evolving nature of data implies a need for continual generation and evaluation of hypotheses so that the new evidence can be accounted for as it arrives and the most likely explanation can be produced at any given time. The incompleteness, unreliability and conflicting nature of the data imply a need for deciding which data sources to query, and what types of analysis to use for collecting, assimilating and abstracting the data into evidence**.** Moreover the analysts tasks are usually time critical and they have to use approaches ranging from quick and dirty methods to detailed, high quality investigations depending on characteristics the task.

Pirolli and Card [Pirolli and Card 2005] describe a model they developed by observing the cognitive task analysis performed by analysts as they did their jobs. They have identified two main, overlapping loops in the analyst's problem solving approach, a foraging loop and a sensemaking loop. Figure 1 describes this process. The foraging loop involves finding the right data sources; searching and filtering the information; and extracting the information. The sensemaking loop involves iterative

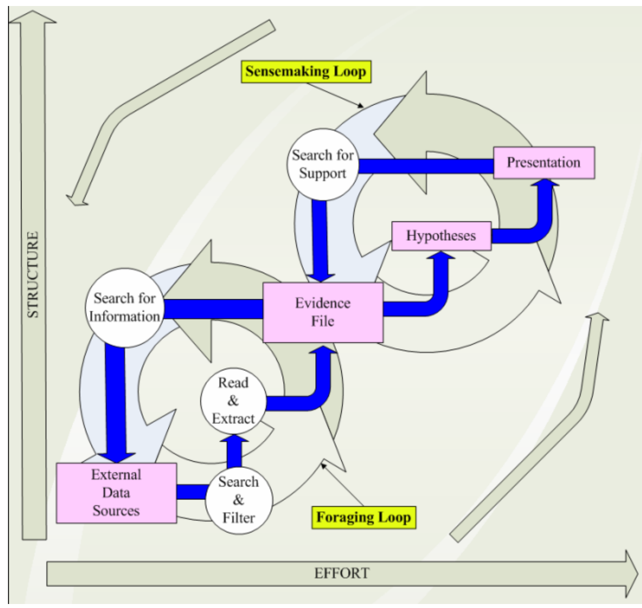

Figure 1: Pirolli and Card's Model of Analyst's Problem Solving Approach

development of a conceptualization (hypothesis) from the schema that best fits the evidence and the presentation of the knowledge product that results from this paper gives an overview of the conceptualization.

This paper gives an overview of an agent designed to handle the information foraging process and assist the analyst in his/her decision making process. Analysis tasks involve identifying and tracking multiple hypotheses by the sensemaking loop. The foraging agent supports the sensemaking loop, by gathering evidence to validate the correct hypotheses and elimination of incorrect hypotheses while solving a query pertaining to Visual Analytics. It uses interactive visualizations visualizations [Luo, H. et al. 2006, Yang, J. et al. 2006] to enable analysts to gather and sift large amounts of evidence in a time-bounded fashion and to collaborate with and, where necessary, to control the analysis domain which is inherently dynamic and uncertain. The contributions of our work are (1) a mixed-

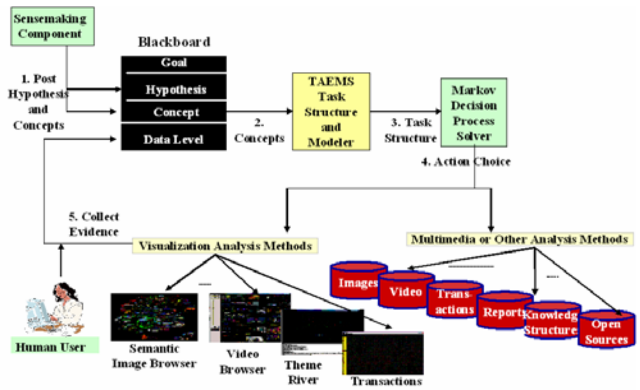

Figure 2: Functional Architecture of Foraging Component

initiative agent architecture to assist analysts in their foraging tasks (2) leveraging sequential decision making and an AI blackboard approach for gathering evidence to (in)validate hypotheses; and (3) supporting interactive visualization and exploration at every step of the problem solving process.

The rest of this paper is organized as follows. We first describe the blackboard-based information foraging agent and the resource-bounded techniques used by the agent. We then describe the current status of the system and present empirical results comparing the sequential decision making approach to a deterministic approach. Finally we summarize related work and present our future work and conclusions.

A Time-Bounded Information Foraging Agent

We have designed TIBOR, a Time BOunded Reasoning agent to handle the complex information foraging process required in analysis domains. TIBOR leverages an AI blackboard system [Morrison, C. and Cohen, P. 2006, Corkill, D. 1991, Lesser, V. et al. 2000] and resource-bounded control mechanisms to support hypothesis tracking and validation in a highly uncertain environment like the analysis domain. Figure 2 describes the TIBOR's agent architecture and control flow. TIBOR handles three types of decisions: gathering of large scale, high dimensional data from a variety of sources, which might be linked multimedia data as found on the web, broadcast video, time-dependent transactional data, telecommunication data, or other types of data; determining the type of processing to extract the data from these sources; and determining appropriate interactive visualization of these data.

The following is a description of TIBOR's decision making process and control flow. The sensemaking component posts a set of hypotheses and concepts to TIBOR's blackboard (Step 1 in Figure 2) and this triggers the TIBOR agent. The concepts are searchable entities that represent a hypothesis. To support reasoning about time and quality trade-offs, and thus a range of different resource/solution paths, TIBOR contains problem-solving models called Taems [Decker, K. 1996] task structures. Taems task structures are abstractions of the low-level execution model and these structures are generated by the Task Structure Modeler sub-component. The Task Structure Modeler takes the concepts as input (Step 2) and generates the corresponding Taems task structure (Step 3) that enumerates multiple different ways (choices for databases, visualization tools and user interactions) to obtain evidence to verify the set of concepts. These different choices are described statistically in the task structure in two dimensions, duration and quality via discrete probability distributions. We provide a detailed description of Taems in the following section. The Taems task structure associated with the concepts is then translated [Alexander, G. and Raja, A. 2006, Raja, A. et al. 2000; 2006; 2007] into a Markov Decision Process (MDP) [Bertesekas, D. and Tsitsiklis, J. 1996] which is also described in detail in the discussion that follows. The Markov Decision Process Solver computes the optimal policy for the MDP given the resource (deadline) constraints and prescribes the best action (Step 4).

User interactions play an important role in the foraging analysis making this a mixed-initiative agent. The analyst must be given the ability to manually direct a search or override actions suggested by the control mechanism, in order for this automated agent to be accepted by the analyst community. The contingency plans built into the MDP policy will allow the control system to adjust dynamically to such overrides. The automatic processing of the visualization tools along with the user interactions will determine the confidence in the concepts (Step 5). These evidential data as well their confidence values are then posted on the blackboard. The blackboard will then propagate the confidence values to verify the associated hypothesis. It is crucial for TIBOR to support both opportunistic and planned verification of hypotheses and concepts. It is probable that while gathering information to verify a concept supporting one hypothesis, the belief for a competing hypothesis increases. TIBOR's control component will model this possibility and then opportunistically redefine the problem solving process to gather evidence to verify the competing hypothesis.

The heart of TIBOR agent is the AI blackboard system [Corkill, D. 1991] that has three main components: the Blackboard; Knowledge Sources (KSs); and the Control Component. The blackboard functions as a multileveled database for the information that has been discovered and produced thus far. This information includes raw data, partial solutions and various problem solving states. The levels in TIBOR's blackboard are Goal, Hypothesis, Concept and Data, in order of decreasing granularity. The Goal level stores the resolution information. The

Hypothesis level has one or more hypotheses. Each hypothesis contains concepts which are represented in the Concept level. The Data level contains the data/evidences gathered to (in)validate the various hypothesis. The layered hierarchy allows for explicit modeling of concurrent top-down and bottom-up processing, while maintaining a clear evidential path for supporting and contradictory information. The information at a given level is thus derived from the level(s) below it, and it in turn supports the hypothesis at higher levels. For example, when evaluating the validity of a particular hypothesis, the system would examine the reliability of the visualizations used to generate the properties of the object upon which the validation will be made, each of which are in turn different types of visual or text data.

The KSs are independent computational modules that together contain the expertise needed to solve the problem. They vary in their internal representation and computational methods and do not interact directly with each other. In TIBOR agent, the KSs include databases of images, video, text and electronic transactions; the visualization tools used to interact with these databases; and the sensemaking component will serve as KSs. Some examples are ImageDBKS, VideoDBKS, IntelReportsKS, SemanticImageBrowserKS, TextKS, and SenseMakingKS etc. A blackboard directs the problem-solving process by allowing KSs to respond opportunistically to changes on the blackboard and chooses the most appropriate KS activation to execute based on the state of the blackboard and the set of triggered KSs [Corkill, D. 1991]. The control component makes runtime decisions about the problem solving process, specifically for a given hypothesis and resource (time) constraints, it will determine the databases and tools that need to be accessed. We have modeled this control process is a Markov decision process-based [Bertesekas, D. and Tsitsiklis, J. 1996] sequential decision problem. The essence of sequential decision problems is that decisions that are made in resource-bounded, uncertain environments can have both immediate and long term effects; the best action choice depends on the types of future situations.

Taems Task Structure

As described earlier, TIBOR receives concepts from sensemaking component and generates the corresponding Taems task structure [Decker, K. 1996]. For the analysis domain, the Taems task structure contains the various choices of databases, visualization tools and levels of user interaction relevant to the particular query/hypothesis. The Taems language models problem solving patterns. We can model the fact that one of several actions may be performed and also that a given method may have several possible outcomes, or that an agent has the option to perform a facilitating task before the actual one. A quantitative representation of the agent's choices using Taems allows the agent to select that which is most appropriate in the given context [Horling et al., 1999].

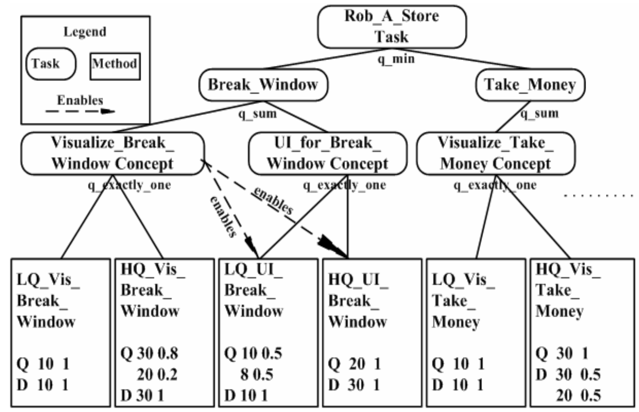

Figure 3: Taems Task Structure for Rob_A_Store Task

Figure 3 describes a simple example Taems task structure that involves finding the evidence for the hypothesis Rob_A_Store. The top-level task is decomposed into two subtasks, *Break_Window* and *Take_Money*. Since TIBOR is a *mixed-initiative* system, user interaction plays an important part and is modeled in the problem solving pattern. Rob_A_Store is modeled in the task structure by the *q_min* quality attribution factor (qaf). The *q_min* qaf states that in order for the Rob_A_Store to achieve quality, both *Break_Window* and *Take_Money* should get non-zero qualities. The task *Break_Window* has two subtasks, the first subtask *Visualize_Break_Window* will determine the mode for the semantic image browser tool [Yang, J. et al. 2006]; and the other subtask *UI_for_Break_Window* will determine the data interaction quality by the user. In order for the *Break_Window* to achieve quality, either the *Visualize_Break_Window Concept* or the *UI_for_Break_Window Concept* or both should get non-zero qualities which is denoted by the *q_sum* qaf. Both *Visualize_Break_Window* and *UI_for_Break_Window* have two more subtasks. A *q_exactly_one* qaf is functionally equivalent to an XOR operator. The quality of the *Visualize_Break_Window Concept* is the quality of any of its subtasks, provided that only one subtask has quality.

Primitive actions in Taems, called *methods*, are characterized statistically in two dimensions: quality and duration. Quality is a deliberately abstract domain-independent concept that describes the contribution of a particular action to overall problem solving. Thus, different applications have different notions of what corresponds to quality. A quality distribution Q 30 0.8 20 0.2 implies that the action will obtain a quality of 30, 80% of the time and a quality of 20, 20% of the time. *LQ_Vis_Break_Window* and *HQ_Vis_Break_Window* in Figure 3 are the two alternative ways of visualizing the data related to the *Visualize_Break_Window Concept*. The former will open up the images quickly in low quality mode using a low pixel rate; the latter will take a longer time to load the images but will have higher precision of the images. The enables non-local relationship from the *Visualize_Break_Window Concept* to the methods *LQ_UI_Break_Window* and *HQ_UI_Break_Window* implies that the *Visualize_Break_Window Concept* has to accrue non-zero quality for the primitive actions related to user interaction to be successful. In other words, the images in the database have to be successfully rendered by the visualization tool for the user to have useful interactions with the data.

In order to determine the optimal action choices, the Taems task structure is translated into a Markov Decision Process by initializing a state set and identifying the possible actions and expanding each possible outcome which are characterized by discrete quality, cost and duration values.

Markov Decision Process

A Markov Decision Process (MDP) [Bertesekas, D. and Tsitsiklis, J. 1996] is a probabilistic model for decision making and planning. It uses dynamic programming to decide on the optimal policy that yields the highest expected utility. MDPs capture the essence of sequential processes and are used to compute policies that identify, track, and plan to resolve confidence values associated with blackboard objects, which in this application correspond to evidence and hypotheses about the evidence.

A Markov Decision Process has four components: a set of actions (A), a set of states (S), a probability function (P), and a reward function (R). $P^a(ss')$ is a probability function denoted as the probability of transitioning from state s to s' via executing action *a*, while $R^a(S)$ is a reward function defined by the reward received for taking action *a*. The solution to a MDP is a policy π which is a mapping from states to actions. The value function $V^\pi(s)$ is the expected cumulative reward of executing policy π starting in state s. It can be expressed as

$$V^\pi(s) = E\,[r_{t+1} + r_{t+2} \ldots | \, s_t = s, \pi]$$

Where r_t is the reward received at time t, s_t is the state at time t.

An optimal policy is one that maximizes the expected value of the policy. The optimal value function, written as V*, is associated with a specialized form of the Bellman equations:

$$V^*(s) = \max \textstyle\sum_{s'} P(s' \mid s, a)\, [R(s' \mid s, a) + V^*(s')]$$

The MDP solver receives a task structure along with an associated deadline from the Taems task structure modeler sends the task structure it generates to the MDP solver. This deadline to validate the hypothesis can be specified by either the user or the sensemaking component. Suppose the deadline is provided as an input to the system and a policy is computed as described in section 2. Figure 4 shows the policy computed for a deadline of 200.

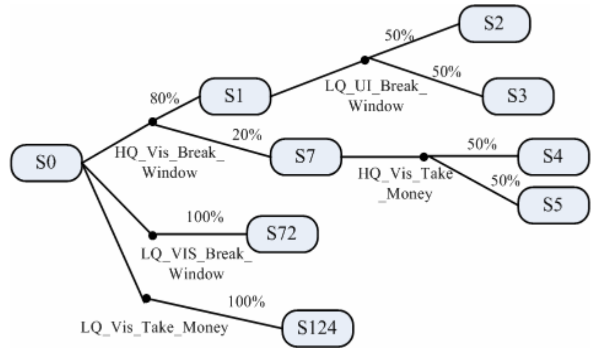

Figure 4: A partial view of the Markov Decision Process describing states, actions and transition probabilities for the Rob_A_Store task

The policy obtained from the MDP assists the analysts in looking at the trade-offs between the greedy choice of actions and the optimal choice of actions in dynamic and uncertain domains.

Runtime System Description

Suppose TIBOR agent receives the hypothesis *Rob_A_Store* from the sense making component. As shown in Figure 3, this top-level hypothesis is decomposed into two searchable concepts, *Break_Window* and *Take_Money*. The sense making component posts the hypothesis and concepts to hypothesis and concept level of the blackboard respectively. When the concepts are posted on the blackboard, it triggers the Taems task structure modeler to generate the corresponding Taems task structure as described in the previous section.

Figure 3 shows the generated Taems task structure. *Break_Window* has two subtasks, *Visualize_Break_Window* and *UI_for_Break_Window*. Suppose the deadline for validity or invalidity of the hypothesis as specified by the sense making component is 200, the MDP solver generates the best policy. The execution of high quality method *HQ_Vis_Break_Window* for *Visualize_Break_Window* opens up the semantic image browser [Yang06] and the images are displayed using a high pixel rate. The quality of *Visualize_Break_Window* is 30 and duration is 30. After completing two actions *HQ_Vis_Break_Window* for *Visualize_Break_Window* and *HQ_UI_Break_Window* for *UI_for_Break_Window*, the quality of *Break_Window* is 50 and duration is 60. Suppose an image of a broken window is found, the blackboard system can determine concept *Break_Window* is valid. The same procedure is followed for the concept *Take_Money*. Suppose no image shows money is taken, the blackboard system determines concept *Take_Money* is invalid. Since *Break_Window* is valid and *Take_Money* is invalid, the blackboard system is able to determine that the hypothesis *Rob_A_Store* is invalid.

Current System Status

We have completed a partial implementation of TIBOR that leverages the MDP-based sequential decision making process. It also supports the semantic image browser [Yang, J. et al. 2006] and allows for user interaction. We are currently in the process of integrating the blackboard into the agent. The following is a description of the implemented system along with screen shots when executing the Rob_A_Store task with a deadline of 200.

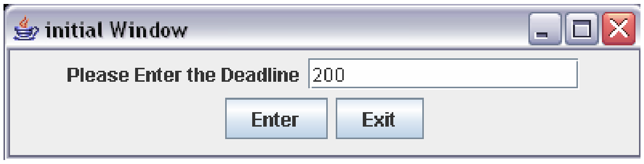

Figure 5: The system being triggered with a deadline of 200

TIBOR has a control panel, shown in the left window of Figure 6 as well as Figure 7 that allows the user to interact with the agent by specifying his/her choices and also to track the progress of the problem solving process. It can be thought of as a dashboard having all the controls to manage the user's decision-making process. The control panel provides two types of views, the current view provides the decisions made by system on sources, analysis tools and visualization tools as well as interaction decisions on interactions made by the user and the future view provides the analyst choices about data sources, analysis tools, visualization tools and hypothesis. The system is designed to run in two modes, Minimized (precognitive) mode and the Maximized mode. The control panel also has a time slot that keeps the user informed about the time used.

The Executed Methods window lists all the actions that have been executed so far. The View Task Structure button displays the TAEMS task structure as a snapshot of the current state of problem solving. The View Markov Process button will enable the users to view the decision process as well as the optimal policy and is under construction. The control panel also provides the user with the accumulated quality to show the quality accrued after each action completes execution. When an action is chosen the control panel triggers the associated visualization tool (the semantic image browser) as shown in the right window of Figure 6 and 7. The agent continues its execution providing recommendations for user interaction along the way**.** The user interaction time allows the user to get a better view by modifying the images like zoom in and out. The semantic image browser provides a VAR view which provides the user with the details of how the information present is grouped together. Figure 6 is the screen shot showing the system with the VAR view. Figure 8 is the screen shot of the control panel when the MDP policy terminates. It provides a list of actions executed, the accumulated quality for the top-level task and total execution time.

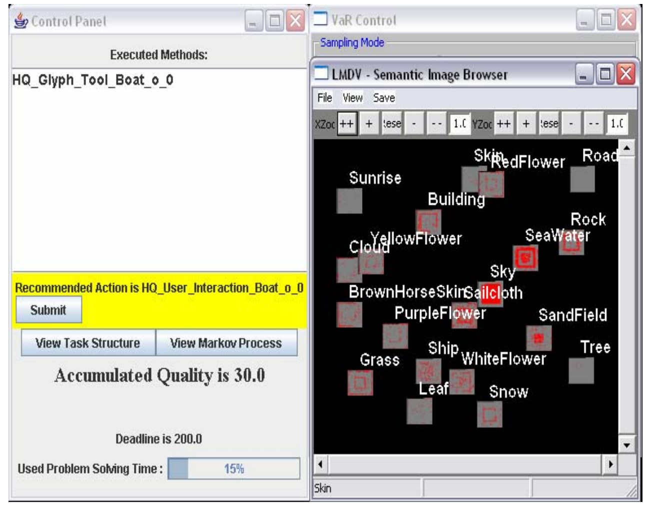

Figure 6: The VAR view showing the grouped data

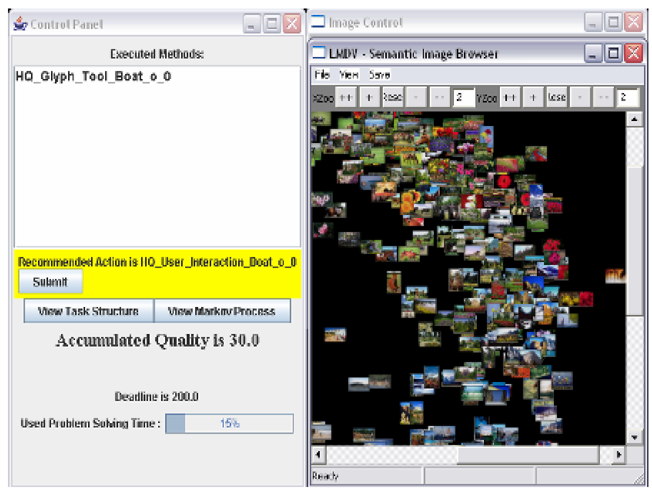

Figure 7: Screenshot of TIBOR in the midst of executing HQ_User_Interaction method: the control panel is on the left and the semantic image browser on the right.

Empirical Results

In this section, we describe our efforts to evaluate the MDP-based decision making mechanism for task structures representing different types of analytical tasks. As described earlier, the MDP approach produces a policy that will dynamically adjust the problem solving process to the deadline and runtime execution characteristics. We define a deterministic method which is commonly used for planning [Lesser, V. et al., 2000] that builds a static schedule with the highest possible quality for the deadline. This method would require rescheduling to adjust to runtime characteristics.

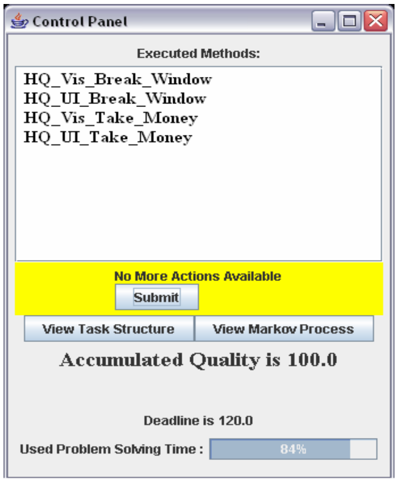

Figure 8: Screen shot displaying accumulated task quality as well as time used when execution terminates.

The experiment was designed to compare the quality of the MDP policy for each task structure to that produced by the deterministic plan. We generated six task structures with deadlines of 20, 40, 60, 80, 120 and 200 respectively. These task structures were similar to the task structure described in Figure 3 but varied in their quality and duration distributions.

Given the duration distributions of the task structures, deadlines in the 20-60 range are considered loose deadlines, 60-120 time units are considered medium tightness deadlines and 200 time units are categorized as a tight deadline. We used the MDP-based controller to generate policies and ran five simulations of each for each task structure and recorded the quality obtained at the top-level task as well as the total execution time for each run. The average quality of the executed policy for each task structure is shown in the lightly-shaded histogram in Figure 9. We then ran five simulations for each task structure using the deterministic scheduled and averaged the quality obtained for each task as well as the average execution time. For each task, the average quality obtained is shown in the dark-shaded histogram in Figure 9. Figure 9 shows that the MDP policy performs better than the deterministic method as it is able to adapt to the uncertainty and dynamism of the runtime environment.

Related Work

Blackboard-based architectures have been previously used for complex information gathering and analysis tasks. Morrison and Cohen [Morrison, C. and Cohen, P. 2006] use a Bayesian blackboard called AIID to serve as a prototype system for analysis and data fusion. BIG (Lesser, V. et al., 2000) is a resource-Bounded Information Gathering agent that combines several sophisticated AI components.

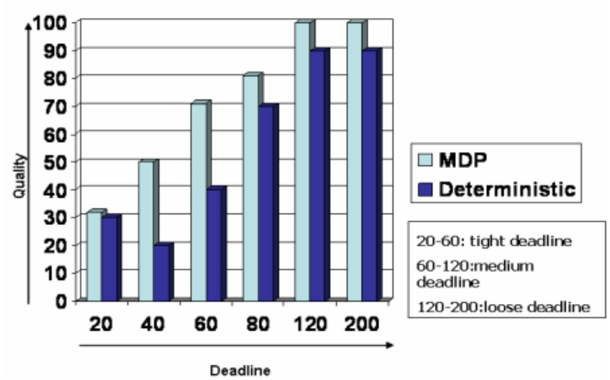

Figure 9: Comparison of Quality

BIG gathers web-based information, extracts information from both unstructured and structured documents, and reaches a decision. BIG uses an opportunistic linear planner and scheduler to direct its activities. TIBOR focuses on the end-to-end decision reasoning of analytical tasks and uses a MDP-based approach to reason about the inherent uncertainty of the domain. TIBOR is designed to be a fully-functional mixed-initiative agent that leverages the state-of-the-art in visualization tools. The control panel in TIBOR also enables the human user to track the problem solving process at various levels of abstraction. .

Conclusions and Future Work

We have described TIBOR, a mixed-initiative agent capable of assisting humans in complex analysis tasks using visualizations. We have identified abstract representations of these tasks to assist in the automated analysis as well as integrated the agent with an image database and the semantic image browser visualization tool. We have also implemented an MDP-based resource-bounded control mechanism that will reason about these tasks. Our next step is to complete integration of the AI blackboard as well a variety of knowledge sources, including a sensemaking knowledge source that uses case-based reasoning and pattern recognition; as well as other visualization tools such as the semantic video browser [Luo H. et al. 2006].

.

Acknowledgement

This work was sponsored by the National Visualization and Analytics Center (NVAC) under the auspices of the Southeastern Regional Visualization and Analytics Center. NVAC is a U.S. Department of Homeland Security Program led by Pacific Northwest National Laboratory.

References

Alexander, G. and Raja, A. 2006 "*The Role of Problem Classification in Online Meta-Cognition",* In Proceedings of the International Conference on Intelligent Agent Technology (IAT 2006).pp. 218-225 Hong Kong, China.

Bertesekas, D. and Tsitsiklis, J. 1996 Athena Scientific, Belmont, MA. Neuro-Dynamic Programming.

Morrison, C. and Cohen, P. 2006. The Hats Simulator and Colab: An Integrated Information Fusion Challenge Problem and Collaborative Analysis Environment. . ISI 2006: 105-116.

Corkill, D. 1991 September, *AI Expert*, 6(9): 40-47.*Blackboard Systems*

Decker, K. 1996 January. *TAEMS: A framework for environment centered analysis and design of coordination mechanisms*, in Foundations of Distributed Artificial Intelligence, Chapter 16, pages 429-448. G O'Hare and N. Jennings (eds.), Wiley Inter-Science.

Horling, B. et al., 1999 The TAEMS White Paper Unpublished, January 1999.

Luo, H. et al., 2006, Exploring Large-Scale Video News via Interactive Visualization, *Charlotte Visualization Center Technical Report*, CVC-UNCC-06-04.

Lesser, V. et al., 2000, BIG: An Agent for Resource-Bounded Information Gathering and Decision Making" In *Journal of Artificial Intelligence, 2000 Special Issue entitled 'Internet Applications'*, May 2000, vol 118, issue 1-2, pages 197-244.

Pirolli, P. and Card, S., 2005 The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. *Proceedings of 2005 International Conference on Intelligence Analysis*, pp. 2-4.

Raja, A., et al. 2000 Robust Agent Control in Open Environments. *Proceedings of the Fourth International Conference on Autonomous Agents*, pp. 84-91, ACM Press.

Raja, A., et al., 2006, "Leveraging Problem Classification in Online Meta-Cognition", *Proceedings of AAAI 2006 Spring Symposium on Distributed Plan and Schedule Management*, pp 97-104, Stanford, CA, March 2006.

Raja, A. and Lesser, V. ""A Framework for Meta-level Control in Multi-Agent Systems" To appear in *Proceedings of Autonomous Agents and Multi-Agent Systems*, 2007.

Yang, J. et al., Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis. *Charlotte Visualization Center Technical Report CVC-UNCC-06-02*, 2006.