Adaptive Knowledge-Based Monitoring for Information Assurance

Jon Doyle, Isaac Kohane, William J. Long, and Peter Szolovits
Massachusetts Institute of Technology
Laboratory for Computer Science
and
Childrens' Hospital (Boston)

This is the main text of a proposal submitted on October 30, 1998 to the Defense Advanced Research Projects Agency in response to BAA #98-34 "Information Assurance (IA) of the Next Generation Information Infrastructure (NGII).

Contents

1 Executive Summary

Monitoring tasks play a central role in information assurance, since it is difficult to respond to attacks one cannot detect. Ensuring effective monitoring is difficult, however, because the threats to which IA monitoring systems must respond evolve continually in the never-ending competition between adversaries and defenders. These changing threats pose important problems for the cost and technological requirements for responding.

The cost of responding to changing threats is great at present, as IA systems require manual coding and configuration on the part of the security personnel of each enclave. These personnel must determine the patterns and circumstances of concern to the particular organization and functions of their own enclave, and how to tailor monitoring mechanisms to provide them the information most relevant to their own situation. Performing such tailoring is difficult at best, and impossible at worst, and sharing the results of such efforts is usually not possible, since nothing enforces any uniformity among the particularizations for different enclaves.

Competition with adversaries also creates demanding technological requirements for monitoring technology since it tends to produce more and more complex attacks that rely on intrusions distributed across facilities. Recognizing such attacks requires technologies capable of correlating events distributed across space and time. Such correlation requires monitoring systems that can adapt their behaviors to information about changes in their situation, for example to increase the degree of scrutiny or engage in proactive investigation when the situation signals increased probability of or danger from attacks.

We propose to build on the MAITA system for knowledge-based monitoring, developed under DARPA's HPKB program, to construct tools and libraries of monitoring knowledge and methods that enable system designers and security analysts to rapidly construct, adapt, share, and reuse networks of distributed monitoring processes. The system architecture provides an open, flexible, and scalable basis for such networks. Libraries describing monitoring methods at different levels of detail and specialized for a wide variety of monitoring tasks and domains are intended to provide near-complete answers or answer components for many needs, and provide for a coherent expression of knowledge about intrusion detection, boundary policies, and the situational in the same vocabulary. Libraries describing the organizational structure, function, and workflow of individual and generic types of enclaves provide the knowledge needed to guide and partially automate system configuration tasks. Advanced methods provide means for adjusting analytical effort to the objective type and degree of danger and for adjusting the character and volume of alerts issued to the needs of the analyst receiving them. A flexible control system provides tools for creating, composing, adjusting, and inspecting monitor models, parameters and real-time data flows. We will combine these elements to demonstrate a security monitoring desk supporting rapid adaptation of monitoring systems, effective sharing of methods contributed by a community of system security personnel, and integration with systems for responding to potential or actual threats and attacks.

2 Innovative claims

Problems of information assurance are not set in a static world in which one can identify threats, develop protective measures, and then sit back in safety; the reality is instead one of an ongoing competition in which each threat and defense is met by new defenses and threats. In this setting, the IA task implies both reactive and proactive defenses, that is, creating defenses against newly observed attacks and imagining new possible attacks and creating defenses against those. This means that IA systems themselves must change as rapidly as their environment if they are to assure performance. Systems that cannot be readily changed to meet new threats will mainly serve to provide a baseless sense of security while letting the intelligent adversary freely pick and choose his actions.

This proposal addresses only detective and defensive aspects of IA, in particular, those of networks of processes for detecting inappropriate information flows and hostile intrusions, and of defensively managing these networks to optimize defensive capabilities. The focus of attention is on techniques for maintaining effectiveness as circumstances and setting change. The expected result of the proposed research is a security monitoring desk that provides direct means for rapidly constructing, adapting, and maintaining effective networks of security monitoring processes.

We consider these problems at several levels or settings: at the level of the individual process performing monitoring or boundary control filtering, at the level of the individual analyst or security officer seeking to address new threats, at the level of the system developer seeking to provide some new enclave with protection, and at the level of the NII community in seeking to share and reuse defenses devised for one enclave with others. The innovative claims highlight issues at each of these levels.

Across all levels, our key innovative ideas are the following:

- We will build monitors and systems of monitors that include *situational awareness* to give a context for interpreting the significance of what they are monitoring.

- We plan to build *formally-defined models* of the systems, enclaves and domains being monitored, the threats offered by potential attackers, and the operations and communications that may be the targets of attack. This explicit knowledge forms a basis for expressing the monitoring plans and processes that are developed for particular tasks and environments.

- We will organizing these formalized plans and processes into *sharable libraries* that support efficient adaptation to new, similar tasks and circumstances.

- We will use *explicitly encoded knowledge* to allow our proposed system to reason about and partly automate the task of constructing or updating monitors as new knowledge is acquired and as circumstances change.

- We apply a *uniform architecture* of monitoring to monitoring tasks at different scales of operation and shared across organizations. This architecture will contribute to a commonly-created set of monitoring solutions that can be efficiently customized and adapted to new needs.

The combination of knowledge-based techniques, intelligent monitoring, and application to information assurance forms the core of innovative claims for this proposal.

IA monitoring processes require situational awareness in varying degrees to provide effective protection. Individual processes that monitor locally-defined streams of data being collected about the operation of particular systems or local segments of networks are not well situated to make accurate judgments about the significance of the data they are examining. In the absence of a broader context, such monitoring processes risk overlooking aspects of their locally-observable state that may, when combined with other monitor processes' observations, indicate something highly significant. Conversely, raising frequent alarms based on local evidence risks overburdening the ability of human observers to respond appropriately to suspicious activities; if too many situations are too often declared suspicious, then the systems that "cry wolf" come to be ignored—at worst, disabled—even when the wolf does show up.

Collections of observations from different monitors become significant only when there is an encompassing theory that lends them common meaning. Thus, anomalies at several nearby sites in a network or a common pattern of unusual activities at systems related by shared function are best interpreted by hypothesizing a potential enemy plan whose components are consistent with the observed activities. In the absence of such knowledge, any system may be likely to miss significant challenges. In addition, if the requisite knowledge is encoded in such a static way that a resourceful attacker can learn its limitations and count on them not changing, then coordination among monitors may again be of limited value because carefully chosen attacks can avoid detection.

To overcome these limitations, locally-focused monitoring processes must interoperate and provide context for each other in which to interpret their observations. Our approach to achieving this is to codify monitor outputs in terms of common knowledge-based concepts that relate even low-level observations to descriptions of the domain being monitored. This use of knowledge will support improved processing, helping to weed out false alarms yet still signal true events.

In addition to carefully coordinating the observations of local monitoring processes, a successful IA architecture must also allow local processes to utilize knowledge about conditions and events external to the enclave being monitored. Without such an ability, local monitor processes will always operate in the same way, regardless of what transpires in the broader world around them. Our MAITA architecture makes it easy to communicate such situational changes to all but the lowest-level detectors (in which performance requirements demanded by high data volume may make dynamic behavior too costly). Some examples of relevant situations in which sensitivity to external events might pay off for local monitors include:

- Conditions that may increase the likelihood of an attack, including news articles about a recent attack or new attack method that might stimulate copycat attacks, news of the arrest of an information terrorist that might stimulate revenge attacks, or increased international tensions with potential enemies.

- Conditions that may increase the difficulty of detecting an attack by increasing noise or loading resources, such as major Internet events (Starr report release, JPL meteorite photos) or propagation of PC viruses designed to launch unwitting attacks.

- Conditions that increase vulnerability or severity of damage in case of successful attack, thus increasing the payoff from attacks, such as a major regional power outage, telephone system failure, or triple witching hour on Wall Street.

Systematic categorization of how external situations vary can help to define the types of potential attacks to which local monitors must be sensitive. Useful dimensions for categorizing potential attacks include the type, likelihood, disutility (resultant damage), source, target, timing, purpose, method, and detectability of attacks, as well as the offensive and defensive capabilities of the attackers and defenders and the potential for interference from third parties.

In addition to requiring sensitivity to local and environmental events, all but the most low-level monitoring processes must respond to changing user needs. Such sensitivity is most obviously necessary in regulating the volume of alerts or other information provided to the user as the user seeks to scrutinize system activity to a greater or lesser extent. Though defenders strive to make their systems perfect, one must expect that some unforeseen threats will not be clearly identified by the monitors in place. At times this requires the user to observe broad classes of event alerts in order to check that the monitors are not missing something important. In many cases this broadening of the events to be observed amounts to changing the alerting thresholds or other bases of decisions to alert of the monitoring processes so that they are more liberal in deciding to report potentially interesting events. In general, such variability requires monitoring processes that take into account an explicit decision model, and so track changing user alerting preferences in the course of their operation.

Rapid response to new threats requires libraries of monitoring procedures and knowledge shared among and easily usable and modifiable by individual analysts. Because of the adversarial nature of the information assurance task, IA attacks and defenses will naturally change over time, and therefore require a high degree of effort devoted to maintenance. Sharing of knowledge and strategies across a knowledgeable user community is the best way to amortize the effort and multiply the benefits across many users. If suitable representations and sharing mechanisms are in place, then knowledge among a community of users can accumulate incrementally and improve everyone's abilities. In particular, a shared knowledge base can become a repository of new relevant knowledge, including an evolving characterization of new methods of attack. A shared library of monitoring methods can also codify approaches that have been found useful by colleagues, either to serve "as is" for protection against known threats or to be the basis for new extended monitoring methods that incorporate earlier ones.

In addition to such valuable sharing, we plan to implement a capability for our system to reconstruct (portions of) IA systems from the shared method libraries either periodically or when suggested by relevant situational changes or changes to the method library. In this way, incremental improvements in some part of a shared monitoring method library can be incorporated in larger monitoring methods, to propagate the improvements to all those monitors where they are relevant. If quality assurance methods are used in admitting library additions to make sure that untested methods do not propagate and worsen other monitors, reconstruction of monitoring processes can leverage contributions from a community of users to improve monitors used by all of them.

Monitoring systems need to be tailored to the particular aspects of each enclave and domain. Different enclaves and domains have different requirements for monitoring, based on differences among the systems they operate, the organizational functions they support, the normal communication flows they use, the risks they face, and other factors. Therefore, it is not possible to design monitoring schemes that are generic over these factors, or even ones that are specialized just to the type of systems they are meant to monitor. In fact, operation of the monitoring processes themselves may require checking the significance of events against a model of the particular organization's structure and function.

Custom creation of monitors tuned to the needs of an individual enclave can be a very costly, labor-intensive task, and this difficulty makes it more likely that inappropriate generic monitors will be adopted across many enclaves, increasing their collective vulnerability.

Our approach to overcoming this difficulty is to build IA systems by composing elements from a rich library of monitoring methods, customizing these against formal descriptions of the characteristics of the target enclave. As in the case of the library compounding the improvements made by a population of individual analysts, as the organizational and monitoring network library grows, it also becomes more likely that reasonable monitors designed for similar enclaves are already cataloged in the library, in which case their adaptation to local needs and conditions becomes much less costly than the construction of brand new monitors.

Self-similar monitoring networks can provide information assurance at all levels from the individual enclave to the national level. Information assurance techniques must scale from the level of individual enclaves up to the national level. In our approach, a uniform architecture for monitoring and a proper design of particular monitoring methods makes scaling of information flows relatively easy. The key is that inputs of each higher level must include the alerts issued by the level just below, in an ordinarily hierarchical fashion. So long as monitoring processes at higher levels correspond well to the monitored organizational and task units ("high-level enclaves"), more aggregate monitors can monitor more aggregate organizations. Our approach to shared libraries permits similar scaling, with the same libraries and monitor construction tools serving both the enclave system designers and individual user analysts at every level.

3 Technical plan

3.1 Technical Approach

Our approach to addressing IA security desk tasks will use the MAITA system for knowledge-based monitoring system construction, maintenance, and control. We will develop a security desk for controlling a distributed network of IA monitoring processes, for exploiting and contributing to shared libraries of IA monitoring procedures and concepts used in rapidly constructing, modifying, and augmenting monitoring networks that counter new actual or contemplated threats, and for tailoring security desk operation to the needs of individual users.

3.1.1 A vision of the future

Monitoring tasks of any scope may benefit from a knowledge-based approach, but the benefits of this approach and the needs for further progress show up most clearly in the broader scope tasks. To see problems before they become imminent requires interpreting a broad stream of data to recognize many different patterns of facts which, in themselves, are only distantly related to the threat of interest through a tangled skien of causes and implications. Lists of indicators and warnings, currently used by analysts, include items both directly and indirectly relevant to the conditions of interest, but are very limited in the sorts of patterns they can detect, since the primary evaluation of such lists is simply to count the number of items which obtain. Improving the effectiveness of such techniques requires the ability to specify different ways the condition might arise, including both particular causes and general patterns of causes, and the ability to easily add new patterns of causes as they occur to the analyst. This ability in turn requires formalization of a broad range of knowledge about the world and means for specifying patterns of causation and interpretation.

To help understand the roles of the technical details to follow, we first describe a possible scenario illustrating the way a security officer might one day use our system. The main point of the scenario is to show how the officer can quickly reconfigure the functionality and adjust the performance of the system to address a new situation.

> *Security Officer MacLean starts his day by relieving Officer MacInnes, who informs him that things have been quiet during her watch, and that the Information Assurance monitoring system is in place and operating properly. As usual, he first readjusts the height of his chair, which MacInnes had adjusted for her shorter stature. He then resets the system alerting parameters, which control how likely and significant different potential conditions have to be to cause an alert to be generated to the watch officer. An old-timer, MacLean knows he can tone down the alerts during quiet times, and quickly see more if things start happening. He knows that MacInnes will eventually find a similar modus operandi when she gains more experience, but having only been on the*

job a month, she still is taking the measure of the task, and dials up a broad stream of alerts just to make sure she isn't missing something.

Officer MacLean next looks over the current activity, sees MacInnes wasn't kidding about it being pretty quiet, and starts to check his email. He quickly sees an article forwarded automatically by his subscription service from the New York Times reporting an intrusion into an Army facility in Texas. The intrusion looks like a standard sort, one he knows is covered by his own monitoring system, but the newspaper description suggests to him that the intruders may have been a new group, not one of the regulars. The hints about possible identities set him thinking. Later in his mail he sees a broadcast from his counterpart at the facility in Texas, giving his colleagues additional details about the attack. Things click for MacLean, and he strongly suspects the perpetrators were a terrorist group based in Sri Lanka operating through Tamil expatriates in Australia and the UK. He sends a note back to Texas raising this possibility, and sets to work to set up new defenses. After calling the IA help desk at the NSA, which supplies the name of a CIA official knowledgeable about the terrorist group, MacLean calls up a red team which serves his and other enclaves on a consulting basis, and in a conference call, identifies a number of the most likely targets and strategies the Sri Lankan group might adopt in light of their Texan foray.

MacLean then starts beefing up his security. He goes into the library of monitoring methods, and finds a method that looks good for detecting one of the identified strategies. He creates a new monitor based on this method, specializes it to the particulars of his enclave by linking it to the enclave organizational model used in specializing all his other monitoring processes, and links it into the operating monitoring network. For another of the identified strategies he finds no canned method, but quickly slaps together three smaller library entries into a new method to handle the second strategy. After storing this new method back into the library, he again creates a new process and incorporates it into his network. The third strategy his team identified requires a bit more thought, and while he is working on it, his board lights up; his enclave is under an attack, one corresponding to the second strategy. He loosens his alerting restrictions to see more, inspects some of the details of the attack, and signals some of his other monitoring processes to move to increased security levels. Those processes will now perform more thorough checks on the streams they monitor to head off attacks seeking to benefit from the distractions posed by strategy number two. Seeing things are under control, MacLean quickly broadcasts an alert to his colleagues about the thwarted attack, refers them to the old and new library entries, and tells them what he will call the third library entry once he finishes it. This he does later in the day after he completes the new defense, sets it going, and stores it in the library.

Some time later, MacLean gets a note from his division's IA chief. The *division-level security desk noted the attack and response in his enclave, and compliments MacLean for the quick thinking on identifying the new threat and developing the countermeasures. Seeing his board quiet again, MacLean throttles back on the alerting levels and prepares advice for his relief.*

3.1.2 Building on the MAITA system

We base our approach on the MAITA system, which provides the framework for realizing the above scenario. The major elements of MAITA are a library of monitoring methods, an architecture for operating networks of monitoring processes, and a flexible, display-oriented control system for quickly constructing, composing, modifying, inspecting, and controlling monitoring networks. The library of the scenario corresponds to the MAITA monitoring library; the operational monitors correspond to the monitoring processes operated by the security officer; and the security desk itself corresponds to the network control system. We briefly describe each of these elements; see [4, 3, 2] for more details.

The central concept in the MAITA architecture is that of a network of monitoring processes, operating under the control of a "monitor of monitors" or "MOM", and constructed from entries in a library describing abstract and concrete monitoring methods.

The set of monitoring processes form the nodes of the network, and the communication paths form the edges or links of the network. Each process in the network may have a number of "terminals", each of which receives or emits streams of reports. The network may exhibit a hierarchical structure, as some monitoring processes may consist of a subnetwork of subprocesses. The metaphor we used for thinking of the operation of these networks is that of electrical networks, in which we "wire together" various processes and network fragments by connecting their terminals together. A MOM provides means for constructing, maintaining, inspecting, and modifying the monitoring network and its operation. We achieve a degree of uniformity in the control process by organizing MOMs as special types of monitoring processes.

The notion of terminals is specific to the MAITA architecture, but the architecture permits connection of its own processes with external data sources or recipients through any one of a number of common protocols. In addition, one may integrate legacy processes more closely into the monitoring network by enclosing them in wrappers to give them terminal functionality of standard MAITA monitoring processes. The MAITA system includes wrapper templates for a variety of types of processes.

Information flows through the network by several different protocols, including socket-based ASCII character streams, HTTP (Hypertext Transport Protocol, used by the World Wide Web), SMTP (the Simple Mail Transport Protocol, used by email systems), Java RMI (Java Remote Method Invocation), ODBC (Open Database Connectivity), and OKBC (Open Knowledge Base Connectivity, a protocol for transmitting logical and frame-structured knowledge to and from knowledge bases), with the system developer or user choosing the protocol appropriate to the volume, regularity, and type of the information being transmitted. Regular and high-frequency transmissions typically go through persistent stream, ODBC, or OKBC connections. Intermittent and low-frequency transmissions probably go on temporary HTTP, SMTP, Java RMI, ODBC, or OKBC connections. Records of information transmitted to input or from output terminals are structured in protocol-dependent formats.

3.1.3 Monitoring libraries

We organize libraries of monitoring procedures and organizational models to express a broad range of knowledge about such procedures and about the environments in which they operate. This requires a rich language in which to express the procedures and world knowledge, a broad body of world knowledge with which to relate different monitoring and organizational concepts, and a clear organization of the procedures themselves that reflects abstraction hierarchies and other dimensions along which to classify the procedures. Reference to a a body of concepts about the world is essential for specifying the intent of different monitoring procedures, and for increasing the coherence of the library. The organization of the monitoring procedures according to different properties and dimensions of variation is essential for reasoning about and manipulating these descriptions in the course of knowledge acquisition, learning, compilation, and other aspects of the process of maintaining a monitoring system. Similar remarks apply to models of organizational structure and function, which monitors may employ in expressing security policies and judging appropriateness of communications and other operations.

The knowledge bases used in MAITA to represent and use the libraries of monitoring element descriptions need not be the same as any KBs used by the monitoring processes themselves.

The descriptions we employ in the libraries do not exercise the more idiosyncratic aspects of some knowledge bases. To ensure this, the MAITA system uses the OKBC protocol for KB retrieval and storage. This is a protocol providing common access means for a variety of KB types. In our implementation, we plan to employ the CYC knowledge base as the primary library KB, in order to make use of the sizable body of knowledge already encoded in CYC as compared with LOOM or other KB systems.

While the main focus of the library is on descriptions of monitoring processes, we will also include descriptions of organizational structure and function. An IA system developer would construct such models of the enclave being protected in order to help identify normal, anomalous, and forbidden system behaviors. These models describe the offices and roles of an organization, the types of information and operations each organizational entity handles, permissible information flows through the organization and permissible transactions among organizational entities. In addition, organizational models should also describe diurnal or calendric variations in information flows and transactions. All these have obvious relevance to IA monitoring, and some current intrusion detection technologies are based on assessing some of these organizational parameters, especially anomalous temporal variation in usage statistics. We propose to base such models on one or more of the recent standards for representing organizational structure and workflow and related notions, such as the SPAR model of activity structure developed in DARPA's ARPI program.

Library entries include two types of descriptions of monitoring methods: competence or capability descriptions, and performance descriptions. The capability descriptions characterize primarily the types of information operated on, the relations of inputs to outputs, and the purposes or principal uses of the method. The performance descriptions, in contrast, characterize the representations or data structures used to encode information and the procedural steps or concrete code used to execute the method. Further details on these descriptions may be found in [2].

The monitoring method library includes entries at all levels of computational detail, from very abstract procedures covering virtually all monitoring tasks, to intermediate-detail procedures capturing more specific algorithmic ideas, information about the class of domain or enclave, or types of information being monitored, to highly detailed procedures involving specific representations, code, domain details, and signal sources. For example, the most abstract levels might speak of constructing and comparing a set of hypotheses about what is going on, without providing any details about how the hypotheses are constructed or compared. At an intermediate level, the $\mathrm{TrenD_x}$ [21, 20, 31, 25, 17] trend monitoring system developed at MIT uses a partial-match strategy operating over a set of trend templates, each of which consists primarily of temporal constraints characterizing some temporal event. More refined monitoring models would amend this procedure to take probabilistic or default information into account, or to embed background knowledge of the domain in the matching strategy (e.g., always try matching location first before bothering with other information). Still more concrete procedures might describe OS-specific methods for recognizing port sweeps or ftp-write attacks. The most abstract control and interpretation procedures serve as a base for more specific ones, but will be rarely used directly. The real strength of the library of monitoring models will be in identifying specific combinations of representations, procedures, and domain characteristics that offer significant power compared to the more abstract procedures.

Another dimension of variation is whether the monitoring represents the activity of a single agent or is distributed across multiple agents. For many target applications, such as battlefield situation awareness and intelligence analysis, the distributed monitoring model arises naturally. The monitoring procedure library will therefore include procedures that distribute the effort in various ways, including fixed distributional arrangements, hierarchical tasking (as is done in military planning at different echelons), and economic models in which analyst processes distribute tasks through a market in intelligence information. We will draw here on both our ongoing research on market-guided planning and computation [11, 12], and on our web-mediated mechanisms for distributed medical record retrieval and analysis.

Combination and cascading of monitoring procedures leads to additional library elements, since one may sometimes combine synergistic but separate monitoring procedures into more effective ones. Some of these combinations of monitoring procedures mirror combinations of trend templates, thus reflecting portions of the trend template library, but the dataflow connections among monitoring procedures mean that the monitoring procedure library must be treated on its own rather than as a derivative of the trend template library.

Procedures for a small number of fairly abstract monitoring procedures have been codified already in the CommonKADS library of problem solving methods [1], but most of these concern fairly active procedures for diagnosing devices for which complete structural and functional information is available. Expanding on this basis to cover the broader tasks not addressed by the very restrictive CommonKADS assumptions will constitute one of the important contributions of this research.

As part of the formalization of the library of monitoring models, we will develop an ontology for monitoring processes, including concepts such as causal structures (chains constitute only the simplest such structures), partial matches, evaluations of significance and likelihood, and focus of attention. We will seek to build on the ongoing ARPI work on planning ontologies in formalizing this ontology. In particular, plans represent a specific type of causal and procedural structure, and plan monitoring (sentinel) tasks constitute a very important class of monitoring applications.

We expect a fully developed library to contain a great many entries, but will only construct those portions of such a library required to characterize IA monitoring methods explored in the course of this research. Our focus is on two classes of monitoring methods: signal transducers, which we view as very specific computational procedures aimed at signal transformation or data reduction that are generally independent of particular tasks, and trend templates or signal correlators, which we view as more abstract multi-input computational methods specific to particular tasks or domains. We describe each of these briefly.

3.1.3.1 Signal transducers: Signal transducers transform a signal into one or more new signals. The most familiar variety of signal transducers all concern continuous or time-series signals. These include linear extrapolation and interpolation, trend line fitting, wavelet decomposition, fourier transforms, summary statistics, outlier detection, threshold detection, and others. We place most low-level pattern-matching procedures into this category as well, including many boundary control policies (e.g., don't pass any incoming requests from `.lk` (Sri Lanka)) and intrusion detection signature-recognizers and statistics collectors (e.g., report all write attempts in `/usr/local`, port or ip sweeps, and syslog and ping of death attacks).

3.1.3.2 Signal correlators and trend templates: Signal correlators take several streams of data as inputs and provide one or more new signals (or propositions) as output—one can think of them as multi-signal transducers—and constitute one of the most important elements of a knowledge-based monitoring system. Events or trends of interest are normally signaled by coordinated changes in different aspects of a situation. For example, common statistical abnormality detectors measure discrepancies between statistics at different time scales. Plan recognizers also are naturally viewed as looking for specific types of correlations between temporal events that characterize one or more execution paths through the plan.

The building blocks of signal correlators include standard continuous-signal operations such as differencing, modulating, and demodulating, but the most interesting building blocks for knowledge-based applications are those correlating propositional and graded information, such as rules, reasons and argument structures, Bayesian probabilistic networks, causal networks, and temporal constraints. Existing monitoring systems show how to combine some of these. Determining how to combine these types of correlations will constitute one of the principal foci of this research.

We will use these signal-correlating building blocks to construct a library of abstract and special signal correlators called *trend templates*, after the representation by that name developed at MIT by Haimowitz and Kohane in the TrenD_x system [21, 20, 31, 25, 17]. A trend template (TT) is an archetypal pattern of data variation in a related collection of data. For example, a particular IA trend template might characterize an event consisting of a port sweep followed by increased traffic using some particular port to a small set of destinations rarely seen before. Each TT has a temporal component and a value component. The temporal component includes landmark time points and intervals. Landmark points represent significant events in the lifetime of the monitored process. They may be uncertain in time, and so are represented with time ranges (min max) expressing the minimal and maximal times between them. Intervals represent periods of the process that are significant interpretation. Intervals consist of begin and end points whose times are declared either as: offsets of the form (min max) from a landmark point, or offsets of the form (min max) from another interval's begin or end point. The representation is supported by a temporal utility package (TUP) that propagates temporal bound inferences among related points and intervals [27, 26]. The value component characterizes constraints on individual data values and propositions and on computed trends in time-ordered data, and specifies constraints that must hold among different data streams.

In matching a trend template to data, two tasks are carried out simultaneously. First, the bounds on time intervals mentioned in the TT are refined so that the data best fits the TT. For example, a TT that looks for a linear rise in a numeric parameter followed by its holding steady while another parameter decays exponentially must find the (approximate) time boundary between these two conditions. Its best estimate will minimize deviations from the constraints. Second, an overall measure of the quality of fit is computed from the deviations. The most appropriate language of trends and constraints will vary from domain to domain, and we expect to build a rich set of capabilities to populate the ontology of trends. For the constraint language, we have so far explored mainly linear and quadratic regression models for numeric data, absolute and relative numerical constraints on functions of the data, and logical combinations of such descriptions and propositions. We plan to develop the ability to build other TTs using descriptions that characterize any outputs of signal transducers and additional models of correlation among signals. The template library will be expanded over the life of the effort, with research and new applications leading to new additions. Moreover, augmenting the library with new templates will form one of the key operations in practical use of the system, allowing analysts to codify new indicators and warnings as they are identified.

The measures of quality that tell how well various TTs fit the monitored data become either time-varying signal or propositional outputs of the signal correlators and trend detectors, and provide the appropriately processed inputs for making monitoring decisions.

3.1.4 Situation-sensitive monitors

Every multi-signal correlation monitor can be viewed as possessing a limited sort of situational awareness, in that the behavior given one signal depends on the situation defined by the information arriving on the other streams. Virtually all interesting monitoring procedures require situational awareness in this sense.

We claim that effective IA monitoring requires that at least some of the monitoring processes exhibit situational awareness in a broader sense, in which changes in monitor behavior are triggered by sporadic receipt of updates about a range of relevant conditions occurring in the environment of the monitor. Such updates might be as simple as a change of "infocon" levels akin to "defcon" levels, or as complex as propositional or probabilistic updates to a situational knowledge base maintained by the monitoring process. We can of course view the sequence of such updates as just another input to the monitor, but the sporadic and nonuniform nature and discontinuous effects of such updates, in which they change the way future inputs are processed, make it more natural to view the updates as changing the situational model employed by the monitor in processing the ordinary input signals.

3.1.4.1 Situational awareness: We can distinguish several forms of situational awareness useful in IA monitoring processes. The first dimension of variation divides monitors according to whether the situation in question is an "objective" situation, such as whether related enclaves are experiencing attacks or whether Internet congestion levels seem abnormally high, and "intentional" situations, such as the preferences of a security officer regarding the seriousness of abnormalities required to justify issuing an alert on the security desk. The second dimension of variation divides monitoring methods according to whether the situational model maintained by the process consists of only summary variables (e.g., overall probability and disutility of an attack at present, source and target of attack, timing, purpose, and method of attack, level of defenses available, etc.) or consists of a threat model of some complexity (e.g., an influence diagram expressing the overall probability and utility of attack in terms of information about attacks on other enclaves, news articles about increased tensions with known adversaries, social and infrastructure disruptions such as power outages and strikes, etc.).

We propose to develop a set of monitor process templates that exemplify the main points within this space of variation, including summary objective situational models, detailed objective situation models, summary intentional situational models, and detailed intentional situation models. The objective models should help represent existing notions like lists of indicators and warnings and causal condition of interest (COI) models. The intentional models, in turn, will help represent user models controlling what is shown to the security desk officer. Such control is critical in rendering the security desk displays and alerts in a comprehensible form, especially when the officer must examine potentially large numbers of uninformed alerts to detect abnormalities or intrusions of sorts that have not yet been identified and covered in the monitoring network. For example, a standard way of organizing the basic security desk alerting displays may be to interpose alerting processes between the objective monitoring network and the user. These alerting processes would consist of little but a detailed or summary model of the user's preferences regarding alerts, and would transmit only those alerts in conformance with the current representation of those preferences. Note that here, as elsewhere, by user preferences we mean true decision-theoretic preference orders and utility representations, not the simple-minded option selection schemes that many COTS software systems call "preferences".

We will also develop standard protocols by which monitoring processes may subscribe to alerts from situation knowledge bases. We will attempt to adopt methods provided by or compatible with current DARPA efforts like Dynamic Databases, Project Genoa, and the crisis management portions of the High Performance Knowledge Bases program.

3.1.4.2 Situation-dependent behaviors: Degree of passivity forms another important dimension of variation. At one extreme, simple monitors based on lists of indicators and warnings may just observe a set of propositional inputs to detect the presence or absence of a set of specific conditions, and the output of the procedure is to simply report the set of present conditions, or perhaps just the number of conditions present at a given time. At the other extreme, active monitors may start with such a list of indicating conditions and continuously actively seek out new information to determine the presence or absence of these conditions, as opposed to simply waiting for notifications of presence to enter as inputs. Intermediate monitoring procedures might simply filter inputs passively until some threshold is reached, and then switch to an active mode to confirm or deny remaining conditions.

Degree of passivity is closely tied to notions of the utility of information. Once an active search is underway, the best strategy is to seek first the information most useful to answering the question in the time allowed, but utility considerations also arise in formulating the thresholds at which monitors "go active". For example, with only a few pieces of information, learning an additional item on an indicators list may not change the quality of the match significantly. But at some point, learning an additional item makes each of the remaining items very significant, and "going active" at that point may well be the appropriate path. Because of this, the library of monitoring models represents in part models of the utility of information. This information is also used in the library of alerting models.

Alerting models describe criteria for deciding what to do with conditions detected by monitoring procedures; whom to notify, when to notify them, and how to notify them. Alerting models are essential since analysts have priorities among the conditions of interest to them, and normally wish to hear about the most urgent and important items right away, with the lesser items deferred for consideration later. Most of the work of alerting models occurs in describing the utility of different results to different agents at different times. These utilities often can be grouped into classes, and the library of alerting models provides templates for specifying utility ascriptions specific to particular alert consumers (human or machine).

We will build the library of alerting models on both extant procedures for making alerting decisions and on methods for convenient specification of utility information. The medical informatics literature contains an unsystematic variety of alerting procedures, but few tied to explicit notions of utility (see, for example, [24, 23, 38, 40, 41]). One element of this research will be to use explicit utility models to develop a systematic collection of alerting procedures that includes the ones already reported in the literature. We will also build on our past work [52, 14, 53, 16] on qualitative representation of utility information, which has developed logical languages that can express generic preferences ("prefer air campaign plans that maintain a center of gravity over those that distribute forces more widely"), and that relate this notion of preference to the notion of problem-solving or planning goals (interpreting goals as conditions preferred to their opposites, other things being equal). We will develop utility models that combine both qualitative preference information with approximate numerical models of common utility structures (e.g., utility models that increase up to some time and then drop off to model deadline goals, as in [18]), along with automatic procedures for combining such information into qualitative decision procedures and numerical multiattribute utility functions suitable for quick evaluation of alternatives.

We expect utility models to account for most of the variation in alerting models, though some variation can arise through the the sets of possible recipients and media used to communicate alerts. Selecting and tailoring utility models will be a key facility in making the monitoring system responsive to individual analysts, since the desired behavior will depend strongly on the utility of the particular conditions being monitored and on the context of other conditions and tasks faced by the analyst.

3.1.5 A security monitoring desk

We propose to implement a security monitoring desk providing mechanisms for command and control of monitoring processes, but not for controlling the enclave or system being monitored.

Making sensible changes in monitoring network structure or operation requires means for telling what the network is doing. This includes both observing the operational parameters of the monitoring processes and observing the data streams coursing through the network. To exercise such control, system developers and users use a special MAITA monitoring process called a "monitor of monitors" or "MOM". A MOM is used by system designers and users to create portions of the monitoring network and to call up process-specific “control panels” for changing operational parameters of the monitoring processes. A MOM can support one or more users at a time. It provides access to the monitoring libraries and means for setting in operation instances of library entries. It provides means for creating connections between operating monitoring processes (including external legacy or non-MAITA systems), and means for storing particular configurations of processes back in the library as new monitoring network descriptions. It allows users to create displays of the information flowing across different connections and of alerts generated by the monitoring processes.

We have exploited ubiquitous web browser technology to simplify the use of a MOM across different computing platforms. Specifically, we have implemented a basic MOM process as a combination of a Java application, which runs on a server machine, and a Java applet, which runs in a browser on possibly a different machine, and which communicates with the MOM server application. The MOM applet provides access to a variety of types of displays. Some of these are provided as part of the main applet (browser) window; others are created upon demand in separate windows.

3.1.5.1 Control actions: The main MOM display features a set of operation buttons and pull-down menus, a textual alert area, a textual help and response line, all organized around a pane providing a pictorial representation of the process-network structure. The user selects and deselects processes and connections depicted in the network window, and then uses buttons and menus to operate on the selected elements. Such operations include several categories of major operations in addition to convenience features like layout-editing actions that change the appearance or arrangement of the network elements in the network display window, including placement, sizes, fonts, and colors. Each of the windows created by the MOM is independently resizable and relocatable by the user.

Users or developers may use a MOM to create additional data connections or to change existing ones; a graphical display of the network topology is provided to facilitate such changes. The MOM also provides means for selecting library elements representing individual processes or network fragments and creating new instances of these in the operating network. In fact, the typical way of building up a monitoring network is to instantiate several processes or network fragments from a library, and then to connect these together.

Users or developers may use a MOM to change operational features of the monitoring processes. (This is in addition to the ways a process may change its behaviors in response to changes in its environment, as discussed below.) MOM-mediated changes may range from changing system parameters (thresholds, scale factors, tolerances, etc.) to complete reinitialization of the monitoring processes. Each type of monitoring process may have a customized “control panel”. A couple mouse clicks through the MOM’s displays brings up such control panels in separate windows.

Each MOM is charged with maintaining the operational status of the monitoring network. Toward this end, it maintains a persistent database giving information sufficient to recreate or restore the monitoring network, including itself and the user displays, when machines or communications networks fail. (This recreation extends only to the structure and operational parameters of the monitoring processes and displays; information internal to those processes is lost unless they store the information persistently themselves.) The MOM also maintains its own functionality by recording all important MOM information in a persistent database, and by setting up a subprocess to check periodically that the MOM is still operating. If a MOM failure is ever detected, the subprocess removes the defunct MOM and starts a new one.

3.1.5.2 Display actions: The MOM provides a variety of types of data displays for presenting different types of information. These fall into several categories: multivariate strip charts of selected streams; two dimensional maps of variables against each other, possibly over a background depicting a geographic map or image; text alerts; and combinations of these types. In addition to these and other visual display types, we expect future enhancements to the MOM to provide audio alerts as well, primarily synthesized speech alerts.

Multivariate strip charts display the values of one or more variables on a rectangular graph, with the displayed variables plotted vertically and time plotted horizontally. The MOM provides both single strips and combination strips, in which several individual strip charts are stacked one on top of the other with a common temporal reference on the horizontal. The MOM provides the ability to create combination and multivariate strip charts by selecting various connections or terminals in the process network diagram.

Two-dimensional (2D) maps display one or more paired variables, with one set of variables plotted on the vertical, and another set plotted on the horizontal. In a 2D map, time does not appear as a dimension of the graph axes. Instead, any temporal window appears only through the number of points plotted; as the window moves, excessively old points are removed from the display, and the new points are added.

Text alert displays simply list sentences, phrases, or words that constitute alerts to the user.

3.1.5.3 Library actions: The MAITA system permits system developers and users to construct monitoring networks and processes from a library of such, and facilitates acquisition of monitoring knowledge by providing the user with means for adding new entries to the library. The simplest means is to add new entries that describe the structure of a portion of the monitoring network operating at the current time. In this method, the user or developer operates on the displayed network topology to compose processes as desired and selects the portions to be encapsulated and stored as a new library entry. The more powerful method is based on conventional knowledge-base and ontology editors, as have been developed for a variety of knowledge representation systems. The MAITA system provides access to such tools through simple Java-coded editors that use the OKBC protocol to browse and edit a wide variety of knowledge representation systems.

The MAITA architecture supports a distributed system of monitoring libraries. While centralizing all libraries into one location makes it easier to maintain the coherence of the entries, it also increases system vulnerability and causes security problems, since classified information from many organizations might then be combined into a single place visible to a wide audience. Accordingly, we permit different enclaves to employ and modify their own copies of libraries, and provide simple mechanisms for propagating updates to libraries among the enclaves via explicit commands issued by the user through the MOM. We plan to expand on these simple mechanisms to formalize policies that specify what sorts of additions can and cannot be moved outside an enclave's own library.

3.1.6 Evaluation

Our innovative claims imply three major dimensions of assessment along which the work proposed here can be evaluated: time or effort needed to construct or extend a monitoring system for new settings or situations; the degree to which the libraries of monitoring knowledge cover IA tasks; and the performance of the systems so constructed. We plan to either perform such assessments ourselves or to provide information to others already engaged in assessment adequate to assess our progress as well. We discuss evaluation along these dimensions in reverse order.

DARPA is already conducting performance evaluations of a number of ID systems. We expect to seek evaluation of MAITA IA monitoring networks along the same terms, that is, accuracy (recall and precision) in recognizing and distinguishing intrusions. However, since our focus will be on constructing situation-sensitive monitoring procedures, the most pertinent performance evaluation involves comparing the accuracy of a basic network of situation-*in*sensitive monitors with the performance of an extended network including situation-sensitive monitors, or alternatively, comparing the accuracy of a single situation-sensitive network with the situation-sensitivity enabled or disabled.

Library coverage may be assessed in two ways. The simplest measure of IA library coverage is provided by periodic tabulation of the attack methods covered and not covered, which provide perhaps the characterizations of most direct interest to system security providers. A more interesting measure is derived from knowledge-base metrics of the sort being formulated in DARPA's HPKB program. Such metrics provide a measure of the growth of knowledge over time. A comprehensive library providing the methods and knowledge needed for every purpose would be characterized by having essentially the same library knowledge content both before and after constructing or adapting a range of monitoring systems. More precisely, the completeness may be measured by the amount of change in library knowledge apart from characterizations of the particulars of a new enclave's structure. (Of course, one could similarly measure the coverage of the library of organization descriptions, but we do not propose to seek any completeness in that portion of library.)

Assessing the time and effort costs of system construction, adaptation, and control may be done both formally and informally. Assessment methodology from the HPKB program is relevant here too. The formal assessment is in terms of time needed to perform representative modifications, control operations, or to construct a new system.

Since we do not propose to construct a large number of complete systems, the small number of data points provided by system-construction evaluations will preliminary but not final assessments. Formal assessments of effort may also be expressed in the same knowledge metrics described above for assessing library coverage. We expect to seek informal assessments of time and effort costs throughout the investigation as a guide to our design and development efforts.

3.2 Comparison to related work

The bodies of ongoing work most relevant to the proposed research are work on intrusion detection, process model libraries, and monitoring systems. We discuss each of these in turn.

Current systems for intrusion detection focus on developing two basic types of mechanisms, plus hybrid methods combining these. The first basic type is that of signature recognition, in which a specific pattern of events signals a possible intrusion. The second basic type is that of statistical anomaly detection, in which comparison of system, user, or operation statistics across different timescales and from different time periods identify the existence of potential intrusions. Methods such as these form the basis of the EMERALD system from SRI and the CMDS system from SAIC. For the purpose of this discussion, we divide intrusion detection systems into architecture and knowledge. We discuss these two aspects of systems separately.

Among intrusion detection systems, the EMERALD system has perhaps the architecture most similar to that used in the MAITA system. The basic EMERALD system provides a distributed set of monitoring processes or capabilities, organized hierarchically in a way that scales with the size of the enclave being protected. EMERALD monitoring processes have a standard form across all levels, one that combines a set of rules with a set of statistical patterns, and incorporates a uniform method by which one process may subscribe to the results of others. Our architecture provides similar capabilities, though allowing somewhat greater flexibility and expressiveness in prescribing the operation of the processes, and apparently allowing a more flexible set of communication mechanisms. Moreover, the EMERALD system is engineered to provide basic protections for its own operations at all levels. We have sought to focus our architecture on the content of the monitoring processes, and plan to piggyback on protections afforded by low-level processing supplied by Emerald or other IA systems. We have plans to extend our architecture to include basic protection mechanisms as these prove necessary in the third year of the proposed work.

The knowledge embodied by intrusion detection systems tends to focus on fairly low levels of signals. In the Emerald system, for example, rules tend to be simple and statistical methods tend to be prominent at the lowest levels of the monitoring hierarchy; at higher levels, rules become more complex, while the contributions of statistical methods diminish. The main bodies of knowledge for rules concern signature of varying degrees of complexity, but most represent fairly local considerations, rather than the wider situational awareness we seek to capture in the proposed monitoring knowledge libraries. The statistical models may be constructed manually or constructed and adapted mechanically, but most are based on fairly gross properties of the system being monitored. Our proposed work develops methods for basing monitors on more complicated statistical and probabilistic models, using Bayesian networks involving terms related to concepts in the situation knowledge base, and exhibiting situational dependence in which the probabilistic network used may itself be changed as the situation changes.

The primary body of work on process models for MAI tasks consists of the problem solving methods present in the CommonKADS library [1], which are also used in the Stanford Protege project and the ISI Expect project. The CommonKADS library contains a good range of abstract procedures for a number of generic tasks, but its coverage of methods for MAI tasks is very limited.

The CommonKADS project puts forward methods for "assessment" and diagnosis as the main methods for MAI tasks. While the methods included for these tasks are all important, they simply do not cover many of the important classes of MAI tasks. The CommonKADS notion of "assessment" consists of taking a "case" and "system description" as inputs and giving a "decision" as outputs. For example, a loan-fraud detection task would involve taking a completed application for a loan and deciding whether the application was legitimate or fraudulent. The CommonKADS library provides an array of different methods for such tasks, but none of these fit general MAI tasks very well. For example, to shoehorn intelligence analysis tasks into this framework requires interpreting the "case" as the current sum of knowledge, so there is no good sense in which one gets different cases, only the same case at different times. The KADS abstractions cover this, but the level of abstraction is much too high, and the CommonKADS library does not include specializations appropriate to the analysis task. Worse still, the "system description" is taken to be static, where in the analysis setting, what is considered to be abnormal or dangerous changes over time and with the new information coming in. That is, the case is the same as the system, and both change together. In addition, there is little or no structure to what is considered a case.

The CommonKADS library also includes more detailed procedures for diagnosis and prediction, but the procedures concern model-based diagnosis and other settings in which the monitor possesses complete or nearly complete information about the structure and intended behavior of the system being monitored. These assumptions are highly inappropriate for the more open-ended range of MAI tasks.

Essentially all of the important structure of MAI tasks lies outside of the extant CommonKADS assessment library, hence the attention of the proposed research to identifying and formalizing the needed extensions to this library.

Specific monitoring systems, as opposed to codifications of libraries of knowledge for constructing monitoring systems, are well represented in the literature and in commercial products. The most relevant work, other than our own, on monitoring knowledge and methods appears in the literature on trend detection and "temporal abstraction", especially in the work of Shahar [47] and Das [5] at Stanford. These efforts focus on representing temporal relationships and on methods for identifying patterns of temporal relationships as instances of more abstract events. This work provides a good foundation for MAI activities, but intelligent monitoring and analysis involve more than just temporal information. Structuring relevant sorts of non-temporal information, especially information about logical implication, statistical correlations, and causation, is crucial, but lacking in most abstraction-based treatments. Statistical trend detection, on the other hand, does not adequately exploit the constraints and structuring information that templates provide. We plan to design representations for monitoring conditions that integrate the best representations devised for each of these separate types of knowledge.

The Guardian project [24, 23] at Stanford has developed a highly dynamic programming environment for the construction of very flexible monitoring systems. It puts very strong emphasis on giving the system the ability to reason, during the monitoring process, about the most appropriate data collection, interpretation and integration strategies. It places correspondingly less emphasis on the ease of constructing relatively simpler monitoring strategies beforehand, and has not developed detailed libraries of monitoring modules to support easy assembly. In our proposed work, we intend the background knowledge about monitoring and about the domain and monitoring task to be used more at the time a monitoring process is assembled and configured, not dynamically during its execution. We believe that this approach will lead to more efficient monitoring systems and greater ease of their development and configuration.

Commercial technology for monitoring and control offers good models of some of the capabilities we seek, but does not offer the flexibility, modularity, or construction tools of interest here. The G2 system offered by Gensym Corporation provides a very good example. This system provides a good base of the "object-level" monitoring capabilities, namely the ability to accept inputs from several types of sources, a library of single-signal filters (linear extrapolation, fourier transforms, etc.), and a knowledge-based reasoning component for constructing multisignal analysis systems. While the library of single-signal filters and the primitives of the multisignal analysis language provide good starting points, they fail to cover some important types of knowledge (probabilities, causality). More importantly, G2 provides only a programming language, and not a structured library of procedures at various levels of specificity. Finally, G2 is structured as a heavyweight, stand-alone application, and does not provide the environment needed to support distributed efforts by multiple collaborating analysts. In G2, adding a new process to monitor some additional threat requires programming the new recognition procedure (without library support) and then recompiling and reinstalling the resulting overall monitoring process. For distributed, collaborative MAI efforts, what is needed instead is the ability to toss a new monitoring element into an ongoing process.

4 Relevant Capabilities

4.1 Previous accomplishments

Our research group has a long history of work in diagnostic and monitoring reasoning in the medical domain, and has made many contributions to complex probabilistic and heuristic reasoning, model-based reasoning, reasoning at multiple levels of abstraction, explanation generation, learning from experience, dealing with time-dependencies systematically, and modeling and using preferences in decision-making. We have also contributed to medical knowledge representation, knowledge representation in general, truth-maintenance, qualitative reasoning, and modeling repetitive decision-making. Most recently, we have developed the core knowledge-based monitoring technologies to be applied in this proposal as part of DARPA's High Performance Knowledge Base (HPKB) program, and have applied them to IA and battlefield situation awareness problems.

Our work on knowledge-based battlefield situation monitoring concerned the interpretation of moving-target indicator (MTI) radar tracks in the context of situational reports including known order-of-battle information, human intelligence reports, signal and electronics intelligence reports, and intelligence reports derived from photographic, infrared, and fixed-target radars. We constructed a network of monitoring processes operating over streams of intelligence reports from these sources that identified a large fraction of the military vehicles, sites, and movements. This network illustrated the use of situational reports in the identification of highly important targets, e.g., locations of displaced mobile anti-aircraft artillery sites. This work provided the basis for the MAITA architecture summarized earlier in this proposal.

Our HPKB work on IA problems is in its early stages, but through our participation in the IA/HPKB meeting held in Monterey, California this year and discussions with IA researchers, we have begun identifying close similarities between IA tasks and a number of the medical monitoring tasks we have addressed. We have also begun formulating some preliminary methods aimed at identifying possible inappropriate actions in the handling of medical records as a familiar proxy for IA tasks.

Our first diagnostic program that attempted to model reasoning processes related to the problems faced in MAI tasks was the "Present Illness Program" (PIP), reported in American J. Med. in 1976 [39]. The present illness problem requires dealing with a stream of often unrelated conditions or reports and trying to determine if there is a problem, and if so, what the problem is. The PIP adopted a hypothetico-deductive framework for diagnostic reasoning, using strong cues from the patient presentation to trigger hypotheses, both logical criteria and a pseudo-probabilistic scoring scheme to confirm or eliminate hypotheses, and explicit differential links to revise hypotheses when discrepant information arose. Later versions introduced a simple model of time, categorizing both patient data and a hypothesis-oriented time line along the dimension: past, recent-past, now, near-future, future. Our interest in temporal reasoning has continued through the doctoral work of Kohane [27, 26], exploring temporal constraints in diagnostic reasoning and Temporal Utility Package (TUP); Russ [42, 43, 44, 45], who designed a control structure that supports reasoning about unreliable streams of time-oriented data and applied it to diabetic ketoacidosis; and Haimowitz [21, 20], who studied trend detection in pediatric growth data and in ICU monitoring in the $\mathrm{TrenD_x}$ system [31, 25, 17].

We have substantial experience in implementing monitoring and analysis environments. In 1991, Dr. Kohane completed the implementation of an on-line medical chart (the Clinician's Workstation–CWS) [33, 28, 32, 29] for the Division of Endocrinology at Children's Hospital. This system has now been in full operation for 5 years and provides on-line access to clinic notes, clinic measurements, demographics, pharmacy data, laboratory results, problem lists and reports from ancillary departmental systems (e.g., radiology) to several clinical divisions at Children's Hospital. Dr. Kohane also designed and led the implementation of a data integration and display system for the Multidisciplinary Intensive Care Unit, and more recently has led development of the W3-ICU web-based ICU monitoring system.

We have a long-term participation in knowledge representation efforts. Hawkinson and Szolovits worked in the mid-1970's on the OWL [49] and BrandX [50] representation schemes that provided great flexibility and opportunities to exploit linguistic analogies but suffered from a lack of semantic rigor. When current more restrictive KR systems were built in the 1980's, we tried to use KL/ONE to represent medical knowledge and found that too much expressive ability had been sacrificed for semantic cleanliness and computational efficiency [19, 22]. Doyle and Patil produced a major and influential critique of this trend for the KR community [13].

Doyle's continuing work on truth maintenance and nonmonotonic reasoning [6, 37, 15, 8, 10, 9, 12] has been complemented in recent years by studies with Michael Wellman (now on the faculty at University of Michigan) of qualitative representations of preference information [52, 14, 53, 16], by studies of the use of economic mechanisms in controlling distributed reasoning and activities [7, 11, 12], and by work on constructing ontologies for plans and the process of planning. The ontology research has been conducted in conjunction with the ARPI Planning Ontology Construction Group.

We are engaged in a number of projects that exploit the revolutionary capabilities of the World Wide Web (W3) in innovative ways. Our W3-EMRS project [34, 30] reengineers electronic medical record systems to use the distributed, multi-platform capabilities of the W3 to build more effective, more flexible, more secure and cheaper to implement record systems. In addition, this project is building virtual records that integrate health information from multiple institutions to reconstruct a patient's longitudinal health history from fragments stored at different hospitals, health centers, doctors' offices, etc. A related project [51] uses similar mechanisms to distribute real-time data via W3, to allow remote monitoring of patients in intensive care from any authorized remote site. Our Guardian Angel project [48] is developing personal health information systems that help patients at home manage significant aspects of their own health care, maintain records on their condition, treatments and responses, communicate with health care providers, and access educational resources that help them understand their conditions, all via the W3.

Long's Heart Disease program (HD) [Long92, Long92a, Long94], addresses the complex treatment of patients with heart failure, providing both a diagnostic and therapy planning component. Diagnosis is based on an approximate probabilistic method that works over a network of clinically-significant causal concepts, and therapy prediction is based on predicting the influence of possible interventions in a complex feedback system by using signal-flow analysis techniques. HF has proven to be quite effective at diagnosis in certain subdomains, and remains under active development to augment its diagnostic acumen and to further develop and test its therapeutic side. Current work includes the creation of W3-based interfaces that allow cases to be entered anywhere in the world and analytical results returned to the widespread community of users.

Former and current students have developed modeling and analysis methods for time-oriented data that are directly relevant to the proposed project. Dr. Tze-Yun Leong, now a professor at Singapore's National University, developed methods of modeling recurring decisions using semi-Markov decision processes [35]. Milos Hauskrecht, now of Brown University, completed a doctoral thesis on the efficient analysis of partially-observable Markov decision processes. Yao Sun, MD, is pursuing his PhD studies and has implemented closed-loop controllers for ventillating infants based on fuzzy control algorithms. Alex Yeh [54] (now at MITRE Corp.) and Elisha Sacks [46] (now a professor at Purdue Univesity) have both developed methods for the analysis of dynamic systems, especially those with repetitive behavior.

4.2 Key Personnel

Jon Doyle, Ph.D., will devote between 50% and 100% of his time to this effort in different years of the covered period. His principal research goal is to develop theories and techniques for representation and reasoning that have a sound basis in decision theory, economics, and logic, and to apply these to practical problems of planning and medical informatics, especially to the representation and use of qualitative and quantitative models of preferences and utilities. He has published widely on the roles that economic notions play in the structure of reasoning and representations, and together with his students, has conducted investigations into means for mechanizing rational reasoning. Dr. Doyle has made many contributions to the theory of rational and economic reasoning, has developed representations for rational agents and market-guided reasoning systems, and has worked on structuring ontologies for planning, the process of planning, and general medical health-maintenance monitoring, all under previous and current DARPA funding. He currently serves as PI of our work on the MAITA system. He is a Fellow and member of the Executive Council of the American Association for Artificial Intelligence, as well as a director of Principles of Knowledge Representation and Reasoning, Inc., which organizes the KR conferences.

Isaac S. Kohane, M.D., Ph.D., will devote 20% of his time to this effort in each contract year through a subcontract to Childrens Hospital. Dr. Kohane is Director of the

Children's Hospital Informatics Program and Assistant Professor of Pediatrics at Harvard Medical School. He is Principal Investigator on several information technology projects funded by the National Institutes of Health and the National Institutes of Standards and Technologies, notably the W3-EMRS project which provides real-time access to multiple heterogeneous clinical databases. Dr. Kohane has also led several projects on real-time trend detection and closed-loop control of physiological systems in the domain of critical care. Dr. Kohane has been a collaborator in the development of $\mathrm{TrenD_x}$, one of the trend detection techniques we will be applying in the proposed project. He currently serves as a key researcher on the MAITA project.

William Long, Ph.D., will devote 20% of his time to this effort in each covered year. He is involved in research in causal and temporal reasoning. This work over the past dozen years has been focused on the cardiology domain and the diagnosis of heart disease. This is a rich domain for causal reasoning because the underlying mechanisms take from seconds to years and the challenge is to fit the findings into a consistent, plausible scenario in time. The strategies for generating and evaluating such causal hypotheses will be useful in the proposed work. Dr. Long has also been involved in the use of classification trees and neural networks for the development of classification tools from data sets. While this was also done in a medical context (detection of cardiac ischemia in the emergency room), the methodology is applicable in a wide range of domains. The third area of expertise is the detection of trends. This work has been carried out in the context of managing therapy, both the adjustment of digitalis and the management of ventricular arrhythmias. Dr. Long is a Fellow of the American College of Medical Informatics. He currently serves as a key researcher in the MAITA project.

Peter Szolovits, Ph.D., will devote 15% of his time to this effort in each contract year. He has worked on problems of knowledge representation, reasoning under uncertainty, and diagnostic and therapeutic planning and monitoring, mostly in applications to medical decision making. He and his students pioneered diagnostic reasoning methods that rely on detailed models of causality and temporal relationships among aspects of a hypothesized disorder, and investigated multi-level reasoning systems that pursue a simple analysis of a problem when all data consistently indicate a single solution but that engage in much more detailed analyses when discrepancies arise between data and expectations. Prof. Szolovits is currently participating in the development of a new architecture for medical record systems that exploit the technologies of the World Wide Web to support sharing and commonality of access to records from multiple institutions, and is engaged in building life-long active patient-centered health information systems that orient medical information processing, decision making, health and treatment monitoring, task-specific education and communication around the individual patient. This project, called Guardian Angel, was begun with DARPA support [48]. Experience from these efforts motivates and contributes to the design of the present proposal, and advances in the ability to support MAI tasks will greatly contribute to the future of these projects as well as to applications in the challenge domains. Prof. Szolovits is a fellow of the American Association for Artificial Intelligence and of the American College of

Medical Informatics. He currently serves as co-PI of the MAITA project.

References

[1] J. A. Breuker and W. Van de Velde, editors. *The CommonKADS Library for Expertise Modelling.* IOS Press, Amsterdam, 1994.

[2] C. Cao, J. Doyle, I. Kohane, W. Long, and P. Szolovits. The MAITA monitoring library and language. In preparation, 1998.

[3] C. Cao, J. Doyle, I. Kohane, W. Long, and P. Szolovits. The MAITA monitoring network architecture. In preparation, 1998.

[4] C. Cao, J. Doyle, I. Kohane, W. Long, and P. Szolovits. The MAITA system: an overview. In preparation, 1998.

[5] A. K. Das and M. A. Musen. A comparison of the temporal expressiveness of three database query methods. In *Nineteenth Annual Symposium on Computer Applications in Medical Care*, pages 331–337, New Orleans, LA, 1995.

[6] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12(2):231–272, 1979.

[7] J. Doyle. Rationality and its roles in reasoning. *Computational Intelligence*, 8(2):376–409, 1992.

[8] J. Doyle. Reason maintenance and belief revision: Foundations vs. coherence theories. In P. Gärdenfors, editor, *Belief Revision*, pages 29–51. Cambridge University Press, Cambridge, 1992.

[9] J. Doyle. Final report on rational distributed reason maintenance for planning and replanning of large-scale activities. Submitted to Rome Laboratory, available via http://www.medg.lcs.mit.edu/doyle, Oct. 1994.

[10] J. Doyle. Reasoned assumptions and rational psychology. *Fundamenta Informaticae*, 20(1-3):35–73, 1994.

[11] J. Doyle. A reasoning economy for planning and replanning. In M. H. Burstein, editor, *Proceedings of the 1994 ARPA/Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, pages 35–43, San Francisco, CA, 1994. Morgan Kaufmann.

[12] J. Doyle. Toward rational planning and replanning: rational reason maintenance, reasoning economies, and qualitative preferences. In A. Tate, editor, *Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative*, pages 130–135. AAAI Press, Menlo Park, California, 1996.

[13] J. Doyle and R. S. Patil. Two theses of knowledge representation: Language restrictions, taxonomic classification, and the utility of representation services. *Artificial Intelligence*, 48(3):261–297, Apr. 1991.

[14] J. Doyle, Y. Shoham, and M. P. Wellman. A logic of relative desire (preliminary report). In Z. W. Ras and M. Zemankova, editors, *Methodologies for Intelligent Systems, 6*, volume 542 of *Lecture Notes in Artificial Intelligence*, pages 16–31, Berlin, Oct. 1991. Springer-Verlag.

[15] J. Doyle and M. P. Wellman. Rational distributed reason maintenance for planning and replanning of large-scale activities. In K. Sycara, editor, *Proceedings of the DARPA Workshop on Planning and Scheduling*, pages 28–36, San Mateo, CA, Nov. 1990. Morgan Kaufmann.

[16] J. Doyle and M. P. Wellman. Representing preferences as *ceteris paribus* comparatives. In S. Hanks, S. Russell, and M. P. Wellman, editors, *Proceedings of the AAAI Spring Symposium on Decision-Theoretic Planning*, 1994.

[17] J. Fackler, I. J. Haimowitz, and I. S. Kohane. Knowledge-based data display using trendx. In *AAAI Spring Symposium: Interpreting Clinical Data*, Palo Alto, 1994. AAAI Press.

[18] P. Haddawy and S. Hanks. Representations for decision-theoretic planning: Utility functions for deadline goals. In B. Nebel, C. Rich, and W. Swartout, editors, *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, pages 71–82, San Mateo, CA, 1992. Morgan Kaufmann.

[19] I. J. Haimowitz. Using NIKL in a large medical knowledge base. TM 348, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, Jan. 1988.

[20] I. J. Haimowitz and I. S. Kohane. Automated trend detection with alternate temporal hypotheses. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 146–151, Chambery, France, 1993.

[21] I. J. Haimowitz and I. S. Kohane. An epistemology for clinically significant trends. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 176–181, Washington, DC, 1993.

[22] I. J. Haimowitz, R. S. Patil, and P. Szolovits. Representing medical knowledge in a terminological language is difficult. In *Symposium on Computer Applications in Medical Care*, pages 101–105, 1988.

[23] B. Hayes-Roth, S. Uckun, J. E. Larsson, J. Drakopoulos, D. Gaba, J. Barr, and J. Chien. Guardian: An experimental system for intelligent ICU monitoring. In *Symposium on Computer Applications in Medical Care*, Washington, DC, 1994.

[24] B. Hayes-Roth, R. Washington, D. Ash, R. Hewett, A. Collinot, A. Vina, and A. Seiver. Guardian: A prototype intelligent agent for intensive-care monitoring. *Journal of AI in Medicine*, 4:165–185, 1992.

[25] I. Kohane and I. Haimowitz. Hypothesis-driven data abstraction. In *Symposium on Computer Applications in Medical Care*, Washington, DC, 1993.

[26] I. S. Kohane. Temporal reasoning in medical expert systems. In R. Salamon, B. Blum, and M. Jørgensen, editors, *MEDINFO 86: Proceedings of the Fifth Conference on Medical Informatics*, pages 170–174, Washington, Oct. 1986. North-Holland.

[27] I. S. Kohane. Temporal reasoning in medical expert systems. TR 389, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, Apr. 1987.

[28] I. S. Kohane. Maintaining alternate interpretations of data from multiple sources in a clinical event monitoring system. In *MEDINFO 92: Proceedings of the Seventh Conference on Medical Informatics*, pages 483–489, Geneva, Switzerland, 1992.

[29] I. S. Kohane. Getting the data in: Three-year experience with a pediatric electronic medical record system. In J. G. Ozbolts, editor, *Symposium on Computer Applications in Medical Care*, Washington, DC, 1994. To appear.

[30] I. S. Kohane, P. Greenspun, J. C. Fackler, C. Cimino, and P. Szolovits. Building national electronic medical record systems via the world wide web. *Journal of the American Medical Informatics Association*, 3(3), 1996.

[31] I. S. Kohane and I. J. Haimowitz. Encoding patterns of growth to automate detection and diagnosis of abnormal growth patterns. *Pediatric Research*, 33:119A, 1993.

[32] I. S. Kohane, R. Lauerman, L. R. First, and J. A. Majzoub. Knowledge-based automated growth monitoring. *Pediatric Research*, 31:124A, 1992.

[33] I. S. Kohane and D. P. McCallie. A dynamically reconfigurable clinician's workstation with transparent access ot remote and local databases. In *AMIA*, 1990.

[34] I. S. Kohane, F. J. van Wingerde, J. C. Fackler, C. Cimino, P. Kilbridge, S. Murphy, H. Chueh, D. Rind, C. Safran, O. Barnett, and P. Szolovits. Sharing electronic medical records across multiple heterogeneous and competing institutions. In J. J. Cimino, editor, *Proceedings 1996 AMIA Annual Fall Symposium*, pages 608–612, Philadelphia, PA, Oct. 1996. American Medical Informatics Association, Hanley and Belfus.

[35] T.-Y. Leong. *An Integrated Approach to Dynamic Decision Making under Uncertainty.* PhD thesis, MIT, Cambridge, MA, May 1994. Laboratory for Computer Science Technical Report MIT/LCS/TR-631.

[36] W. J. Long, H. Fraser, and S. Naimi. Web interface for the heart disease program. In J. J. Cimino, editor, *Proceedings 1996 AMIA Annual Fall Symposium*, pages 762–766, Philadelphia, PA, Oct. 1996. American Medical Informatics Association, Hanley and Belfus.

[37] D. McDermott and J. Doyle. Non-monotonic logic—I. *Artificial Intelligence*, 13:41–72, 1980.

[38] S. M. Ornstein, D. R. Garr, R. G. Jenkins, P. F. Rust, and A. Arnon. Computer-generated physician and patient reminders. *Journal of Family Practice*, 32:82–90, 1991.

[39] S. G. Pauker, G. A. Gorry, J. P. Kassirer, and W. B. Schwartz. Towards the simulation of clinical cognition: Taking a present illness by computer. *American Journal of Medicine*, 60:981–996, 1976.

[40] D. M. Rind, C. Safran, R. S. Phillips, W. V. Slack, D. R. Calkins, T. L. Delbanco, and H. L. Bleich. The effect of computer-based reminders on the management of hospitalized patients with worsening renal function. In P. Claytons, editor, *Proceedings Symposium Computer Applications in Medical Care*, pages 28–32, Washington, DC, 1991. McGraw-Hill.

[41] D. M. Rind, C. Safran, R. S. Phillips, Q. Wang, D. R. Calkins, T. L. Delbanco, H. L. Bleich, and W. V. Slack. Effect of computer-based alerts on the treatment and outcomes of hospitalized patients. *Archives of Internal Medicine*, 154:1511–1517, 1994.

[42] T. A. Russ. Temporal control structure reference manual. TM 331, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, June 1987.

[43] T. A. Russ. Using hindsight in medical decision making. In *Symposium on Computer Applications in Medical Care*, 1989. Also published in *Computer Methods and Programs in Biomedicine*, 32(1): 81–90, May 1990.

[44] T. A. Russ. Using hindsight in medical decision making. *Computer Methods and Programs in Biomedicine*, 32(1):81–90, May 1990. Also published in *Proceedings of the Symposium on Computer Applications in Medical Care*, pp. 38–44, 1989.

[45] T. A. Russ. *Reasoning with Time Dependent Data*. PhD thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, Cambridge, MA, Sept. 1991. To appear as MIT/LCS/TR-545.

[46] E. P. Sacks. Automatic analysis of one-parameter planar ordinary differential equations by intelligent numerical simulation. *Artificial Intelligence*, 48(1):27–56, 1991.

[47] Y. Shahar. *A Knowledge-Based Method for Temporal Abstraction of Clinical Data*. PhD thesis, Stanford University, Stanford, CA, 1994. Available as Computer Science Department report CS-TR-94-1529.

[48] P. Szolovits, J. Doyle, W. J. Long, I. Kohane, and S. G. Pauker. Guardian Angel: Patient-centered health information systems. Technical Report MIT/LCS/TR-604, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, May 1994.

[49] P. Szolovits, L. Hawkinson, and W. A. Martin. An overview of the OWL language for knowledge representation. In *Proceedings of the Workshop on Natural Language Interaction with Databases*, Schloss Laxenburg, Austria, Jan. 1977. Also appeared as MIT/LCS/TM-86(1977).

[50] P. Szolovits and W. A. Martin. BRAND X: Lisp support for semantic networks. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 940–946, 1981.

[51] K. Wang, I. Kohane, K. L. Bradshaw, and J. Fackler. A real time patient monitoring system on the world wide web. In J. J. Cimino, editor, *Proceedings 1996 AMIA Annual Fall Symposium*, pages 729–732, Philadelphia, PA, Oct. 1996. American Medical Informatics Association, Hanley and Belfus.

[52] M. P. Wellman and J. Doyle. Preferential semantics for goals. In *Proceedings of the National Conference on Artificial Intelligence*, pages 698–703, 1991.

[53] M. P. Wellman and J. Doyle. Modular utility representation for decision-theoretic planning. In *Proceedings of the First International Conference on AI Planning Systems*, 1992.

[54] A. S. Yeh. PLY: A system of plausibility inference with a probabilistic basis. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1983.