KARMEN: Multi-agent Monitoring and Notification for Complex Processes

Larry Bunch, Maggie Breedy, Jeffrey M. Bradshaw, Marco Carvalho, and Niranjan Suri

Florida Institute for Human and Machine Cognition,
40 S. Alcaniz St. Pensacola, FL 32563
{lbunch, mbreedy, jbradshaw, mcarvalho, nsuri}@ihmc.us
http://www.ihmc.us/

Abstract. Early and consistent detection of abnormal conditions is important to the safe and efficient operation of complex industrial processes. Our research focuses on enabling the operators and engineers who control and maintain such systems to describe process conditions to software agents, deploy such agents to continuously monitor live process data, and receive appropriate notification from their personal agents concerning the process state. The resulting dynamic population of monitoring agents is managed by our agile computing framework according to policies that define computing and networking resource restrictions as well as user notification requirements and preferences.

1 Introduction

The Florida Institute for Human and Machine Cognition (IHMC) is conducting research and development for automated safety and health monitoring of industrial chemical processes [1] as well as the NASA space shuttle fueling and launch process. Our current KAoS Reactive Monitoring and Event Notification (KARMEN) multi-agent system enables users to perform automated live monitoring of complex process conditions that were typically only detected manually or during post-analysis of recorded process data [2]. This research also extends to the complementary area of remote user notification, most notably concerning adapting the notification mode and salience based on the event context.

We have taken a human-centered approach to monitoring automation that complements the user's ability to identify relevant monitoring contexts with the software agent's ability to rapidly and vigilantly assess the process state. This frees our agents from needing complete and accurate models of the systems being monitored and also distinguishes our work from related multi-agent approaches to chemical and manufacturing systems diagnostics [3-5].

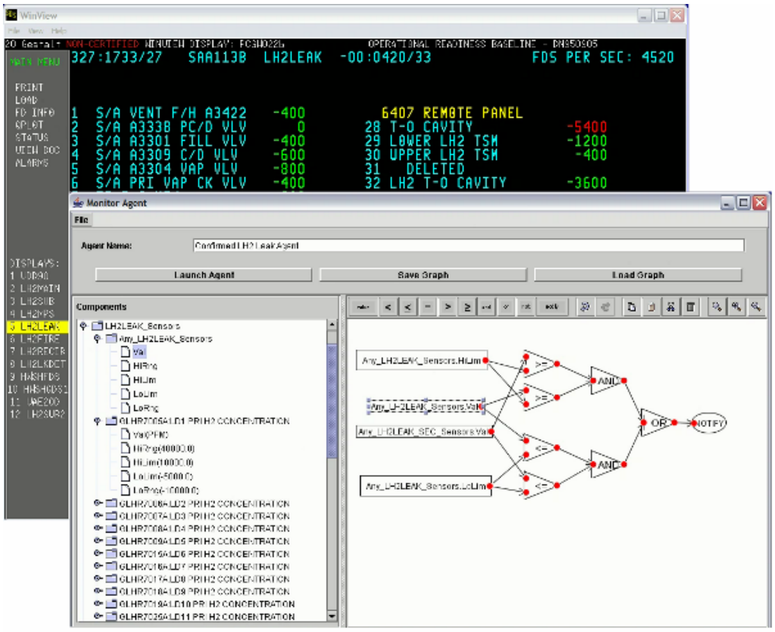
A brief description of KARMEN is followed by more detailed coverage of recent results involving how users describe process conditions to agents, how the resulting ad-hoc population of agents are deployed and managed by the framework, and the role of ontologies and policies in creating and controlling these agents.

© Springer-Verlag Berlin Heidelberg 2005

2 KARMEN Overview

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.

From the users' perspective, KARMEN is a desktop application for creating and deploying software agents to continuously monitor a process and notify the user as the conditions obtain and abate. Agent creation begins with condition specification. The user browses the space of sensors, valves, and other control elements comprising the process and selecting among the data streams each component provides.



```
20 Gestalt NON-CERTIFIED WINVIEW DISPLAY: FCGW0226        OPERATIONAL READINESS BASELINE - DN950905
327:1733/27   SAA113B  LH2LEAK  -00:0420/33              FDS PER SEC: 4520

1  S/A VENT F/H A3422    -400        6407 REMOTE PANEL
2  S/A A333B PC/D VLV       0      28 T-O CAVITY              -5400
3  S/A A3301 FILL VLV    -400      29 LOWER LH2 TSM           -1200
4  S/A A3309 C/D VLV     -600      30 UPPER LH2 TSM            -400
5  S/A A3304 VAP VLV     -800      31   DELETED
6  S/A PRI VAP CK VLV    -400      32 LH2 T-O CAVITY          -3600
```

Agent Name: Confirmed LH2 Leak Agent

```
LH2LEAK_Sensors
  Any_LH2LEAK_Sensors
    Val
    HiRng
    HiLim
    LoLim
    LoRng
  GLHR7005A1LD1 PRI H2 CONCENTRATION
    Val(PPM)
    HiRng(40000.0)
    HiLim(10000.0)
    LoLim(-5000.0)
    LoRng(-10000.0)
  GLHR7006A1LD2 PRI H2 CONCENTRATION
  GLHR7007A1LD3 PRI H2 CONCENTRATION
  GLHR7008A1LD4 PRI H2 CONCENTRATION
  GLHR7009A1LD5 PRI H2 CONCENTRATION
  GLHR7015A1LD6 PRI H2 CONCENTRATION
  GLHR7016A1LD7 PRI H2 CONCENTRATION
  GLHR7017A1LD8 PRI H2 CONCENTRATION
  GLHR7018A1LD9 PRI H2 CONCENTRATION
  GLHR7019A1LD10 PRI H2 CONCENTRATION
  GLHR7025A1LD11 PRI H2 CONCENTRATION
```

Fig. 1. Users construct graphical expressions to describe process conditions to monitoring agents, in this case involving the set of values from the operator screen shown in the background

$$(P_1 \geq P_{hi} \;\&\; S_1 \geq S_{hi}) \mid (P_1 \leq P_{low} \;\&\; S_1 \leq S_{low}) \mid (P_2 \geq P_{hi} \;\&\; S_2 \geq S_{hi}) \mid \ldots) \quad (1)$$

Part of our research includes applying semantic web techniques to provide rich and flexible descriptions and classifications of sensors through both enumeration and common properties such as type, limits, capabilities, location, and status. The user next selects the control elements to monitor, then uses a graph-based interface to assemble the process variables into a logical expression. For example, liquid hydrogen (LH2) leak sensors are placed at intervals along a shuttle fueling line in pairs of primary and secondary (SEC) detectors. In this case there are over 50 sensors to monitor that are individually very sensitive and therefore prone to false positives. The graph in figure 1 expresses the condition that the high or low limit has been exceeded for a pair of primary and secondary LH2 leak sensors concurrently. The same expression is also depicted in formula 1 below. The user launches the agent into the KARMEN system where the agent survives indefinitely monitoring the process conditions and notifying the user. Perhaps through the KARMEN application if it is running, otherwise through email, text pager, or other modes.

From a systems perspective, KARMEN is a set of host environments configured to run these agents as well as a collection of services for managing them and the resources they consume. When a new agent is requested a central coordinator service determines where the agent is deployed based on the components referenced, the monitored condition expression and its sub-expressions, as well as available host resources and overlap with existing agents. The coordinator service establishes data feeds among agents and can move agents from host to host to balance processing and network load as well as keeping the agents running. Policies play an important role in restricting user access to sets of components, setting constraints on the resources agents are allowed to consume, and defining preferences concerning how the notification service will deliver messages.

3 KARMEN Architecture

KARMEN relies upon multi-agent frameworks developed at IHMC to manage and control its Java agent population as shown in figure 2. The FlexFeed agile computing framework provides mobile agent hosting, networking, and coordination. The KAoS policy framework provides tools and services for defining polices that constrain agent actions and resource usage as well as oblige agent actions including user notification. KARMEN and KAoS both rely upon semantic web ontologies developed using the W3C standard Web Ontology Language (OWL). KAoS also provides an extensible base ontology with services to query the ontology.

3.1 FlexFeed Agent Networking Framework

FlexFeed is a Java agent framework that facilitates communication between agents and manages the computing and network resources within a distributed multi-agent system [6, 7]. KARMEN agents rely on the FlexFeed API for mobile deployment and access to information feeds among heterogeneous sensor, intermediate, and user

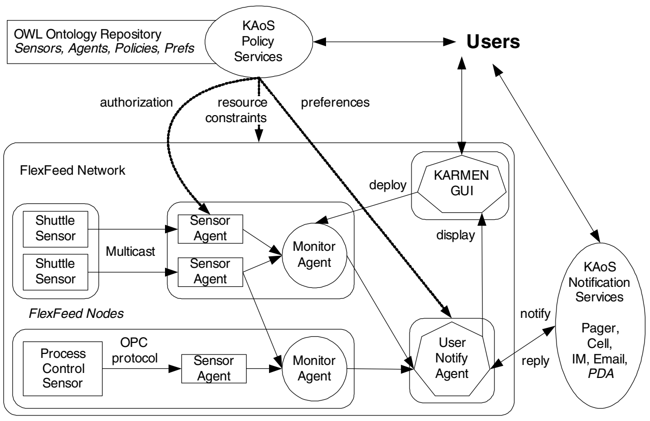


Fig. 2. A high-level depiction of the relationships among KARMEN system components

nodes. FlexFeed supports policies that restrict communication among agents and limit agent resource usage. 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 which is the intelligent component that is responsible for establishing and maintaining data streams in the framework. The Coordinator distributes processing load and bandwidth consumption across the framework preserving the resources on the nodes. Upon multiple requests on the same sensor, the Coordinator has the ability to discover and use intermediate processing nodes to minimize the network bandwidth and improve load balancing.

3.2 KAoS Policy Services

KAoS is a collection of policy services compatible with several software agent and robotic frameworks, as well as traditional distributed services platforms (e.g., CORBA, Web Services, Grid Computing) [8-10]. In the context of KARMEN, KAoS is used to define, manage, deconflict, and enforce policies restricting agent access to sensor data, bounding agent resources, and governing the mode of notification to users. The KAoS Policy Ontologies (KPO; http://ontology.ihmc.us/) are represented in the W3C standard Web Ontology Language (OWL) [11]. 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.

3.3 OWL Ontology Representation and Reasoning

Our system employs OWL to organize and classify process components, 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.

We employ ontological classifications of process control components and events for expressing potentially large and complex aggregate monitoring conditions. to dynamically define custom limits and alarm conditions that were previously only pre-defined in the control system.

4 Process Monitoring

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.

4.1 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.

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.

4.2 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.

4.3 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), 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.

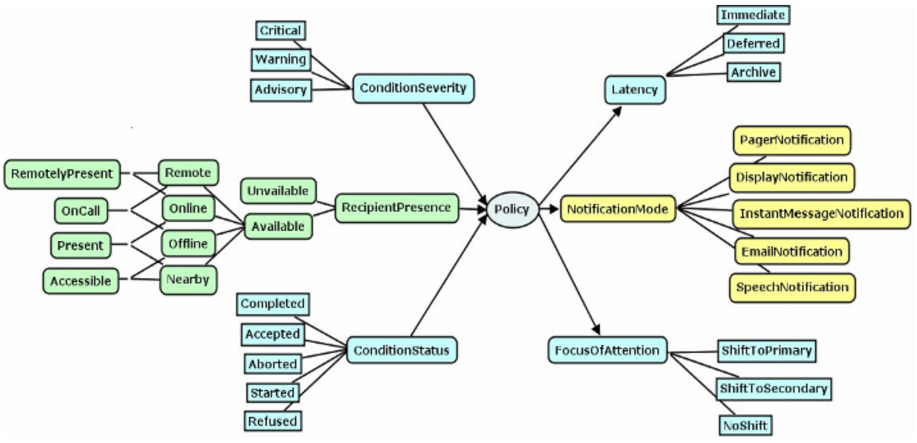
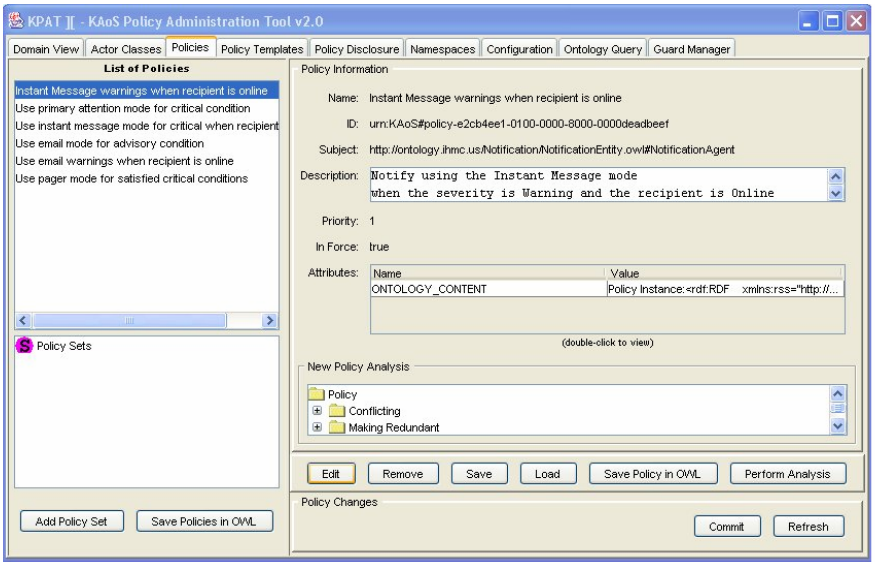


Fig. 3. OWL ontologies used by the KARMEN system for notification are graphically depicted



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 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

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.

5 Summary

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. Bunch, L., Breedy, M., Bradshaw, J., Carvalho, M., Suri, N., Uszok, A., Hansen, J., Pechoucek, M., and Marik, V. 2004. Software Agents for Process Monitoring and Notification. In Proceedings of the ACM Symposium for Applied Computing, Nicosia, Cyprus, 94-99. New York: ACM.
2. 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.
3. N. Hamdy and R. Fulvio, "Abnormal Condition Management with Real-time Expert System and Object Technology," *PCAI*, vol 17(1), pp. 28-35, 2003.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.

12. 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.
13. 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.
14. S.D.J. McArthur, E.M. Davidson, J.A. Hossack and J.R. Mc Donald. Automating Power System Fault Diagnosis through Multi-Agent System Technology. In Proceedings of the Hawaii International Conference on System Sciences. 2004.