Flexible Automated Monitoring and Notification for Complex Processes using KARMEN

Larry Bunch, Maggie Breedy, Jeffrey M. Bradshaw, Marco Carvalho, David Danks, & Niranjan Suri

Abstract— **Automated monitoring for complex systems such as the space shuttle fueling and launch process can increase the effectiveness of operators in detecting abnormal conditions before become hazardous to the point they may compromise the mission. We have developed the KAoS Reactive Monitoring and Event Notification (KARMEN) multi-agent system to allow users to describe and change monitoring conditions at any time in order to effectively monitor such processes and appropriately notify key operations personnel about off-nominal conditions. These notification actions are overseen by policies that contain process requirements and users preferences.**

Index Terms—**KAoS, KARMEN, OWL, Monitoring, Multi-Agent System, Notification**

I. INTRODUCTION

SAFETY remains one of NASA's primary concerns in all aspects of hydrogen production, storage, delivery, and use. The Institute for Human and Machine Cognition (IHMC) has developed an innovative process monitoring system for the space shuttle fueling and launch process as well as planned liquid hydrogen production facilities at the Kennedy Space Center (KSC). We have taken a human-centered approach to monitoring automation that complements the engineer's ability to identify relevant monitoring contexts with the software agent's ability to rapidly and vigilantly assess the process state.

Our KAoS Reactive Monitoring and Event Notification (KARMEN) multi-agent system enables any local or remote user to describe a set of process conditions to a personal software agent, deploy the agent to monitor the process as it runs, and have the agent notify the user appropriately as the conditions change. This research is advancing the state of the art for automated process monitoring. The main goal is improving the awareness and responsiveness of operational personnel concerning abnormal conditions that may compromise safety or efficiency of complex systems.

We have built upon the foundation of a flexible agent networking and security architecture (FlexFeed), a rich declarative ontology represented in the powerful W3C standard Web Ontology Language (OWL), comprehensive reasoning tools, and state-of-the-art policy and domain services (KAoS). The integration of these tools provides a solid base for effective agent configuration, deployment, communication, and control. The KARMEN process monitoring tools include a graph-based interface for users to describe process conditions to personal agents concerning individual sensors or ontological classes of sensors. The tools also provide adaptive multi-modal user notification including graphical interfaces for the live display of agent status and communications.

II. MOTIVATION AND RELATED WORK

Early detection of abnormal conditions can help engineers and operators prevent costly or hazardous results in complex processes [1, 2]. With over 40,000 sensors in systems such as the space shuttle, manually monitoring for precursors to failure quickly becomes intractable for operators. Current methods for system safety and health monitoring typically involve alarms based on predefined limits for individual sensors or post-analysis of recorded operational data. Each of these approaches has both advantages and limitations in the control of complex systems. Moreover, both approaches assume that we have in hand a well-understood, accurate model of the system. We are combining the power and flexibility of the post-analysis approach with the real-time responsiveness of commercial alarm systems. Software agents provide an excellent mechanism to deliver these capabilities in an agile, personalized, and effective manner.

Other researchers have applied multi-agent systems to monitoring and diagnosis of chemical and industrial processes [3, 4]. These systems have focused primarily on the important task of autonomously diagnosing sensor state using agents to model system components. The KARMEN system is distinguished by its human-centered approach of providing tools to create personal monitoring agents with rich semantic descriptions of process state and salient user notification. Our agent-based tools complement and amplify the expertise of the process engineers and operators by enabling them to create and refine personally relevant assessments of current process conditions. Although future plans include developing autonomous monitoring for KARMEN, the focus of this research is not diagnosis, but rather collaborating with users to keep them informed about aspects of the process that are specifically relevant to each person.

Manuscript received October 1, 2004. This work was supported in part by the Hydrogen Research at Florida Universities program sponsored by the National Aeronautics and Space Administration grant number NAG-3-2751.

Larry Bunch is with the Institute for Human and Machine Cognition, Pensacola, FL 32502 USA (phone: 850-202-4475; fax: 850-202-4440; e-mail: lbunch@ihmc.us).

Maggie Breedy, Jeffrey M. Bradshaw, Marco Carvalho, David Danks, and Niranjan Suri are also with the Institute for Human and Machine Cognition, Pensacola, FL 32502 USA (e-mail: {mbreedy, jbradshaw, mcarvalho, ddanks, nsuri}@ihmc.us).

0-7803-8125-4/03/$17.00 © 2005 IEEE

III. FOUNDATION

KARMEN is built upon multi-agent frameworks and services for networking, reasoning, security, and control. We also rely on simulations to develop and experiment with the system. For the space shuttle launch process, we use trace-based simulations provided by KSC. This is a playback of actual data recorded during previous shuttle launches. For chemical processes, we run dynamic model-based simulations in AspenTech's Hysys environment [5] and interface these with the Emerson DeltaV commercial control system [6] to create a virtual plant. Both of these simulated environments support distributed processing for evaluating our multi-agent system. A primary objective for 2005 is to begin using KARMEN on live systems at KSC.

A. FlexFeed Agent Networking Framework

FlexFeed is a Java agent framework that provides efficient communication and resource distribution for networks of collaborating agents [7, 8]. KARMEN agents and user displays rely on the FlexFeed API for mobile deployment and communication among heterogeneous sensor, intermediate, and user nodes. FlexFeed also supports information release policies between agents. The transport mechanism, message distribution, and filtering are each handled at the framework level, hiding these implementation details from the data producers and consumers. This architecture allows the framework to transparently customize the routing and transformation data streams while abstracting from the agent the tasks associated with the protocol selection, policies, and load balancing. Multiple communication protocols and lookup services can coexist in the network and FlexFeed will determine what protocols to use in order to distribute messages between any two nodes. This API provides two main components: the FlexFeed Manager, that handles agent lookup and delivery of data, and the FlexFeed Coordinator that distributes processing load and bandwidth consumption across the framework preserving the resources on the nodes. The coordinator also ensures that the data feeds between agents are authorized using KAoS policies, as described in the next section. To interact with other agents via the framework, an agent registers itself either as a sink node (one that will potentially work as a client of data feeds), or as a source node (sensor or source of data feeds), or as both (e.g., a relay or transformation node). After the nodes are registered, they can interact directly through message passing, and listener interfaces. All the routing and policy constraints are enforced by the framework, and can be changed and distributed at run-time.

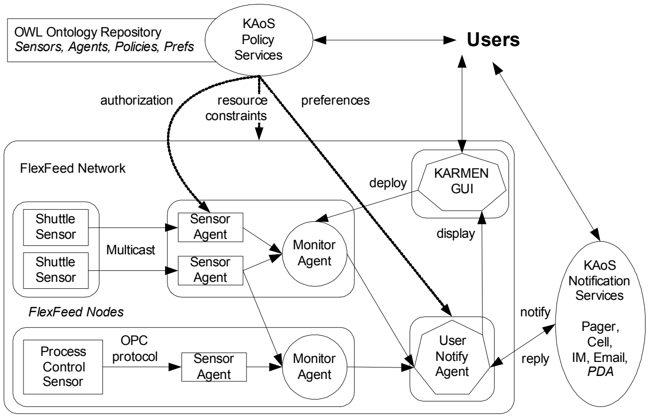


Fig 1. A high-level diagram that describes the relationships among the KARMEN system components.

B. KAoS Policy Services

KAoS is a collection of componentized services compatible with several popular software agent and robotic frameworks, as well as traditional distributed services platforms (e.g., CORBA, Web Services, Grid Computing) [9-11]. In the context of KARMEN, KAoS policy and domain services are used to define, manage, deconflict, and enforce policies that provide secure agent access to sensor data and govern the mode of notification to users. The KAoS Policy Ontologies (KPO; http://ontology.ihmc.us/) are represented in W3C standard Web Ontology Language (OWL; http://www.w3c.org/TR/owl-features/) [12]. KAoS relies on an integrated theorem prover along with KAoS-specific extensions to support representation and reasoning about policies.

The current version of KPO defines basic ontologies for actions, actors, groups, places, various entities related to actions (e.g., computing resources), and policies. As the application runs, classes and individuals corresponding to new policies and instances of application entities are also transparently added and deleted as needed. Through various property restrictions, a given policy can be appropriately scoped to various domains, for example, either to individual agents, to agents of a given class, to agents belonging to a particular group, or to agents running in a given physical place or computational environment. Additional aspects of the action context can be precisely described by restricting values of its properties. Groups of people, agents, and resources are also structured into ontologies to facilitate policy administration.

C. OWL Ontology Representation and Reasoning

Our system employs the OWL to organize and classify sensors, monitoring states, notification modes and salience, as well as users and organizational roles. OWL is a powerful description logic-based language developed for the semantic web. It provides vocabulary for describing properties and classes including relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, rich typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. Combined with the reasoning capability of Stanford's Java Theorem Prover (JTP; http://www.ksl.stanford.edu/software/jtp/), these ontologies enable users to effectively describe sophisticated monitoring conditions in a way that is accessible to agents. To make the use of OWL simple to non-specialist users, a number of graphical user interfaces have been defined (see below).

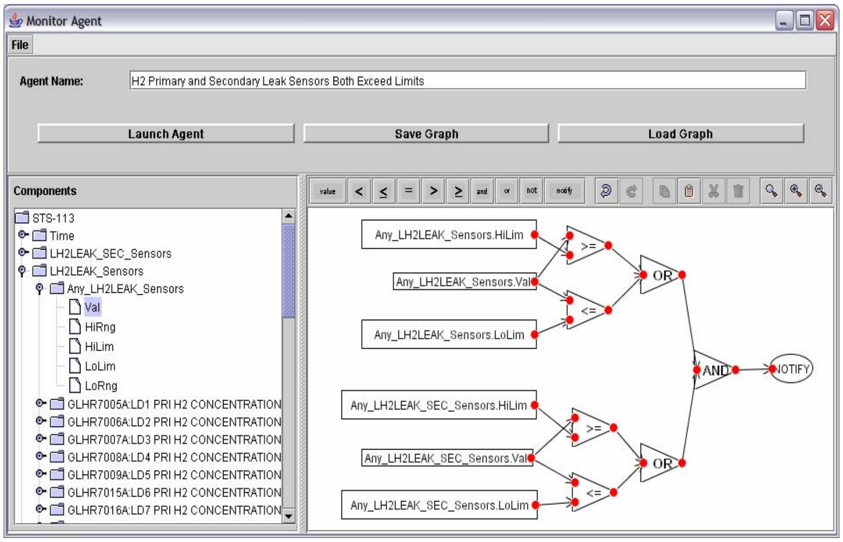
IV. PROCESS MONITORING

The health monitoring and process control of a complex system involves an extensive network of sensors, processors, and actuators. The elements of this process intranet may be linked together wirelessly or by more conventional means. In either case, unique opportunities for process control and health monitoring are offered by software agents that can migrate within the network to accomplish tasks specified by the human operators. Conceptually, a number of collaborating agents seek, collect, and evaluate data from individual sensors, interacting with other agents to form a composite picture of system state, and interacting with the human operators to provide information that is critical for system safety. The mobility of such agents (their ability to migrate within the system as required to accomplish tasks) introduces a previously unavailable degree of flexibility in the development of safety and health monitoring systems. The FlexFeed framework allows us to deploy and migrate agents in a way that optimizes bandwidth, power, processing, and other resources within the network.

Some key challenges in monitoring automation include enabling users to easily describe conditions of interest to the monitoring software, allowing users to dynamically change the conditions being monitored without affecting the process control, automatically changing the monitored conditions in response to changes in the process state, efficiently evaluating the system state for the given conditions, and effectively communicating the process state back to the user.

A. *Describing Process Conditions*

KARMEN users define process conditions for agents using a graph-based tool to build expressions concerning process state as shown in figure 2. Users browse for individual sensors or classes of sensors and add these inputs to the graph. Nodes are then selected to compare, combine, and transform these sensor inputs into a logical expression. When the user launches the agent, each sub-expression can be assigned to an existing agent in the FlexFeed network for evaluation or new agents be created as needed.



Agent Name: H2 Primary and Secondary Leak Sensors Both Exceed Limits

Fig 2. Users construct graphical expressions describing process conditions for agents to monitor.

One particularly valuable aspect of the research involves enabling users to monitor complex and aggregate process conditions that could not previously be monitored at runtime. Defining *ontologies of process variables* in OWL enables users to organize and classify sensors by relevant properties to easily express complex monitoring conditions for groups of related sensors (e.g. monitor for *any sensor value* from *shuttle main engine one* that *exceeds 90%* of its *associated high alarm limit*). Using ontological classes in monitoring expressions allows users to define complex aggregate conditions concisely. For example, the class of "all *sensors* on *main engine one* with a *high limit value*" can be constructed in the ontology based on the common properties of individual sensors such as location and limits. Such an ontology class can then be incorporated into a monitoring expression such as "sensor current value greater than 90% of sensor's high limit". This allows users to define conditions at a variety of scopes from the very narrow and specific to system-wide.

B. *Monitoring Capabilities*

The most basic capabilities of the process monitoring agents for this system include comparing process variables to scalar values and other process variables (e.g. monitor for a *valve's actuator position greater than its predefined high alarm limit*). We effectively extend the alarm functionality commercial control systems provide with the added value of making this capability available for ad-hoc and remote use. The ability to inject new conditions non-intrusively into an operational environment is critical. We can further incorporate monitoring statistical summaries of sensor behavior including standard deviation, variance, mean, and rate of change over a given time period or number of samples. Users can also employ mathematical expressions to derive new aggregate conditions (e.g., monitor the *product of pressure and temperature sensor readings*), annotate process variables such as defining progressive high and low limits, and access *system annotations* such as maximum, minimum, and average observed values from historical data. These feature support live, flexible monitoring using new combinations of parameters not inherent in the control system.

Adding remote monitoring capabilities carries the responsibility to control access to sensitive data. The KAoS policy services leverage the ontologies defined for classifying sensors to also define and enforce authorization policies that restrict access to process data such as "*IHMC personnel* can only access sensors in the *shuttle main engine* class" which will deny authorization to access feeds from these sensors to all agents created by IHMC personnel. Such policies could also describe reductions of sampling rate or precision which the agents would enforce.

V. USER NOTIFICATION

Notifications are generated as the monitored conditions obtain and abate. The mode, salience, and recipients of each notification are governed by KAoS policies representing organizational requirements and users' personal preferences. Notification modes may include E-mail, Instant Message, Pager, and Operator Displays. Notification policies typically cover such factors as event type, severity, the recipient's organizational role and presence, and the plant area in which the event occurred. For example, a policy might be to "page an onsite Field Operator immediately when a critical H2 Plant monitored condition is satisfied and the Process Engineer is unavailable". The selection of mode, recipients, and salience is made at runtime based on information gathered about the user's presence and the modes available (e.g. the user's instant message client indicates user is available and the user's schedule indicates she is onsite).

The default behavior of the Notification agent is to display messages in the KARMEN application interface. All other notification actions are governed by KAoS policies representing organizational requirements and personal preferences. Each policy obliges the notification agent to take certain actions based on the qualities of the monitored event and the current disposition of the concerned personnel. We have developed a set of initial ontologies depicted in figure 3 for notification that draws heavily on the work of Schrekenghost and colleagues [13].

The current event characteristics that can trigger a policy include the event type (satisfied/unsatisfied condition, activated/deactivated alarm: see ConditionStatus in figure 3),

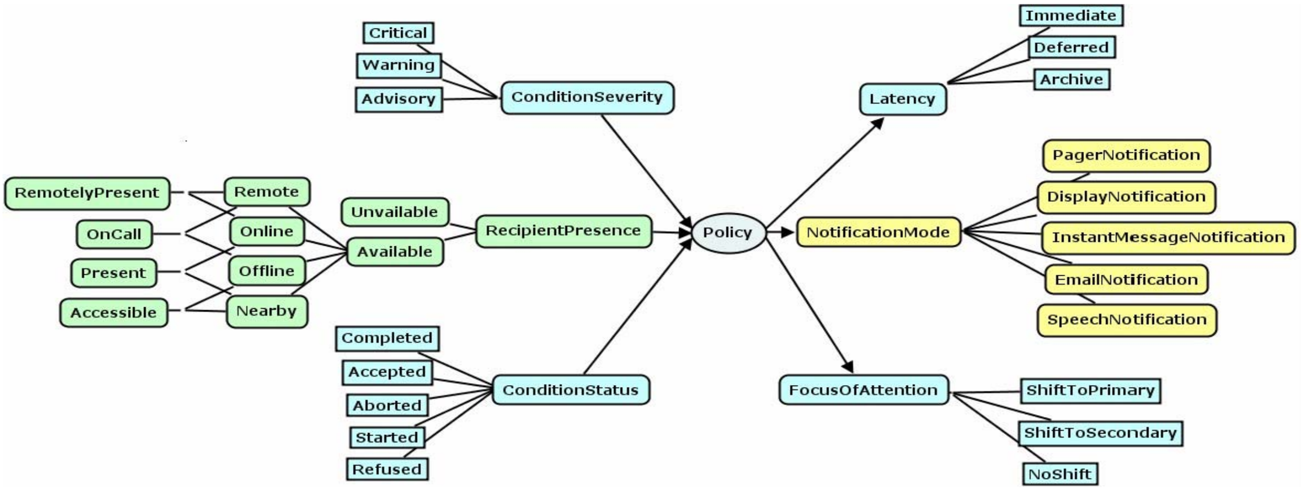
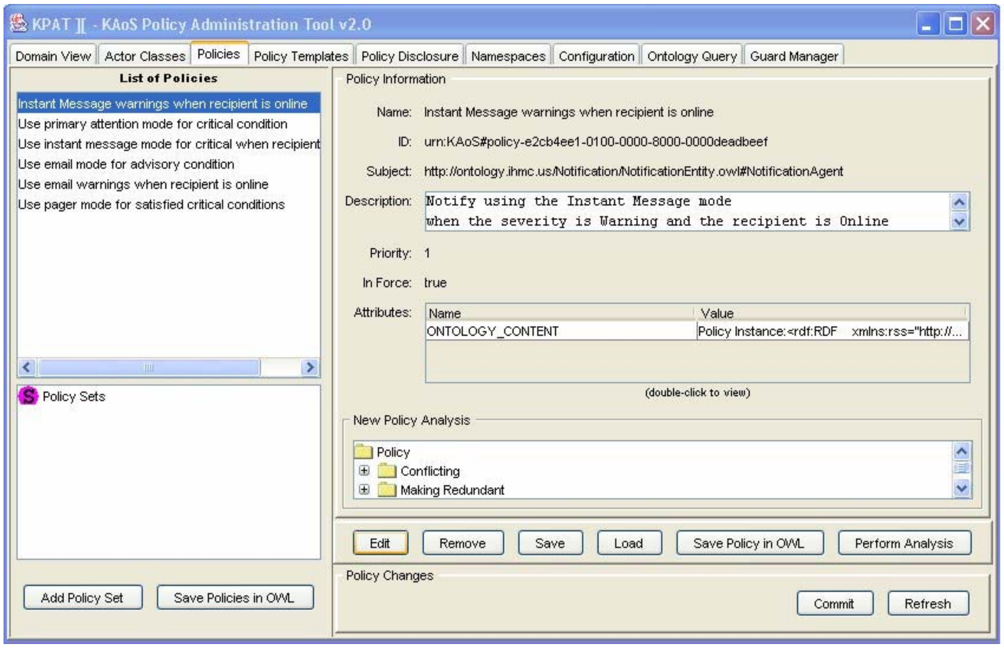


Fig 3. OWL ontologies used by the KARMEN system for notification are graphically depicted.



List of Policies

- Instant Message warnings when recipient is online
- Use primary attention mode for critical condition
- Use instant message mode for critical when recipient
- Use email mode for advisory condition
- Use email warnings when recipient is online
- Use pager mode for satisfied critical conditions

Policy Sets

Policy Information

Name: Instant Message warnings when recipient is online

ID: urn:KAoS#policy-e2cb4ee1-0100-0000-8000-0000deadbeef

Subject: http://ontology.ihmc.us/Notification/NotificationEntity.owl#NotificationAgent

Description: Notify using the Instant Message mode when the severity is Warning and the recipient is Online

Priority: 1

In Force: true

Attributes:

Name	Value
ONTOLOGY_CONTENT	Policy Instance:<rdf:RDF xmlns:rss="http://...

(double-click to view)

New Policy Analysis

- Policy
  - Conflicting
  - Making Redundant

Policy Changes

Fig 4. The KAoS Policy Administration Tool (KPAT) screen displaying a sample set of policies that govern notification modes in the KARMEN system.

the assigned event severity (critical, warning, advisory, log), and the plant area in which the event occurred based on the component hierarchy defined in the ontology. The user characteristics that can trigger a policy include the user's organizational role (operator, process engineer, area manager, etc.) and the user's current physical and computational presence (nearby/remote, online/offline: see figure 3). The qualities of the notification action that policies can oblige include the mode, latency, and focus of attention. Notification modes currently include e-mail, instant message, pager, operator displays, and the IHMC Monitor application. The latency quality controls how quickly the user is notified (immediate, deferred, archive). The focus of attention quality controls how forcefully the user's attention is obtained and depends on the features available in each notification mode (e.g. instant message chat session that interrupts the user vs. a queued message in the background).

The notification obligation policies are created using the KAoS Policy Administration Tool (KPAT) shown in figure 4. The attributes of the policy are from the ontological concepts shown in figure 3. Multiple policies can apply to a single event such as using the pager mode for critical events, using the instant message mode for critical events when the user is online, and using the primary focus of attention for critical events. Each policy is assigned a priority. KAoS uses the priority to resolve policy conflicts thereby enforcing organizational policies over personal preferences. The user notification agents can require and obtain acknowledgement of notifications and escalate the notification mode and recipients when acknowledgement is not received in a specified timeframe. In the near future, agents will select notification mode and salience based on the recipient's responsiveness to previous notification attempts by learning the most effective mode of contacting each user according to the time, user presence, and condition severity. Monitoring agents can begin recording a set of sensor values when the specified conditions obtain, stop recording when the conditions abate (or after a certain duration), then include a graph of the recorded data as an attachment to electronic notifications. Summaries could be extended to several formats including movies, spreadsheets, and PDF files.

VI. FUTURE WORK

A. Agent Modeling and Learning

We plan to enhance our agents with automated methods to model these highly non-linear, dynamic, possibly changing relationships among many distinct components and sensors, given live, streaming input data containing relatively few instances of anomalous system performance. The learned models should support automated response systems; that is, they should support decisions about which system changes e.g., a valve opening or closing) will bring the system back into normal operating range. More succinctly, the models must support causal inferences about the complex system. A suite of algorithms and techniques has been developed at IHMC and Carnegie Mellon University including the TETRAD IV software system that can serve as a substantial basis for an integrated KDD system capable of handling all of these features simultaneously [14]. The KDD algorithms will be integrated into the software agents framework, so that they can be used for automated learning of appropriate causal models based on liquid hydrogen sensor measurements from recent shuttle launches.

B. Novel Operator Displays

The operation of these complex plants or systems is typically overseen by human experts receiving information of a variety of types from numerous sources. Additional information, even relevant information, sometimes degrades the human experts' performance. A range of approaches to displaying complex information in challenging environments have been developed at IHMC by understanding both the task to be performed and the human cognitive processes involved. Methods to best display health and safety information to plant operators are being considered. One compelling display for use with KARMEN shown in figure 5 employs a radar-screen metaphor where the current status of a process monitoring agent is displayed live as a 'blip' on the screen.

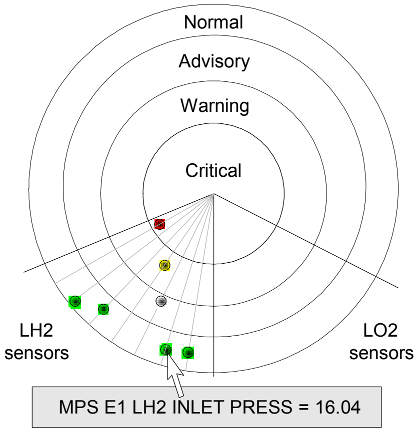


Fig 5. An alarm display design based on a radar screen metaphor.

VII. CONCLUSION

The KARMEN system is distinguished by its human-centered approach of providing tools to create personal monitoring agents with rich semantic descriptions of process state and salient user notification. These agent-based tools complement and amplify the expertise of the engineers and operators with the ability to create and refine personally relevant assessments of live process conditions. KARMEN focuses on supporting users in the difficult task of safely and effectively operating complex processes. We enable operators to specify complex monitoring conditions by using intuitive graphical tools; these conditions can be changed at any time without affecting the process control. Users also can apply ontological classes to define complex aggregate conditions that have not been previously specified in real-time.

REFERENCES

[1] T. Blevins, G. McMillan, W. Wijsznis, and M. Brown, *Advanced Control Unleashed: Plant Performance Management for Optimum Benefit*. Research Triangle Park, NC: The Instrumentation, Systems, and Automation Society, 2003, pp. 163-182.

[2] N. Hamdy and R. Fulvio, "Abnormal Condition Management with Real-time Expert System and Object Technology," *PCAI*, vol 17(1), pp. 28-35, 2003.

[3] F. Heck, T. Laengle, H. Woern, "A Multi-Agent Based monitoring and Diagnosis System for Industrial Components," University of Karlsruhe, Institute for Process Control and Robotics. Karlsruhe, Germany.

[4] I.A. Letia, F.Craciun, Z. Kope, A. Netin, "Distibuted Diagnosis by BDI Agents". In Proceedings of the IASTED International Conference APPLIED INFORMATICS. Innsbruck, Austria. 2000.

[5] Aspen Technologies, Hysys Dynamics. Available: http://www.aspentech.com/.

[6] Emerson Process Management DeltaV Distributed Control System. Available: http://www.easydeltav.com/.

[7] M. Carvalho and M. Breedy, "Supporting Flexible Data Feeds in Dynamic Sensor Grids Through Mobile Agents". In *Proceedings of the 6th International Conference in Mobile Agents*, Barcelona, Spain, October 2002.

[8] N. Suri, J.M. Bradshaw, M. Breedy, P. Groth, G. Hill, R. Jeffers, and T. Mitrovich, "An Overview of the NOMADS Mobile Agent System," Sixth ECOOP Workshop on Mobile Object System. Available: http://cui.unige.ch/~ecoopws/ws00.

[9] A. Uszok, J.M. Bradshaw, R. Jeffers, N. Suri, P. Hayes M Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott, "KAoS policy and Domain Services: Toward a Description-logic Approach to Policy Representation, Deconfliction, and Enforcement," In Proceedings of IEEE Fourth International Workshop on Policy. Lake Como, Italy, June 2003, pp. 93-98.

[10] J.M. Bradshaw, A. Uszok, R. Jeffers, N. Suri P. Hayes, M. Burstein, A. Acquisti,, B. Benyo, M. Breedy, M. Carvalho, D. Diller, M. Johnson, S. Kulkarni, J. Lott, M. Sierhuis, & R. Van Hoof, "Representation and Reasoning for DAML-based Policy and Domain Services in KAoS and Nomads," In Proceedings of the Autonomous Agents and Multi-Agent Systems Conference. Melbourne, Australia. ACM Press, New York, NY, 2003, pp. 835-842.

[11] J.M. Bradshaw, P. Beautement, M. Breedy, L. Bunch, S. Drakunov, P.J. Feltovich, R.R. Hoffman, R. Jeffers, M. Johnson, S. Kulkarni, J. Lott, A. Raj, N. Suri, & A. Uszok, "Making Agents Acceptable to People," In Intelligent Technologies for Information Analysis: Advances in Agents, Data Mining, and Statistical Learning, N. Zhong and J. Liu, Eds. Berlin, Germany, Springer Verlag, 2004, pp. 361-400.

[12] J. Hendler, T. Berners-Lee and E. Miller, "Integrating Applications on the Semantic Web," Journal of the Institute of Electrical Engineers of Japan, vol 122(10), October, 2002, pp. 676-680.

[13] D. Schreckenghost, C. Martin, C. Thronesbery, "Specifying Organizational Policies and Individual Preferences for Human-Software Interaction," In Etiquette for Human-Computer Work, Papers from the AAAI Fall Symposium. Technical Report FS-02-02, AAAI Press, 2003.

[14] C. Glymour, K. McGlaughlin, "Analyzing A Data Lookup Method for Machine Learning in Monitoring and Fault Localization for Hydrogen Generation Plants, Chemical Processing Plants and Other Complex Systems," Final Report for UCF contract 26-56-208. Pensacola, FL, September 2003.