OmniSeer: A Cognitive Framework for User Modeling, Reuse of Prior and Tacit Knowledge, and Collaborative Knowledge Services

John Cheng and Ray Emami
Global InfoTek
jcheng@globalinfotek.com

Larry Kerschberg
KRM, Inc
kersch@gmu.edu

Eugene Santos, Jr,. Qunhua Zhao, Hien Nguyen, and Hua Wang
University of Connecticut
eugene@cse.uconn.edu

Michael Huhns, Marco Valtorta (corresponding author), Jiangbo Dang, Hrishikesh Goradia, Jingshan Huang, and Sharon Xi
University of South Carolina
mgv@cse.sc.edu

Abstract

This paper describes the current state of the OmniSeer system. OmniSeer supports intelligence analysts in the handling of massive amounts of data, the construction of scenarios, and the management of hypotheses. OmniSeer models analysts with dynamic user models that capture an analyst's context, interests, and preferences, thus enabling more efficient and effective information retrieval. OmniSeer explicitly represents the prior and tacit knowledge of analysts, thus enabling transfer and reuse of such knowledge. Both the user and cognitive models employ a Bayesian network fragment representation, which supports principled probabilistic reasoning and analysis. An independent evaluation of OmniSeer was carried out at NIST and will be used to guide further development.

1. Introduction

Perhaps the two most daunting issues facing intelligence analysts today are: (1) the extremely large amount of data available for analysis, and (2) the analysts' inherent cognitive limitations and biases. Data volume makes it almost impossible for analysts to find nuggets of information relevant to specific hypotheses or to construct novel hypotheses from information fragments – analysts are "drowning in data and starved for information." Cognitive limitations increase the likelihood that analysts do not correctly or adequately exploit the information that they actually receive. The OmniSeer system provides intelligent, automated tools that address both issues. In particular, both users and data are explicitly modeled. This enables OmniSeer to discover analyst preferences, interests, contexts, and biases, find relevant – such as surprising or high-value – information, and form plausible hypotheses about imminent, significant events.

OmniSeer is comprised of three major research thrusts:

- Bayesian network representation of user models,
- Bayesian network representation of prior-and-tacit knowledge, and
- Software agency-based cognitive framework for knowledge services.

The Bayesian network (BN) representations of user models and prior-and-tacit knowledge are an innovative, principled way to represent uncertain information. BN representations enable the system to: (1) model user states and contexts, (2) manage prior shared knowledge, (3) capture tacit knowledge, (4) generate multiple hypotheses about possible threats given evidence, (5) reason efficiently to select the most relevant and plausible hypotheses, (6) guide the search for additional evidence to sharpen conclusions, (7) identify novel situations, and (8) help analysts justify their confidence in conclusions reached. The agency-based cognitive framework enables OmniSeer to support multiple software agents, software services and classes of agents (e.g., data-finding agents) in a scalable, easy-to-integrate fashion.

2. OmniSeer System Concept

The OmniSeer system concept consists of software agencies that manage agents or services in support of analysts as they sift and winnow massive amounts of data in search of data that fit situation-specific scenarios. The remainder of this section deals with agencies supporting the User Model, Prior and Tacit Knowledge, and the Cognitive Framework for Knowledge Services.

2.1 Analyst Agency for User Modeling

An important research goal is to understand and develop the structure of the user/analyst model, the dynamic nature of the construction process of that model, and the services offered to other analysis processes. To capture the analyst's intent, the OmniSeer dynamic user model clearly delineates a user's interests, preferences,

and context, focusing especially on the interactions among them as they change over time. The services provided by the analyst model are used by other processes in OmniSeer to better serve the analyst in analyzing and detecting important and critical information. By constructing the user model "on-the-fly," the model adapts quickly to the changes in the goals and the approaches used by the analyst, thereby improving the quality of the information seeking process and the analyst's experience with the system.

2.2 Decision Agency for Knowledge Reuse

The OmniSeer approach for using prior knowledge is to capture an analyst's experience and knowledge in the form of a repository of Bayesian network fragments. These are then used to process intelligence reports. Analyses of, and conclusions about, the reports are presented to the analyst. If the analyst characterizes the reports differently than the OmniSeer system, these differences are captured from the analyst. The differences represent errors (normally on the part of OmniSeer), or that analyst's tacit knowledge – knowledge that was not previously represented or available to OmniSeer. Tacit knowledge is represented as Bayesian network fragments and is added to the repository.

When information – e.g., incoming messages or results from an open-source search – is to be processed by OmniSeer, it is compared and matched to the Bayesian network fragments. Matched fragments are then composed into larger and more complete scenarios, representing possible terrorist activities. (We use terrorist activity as a generic context in examples.) The scenarios are evaluated for surprise and robustness and, under user guidance, for what additional information is most needed to confirm uncertain conclusions.

An analyst interacts with the system via a graphical interface that displays the scenario as a set of events and the causal links among them. (Cf. Figure 7.) The events and links are derived from the original messages and input information, and from prior knowledge that the system has acquired.

2.3 Agency-Based Cognitive Framework

The OmniSeer cognitive framework is depicted in Figure 1. It consists of four software agencies that focus on supporting the agents and services of that particular agency. They are the Analyst, Decision, Query, and Core Agencies. The agencies communicate via a Knowledge Bus that allows for the exchange of large-grain knowledge "nuggets."

The agents/services that reside within an agency have access to multiple knowledge sources and these in turn access multiple data sources including domain specific documents (e.g., weapons of mass destruction (WMD)), open source databases, the World Wide Web (Web), and real-time message streams. We now discuss the various agencies, their associated agents and services, and the knowledge/databases they use [1-3].

2.3.1 Analyst Agency. This agency deals with the user and provides services such as User Model Management, the User Model, and Explanation services. The services create knowledge regarding user interests, preferences, the action net and user context. This knowledge is used to focus OmniSeer resources on the task at hand by narrowing the search space to topics of interest and relevant supporting evidence.

2.3.2 Decision Agency. This agency is responsible for services related to Bayesian reasoning, the matching of BN fragments with incoming evidence, the composition of BN fragments into a scenario, the assessment of the value of new information, a determination of the scenario's sensitivity to the quality of available information, and a calculation of how well evidence matches an expected scenario. The mismatch between the evidence and a scenario measures the surprise of the situation.

2.3.3 Query Agency. This agency is responsible for accepting a search request based on user interests a hypothesis, and, possibly, a scenario supporting the hypothesis, providing generalization or specialization services to expand or focus the query, and decomposing it for submission to target databases. The graph generator agent works closely with the Analyst and Decision Agencies to pose the query as a subgraph of the domain ontology, user interests, the hypothesis, and the scenario.

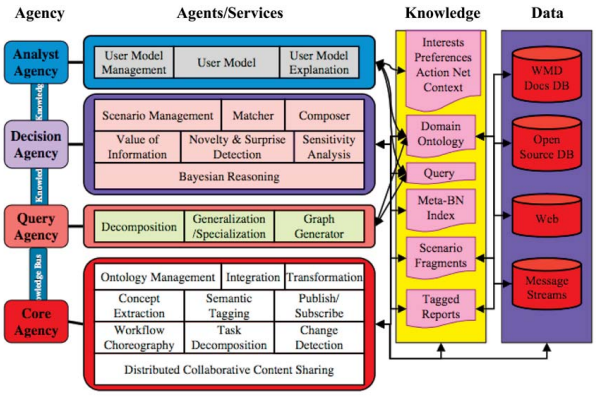

Figure 1. OmniSeer Software Agency-Based Cognitive Framework

2.3.4 Core Agency. This agency provides a collection of cognitive and infrastructure services that enable OmniSeer to support the Analyst, Decision, and Query Agencies. There are services that maintain and evolve the knowledge base components, including the domain

ontology, the queries submitted by analysts, standing queries that monitor data sources for new evidence culled from internal reports, open source databases, the Web, and real-time message streams. The core services also deal with concept extraction and semantic tagging, although for this project, we assume that tools are available, or will soon be available to perform these tasks. The core agency also provides for workflow choreography to decompose the tasks, coordinate their execution, and manage the information flow between agents and agencies in processing user requests.

2.4 OmniSeer Functional Architecture

The OmniSeer functional architecture is shown in Figure 2. The inputs to OmniSeer are message traffic, open-source news feeds, and documents resulting from searches on the Web. The outputs are intelligence reports in the form of analyzed terrorist scenarios and indications of which additional information is most needed. An additional outcome is an accumulation of reusable knowledge, representing an analyst's expertise, biases, and judgments.

When information arrives, it is first filtered by a user model that expresses the preferences and interests of an analyst. The arriving information is processed to extract its concepts and represent them in document graphs, which are then matched with user interests expressed as queries. The relevant information that passes the filtering is then matched with Bayesian network fragments representing common parts of terrorist activities, events, and situations. The syntax of the fragments is XMLBIF, and the semantics is determined by a domain ontology, consistent with Cyc, that was developed in Protégé and represented in RDF (Resource Description Framework).

Information that completely or partially matches the BN fragments is stored in a repository as instantiated fragments and made available to a composer. The composer combines the instantiated fragments into a situation-specific scenario, which is then analyzed to determine (1) if the situation is novel or surprising, (2) if the conclusion is particularly sensitive to any of the original pieces of information, and (3) if there is additional information that would be especially valuable in strengthening or changing the conclusion. The third kind of analysis is illustrated in Figure 9, for an example scenario.

If the analyst disagrees with OmniSeer's conclusion, the analyst is prompted for additional information that he/she might have but the system does not. This tacit knowledge is captured by OmniSeer in the form of additional BN fragments.

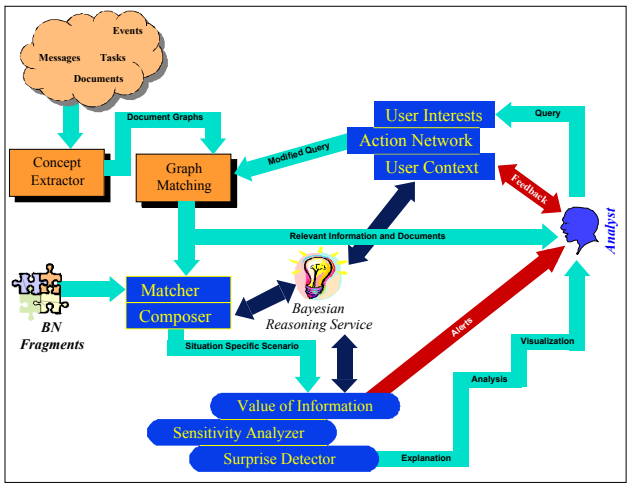

Figure 2: OmniSeer Functional Architecture

3. OmniSeer Research Approach and Accomplishments

3.1 Role of Bayesian Networks for Both User Modeling and Prior-and-Tacit Knowledge

The two subsystems share probabilities, thereby strengthening the relationship between them and reinforcing the dual use of Bayesian networks for representation and reasoning. For example, the user model for a particular analyst might indicate that the analyst is interested with high probability in illegal financial transactions. Because of this, intelligence reports dealing with financial transactions would more likely be processed by the prior-and-tacit-knowledge subsystem. Once the prior-and-tacit-knowledge subsystem detects an interesting financial scenario and suspicious people associated with it, it might suggest to the analyst that these people be investigated. If the analyst concurs, then these people might be added to the analyst's interest list in the user model.

3.2 User Modeling for Knowledge Capture of Interests, Context, and Queries

At the heart of this effort are the user model network components, which provide the dynamic user modeling capabilities. The user/analyst model is situated between the analyst and the intelligent software tools which in OmniSeer, are tailored to the specific analyst's needs to better enhance and support analytic activities [5, 7-10]. The user model continuously monitors these activities and proactively predicts and explains analyst goals and intentions. The results from intelligent software tools or subsystems (such as the prior and tacit knowledge subsystem) can also provide essential domain feedback and knowledge for the user model to better model the

analyst with respect to his/her tasks and specific domains of operation. As the analyst performs his/her functions, the results from the user model will also serve as input to the various software tools. The next section discusses important aspects of the User Model.

3.2.1 Analyst Intent Prediction. The primary objective is to develop a user model that can predict the analyst's goal and intent. The user model then assists the analyst with his/her information retrieval tasks and also provides services to other intelligent services. The services provided by the User Model include:

User Model Component Networks: Interests, context and preferences. There are three basic components in the user model structure: Interests (What is the analyst focusing on?), Context (Why is the analyst focused on these interests?), and Preferences (How does the analyst seek and view information?)

User Model Loading: Initialization of User Model. The user model is initialized either by loading a prior model for a returning analyst, or selecting a user model template for a new analyst.

User Model Updating: Feedback-based updates. The user model is updated based on the observation of an analyst's activities, and feedback from the analyst and other services.

User Model Query Service. The user model supports other intelligent services by providing information on an analyst's current goals and interests, and disseminates "knowledge nuggets" regarding an analyst's intentions. The term "query" here refers to the queries initiated by other services, such as value information or surprise detection.

User Model Explanation. Provides detailed feedback to analysts and other services regarding decisions made by the user model and facilitates the exchange of knowledge among analysts.

Query Modification Service. Analyst's information queries are proactively modified and new queries may be recommended to the analyst based on the current user model. The term "query" here refers to the queries issued by an analyst in an information-seeking task.

3.2.2 Dynamic Cognitive User Modeling. Our approach is to build the user/analyst model based on Bayesian networks, which capture the uncertain nature of knowledge acquisition and the human reasoning process. The user model also focuses on the interactions among different components in a dynamic fashion, which reflects the nature of the dynamic environment in which the analyst performs his/her tasks.

Role of "knowledge nuggets" for transmitting knowledge via message communication. "Knowledge nuggets" are the basic informational units in the messages communicated to and from the user model. This enables the user model to provide assistance to other intelligent services and to learn from feedback. It also provides a mechanism by which analysts may share knowledge.

Definition of "knowledge nuggets" as directed graphs with probabilities. Knowledge nuggets are directed graphs that are probabilistic in nature, and contain concepts and relations among the concepts. These include messages from services such as behavioral explanation, prediction, external feedback, etc. Typical knowledge nuggets are derived from the individual behavioral model as well as domain model information and ontological information.

Bayesian Network approach to user models. The analyst acts based on the goal that he/she wants to achieve and the environment in which the action is going to be carried out. A Bayesian Network model is appropriate here to capture the uncertain, causal relationship between the pre-conditions, goals and actions.

Context networks of user interests. The user context network captures the reasons why the analyst focuses on a certain set of interests. It contains the knowledge the analyst learned from previous retrieved information. A context network is constructed "on-the-fly" by finding the common subgraphs among the graph representations of relevant documents. Figure 3 shows the construction process of the context network.

Algorithm for Construction of a User Context Network:

Input: a list of documents indicated by the analyst as relevant, which are in the form of document graphs (DGs) [8].

Output: an updated context network.

Process:

- Select a sub-graph X in the first document graph (DG)
- Calculate the relevancy weight of sub-graph X, which is the ratio of the number of DGs containing X to the to the total number of relevant documents
- If the relevancy weight of sub-graph X exceeds a given threshold, then compose it into Context Network.
- Continue previous steps until all the sub-graphs have been processed.
- Return the updated context network.

Note:

- Sub-graph matching is labeled; not as hard as general graph isomorphism problem.
- Size of sub-graph is bounded for complexity purposes.

Figure 3: User Context Network Construction Algorithm

Context network reasoning is currently based on spreading activation. Each node in the network has an associated weight that indicates the level of user interest in the concept/entity represented by this node. Nodes in the network representing evidence are initially activated. These activations will propagate through the network based on node connectivity/linkage patterns and their weights; nodes that are connected to a sufficiently large (threshold-based) weighted number of activated nodes will themselves be activated.

Preference networks for user query modification. The preference network captures the user's actions, the pre-

and post-conditions of the environment, and gives suggestions to the analyst on the decision being made. It is represented by a Bayesian network, which contains four kinds of nodes: pre-condition nodes (user interests or related queries), goal nodes (filtering or broadening suggestions), action nodes (modified queries) and intermediate nodes (AND gates). Each goal node is associated with a set of pre-condition nodes, and each action node is associated with one goal node.

Algorithm for Creating and Updating the Preference Network:

Input: a new query in the form of a document graph [8] and a set of concepts/entities.

Output: an updated preference network.

Process:

- If the query or part of the query exists in the preference network, the corresponding pre-condition nodes will be set as evidence.
- Create a goal node representing a tool (filter or broadener).
- Add a node for the query and nodes for the concepts/entities as preconditions.
- Associate the pre-condition nodes to the goal nodes.
  - If there are more than 2 pre-condition nodes, then insert AND gates as intermediate nodes for the purpose of reducing the size of the network.
- Add an action node that represents one way to modify the query.
- Associate the goal node to the action node.
- Compute the probability of different tools that will help to improve information retrieval results based on the history/record of user feedback.
- Return the network with the highest probability that will improve the search results.

Figure 4. Algorithm for creating or updating the user model preference network

3.3. Prior and Tacit Knowledge (PTK)

The overall approach for utilizing prior knowledge is to represent it in the form of Bayesian network fragments and store the fragments in a repository. The fragments are used to process intelligence reports and construct interpretation models that are analyzed by Bayesian reasoning methods. The results are presented to an analyst if they match the analyst's interests with sufficiently high probability and if they are sufficiently novel or surprising. If the analyst disagrees with the results of this subsystem, because the analyst possesses knowledge that OmniSeer does not have (*tacit knowledge*), the specific points of disagreement are captured in the form of BN fragments and added to the repository for future use by OmniSeer.

3.3.1. Representation Issues

Using Bayesian network (BN) fragments. Incoming intelligence data is first filtered by preferences and interests specified in the user model. The intelligence data is matched, either completely or partially, with Bayesian network (BN) fragments. The BN fragments, stored in a repository, represent an analyst's prior knowledge about terrorist activities or other domains of interest specified in the user model. Relevant facts extracted from the documents and messages fill in the details of the BN fragments of interest. The matched fragments, whether completely or just partially matched, are stored in the repository. Figure 5 shows the matching algorithm.

Matcher Algorithm:

Input: Bayesian network fragment repository, data report to be matched.

Output: List of newly matched fragments.

Process:

- The system represents each incoming data report as a mapping between ontology concepts (including their state and attributes) and values.
- Every BN fragment in the repository is tested for *unification* with the report.
  - Every variable in the fragment is checked with the data report to determine if the report contains any information about the ontology concept that the variable refers to. The state value is also checked if the variable is instantiated.
  - If the above test is successful then we determine if the variable's attribute instantiation is consistent with the information in the data report. A variable is considered to unify with the data report if no violations are detected.
  - A BN fragment is considered to successfully unify with the data report if at least one non-instantiated variable in the fragment unifies with the report and no variable reports any inconsistencies with its attribute assignments.
- Every BN fragment that successfully unifies with the data report is *bound* with the information in the report.
  - Every ontology concept in the data report is checked for the existence of a corresponding variable in the fragment. The variable's attributes are updated with the new bindings from the report.
- The newly matched BN fragment is tested for *eligibility*. A fragment is considered eligible if all its essential variables are instantiated.
- Eligible fragments are added to the list of newly matched fragments. The list is returned as the output after all the existing fragments in the repository are considered for matching.

Figure 5. PTK BN-Information Matching Algorithm

Developing an ontology of events and activities, which is needed for Bayesian reasoning, and compatible with the Cyc ontology. The nodes of the fragments are based on a domain ontology that we have developed for describing terrorist activities, based on a structure provided by an analyst from the intelligence community and other documents provided by the program manager. The ontology is represented in RDF with terms that are compatible with the Cyc ontology. Details about each node are represented by associated attributes. The matcher makes use of the attributes in its operation.

Composing BN fragments that have attributes attached to each node. Instantiated BN fragments are retrieved from the repository and composed into scenarios specific to the situation at hand, termed *situation-specific scenarios*. The composition is based on matches among instantiated nodes, where the values of the attributes are compatible. Figure 6 shows the algorithm for composition.

A large repository can result in computational performance problems. To prevent these, a component must be developed to enable the "forgetting" of partially instantiated BN fragments that are no longer needed. This will enable OmniSeer's efficiency to be maintained. Note that a form of this forgetting is achieved by the pruning

that occurs in the last step of the composing algorithm in Figure 4.

Composer Algorithm:

Input: Bayesian network fragment repository, list of newly matched fragments.

Output: Repository updated with new fragments.

Process:

- For each BN fragment *bnf* in the newly matched fragments list *newFrags*:
  - Test if *bnf* is a duplicate. If so, then delete *bnf* from *newFrags* list.
  - For each BN fragment *repFrag* in the repository:
    - Test if *bnf* is a subset of *repFrag* or vice-versa. If so, then *repFrag* and bnf are not composed together.
    - If *bnf* and *repFrag* share at least one node that unifies, then construct a new situation specific scenario, *sss*, by merging the two fragments. The shared node is updated to reflect the bindings of both the contributing fragments. The probability distribution table for the shared node is also modified appropriately to accommodate new parents (if they exist).
    - Check the validity of the new scenario *sss* to ensure that the two contributing fragments relate to the same set of events, and that *sss* is not a duplicate. If valid, then add *sss* to the *newFrags* list.
  - Add *bnf* to the fragment repository, *repFrag*.
  - Sort *repFrag* to propagate the better scenarios to the top. Prune the repository to prevent the solution from becoming intractable.
- Return the fragment repository, *repFrag*.

Figure 6. PTK BN Composition Algorithm

3.3.2. Processing and Reasoning Issues. Situation-specific scenarios provide a rich representation for analysis by Bayesian reasoning services, which include a Bayesian network shell for probability update, a value of information (VOI) computation service, a surprise detection service, and a sensitivity analysis service. Scenarios are built in such a way that they are immune to the anchoring bias and share the desirable properties of Bayesian networks in eliminating other the vividness bias and other kinds of cognitive bias.

First, the probability of the variables represented in a situation-specific scenario is updated to be consistent with the evidence at hand. In this way, the situation-specific scenario tracks the variables of interest to an analyst. When the probability of a particular value of a variable of interest becomes sufficiently high, an alert is issued to the analyst.

Second, a value of information analysis is carried out, in order to identify the variables that have the most potential impact on the probability profile of a variable of interest. Such especially informative variables can then become the subject of focused queries.

Third, the surprise detection service continuously looks for situations in which the existing evidence is not well-explained by the existing situation-specific scenarios. Evidence that is not explained by any scenarios is a sign that new Bayesian network fragments are needed, and an analyst may need to be alerted of the inability of the existing models to explain a novel pattern of evidence [11].

Fourth, the sensitivity analyzer assesses the robustness of the analytical conclusions with respect to parametric assumptions contained in the model and with respect to the evidence. This assessment is normally driven by an analyst's request.

3.3.3. Graphical Interface for Analysts. The prior-and-tacit knowledge subsystem produces a mathematically correct analysis of available of events and evidence, but the results of the analysis need to be presented in a form that is readily understandable by analysts and can form a plausible basis for their decisions about terrorist threats, actions, or predicted events. Our graphical interface, shown in Figure 7, enables analysts to interact with computed terrorist scenarios, determine their robustness and novelty, and decide whether to seek additional information.

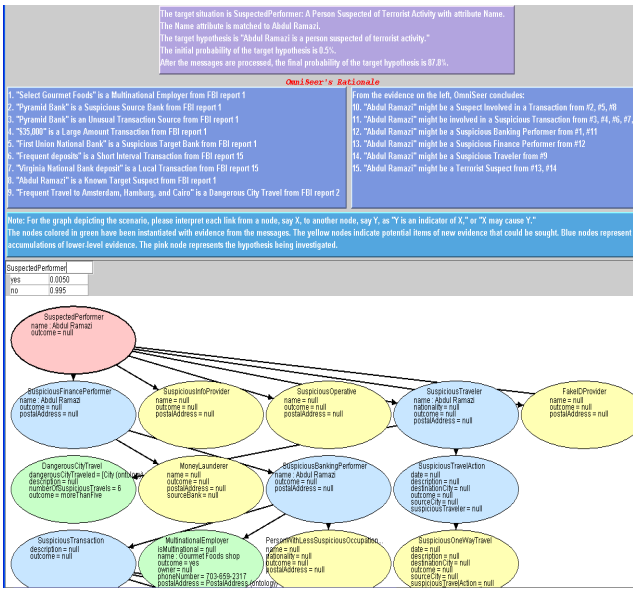

Figure 7. GUI for viewing and analyzing situation-specific scenarios

4. Examples

4.1 End-to-end Analyst Interaction Example

Consider the following scenario, which is based on "The Sign of the Crescent," a tutorial example developed by Frank Hughes, a professor at the Joint Military Intelligence College. Imagine that an analyst is required to monitor banking transactions with the objective of identifying patterns of suspicious activity. In OmniSeer, this would result in the prior-and-tacit knowledge subsystem sending to the used modeling subsystem a standing query of the form

SELECT * FROM Messages WHERE MessageContent concerns BankingTransaction

With the help of an ontology, as described in the following section, any message that mentions a bank account, a deposit, a withdrawal, a transfer, etc. is retrieved. The result is FBI Messages #1, #15, #16:

1) Report Date: 1 April, 2003. FBI: Abdul Ramazi is the owner of the Select Gourmet Foods shop in Springfield Mall. Springfield, VA. (Phone number 703-659-2317). First Union

National Bank lists Select Gourmet Foods as holding account number 1070173749003. Six checks totaling $35,000 have been deposited in this account in the past four months and are recorded as having been drawn on accounts at the Pyramid Bank of Cairo, Egypt and the Central Bank of Dubai, United Arab Emirates. Both of these banks have just been listed as possible conduits in money laundering schemes.

15) Report Date 20 April. 2003: FBI: Mukhtar Galab has an account at the Virginia National Bank in Charlottesville, VA. Bank records say he has deposited several checks in the last three months, totaling $13,000, drawn on account number 1070173749003 held by Abdul Ramazi at the First Union Bank in Springfield, VA.

16) Report Date 22 April. 2003. FBI: Hani al Hallak, of North Bergen NJ, has deposited checks in his bank account that were drawn on First Union Bank account number 10701737490Q3 in Springfield VA in the name Abdul Ramazi. The latest check is dated 16 April, 2003 and was in the amount of $8500.

Analysis of the resulting situation-specific scenario by the Bayesian reasoning services would reveal that there are several possible suspicious people. Using VOI and the fact that Ramazi is mentioned in all three messages, the system would decide that Ramazi is the most likely suspect and would request additional information about Ramazi. So the second step is:

The Prior and Tacit Knowledge subsystem sends to the User Model subsystem a standing query of the form:

SELECT * FROM Messages WHERE MessageContent concerns Ramazi

The result would be FBI Messages #1, #2, #15, #16, and #20. The messages that were not considered before, i.e. #2 and #20 are copied below:

2) Report Date: 5 April. 2003. FBI: Passport control at Dulles Airport in Wash DC records that Abdul Ramazi; holder of US passport # 177183634 (issued by Passport Agency, Wash. DC on 12 Feb. 1997) has made three trips to Amsterdam, two trips to Hamburg, Germany, and three trips to Cairo, Egypt in the last five months. The address given by Ramazi on his passport is 1176 Floyd Ave., Springfield, VA. Phone number at this address is 703-734-0104.

20) Report Date 26 April, 2003. FBI: A check of rented storage facilities in the Richmond and Charlottesville areas reveals that a man giving his name as Abdulla Ramzi rented storage unit # 174 on 10 April, 2003 at the Budget Storage Units in Keswick, VA. Ramzi gave his address as 2932 University Drive, Charlottesville, VA. Ramzi paid in cash for a month's rental.

From these messages, the situation-specific scenario in Figure 8 is assembled. The Bayesian reasoning service for probability update would conclude that Ramazi is a suspicious person and bring this to the analyst's attention.

The user interacts with the Bayesian reasoning services to analyze the scenario. In particular, the Value of Information Service suggests following up on Ramazi's travels, because information about his travels would decrease the expected uncertainty about whether Ramazi is a terrorist, as shown in Figure 9. The user then interacts with the user model to express a refocusing of the preference network. New messages continue to stream in, and OmniSeer supports various types of reasoning and multiple-hypothesis tracking.

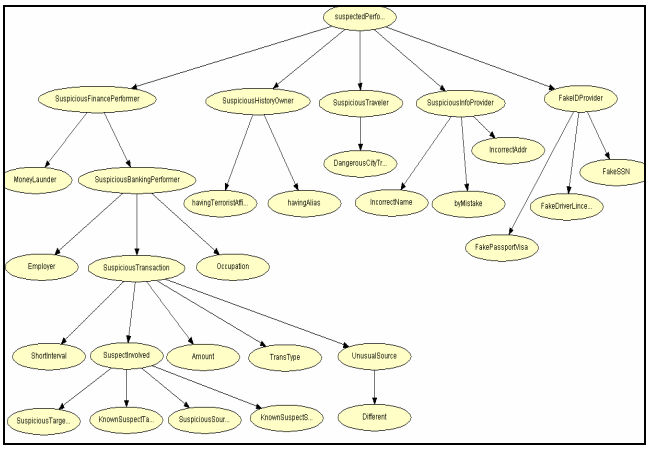

Figure 8. A Situation-Specific Scenario

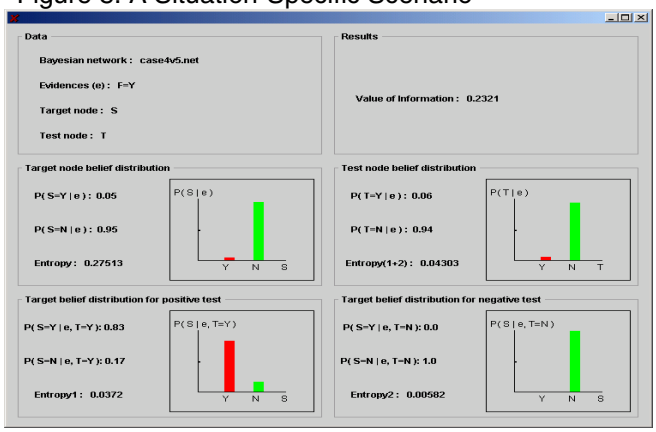

Figure 9. Value of Information Computation

4.2. Domain Modeling and Message Classification

In order for OmniSeer to handle message traffic from heterogeneous sources such as reports, communiqués, intercepts, etc., we posit the need for a domain ontology, as shown in the UML language in Figure 10. This is a very general model which denotes that an Intelligence Analyst has a Profile, and defines one or more scenarios. A scenario might represent a query or, more generally, provide context and support for a hypothesis. A scenario is comprised of one or more Item of Interest and each such Item of Interest has information provided by several Information Sources. An Item of Interest may be specialized to Person, Organization, Event, Place, and of particular interest are occurrences relating events involving people at particular places. Information sources can be maps, images, reports video, audio, email, web sites and database records. Typically, an item of interest would have many information sources describing aspects of that item, for example a meeting held be members of a suspected terrorist organization might have a news article, reports by insiders, audio, video and email surveillance.

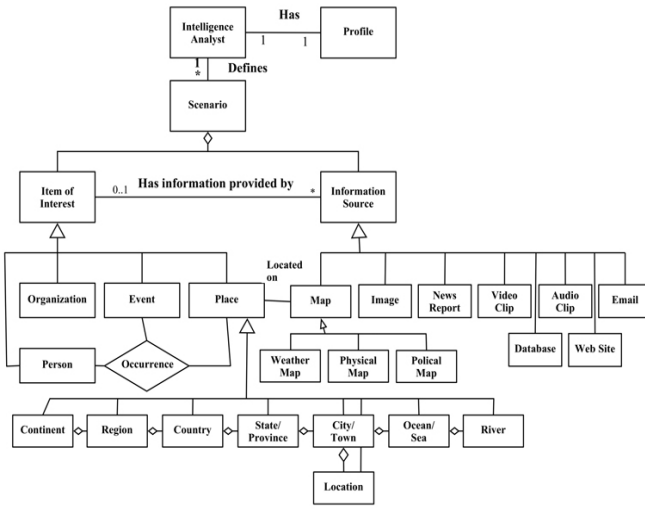

Figure 10. Domain Ontology for People, Places, Events and the Information Sources providing supporting evidence.

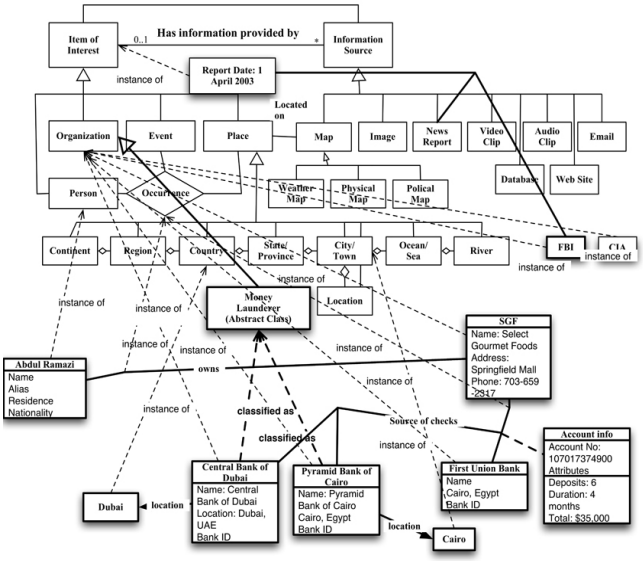

Figure 11. FBI Report Classified According to Domain Ontology

To illustrate how the domain ontology can be used for message classification, we show in Figure 11 how Report 1 dated April 1, 2003, whose full text is given is section 4.1, would be modeled.

The square boxes represent instances of classes together with their attributes. Dotted lines denote the 'instance of' relationship. Note that 'Report Date: 1 April 2003' is an instance of Item of Interest and relates the FBI, an instance of Organization, originated the report which is of type News Report.

The other reports can be similarly classified according to the domain ontology. In order for this classification to be done automatically, or semi-automatically, these reports must be tagged appropriately, preferably using an XML-based markup language. The tagged reports then become part of the knowledge base shown in Figure 1.

The domain ontology can be used to model the domain of discourse, which helps to focus on those concepts and relationships of interest. Further, the domain ontology and instances can be represented in XML, thereby allowing the metadata and data to be shared among OmniSeer subsystems.

Lastly, the concepts of the domain ontology can be mapped to other representations such as Bayesian Net Fragments. In fact the domain ontology allows indexing the messages, and fragments, according to various facets, by person, by place, by event, etc.

5. Future OmniSeer Research

Scalability Issues for BN Fragments. Success in discovering imminent terrorist activities depends on the ability to recognize patterns of suspicious and dangerous behavior by terrorists. Within OmniSeer, such patterns are represented by BN fragments,which must cover various actions and events, including both common and unusual activities. Both the ontology and the repository of BN fragments need to expand to useful initial sizes. When OmniSeer enters routine use, its repository will grow as analysts provide it with their tacit knowledge.

Scalability Issues for User Context and Preference Network in User Modeling. As the analyst interacts with the system, the user model keeps on learning from the analytic process. We will focus on the study that tries to identify an active set of the expert's knowledge that is relevant to current user's interest. The active set is defined as the minimum knowledge that is necessary for the reasoning purpose.

Study on Knowledge Learning and User Model Adaptation. It is important to evaluate how well the user model adapts based on the analyst's searching process. We will test the learning accuracy and convergence of user context modeling, by going through an information scenario, hand building document graph for each selected document, and comparing context network built by system versus target context network.

OmniSeer Agent-Based Architecture. We intend to explore the role of proactive agents in the context of the cognitive framework of agencies, and to investigate those agents and services that complement the needs of the User Model and Prior and Tacit Knowledge subsystems. We will deploy the components of OmniSeer's subsystems as agents and services on the CoABS Grid.

6. Evaluation

A formative evaluation of OmniSeer was conducted at NIST in early May, 2004. The evaluators (three naval reservists with a background in intelligence analysis)

tested the system for eight hours, with four hours allocated to each of the hypotheses generation (OmniSeer PTK subsystem) and user modeling (OmniSeer UME subsystem) parts. In the hypothesis generation test, the analysts were presented several pieces of information (similar to the messages of section 4.1) and asked to generate hypotheses. After they had finished, they were shown hypotheses generated by OmniSeer (using the interface shown in Figure 7) and were asked to rate these hypotheses (using an interface similar to one in Figure 12) in comparison to the ones they had generated. The NIST summary of this part of the evaluation follows. "In general, the analysts felt that they generated more hypotheses than OmniSeer. The analysts noted a number of things that were missing in the hypotheses proposed by OmniSeer as well as a number of hypotheses proposed that did not take into account all the possible variables. They also felt that often OmniSeer did not pick the most important information. However, analysts' ratings for OmniSeer generated hypotheses are equal to the ratings for the Analyst generated hypotheses in 1/3 of the cases. In 7/9 cases the ratings for the OmniSeer generated hypotheses were given mid-level ratings or higher."

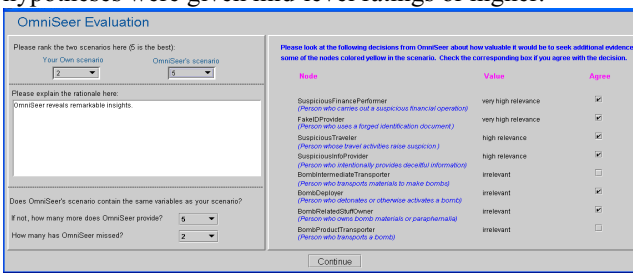

Figure 12. GUI for Evaluation of OmniSeer's Prior/Tacit Knowledge subsystem

We need to improve the scenario-construction subsystem of OmniSeer to support analysts better. The evaluation report includes detailed comments from the analysts that we plan to address in a later release.

The User Modeling subsystem has been successfully applied to different types of data collections, such as Case Study 4 and the CRANFIELD collection. Case Study 4 is a collection of messages and reports; the CRANFIELD collection is the oldest and most widely used test bed in the information retrieval community, and contains 1398 papers on aerodynamics in addition to 225 queries with document relevancy assessments.

The CRANFIELD collection has been used to evaluate the performance of the User Modeling subsystem. The results show that the User Modeling helps to improve the retrieval results, and produces results competitive with the best traditional information retrieval approach Ide dec-hi [4, 6] without exploiting the power of negative feedback in the current version [5]

The OmniSeer user modeling subsystem was tested using analyst information collected in a glass-box environment. In the glass-box evaluation for this subsystem, our user modeling approach is compared against other keyword matching algorithms in a standard information retrieval process. The CNS data collection (distributed on September 2003) was chosen as the experimental dataset, and consists of thousands of documents relevant to the topics of "WMD Terrorism", "WMD Country Profiles" and "China WMD and Arms Control".

An interface (Figure 13) was developed for this evaluation, whose intent was to reduce the possible bias caused by different user interfaces.

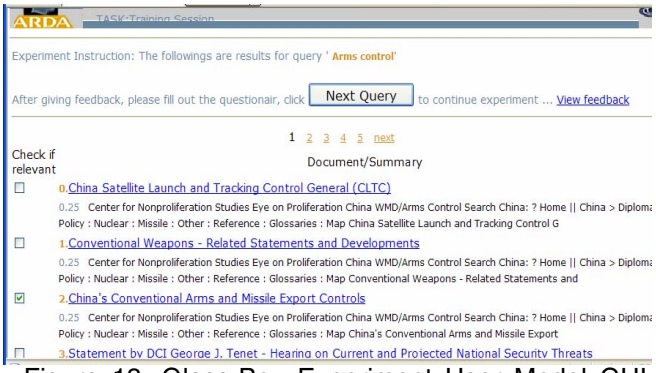

Figure 13. Glass-Box Experiment User Model GUI for Information Retrieval (cropped for readability)

The analysts were presented a set of queries and asked to retrieve information addressing the queries both through OmniSeer and, for comparison, through a baseline commercial information retrieval system by Verity, Inc. (www.verity.com). The analysts then gave feedback to the evaluators, in the form of relevance ratings, using the interface described in Figure 13. NIST summarized this part of the evaluation as follows: "The analysts rated 44% more documents relevant with the OmniSeer system than for the Verity system. The OmniSeer system also returned more unique documents than Verity." We emphasize the encouraging finding, detailed in the table below, that 24 documents retrieved by OmniSeer were ranked as relevant by only one analyst as opposed to eight for Verity, indicating that the UME subsystems worked well in modeling the individual analysts. NIST assessed the user modeling part subsystem of OmniSeer as successful, noting that the individualized support afforded by the active user interface is useful.

Relevance of documents	OmniSeer	Verity
Documents marked relevant by all three analysts	8	3
By two or more	15	19
By only one	24	8

7. Conclusions

We have achieved several significant technical capabilities to date: 1) We are able to capture an analyst's

prior knowledge in the form of Bayesian network fragments, where the nodes in the fragments have attributes; 2) We can process RDF-annotated messages by matching them with Bayesian network fragments; 3) We can compose the matched fragments into *situation-specific scenarios* that represent terrorist activities; and 4) We can analyze situation-specific scenarios by calculating *value of information* and measuring *surprise* and *sensitivity to parametric assumptions*.

The scientific knowledge gained to date includes: 1) To support Bayesian reasoning, the ontology must contain causal relationships among events and activities, rather than the subclass and part-of relationships in conventional ontologies; 2) It is important to distinguish between defining attributes and nonessential attributes when analyzing message traffic and other intelligence information; and 3) Additional research is needed to combine first-order logic with Bayesian reasoning that goes beyond the existing work on object-oriented Bayesian networks.

To evaluate the above technical and scientific accomplishments we have applied our technology and methodology to a substantial, realistic case study, and we have captured the concepts in a decision scenario for monitoring terrorist activity.

In the near term, we expect to achieve the following capabilities and knowledge: 1)We will demonstrate the capabilities of the User Model in capturing and representing user interests, preferences and contexts, and use these to focus the specification and execution of queries against reasonably large collections; 2) We will be able to capture and exploit analysts' tacit knowledge, as well as their prior knowledge; 3) Analysts will be able to analyze situation-specific scenarios in terms of sensitivity to assumptions and the validity of evidence; and 4) Agents will facilitate the interchange of knowledge nuggets among OmniSeer agencies so as to provide tight-coupling among system components.

OmniSeer has recently been the object of an evaluation carried out at the National Institute of Standards and Technology, whose results will guide future development.

8. Acknowledgments

This work was supported in part by the Advanced Research and Development Activity (ARDA), an entity of the U.S. Government. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. government. We thank in a special way Jean Scholtz and Emile Morse of NIST for conducting the evaluation of OmniSeer.

9. References

[1] L. Kerschberg, "The Role of Intelligent Agents in Advanced Information Systems," in *Advanced in Databases*, vol. 1271, *Lecture Notes in Computer Science*, C. Small, P. Douglas, R. Johnson, P. King, and N. Martin, Eds. London: Springer-Verlag, 1997, pp. 1-22.

[2] L. Kerschberg, "Knowledge Management in Heterogeneous Data Warehouse Environments," presented at International Conference on Data Warehousing and Knowledge Discovery, Munich, Germany, 2001.

[3] A. Brodsky, L. Kerschberg, and S. Varas, "Resource Management in Agent-based Distributed Environments," in *Cooperative Information Agents III*, vol. 1652, *Lecture Notes in Artificial Intelligence*, M. Klusch, O. Shehory, and G. Weiss, Eds. Berlin, et al.: Springer-Verlag, 1999, pp. 50-74.

[4] Cleverdon C. 1967. The Cranfield test of index language devices. *Reprinted in Reading in Information Retrieval Eds.* 1998. Pages 47-59.

[5] N. Hien, E. Santos, Jr., Q. Zhao and H. Wang "Capturing User Intent for Information Retrieval" to be appear in *Proceedings of the 48th Annual Meeting for the Human Factors and Ergonomincs Society, Sept, 2004*

[6] Salton G. and Buckley C. 1990. Improving Retrieval Performance by Relevance Feedback. In *Journal of the American Society for Information Science*. Vol 41(4), 288-297.

[7] E. Santos, Jr., S. M. Brown, M. Lejter, G. Ngai, S. B. Banks, and M. R. Stytz, "Dynamic User Model Construction with Bayesian Networks for Intelligent Information Queries," presented at Proceedings of the 12th International FLAIRS Conference, Orlando, FL, 1999.

[8] E. Santos, Jr., H. Nguyen, and S. M. Brown, "Kavanah: An Active User Interface Information Retrieval Application," presented at 2nd Asia-Pacific Conference on Intelligent Agent Technology, 2001.

[9] E. Santos, Jr., H. Nguyen, Q. Zhao, and H. Wang, "User Modelling for Intent Prediction in Information Analysis," presented at 47th Annual Meeting for the Human Factors and Ergonomics Society (HFES-03), Denver, CO, 2003.

[10] E. Santos, Jr., H. Nguyen, Q. Zhao, and E. Pukinskis, "Empirical Evaluation of Adaptive User Modeling in a Medical Information Retrieval Application," presented at User Modeling 2003, Johnstown, PA, 2003.

[11] Y.-G. Kim and M. Valtorta, "On the Detection of Conflicts in Diagnostic Bayesian Networks Using Abstraction, " in P. Besnard and S. Hanks (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Eleventh Conference*, San Francisco, CA: Morgan-Kaufmann, pp. 362-367, 1995.

[12] M. Singh and M. Valtorta, "Construction of Bayesian Belief Networks from Data: A Brief Survey and an efficient algorithm," *International Journal of Approximate Reasoning*, vol. 12, pp. 111-131, 1995.

[13] K. Laskey and S. Mahoney, "Network Engineering for Agile Belief Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, pp. 487-498, 2000.

[14] J. Yoon, V. Raghavan, V. Chakilam, and L. Kerschberg, "BitCube: A Three-Dimensional Bitmap Indexing for XML Documents," *Journal of Intelligent Information Systems*, vol. 17, pp. 241-254, 2001.