Published in ICCBR'03 (5th International Conference on Case-Based Reasoning; Trondheim, Norway; June 23-26, 2003)

Assessing Elaborated Hypotheses: An Interpretive Case-Based Reasoning Approach

J. William Murdock, David W. Aha, & Leonard A. Breslow

Intelligent Decision Aids Group
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory, Code 5515
Washington, DC 20375
lastname@aic.nrl.navy.mil

Abstract. Identifying potential terrorist threats is a crucial task, especially in our post 9/11 world. This task is performed by intelligence analysts, who search for threats in the context of an overwhelming amount of data. We describe AHEAD (Analogical Hypothesis Elaborator for Activity Detection), a knowledge-rich post-processor that analyzes automatically-generated hypotheses using an interpretive case-based reasoning methodology to help analysts understand and evaluate the hypotheses. AHEAD first attempts to retrieve a functional model of a process, represented in the Task-Method-Knowledge framework (Stroulia & Goel, 1995; Murdock & Goel, 2001), to identify the context of a given hypothesized activity. If retrieval succeeds, AHEAD then determines how the hypothesis instantiates the process. Finally, AHEAD generates arguments that explain how the evidence justifies and/or contradicts the hypothesis according to this instantiated process. Currently, we have implemented AHEAD's case (i.e., model) retrieval step and its user interface for displaying and browsing arguments in a human-readable form. In this paper, we describe AHEAD and detail its first evaluation. We report positive results including improvements in speed, accuracy, and confidence for users analyzing hypotheses about detected threats.

1. Introduction

Terrorist activities are examples of *asymmetric threats*, which occur when a small, secretive group engages in a conflict with a large, powerful (e.g., military, law enforcement) group. Preventing asymmetric threats requires their *detection*. For example, if a law enforcement group detects an attempt by an organized crime group to take over a commercial industry in some region, the law enforcement group can then attempt to stop the takeover or reverse it. Unfortunately, detection is exceedingly difficult for many asymmetric threat domains because their data sets are both large and complex, involving many types of relationships among entities. Thus, detection can require an enormous amount of time.

The DARPA Evidence Extraction and Link Discovery (EELD) program is trying to speed the detection process and increase its reliability by creating software that automatically discovers potential asymmetric threats. EELD consists of research and development in three primary areas: evidence extraction, link discovery, and pattern learning. *Evidence extraction* tools convert unstructured data (i.e., raw text) into structured data (e.g., semantic networks or databases). *Link discovery* tools match collections of structured data to known patterns of asymmetric threats. Finally, *pattern learning* discovers new patterns of asymmetric threats. EELD is integrating these three areas to perform fast and accurate detection of threats from organized crime, terrorist groups, etc.

This integrated EELD system runs the risk of generating hypotheses of varying credibility (e.g., false positives). Consequently, an additional challenge arises, namely *elaboration*: providing information to help an intelligence analyst determine whether a hypothesized threat is genuine and decide how to respond to it. To address this, we are developing AHEAD (Analogical Hypothesis Elaborator for Activity Detection), the EELD component that performs hypothesis elaboration. AHEAD takes as input a hypothesis from EELD's link discovery components, along with the evidence used to create that hypothesis, and outputs a structured argument for and/or against that hypothesis. These arguments should help a user (e.g., an intelligence analyst) to quickly and confidently decide whether and how to respond to hypothesized asymmetric threats.

We introduced AHEAD in (Murdock *et al*., 2003); it uses an interpretive case-based reasoning process consisting of three steps: *case retrieval*, *solution proposal*, and *solution justification*. These steps are part of the general process for case-based reasoning defined by Kolodner & Leake (1996). Currently, we have implemented only AHEAD's case retrieval step and user interface, which permits an analyst to examine and browse the given hypotheses and the arguments generated by AHEAD. In this paper, we elaborate AHEAD's design and detail its first evaluation. In particular, we test whether its interface can assist the analyst in accurately determining the hypothesized threat's validity, increasing the analyst's confidence in this assessment, and reducing the time required to study the hypothesis before making the assessment. Section 7 describes this experiment, an initial pilot study, and its encouraging results.

2. Motivations and Related Work

In any asymmetric threat domain (e.g., terrorism, organized crime), threats are relatively infrequent and are sufficiently complex that a virtually limitless range of variations exists. Thus, any new threat that arises is unlikely to be an exact or near-exact match to some past instance and is therefore unlikely to be detected or elaborated through using specific concrete cases. Consequently, we are employing *generalized cases* (Bergmann, 2002) to represent asymmetric threats. Specifically, we use functional process models; a single model encodes an abstract representation of a hostile process, such as a takeover of an industry by an organized crime group, and multiple instances of takeovers could match to a single model. Many other systems integrate CBR with other reasoning approaches (e.g., Rissland & Skalak, 1989; Branting, 1991; Goel, Bhatta, & Stroulia, 1997), and some include processes as cases (e.g., Cox 1997; Tautz & Fenstermacher, 2001). AHEAD combines these characteristics in an interpretive process that elaborates hypotheses regarding asymmetric threats.

Sibyl (Eilbert, 2002) is another CBR approach in the EELD program. Sibyl uses CBR for hypothesis generation; it uses generalized cases (to ensure close matches exist for a new input), and its cases closely resemble the evidence in structure and content (to enable fast matching of cases to large bodies of unorganized relational data). AHEAD's cases differ significantly from Sibyl's because they are used for different purposes that impose different demands. Whereas Sibyl searches for threats, AHEAD does not. Instead, it is given a threat hypothesis, which is directly tied to relevant pieces of evidence, and focuses on elaboration of this hypothesis. Thus, AHEAD's cases do not need to be structured for efficient matching to large bodies of evidence. However, they do need to include information not only on *what kinds* of evidence are consistent with a given hypothesized threat, but also on *why* that evidence is consistent with it. Consequently, AHEAD uses *functional* models of processes as cases; such models describe both the actions performed in the process and how those actions contribute to the overall effect.

Although some previous CBR research projects have employed functional process models for explanation, they were not used to generate arguments concerning whether a process is occurring. Instead, functional process models have generally been used to explain a process that the system performed itself (e.g., Goel & Murdock, 1996). AHEAD represents a novel application of model-based CBR to help generate arguments concerning detected activities.

While previous work has studied argumentation in interpretive CBR, that work focused on domains in which detailed models of the processes under examination do not exist (e.g., Aleven & Ashley, 1996) or are best defined in terms of concrete examples (e.g., McLaren & Ashley, 2000). AHEAD employs an innovative structure for the generated arguments, derived from the capabilities provided by functional process models and from the goal of helping an analyst to accept or reject a complex detected hypothesis.

3. Case Representation: TMK Models

Cases in AHEAD are generalizations of concrete event descriptions. For example, instead of describing a single specific industry takeover by a criminal group, a case in AHEAD provides an abstract description of the process by which criminal groups take over industries. AHEAD's representation of processes includes information about how the process is performed and why portions of the process contribute to its overall objective. This representation is known as the TMK (Task-Method-Knowledge) modeling framework (Stroulia & Goel, 1995; Murdock & Goel, 2001). A TMK model is divided into *tasks* (defining what the process is intended to accomplish), *methods* (defining how the process works), and *knowledge* (information that drives the process by providing context).

Figure 1 displays a high-level overview of a sample TMK model that can be used in AHEAD. The rectangles represent tasks, the rounded boxes represent methods, and the oblique parallelograms represent parameters in the knowledge base. Methods include state transition machines that impose ordering constraints on subtasks. Labeled links denote relational information encoded in the tasks and methods. These links connect tasks, methods, parameters, and other links. For example, there is a link labeled *makes* from the *Industry-Takeover* task to the link labeled *controls* from the *Mafiya* parameter to the *Target-Industry* parameter. Those links indicate that an industry takeover produces a state in which the involved mafiya controls the target industry. The bottom of Figure 1 shows ellipses, indicating that those tasks can be further decomposed by additional methods into lower-level tasks.

Because TMK is a functional process modeling language (i.e., it encodes not only the elements of the process but also the purposes that those elements serve in the context of the process as a whole), an argument based on a TMK model can both indicate *which* pieces of evidence are consistent with the given hypothesis and also identify *why* that evidence supports the hypothesis. Consequently, TMK is well suited to addressing AHEAD's knowledge requirements. Models in AHEAD are currently composed manually using domain expertise developed within the EELD program. In future work, we intend to study automatic learning of models from instances and/or interactive support for graphical authoring of models.

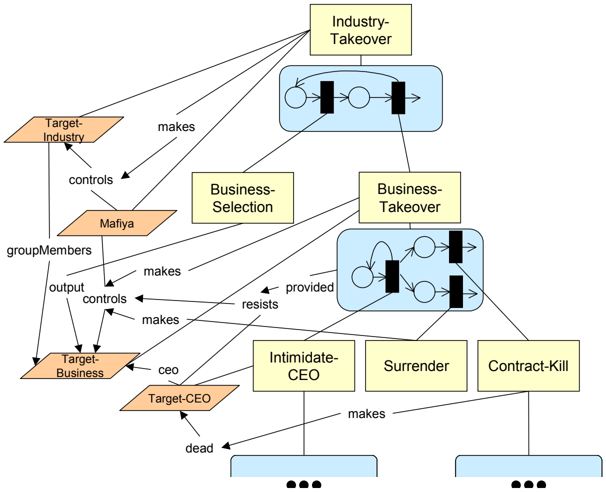


Figure 1: A partial TMK model of an industry takeover

4. Output: Structured Arguments

AHEAD's argumentation structure is inspired by Toulmin (1958), and specifically concentrates on relating facts from the evidence to specific assertions about the process being performed (Murdock *et al*., 2003). For example, an argument involving an industry takeover would step through the various events in that takeover (e.g., selecting business, intimidating CEO's). At the root of the argument is the original *hypothesis* that was provided as input to AHEAD. Associated with that hypothesis are individual, atomic *arguments for* the hypothesis and *arguments against* the hypothesis.

Each argument for the hypothesis is an assertion that some portion of the retrieved model is likely to have occurred. Arguments against the hypothesis assert that some portion of the model has not occurred. Some arguments against the hypothesis include links to evidence indicating that the portion of the model did not occur, while others simply indicate a lack of evidence for the event. Each argument for or against includes a statement of what happens in the model and (when applicable) a statement of the purpose of what happens. For example, an industry takeover model states that (under certain circumstances) an organized crime group will kill a CEO of a business because that CEO resists a takeover of that business. In a hypothesis that said (for example) that a particular CEO was killed as part of an industry takeover, there would be an argument for or against involving the assertion that this CEO was killed because he or she resisted an attempt to control the business. That assertion would be included in an argument for the hypothesis if the evidence supported the claim (e.g., if there was a police report saying that a member of that crime group killed that CEO). It would be included in an argument against the hypothesis if there were no supporting evidence or there were evidence contradicting the claim.

5. An Interpretive Case-Based Reasoning Methodology

The AHEAD methodology partially implements *interpretive* CBR (Kolodner & Leake, 1996). Interpretive CBR differs from problem-solving CBR in that it analyzes a given situation (here, a paired hypothesis and its evidence). After case *retrieval*, interpretive CBR *proposes* a solution, which is then *justified* prior to *critiquing* and *evaluation*. Following evaluation, a justified solution may require *adaptation* and then further critique and evaluation. A distinctive element of interpretive CBR is its justification step, which creates an argument for a given interpretation by comparing and contrasting the current situation with the interpretation of the stored situation (to determine whether the interpretation holds for the current situation). The critiquing step tests a justification's argument by applying it to hypothetical situations, prior to evaluation. AHEAD implements retrieval, solution proposal, and solution justification but leaves critiquing, evaluation, and adaptation to the user; we discuss this further in future work (Section 8).

Figure 3 displays AHEAD's functional architecture. Briefly, AHEAD's algorithm consists of three primary steps:

1. *Retrieve*: Given a hypothesis (i.e., a possible terrorist activity) and a library of TMK models representing types of these activities, retrieve the model that best matches that hypothesis.
2. *Propose*: Given the matched model and the evidence leading to the hypothesis, generate the instantiation of that model (i.e., a *trace* in that model) that best matches the evidence. Instantiation is needed because AHEAD's process models are generalizations of concrete cases.
3. *Justify*: Given a model trace and the evidence, analyze the situation described by this evidence and create arguments (both pro and con) explaining why that situation does or does not match that trace.

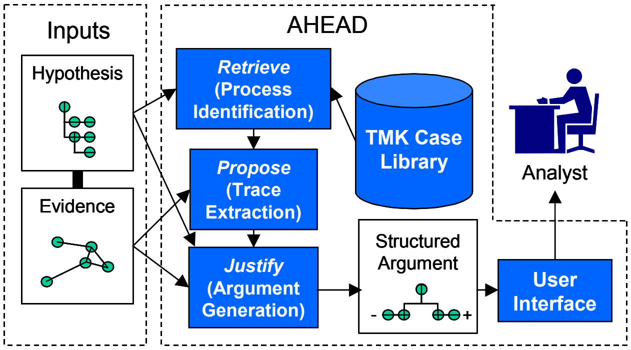

Inputs
Hypothesis
Evidence
AHEAD
Retrieve (Process Identification)
Propose (Trace Extraction)
Justify (Argument Generation)
TMK Case Library
Structured Argument
- +
User Interface
Analyst

Figure 3: Functional architecture for AHEAD.

Figure 4 displays procedural pseudocode for these three steps. The three steps are detailed in the following subsections.

```
h: Hypothesis, e: Evidence, L: Library of process models, p: Process model,
t: Trace (in p), a: Argument (both pro and con)

AHEAD(h,e,L) =
        p = identify_process(h,L)        ; Retrieve
        t = extract_trace(e,p)           ; Propose
        a = generate_argument(h,e,t)     ; Justify
        return(a)
```

Figure 4: AHEAD's pseudocode.

5.1 Retrieve: Process identification

The first phase relates the given hypothesis to a TMK process model. It implements a *case retrieval* step by identifying which model in AHEAD's library of TMK models is most relevant to the given hypothesis. AHEAD uses an off-the-shelf analogical mapping tool for retrieval, namely the FIRE Analogy Server (from the Institute for Learning Sciences' Qualitative Reasoning Group at Northwestern University), a general-purpose analogical reasoning tool (Forbus, 2001). Some automated syntactic transformation is required to represent hypotheses and models in the Analogy Server's formalism. The portion of the Analogy Server that AHEAD directly invokes is MAC/FAC (Gentner & Forbus, 1991), which yields for AHEAD (1) the case (i.e., a TMK model) that most closely matches the input hypothesis, and (2) a mapping between elements of the input and the case.

Consider, for example, the following hypothesis: a local organized crime group has taken over the cigarette industry in Kaliningrad and has killed two people during that takeover. AHEAD would invoke MAC/FAC on that hypothesis using the case library of TMK models to retrieve a TMK model of industry takeovers and to map parameters of that model to entities in the hypothesis (e.g., the parameter for target industry would be mapped to the Kaliningrad cigarette industry).

If no model exactly matches the type of activity being performed, MAC/FAC would retrieve an approximate match. For example, if AHEAD receives a hypothesis concerning an organized crime takeover of postal service in some area, MAC/FAC would recognize that the overall structure of the hypothesis resembles the structure of industry takeovers, even though a postal service is a government organization, not an industry. The specific entities in the hypothesis can then be mapped to analogous model elements. This would allow AHEAD to then perform trace extraction and argument generation using this partially relevant model. If there is no model that even comes close to the hypothesis, AHEAD would skip over the trace extraction portion and proceed directly to argument generation (see Section 5.3).

5.2 Propose: Trace extraction

In the second phase, AHEAD constructs a *trace*: a permissible temporal path through the model with parameter bindings and links to supporting evidence. Because the trace constitutes an elaboration of the hypothesized case, trace extraction is the portion of the general interpretive CBR process (Kolodner & Leake 1996) in which an interpretation is proposed. To illustrate, if an input hypothesis posited an industry takeover, the trace extraction process would start with the mapping between the specific hypothesis and a general model of industry takeovers. It would then determine a temporal path through the model that could produce the observed evidence. Insights from this process would include inferences about what parts of the model have been completed and what parts are underway (e.g., that one company in the industry is being threatened but has not yet been taken over). The trace would include direct links to supporting or contradicting evidence. AHEAD quickly finds relevant evidence because the model is linked to the hypothesis (during analogical retrieval) and the hypothesis is linked to the evidence (in the input).

For example, the model of industry takeovers (Figure 1) involves attempts to take over multiple businesses within the industry, and a business takeover is further decomposed into lower level tasks involving intimidating the CEO of the business and possibly killing the CEO. There are multiple possible paths through the model. For example, if the criminal organization succeeds in taking over a business after intimidating a CEO, then it has no need to kill that CEO. Thus, if the evidence for a particular business takeover suggests that the crime group succeeded through CEO intimidation, then there will be no step in the trace that encodes the killing. However, if the crime group failed to intimidate but did not kill the CEO, then the trace would contain a step that represents the killing (because the model states that it should occur) along with evidence that the step did not occur; this trace step is used during argument generation to create an argument against the hypothesis.

The details of the trace extraction process are illustrated in Figure 5. The inputs to this process are the outputs of the Analogy Server: the model and the mapping between the model and the given hypothesis. The first step in trace extraction is the production of an empty trace (i.e., a trace of the model that asserts that no actions in the model have been performed and that no parameters in the model have been bound). This empty trace is provided to an *atemporal trace-evidence unifier*, which adjusts the trace to be consistent with the evidence at a fixed moment in time. For the empty trace, the unifier adjusts the trace to reflect the world state prior to any actions being taken (i.e., it produces a trace reflecting the initial state). The initial trace is then passed to a *temporal trace iterator*. This subcomponent moves through a single step in the model. It produces a potential updated trace, which indicates that an additional step has been performed, but does not include information about how the evidence supports or contradicts that step. This trace is passed back to the unifier, which does connect the trace at that step to the evidence. The loop between the iterator and the unifier continues until the unifier determines that the final state of the trace corresponds to the current world state. The trace may cover the entire model at this point (suggesting that the process has been completed) or it may only cover part of the model (suggesting that the process is still ongoing).

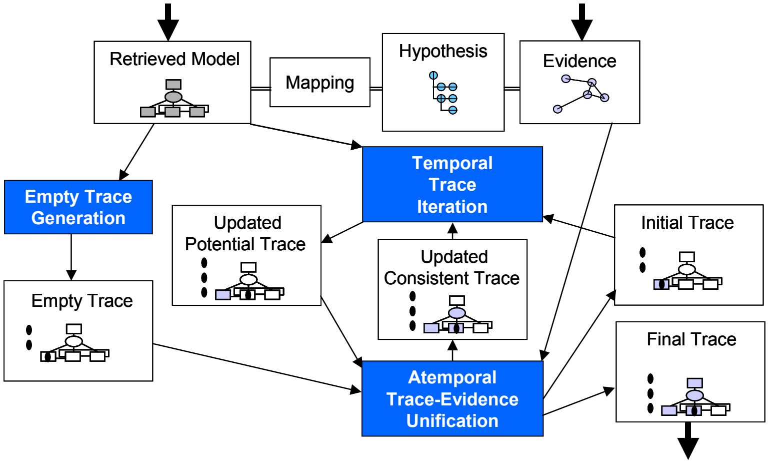


Figure 5: Details of the trace extraction process.

5.3: Justify: Argument Generation

Finally, AHEAD constructs arguments concerning the hypothesis based on the extracted trace; this constitutes a *justification* step of the general process for interpretive CBR (Kolodner & Leake, 1996). More specifically, the argument generation process steps through the extracted trace and produces arguments for or against the input hypothesis based on the evidence

For example, evidence from a business takeover may suggest that a group intimidated the CEO, did not take over the business, and did not kill the CEO. In this example, one portion of the argument AHEAD produces would support the overall hypothesis of an industry takeover (because intimidating a CEO is part of industry takeovers), while another portion of the argument would contradict the claim (because killing the CEO would be expected under the circumstances but did not occur). A user examining the argument could decide that the latter evidence is strong enough to conclude that an industry takeover has not occurred (i.e., that the intimidation of the CEO was part of some other kind of activity). Alternatively, the user might conclude that the crime group simply acted in an atypical manner or that the activity is still taking place. AHEAD does not draw conclusions of these sorts; it simply presents the relevant information supporting and/or contradicting the hypothesis so that the analyst can make a final judgment.

Figure 6 displays the details of the argument generation process. Each element of the trace is analyzed. Elements of the trace include links to facts that support or contradict them; supporting facts lead to arguments for the hypothesis while opposing facts (or a lack of facts) lead to arguments against the hypothesis. Once the analysis is complete, AHEAD has built a formal structured argument consisting of logical assertions. This formal structured argument (including the original hypothesis, the individual arguments for and against, and the facts supporting those arguments) is translated into semiformal and informal versions via a text generation module. The semiformal version is structured as a nested tree composed of small chunks of English text (e.g., "targeted industry: Kaliningrad cigarette market"). The informal version is structured as full sentences and paragraphs; it is much less concise than the semiformal version but may be helpful for users who are unfamiliar with the exact semantics of the semiformal tree. Users can browse both the informal version and the semiformal version of the arguments (see Section 6); the formal version is intended only for use in automated processing.

There are two extreme circumstances for the execution of the argument generator. In the first, retrieval of the model has failed and thus no trace extraction has been performed. In this situation, the trace element analysis loop runs zero times and no arguments for or against the hypothesis are produced; text generation then operates only on the hypothesis (as the root of the structured argument). Thus AHEAD's user interface is still able to present the hypothesis in an organized, textual format even when it fails to produce any analysis of that hypothesis. The second extreme condition for argument generation is one in which no evidence supporting the

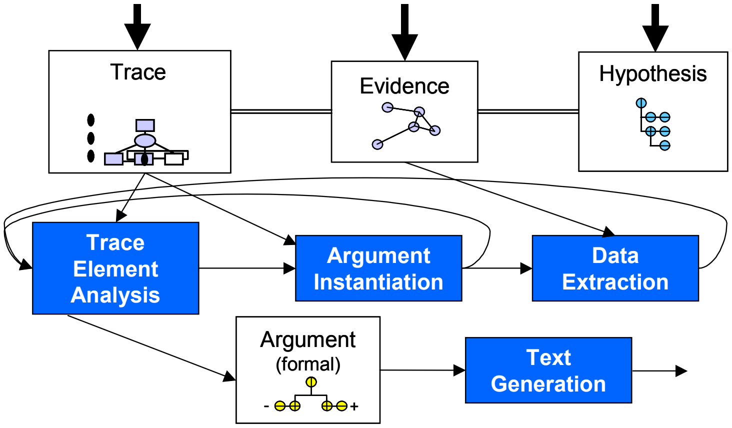


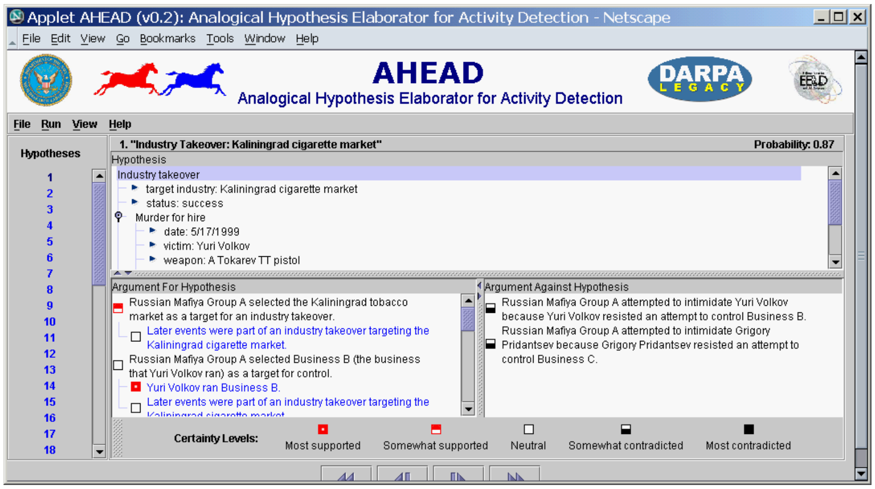
Figure 6: Details of the argument generation process.

hypothesis was found during trace extraction. In this situation, every step in the model will have a corresponding argument against, pointing to a lack of evidence. This result is slightly more informative than the former (i.e., it does give the user a sense of what evidence would be relevant if it were available).

6. Graphical User Interface

AHEAD's user interface allows the user to browse through the semiformal and informal versions of the arguments associated with each hypothesis. Whenever the Argument Generator produces an argument, that argument and the hypothesis that led to it are stored in a library of bindings between hypotheses and arguments. An argument server provides access to this library; it sends the arguments to an argument browser (a web applet). The analyst may then navigate among different hypotheses and related arguments. The browser also has features for switching among semiformal and informal versions of the hypothesis and argument. Furthermore, the browser allows the various elements of the tree representation to be expanded and collapsed, enabling a user to view an abstract overview of the entire argument and then zoom in for details. Finally, the arguments include links to original sources, allowing an analyst to see the arguments' evidential support.

Figure 7 shows a screen shot of AHEAD's argument browser. In this figure, the hypothesis, argument for, and argument against areas are all being presented in semiformal notation. The red and black icons accompanying each argument denote qualitative degrees of certainty (based on source reliability and qualitative heuristics) as indicated by the key at the bottom. For example, complete red squares represent extremely well supported statements while half-filled black squares represent moderately contradicted statements. The example shown in the figure involves an


Applet AHEAD (v0.2): Analogical Hypothesis Elaborator for Activity Detection - Netscape
File Edit View Go Bookmarks Tools Window Help
AHEAD
Analogical Hypothesis Elaborator for Activity Detection
DARPA LEGACY
File Run View Help
Hypotheses
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

1. "Industry Takeover: Kaliningrad cigarette market" Probability: 0.87

Hypothesis

Industry takeover
- target industry: Kaliningrad cigarette market
- status: success

Murder for hire
- date: 5/17/1999
- victim: Yuri Volkov
- weapon: A Tokarev TT pistol

Argument For Hypothesis

⬓ Russian Mafiya Group A selected the Kaliningrad tobacco market as a target for an industry takeover.
- □ Later events were part of an industry takeover targeting the Kaliningrad cigarette market.

□ Russian Mafiya Group A selected Business B (the business that Yuri Volkov ran) as a target for control.
- ▣ Yuri Volkov ran Business B.
- □ Later events were part of an industry takeover targeting the Kaliningrad cigarette market

Argument Against Hypothesis

⬒ Russian Mafiya Group A attempted to intimidate Yuri Volkov because Yuri Volkov resisted an attempt to control Business B.

⬒ Russian Mafiya Group A attempted to intimidate Grigory Pridantsev because Grigory Pridantsev resisted an attempt to control Business C.

Certainty Levels: ▣ Most supported ⬓ Somewhat supported □ Neutral ⬒ Somewhat contradicted ■ Most contradicted

Figure 7: Screenshot of AHEAD's GUI.

argument in the domain of Russian organized crime, one of the challenge problems in the EELD program. In particular, the hypothesis presented to AHEAD involves an industry takeover decomposed into a set of murder-for-hire events. The argument produced by AHEAD includes concrete steps in this industry takeover and the data that indicates whether those steps occurred in this specific instance. The references to evidence (which appear in blue) are hyperlinks that allow a user to directly access the original source or sources for that evidence. In addition to the view shown in the figure, AHEAD also provides a less compact, informal view in which relationships between elements are written out explicitly (e.g., "An attempt to control a business typically includes an attempt to intimidate the CEO. There is a lack of support for the belief that Russian Mafiya Group A attempted to intimidate Yuri Volkov.").

7. Evaluation

We recently conducted an internal pilot study focusing on AHEAD's user interface and the arguments presented in that interface. In this study users were asked to rank the credibility of presented hypotheses, where only some of the hypotheses were accompanied by their arguments. Because some of AHEAD's automated components (i.e., the Trace Extractor and parts of the Argument Generator) are not yet functional, we constructed outputs for these processes by hand. However, we did follow AHEAD's overall process in producing the arguments for the evaluation, so we expect that when the automated components are complete they will produce outputs comparable to the ones we used in this experiment. Consequently, while this experiment provides an evaluation of AHEAD's GUI and the content of the arguments that AHEAD will produce, it does not investigate the computational costs or accuracy of automatically producing these arguments. We will conduct future experiments that address these issues.

7.1 Methodology

In our experiment, we gave six hypotheses to each of six subjects. The subjects were computer scientists but were not participants in this research project. The subjects were not experts in organized crime or intelligence analysis. Each hypothesis concerned a potential Russian organized crime activity, drawn from evidence produced by a simulator developed for the EELD program (IET, 2002). The simulator steps through a declarative representation of various activities in a domain (e.g., contract killings, industry takeovers) to produce "ground truth" information about some concrete simulated events. Once it has the ground truth, it randomly corrupts portions of the information to simulate incompleteness and inaccuracy in gathering evidence. The resulting corrupted evidence is provided as an input for both the hypothesis generation systems and AHEAD.

Three of the hypotheses concerned a single, isolated contract killing, while the other three involved industry takeovers (i.e., larger, more complex activities that include contract killings as subcomponents). Two sources were used for the hypotheses: the ground truth answer key provided by the simulator (which is, by definition, absolutely correct) and the SCOPE iGEN module (Eilbert, 2002), an EELD pattern matching system.

In all trials, subjects had access to the *hypothesis* displayed in the top portion of AHEAD's user interface (see Figure 7) and the original *evidence* (in a separate data file). In some trials, subjects also had access to the *arguments* displayed in the bottom portion of the AHEAD user interface. The independent variable studied in this experiment is the presence or absence of the argument. The dependent variables correspond to responses to questionnaires. Each subject was given six hypotheses to evaluate. All subjects received the same six hypotheses, but each one had a different (randomly assigned) subset for which the argument was also presented. For each hypothesis, subjects were asked the following questions:

- How valid is this hypothesis? (1-10)
- How confident are you of your hypothesis validity assessment? (1-10)
- How much time did you spend studying this hypothesis?

At the end of the experiment, subjects were asked to indicate how much they liked the following features of AHEAD, each on a scale of 1-10:

- Presentation of the hypothesis
- Presentation of the arguments for the hypothesis
- Presentation of the arguments against the hypothesis

Finally, participants were asked to provide any additional comments concerning the interface and the displayed content.

7.2 Results

Table 1 displays the mean response values for the questions about the individual hypotheses. On average, users with arguments took 10% less time and indicated a confidence of approximately half a point higher. These are both encouraging results because increasing the speed with which analysts operate and their assessment confidence are primary objectives of this research.

Table 1: Mean results for the two experimental conditions.

Metric	With Argument	Without Argument
Elapsed Time	5:18	5:55
Confidence	7.40	6.86
Error in judgment	**1.83**	**3.01**
Error in confidence	1.70	2.26

Two other values are listed in Table 1. The first is *error in judgment*: a measure of how far the user's estimate of a hypothesis' validity is from the actual validity of that hypothesis. Actual validities were computed by comparing the hypothesis to the original ground truth using the scoring module associated with the EELD simulator; the scores were scaled to a one to ten range for direct comparison with the users' judged validity. We define error in judgment for a hypothesis as the absolute difference between the judged validity and the scaled (one to ten) actual validity of that hypothesis. The last entry in Table 1 displays the *error in confidence*. Specifically, we define error in confidence for a hypothesis as the absolute difference between how far the user was from being certain (i.e., ten minus the confidence) and how far the user was from being correct (i.e., the error in judgment). On average, users showed lower errors in judgment and lower errors in confidence when they did have arguments than when they did not arguments. These are also important and encouraging results.

All of the results shown in Table 1 were tested for statistical significance using a one-tailed *t* test assuming unequal variance. The result for error in judgment was found to be statistically significant with $p<.05$ (as indicated by boldface in Table 1); this was very encouraging because correctness is arguably our most important objective. The other results were not statistically significant in this pilot study. Given the observed variances and differences in means, a *t* test would require about three times as much data to get statistically significant results for confidence and error in confidence and about fifteen times as much data to get statistically significant results for the elapsed time. It may be possible to reduce the amount of data needed in future experiments by having more constrained tasks.

The results for the summary questions were also encouraging. Presentation of the hypothesis and the arguments against the hypothesis received an average rating of 7.5; presentation of the arguments for the hypothesis received an average rating of 8.6. These results suggest a reasonably favorable impression. Some additional comments addressed specific concerns regarding the interface (e.g., layout of the arguments); these comments will help us in designing future versions of the interface.

8. Future Work

We intend to extend the range of capabilities that AHEAD can provide in a purely automated context. The current AHEAD methodology does not include any step in the process in which some conclusion is drawn about whether the hypothesis is valid; the information in the argument is only expected to help a user determine the hypothesis' plausibility. Automated evaluation of structured explanations is a topic that has been addressed in previous CBR research (Leake, 1992). We may build on this work to enable automated evaluation of hypotheses using the arguments that AHEAD constructs. This hypothesis evaluator could be used as a filter (i.e., users would only see those hypotheses that AHEAD assigned a credibility rating above a certain threshold). Furthermore, this ability could be used to identify critical weaknesses in a hypothesis that could then be sent back to the original hypothesis generation system to drive a search for a stronger hypothesis. In our first round of development, we intend to perform automated evaluation using three interrelated Russian organized crime models (contract killing, industry takeovers, and gang wars). In later work we will increase the number of models and address other domains.

Another key element for future work concerns more elaborate evaluation. After completing AHEAD's implementation, we will conduct increasingly informative evaluations and make incremental refinements based on the results. We will pursue three primary improvements in the evaluation process. First, we will increase the number of subjects that perform the experiments to obtain more statistically meaningful results. Second, we will examine a wider variety of experimental conditions; in the current evaluation we compared performance only with and without arguments, but in later evaluations we will compare performance with different parts of the arguments and/or different variations on argument generation and presentation. Third, we will conduct experiments with real-world data and subject matter experts (i.e., analysts). The third improvement is potentially difficult and costly, but it is crucial for realistically evaluating how well AHEAD addresses its motivating problems.

9. Conclusions

Our pilot study provides preliminary support for the following hypotheses concerning the content and presentation of AHEAD's arguments:

- The arguments allow users to make judgments about hypotheses **faster**.
- The arguments enable more **accurate** judgments about hypotheses.
- The arguments give users more **confidence** in their judgments.
- The arguments lead to more **reliable** reports of confidence.

We expect additional development to enable fully automated production of these arguments. Once that work is complete, we will develop new variations on the content and organization of our arguments. We are also in contact with intelligence analysts for their feedback and suggestions on user interface design and argument content. These efforts will enable us to conduct future experiments that contrast different types of arguments and see how each performs along the different metrics we have considered. Such experiments will enable us to produce an increasingly beneficial tool for enabling analysts to understand and react to hypotheses in a wide variety of asymmetric threat domains.

Acknowledgements

Thanks to Dan Fu and Jim Eilbert for providing hypotheses for development and testing of AHEAD, to our subjects, and to Kalyan Moy Gupta and David Leake for discussions. The lead author holds a National Research Council Associateship Award at the Naval Research Laboratory. The work of the other authors is sponsored by the Defense Advanced Research Projects Agency. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of DARPA or the United States Government.

References

Aleven, V., & Ashley, K. (1996). How different is different? Arguing about the significance of similarities and differences. *Proceedings of the Third European Workshop on Case-Based Reasoning* (pp. 1-15). Lausanne, Switzerland: Springer.

Bergmann, R. (2002). *Experience management: Foundations, development methodology, and Internet based applications*. New York: Springer.

Branting, K. (1991). Reasoning with portions of precedents. *Proceedings of the Third International Conference on AI and Law* (pp. 145-154). Oxford, UK: ACM Press

Cox, M. (1997). An explicit representation of reasoning failures. *Proceedings of the Second International Conference on Case-Based Reasoning* (pp. 211-222). Providence, RI: Springer.

Eilbert, J. (2002). *Socio-culturally oriented plan discovery environment (SCOPE)*. Presentation at the Fall 2002 EELD PI Meeting. San Diego, CA: Unpublished slides.

Forbus, K. (2001). Exploring analogy in the large. In D. Gentner, K. Holyoak, & B. Kokinov (Eds.) *The analogical mind: Perspectives from cognitive science*. Cambridge, MA: MIT Press.

Gentner, D., & Forbus, K. (1991). MAC/FAC: A model of similarity-based retrieval. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 504-509). Chicago, IL: Lawrence Erlbaum.

Goel, A., Bhatta, S. & Stroulia, E. (1997). Kritik: An early case-based design system. In M. Maher and P. Pu. (Eds.) *Issues and Applications of Case-Based Reasoning in Design*. Mahwah, NJ: Erlbaum.

Goel, A.K., & Murdock, J.W. (1996). Meta-cases: Explaining case-based reasoning. *Proceedings of the Third European Workshop on Case-Based Reasoning* (pp. 150-163). Lausanne, Switzerland: Springer.

IET [Information Extraction & Transport, Inc.] (2002). *Task-based simulator version 9.1*. Unpublished user's manual. Arlington, VA.

Kolodner, J., & Leake, D. (1996). A tutorial introduction to case-based reasoning. In D. Leake (Ed.) *Case-based reasoning: Experiences, lessons, & future directions*. Cambridge, MA: MIT Press & AAAI Press.

Leake, D. (1992). *Evaluating explanations: A content theory*. Mahwah, NJ: Erlbaum.

McLaren, B.M. & Ashley, K.D. (2000). Assessing relevance with extensionally defined principles and cases. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*. Austin, Texas: AAAI Press

Murdock, J.W., Aha, D.W., & Breslow, L.A. (2003). Case-based argumentation via process models. To appear in *Proceedings of the Fifteenth International Conference of the Florida Artificial Intelligence Research Society*. St. Augustine, FL: AAAI Press.

Murdock, J.W., & Goel, A.K. (2001). Meta-case-based reasoning: Using functional models to adapt case-based systems. *Proceedings of the Fourth International Conference on Case-Based Reasoning* (pp. 407-421). Vancouver, Canada: Springer.

Rissland, E.L & Skalak, D.B. (1989). Combining case-based and rule-based reasoning: A heuristic approach. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (524-530). Detroit, MI: Morgan Kaufmann.

Stroulia, E., & Goel, A.K. (1995). Functional representation and reasoning in reflective systems. *Applied Intelligence*, 9, 101-124.

Tautz, C. & Fenstermacher, K. (Eds.) (2001). Case-base reasoning approaches for process-oriented knowledge management. In R. Weber & C.G. von Wangenheim (Eds.) *Case-Based Reasoning: Papers from the Workshop Program at ICCBR-2001* (Technical Note AIC-01-003, pp. 6-28). Washington, DC: Naval Research Laboratory.

Toulmin, S. (1958). *The uses of argument*. Cambridge, UK: Cambridge University Press.