Introspective Self-Explanations for Report Generation in Intelligence Analysis

Ashok K. Goel[1], Emile Morse[2], Anita Raja[3], Jean Scholtz[4] & John Stasko[1]

[1]School of Interactive Computing, Georgia Institute of Technology
[2]Visualization and Usability Group, Information Access Division, National Institute of Standards and Technology
[3]Department of Software and Information Systems, University of North Carolina at Charlotte
[4] National Visualization and Analytics Center, Pacific Northwest National Laboratory

Abstract. Report generation is an integral part of many analytical tasks such as intelligence analysis. Thus, an important issue in developing cognitive assistants for analytical tasks is how the cognitive assistant may help a human analyst in generating reports. In this paper, we first describe the task of report generation in intelligence analysis. Then, we describe a scheme for enabling a cognitive assistant to generate self-explanations. The proposed scheme uses introspection over the knowledge, reasoning, and conclusions of the cognitive assistant. Finally, we analyze the introspective scheme for generating self-explanations from the perspective of report generation.

1. Introduction

Since intelligence analysis is in part an information-processing activity, information technology is playing an increasingly important role in many tasks of intelligence analysis. In particular, research in human-centered computing is developing cognitive assistants for a variety of high-level tasks of intelligence analysis such as sensemaking and dataforaging, e.g., Jigsaw [31], a tool for visualizing information about relations among entities such as people, organizations, places and times, ACH [24], a tool for analyzing competing hypotheses based on a set of evidence, AHEAD [23] that uses past cases to build evidence for and against a hypothesis in a new intelligence situation, Disciple-LTA [33] that helps analysts build structured arguments, and RESIN [19,36] that forages data from multiple multimedia data sources.

Report generation is another important high-level task in intelligence analysis [18, 34]. However, insofar as we know, report generation so far appears to have received little attention in the development of cognitive assistants for intelligence analysis. This raises two sets of questions for our work. Firstly, what are the purposes, contents and structure of intelligence reports? As one may expect, a report may contain findings, assessments, and possibly recommendations, as well as explanations and justifications of the assessments and the conclusions. Thus, a second set of questions pertains to how a cognitive assistant may explain and justify its conclusions. In particular, what is the relationship between current introspective schemes for generating explanations of the knowledge, reasoning and conclusions of a cognitive agent and the information requirements of report generation?

2. Threat Assessment in Intelligence Analysis

The literature on intelligence analysis includes both cognitive accounts of intelligence analysis (e.g., [14, 25]) and analyses of information-processing techniques used in intelligence analysis (e.g., [18, 34]). We know that sensemaking in intelligence analysis involves the tasks of recognizing and characterizing a problem based on some initial evidence about an event or activity; generating multiple explanatory hypotheses based on the evidence; collecting and assimilating additional data; evaluating the multiple explanatory hypotheses; and selecting the most plausible hypothesis. This analytical task is complex because of the constantly evolving, and often unreliable and conflicting nature, of the data [28]. The evolving nature of data implies a need for ongoing monitoring and continual generation and evaluation of hypotheses so that new evidence can be accounted for as it arrives and the most likely explanation can be produced at any time.

Table 1 shows a taxonomy of intelligence tasks based on Jones [15]. According to Jones, simplistic tasks seek to find a fact; deterministic tasks seek a single answer but the answer is the result of a calculation; random tasks look for answers where there are multiple answers but all those possibilities can be identified; and indeterminate tasks are those for which there are multiple possible answers but not all possible answers can be identified. Table 1 suggests that random and indeteminate tasks, such as threat assessment, are the ones that are most likely to benefit from the technological support that automated agents promise to provide. Agents that help analysts avoid the common biases will most certainly be perceived to be valued cognitive assistants when tasks are hard, the number of constraints is high, and the limits of cognitive resources are exceeded.

3. Intelligence Reports

The purposes, contents, structure and appearance of intelligence reports are diverse [18]. The desired content and format are specified by the customer and communicated during the actual tasking to the analyst. Reports may be hard and/or soft copy, oral and/or written, and even video. They may be formal or informal, and textual or graphical. Whether an analyst is tasked to find a fact (i.e. simplistic task in Table 1), or to produce a one-paragraph inclusion in a Presidential Daily Brief, or to contribute to a community-wide National Intelligence Estimate, the underlying analysis is expected to adhere to rigorous standards.

Figure 1 illustrates a template that has been extensively used for intelligence reports in a variety of studies [17,29] with all-source analysis. In the Discussion section in this template analysts are required to indicate gaps in the material available for their reports; there are mechanisms within each agency for initiating a search for material that might resolve the gap. Explicit acknowledgment of a gap is helpful to the consumer in interpreting the report. The appendix materials in this template encourage the analyst to use structured methods for tracking the hypotheses, for evaluating the evidence for and against the hypotheses, and externalizing assumptions.

Table 1: Taxonomy of Intelligence Analysis Tasks

	Simplistic	**Deterministic**	**Moderately Random**	**Severely Random**	**Indeterminate**
Role of facts	Highest	High	Moderate	Low	Lowest
Role of judgment	Lowest	Low	Moderate	High	Highest
Analytical task	Find information	Find/create formula	Generate all outcomes	Define potential outcomes	Define futures factors
Analytical method	Search sources	Match data to formula	Decision theory; utility analysis	Role playing and gaming	Analyze models and scenarios
Analytical instrument	Matching	Mathematical formula	Influence diagram, utility, probability	Subjective evaluation of outcomes	Use of experts
Analytic output	Fact	Specific value or number	Weighted alternative outcomes	Plausible outcomes	Elaboration on expected future
Probability of error	Lowest	Very low	Dependent on data quality	High to very high	Highest
Follow-up task	None	None	Monitor for change	Repeated testing to determine true state	Exhaustive learning

Task Description:
Outline:
- Executive Summary:
- Discussion (include: 1. Intelligence gaps and 2. Potential for Denial and Deception)
- Assessment
  - End Notes/Citations

Appendix:
- Hypotheses Investigated/ Key questions
- Key Assumptions:
- Key pieces of evidence

Confidence in assessment on a scale of 1-10

1	2	3	4	5	6	7	8	9	10
not at all confident									very confident

Figure 1: A Template for Intelligent Reports

Until the United States Office of the Director of National Intelligence (ODNI) assumed overall control of all 16 intelligence agencies, each agency employed its own methods for assessing the quality of the reports generated by its analysts. One of the first efforts of the ODNI was to establish community-wide criteria for judging analytic quality; the guidelines were distributed as Intelligence Community (IC) Directive 203, effective June 21, 2007. These criteria are intended to be used by all agencies and for all activities including analytic production, analysis teaching modules and case studies used throughout the IC. The IC Analytic Standards are:

A. Objectivity – requires analytic and informational functions be done from an unbiased perspective.
B. Independent of Political Considerations – provide assessments that are not distorted or altered to support a particular policy, political viewpoint, or audience.
C. Timeliness – reports are to be produced in time to allow customers adequate time to factor them into their required decision-making.
D. Based on All Available Sources of Intelligence – include all relevant sources; identify critical gaps and work with collectors to remedy those gaps.
E. Exhibits Proper Standards of Analytic Tradecraft
   1. Properly describes quality and reliability of sources
   2. Properly caveats and expresses uncertainties or confidence in analytic judgments
   3. Properly distinguishes between underlying intelligence and analysts' assumptions and judgments
   4. Incorporates alternative analysis where appropriate
   5. Indicates relevance to US national security
   6. Uses logical argumentation
   7. Points to consistency with earlier reports or highlights change
   8. Makes accurate judgments and assessments

Since the promulgation of IC Directive 203, agencies have begun to regularize their quality standards. This process is likely to take some time as each agency will want to ensure that its prior assessment methods are captured by implementation of the new criteria.

4. A Computational Assistant for Threat Assessment

As part of a project sponsored by the United States Department of Homeland Security under the auspices of the Pacific Northwest National Laboratory, we are presently developing component technologies for an interactive cognitive assistant for threat assessment. The components include an automated sense-making agent called STAB [1, 2], an automated data-foraging agent called RESIN [19,36], and a visual assistant for human analysts called Jigsaw [31]. Here, due to limitations of space, we briefly describe only STAB, but it is important to note that STAB is intended as one tool in a suite.

STAB views the task of threat perception as that of abducing a story such that the intentional and causal relationships in the story explain the current data and make verifiable predictions about both the future and the past. The generic stories are organized as hierarchical scripts and represented in the Task Method Knowledge knowledge representation language (TMKL) [20,21,22]. Thus, each script is organized in a task-method-subtask hierarchy, where the primitive tasks at the leaf node correspond to primitive actions. Methods in TMKL are state transition machines that recursively compose primitive tasks into compound tasks, where the primitive tasks correspond either to a primitive action in the world, to a chunk of knowledge available in memory, or an executable procedure.

STAB operates in the domain of Visual Analysts Science and Technology (VAST) datasets [26,35]. The VAST datasets have been generated for the VAST symposium contests by the Threat Stream Generator project at the Pacific Northwest National Laboratory. Each dataset contains over a thousand news stories and a small number of other data such tables, maps and photographs. We manually extract the facts and events in the dataset into datastreams that are input into STAB.

As events in the input data stream arrive incrementally, STAB matches the events with the leaf task nodes in the scripts. This matching is done using feature vectors. Figures 2 illustrates the script Robbing a Store, whose task nodes match the input event Break(Window). The matching task nodes are shown in yellow. STAB uses the hypotheses to generate expectations about additional data not yet seen by STAB.

Commit Vandalism (not shown here) is another script containing the matching task node of Break(Window). While the hypothesis of Robbing a Store generates the expectation of Take(Money) and Commit Vandalism generates the expectation of Stole(Null). As additional data in the form of event Take(Money) arrives as input to STAB, the system matches the data with the expectations generated by the candidate hypotheses. This may lead to abandonment of some hypotheses. For example, in the current scenario, the event Take(Money) results in the refutation of the Committing

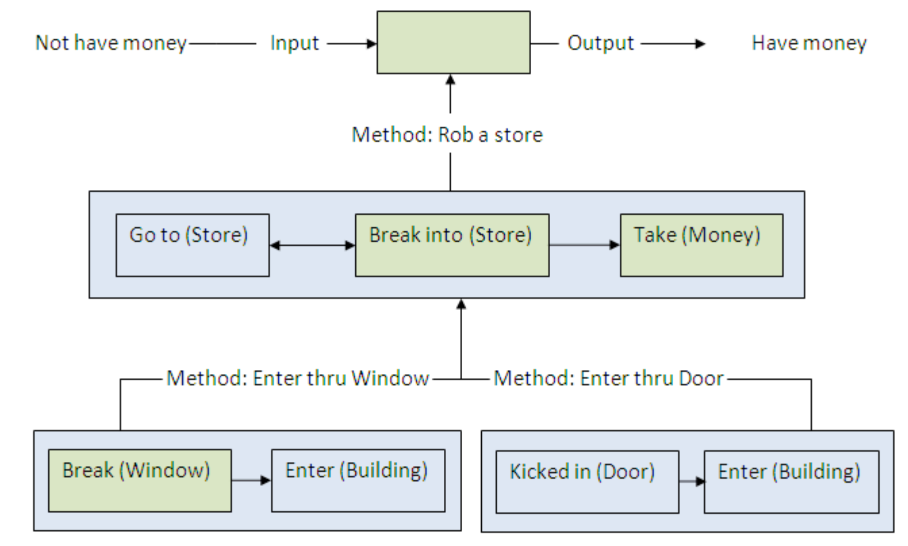

Figure 2: A Hierarchical Script from STAB

Vandalism hypothesis (shown in red/darker shade), and the acceptance of the Robbing a Store hypothesis. The confidence value of the hypothesis Robbing a Store measures how many of the task nodes in a task-method tree of the hypothesis were matched by the input data. STAB keeps track of all competing hypotheses, the evidence for and against each hypothesis as well as the confidence value for each hypothesis.

5. Introspective Self-Explanation in the Sensemaking Agent

In general, explanations can be of two types [16]: abductive explanations and self-explanations. Abductive hypotheses provide explanations for a set of evidence, and typically have associated confidence values. In threat assessment, for example, an abductive hypothesis may explain how a sequence of apparently unrelated events forms a pattern of activity. An agent's self-explanation describes the agent's reasoning in reaching a conclusion. A self-explanation in general may have three components [7]: justification of the conclusions, explanation of the reasoning process, and justification of the domain knowledge. In threat perception, for example, a conclusion about a specific pattern of activity may be justified by the evidence supporting it and the evidence that does not support other hypotheses, the reasoning process may be explained in terms of the steps of the process including collection of evidence and analysis of competing hypotheses, and the domain knowledge may be justified in terms of past cases of threat perception.

In earlier work on introspective self-explanations [13], we posited that the generation of useful self-explanations in automated agents requires introspection by the agent over its own reasoning, knowledge and conclusions. Introspection in turn requires a representation of the tasks the agent addresses, the methods it uses to address them, and the knowledge used and created by the methods. Thus, we postulate that introspection over the task structure of the analytical process may enable self-explanations in automated agents. Task structure refers to the recursive task-method decomposition of an analytical task into primitive tasks. The task structure also relates the methods with domain knowledge. Introspection pertains to the ability to capture the invocation of tasks and methods, and the use of knowledge by them. Since the task structure is a recursive decomposition, introspection over the task structure may enable generation of explanations at multiple levels of abstraction. An opposing hypothesis might be that the self-explanation of an agent can be in the form of the trail of decisions made by the agent (as, for example, in audit trails). If our postulate is correct, then we should find that introspection over the task structure of threat perception provides the right level of abstraction for the explanations to be useful for human analysts.

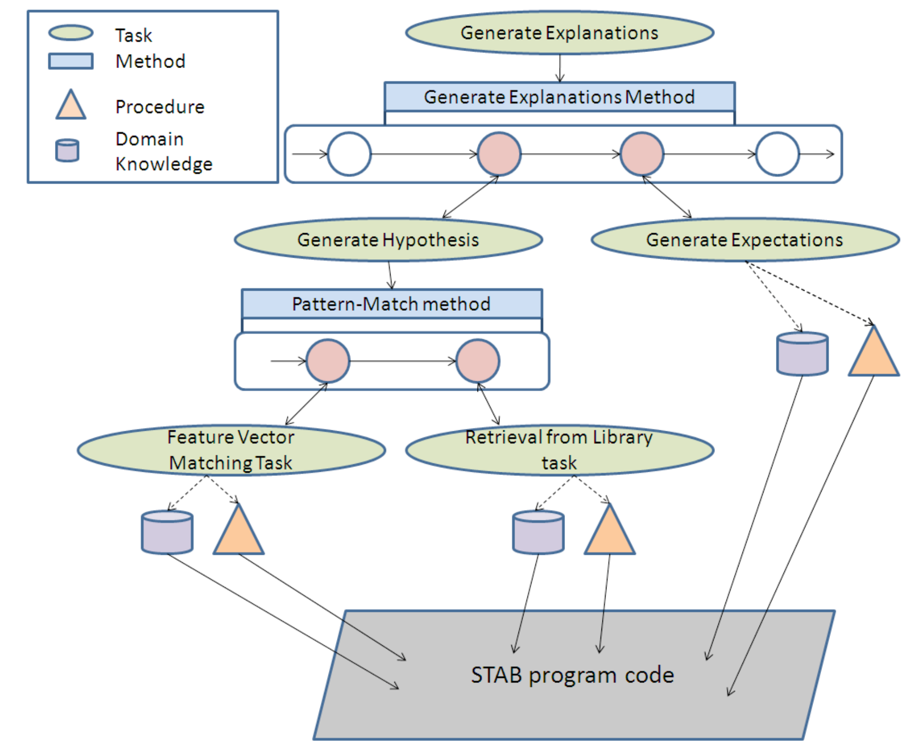

Figure 3: Meta-STAB's Task Structure of STAB

The origin of this postulate can be traced to early work on knowledge systems. Chandrsekaran, Tanner & Josephson [6] proposed that the task-level was the right level for generating computational explanations. NEOMYCIN [9], a reimplementation of MYCIN [3] in terms of the task structure of diagnosis, provided explanations at a more useful level of abstraction for human comprehension. XPLAIN [32] contained a declarative model of clinical decision making for generating explanations. Research on knowledge systems since then has developed several languages for describing problem-solving tasks and methods, including Generic Tasks [4, 5, 8], CommonKADS [30], Protege [11], TAEMS [10], and UPML [12]. We now know that because problem-solving methods relate knowledge, inference and control, declarative representations of the methods in the task structure of an agent enable deeper and more perspicuous explanations.

We envision a Meta-STAB component based on the Reflective Evolutionary Mind (REM) functional architecture [20, 21, 22]. REM encodes the meta-reasoner in the same TMKL language as the reasoner itself, which provides uniformity of knowledge representation. Figure 3 illustrates Meta-STAB's encoding for a small portion of STAB. The ovals in the figure represent tasks; thus, the highest-level task is Generate Explanations. The rectangles represent methods used by STAB; thus the Pattern-Match Method addresses the task of Generate Hypotheses. This Pattern-Match Method decomposes the Generate Hypotheses task into two simple subtasks: Feature Vector Matching and Retrieval from Library. The transition machine for the Pattern-Match Method depicted in the rounded rectangle represents the control of processing of the subtasks. In general, the control of subtasks need not be linear; similarly, in general, multiple methods may be available for addressing a particular task. The Retrieve from Library Task is directly encoded in STAB; a primitive, or directly encoded, task may use both some domain knowledge, K, and some procedure, P. In this way, the TMKL model of STAB in Meta-STAB explicitly captures the entire task structure of STAB including the relationship between tasks, control, inference and domain knowledge.

When input data arrives, Meta-STAB executes its task structure, dynamically selecting and invoking tasks and methods up to the level of primitive tasks. This selection and invocation of tasks and methods depends on the knowledge conditions generated by the preceding tasks. The primitive tasks in turn execute the corresponding code in STAB. Meta-STAB keeps track of the trace of processing in the vocabulary of tasks and methods.

6. Discussion

How well do the explanatory capabilities of STAB and Meta-STAB match the needs of report generation in intelligence that we described earlier in Section 3? The template in Figure 1 refers to both the contents and the format of an intelligence report. Clearly, explanatory capabilities of automated agents such as STAB and Meta-STAB are confined to the issue of content. However, contents of explanations generated by automated agents appear closely aligned with the criteria for a report. In particular, generation of a report requires explanations of several different kinds: (1) Justification of the conclusions, (2) Confidence values for the conclusions, (3) Explanation of reasoning, (4) References to alternative hypotheses, and (5) References to data sources including the reliability of those sources.

STAB itself generates the justifications for its conclusions in the form of evidence in support of each conclusion. Indeed, STAB also collects and reports the evidence against each of its conclusion, which helps capture some of the uncertainty in the conclusions. STAB also generates confidence values for each of its conclusions based on the degree of match between the input events and the tasks in the task structure of the story plots.

Meta-STAB generates explanations of the analytical process in the task-method language. The self-explanation references the alternative hypotheses considered and the reasons for rejecting them. In addition, since TMKL directly captures the relationship between tasks and knowledge, Meta-STAB references the data sources and justifies its decisions by relating them to its domain knowledge and data sources.

Further, the explanations generated by Meta-STAB are both causal and intentional. The explanations are causal because the execution of a task in the task structure sets up the knowledge conditions for the selection, invocation and execution of the succeeding task. Thus, the execution of various tasks is linked by the knowledge states they take as inputs and give as outputs. The explanations are intentional because the execution of a task (except the dummy task at the root of the task structure) takes place in the context of some higher-level task. Thus, this scheme can help answer not only the question of what STAB is doing at any given state of processing (the task), but also how (the method), and why (the higher-level task). Further, since the analytical process can be explained to different levels of depth in the task structure of STAB, the level of abstraction of the explanations is tunable to different goals and queries of the human analyst.

Thus, we posit that the self-explanations of the automated agents may provide a basis for interactive generation of intelligence reports. This is because of the similarity between the requirements for the self-explanations generated by automated agents and the requirements for intelligence reports generated by human analysts. If this postulate is correct, then we should find that while the structure of self-explanations generated by the agents and the intelligence reports generated by an analyst may be different, their contents will be similar, and, in a mixed initiative system, the agents' self-explanations may directly provide much of a report's contents.

Acknowledgements. Our work on the STAB, RESIN and Jigsaw projects is sponsored by the National Visual Analytics Center (NVAC) under the auspices of SouthEast Regional Visualization and Analytics Center (http://srvac.cc.gatech.edu/). NVAC is sponsored by the U.S. Department of Homeland Security and located at the Pacific Northwest National Laboratories.

References

1. Adams, S. and Goel, A. (2007a): STAB: Making Sense of VAST Data. In Proc. IEEE Conference on Intelligence and Security Informatics. May 2007.
2. Adams, S. and Goel, A. (2007b): A STAB at Making Sense of VAST Data. In Proc. AAAI-07 Workshop on Plan, Intent and Activity Recognition. Vancouver, Canada. July 2007.

3. Buchanan, B., & Shortliffe, E. (1984): Rule-based expert systems : the MYCIN experiments of the Stanford Heuristic Programming Project. Addison-Wesley, Reading, MA.
4. Chandrasekaran, B. (1986): Generic tasks in knowledge-based reasoning: High-level building blocks for expert systems design. IEEE Expert. 1, 3, 23-30, 1986.
5. Chandrasekaran, B. (1988): Generic Tasks as Building Blocks for Knowledge-Based Systems: The Diagnosis and Routine Design Examples. Knowledge Engineering Review. 3 (3).
6. Chandrasekaran, B., Tanner, M., & Josephson, J. (1989) Explaining Control Strategies in Problem Solving. IEEE Expert. 4:1, pp. 9-24, Spring 1989.
7. Chandrasekaran, B. & Swartout, W. (1991): Explanations in Knowledge Systems: The Role of Explicit Representation of Design Knowledge. IEEE Expert. Vol. 6, No. 3, June 1991, pp. 47-49.
8. Chandrasekaran, B., Johnson, T., & Smith, J.W. (1992): Task Structure Analysis for Knowledge Modeling. Communications of the ACM. Vol. 33, No. 9, September 1992, pp. 124-136.
9. Clancey, W. (1983): The Epistemology of a Rule-Based Expert System - A Framework for Explanation. Artificial Intelligence 20(3): 215-251.
10. Decker, K. (1996): 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, January 1996.
11. Eriksson, H., Shahar, Y., Tu, S., Puerta, A., & Musen. M. (1995): Task Modeling with Reusable Problem-Solving Methods. Artificial Intelligence. 79(2):293-326, 1995.
12. Fensel, D. et al. (2003): The Unified Problem-Solving Method Development Language UPML. Knowledge and Information Systems, An International Journal. 5, 83-127, 2003.
13. Goel, A., & Murdock, J.W. (1996): Meta-Cases: Explaining Case-Based Reasoning. In: Lecture Notes in Artificial Intelligence 1168: Advances in Case-Based Reasoning. I. Smith & B. Faltings (eds.), Berlin: Springer-Verlag, 1996.
14. Heuer, R. (1999): Psychology of Intelligence Analysis. CIA Center for the Study of Intelligence, 1999. [https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/psychology-of-intelligence-analysis/PsychofIntelNew.pdf]
15. Jones, M.D. (1995): The Thinker's Toolkit, Three Rivers Press, 1995.
16. Josephson, J., & Josephson, S. (1994): Abductive Inference: Computation, Philosophy, Technology. Cambridge University Press, 1994.
17. Kelly, D., Kantor, P., Morse, E., Scholtz, J., & Sun, Y. (2007): Questionnaires for Eliciting Evaluation Data from Users of Interactive Question Answering Systems. Special issue of the Journal of Natural Language Engineering on Interactive Question Answering Systems.
18. Krizan, L. (1999): Intelligence Essentials for Everyone, Joint Military Intelligence College, 1999.
19. Liu, D., Raja, A., & Vaidyanath, J. (2007): TIBOR: A Resource-bounded Information Foraging Agent for Visual Analytics. In: Proceedings 2007 IEEE/ WIC/ ACM International Conference on Intelligent Agent Technology (IAT 2007), Silicon Valley, CA, Nov 2-5, 2007.
20. Murdock, J.W., & Goel, A. (2001a): Meta-case-Based Reasoning: Using Functional Models to Adapt Case-Based Agents. In: Proceedings ICCBR 2001: 407-421.
21. Murdock, J.W. & Goel, A. (2001b): Learning about Constraints by Reflection. In: Lecture Notes in AI - 2006, Springer, pp. 131-140.
22. Murdock, J.W. & Goel, A. (2003): Localizing Planning using Functional Process Models.

In: Proceedings International Conference on Automated Planning and Scheduling (ICAPS-03), June 2003.

23. Murdock, J.W., Aha, D., & Breslow, L. (2003): Assessing Elaborated Hypotheses: An Interpretive Case-Based Reasoning Approach. In: Proceedings ICCBR 2003: 332-346.
24. PARC AI Team & Heuer, R. (2004): ACH: A Tool for Analyzing Competing Hypotheses. PARC Technical Report, October 2004.
25. Pirolli, P. & Card, S. (2005): The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In: Proceedings 2005 International Conference on Intelligence Analysis. pp. 2-4, May 2005.
26. Plaisant, C. et al. (2008): Evaluating Visual Analytics at the 2007 VAST Symposium Contest. IEEE Computer Graphics and Applications. Vol. 28, No. 2, March/April 2008, pp. 12-21.
27. Raja, A., & Goel, A. (2007): Introspective Self-Explanation in Analytical Agents. In: Proceedings of AAMAS 2007 Workshop on Metareasoning in Agent-based Systems. pp 76-91, Hawaii, May 2007.
28. Russell, D., Stefik, M., Pirolli, P., & Card, S. (1993): The cost structure of sensemaking. Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems, Amsterdam. pp. 269 – 276.
29. Scholtz, J., Morse, E., & Hewett, T. (2005): An Analysis of Qualitative and Quantitative Data from Professional Intelligence Analysts. In: Proceedings 2005 International Conference on Intelligence Analysis. May 2-6, McLean VA.
30. Schreiber, G., Akkermans, H., Anjewierden, A., De Hoog, R., Shadbolt, N., Van De Velde, W. & Wielinga, B. (2000): Knowledge Engineering and Management: The CommonKADS Methodology, MIT Press.
31. Stasko, J., Gorg, C., & Liu, Z. (2008): Jigsaw: Supporting Investigative Analysis through Interactive Visualization. Information Visualization. Vol. 7, No. 2, Summer 2008, pp. 118-132.
32. Swartout, W. (1983): XPLAIN: A system for Creating and Explaining Expert Consulting Programs. Artificial Intelligence. 21(3), pp.285-325, 1983.
33. Tecuci, G., Boicu, M., Marcu, D., Boicu, C., & Barbulescu, M. (2008): Disciple-LTA: Learning, Tutoring and Analytic Assistance. Journal of Intelligence Community Research and Development (JICRD). July 2008.
34. Thomas, J., & Cook, K. (2005): Illuminating the Path.: The R& D Agenda for Visual Analytics. IEEE Press.
35. VAST Contest Datasets. http://www.cs.umd.edu/hcil/VASTcontest07/
36. Yue, J., Raja, A., Liu, D., Wang, X., & Ribarsky, W. (2009): A Blackboard-based Approach towards Predictive Analytics. In: Proceedings AAAI Spring Symposium on Technosocial Predictive Analytics. pp 154-161, Stanford University, CA, March 23-25, 2009.