Intelligent Information, Monitoring, and Control Technology of Industrial Process Applications

James H. Taylor[1] and Atalla F. Sayda[2]

Department of Electrical & Computer Engineering
University of New Brunswick
PO Box: 4400, Rm. H113 Head Hall
Fredericton, NB CANADA E3B 5A3

ABSTRACT

Abnormal event management (AEM) in large manufacturing plants has evolved as a higher and increasingly vital function of process control. In this paper, an intelligent information management and control system is introduced. The different computational agents (i.e., modules) of the system are embodied in a three-layered cognitive hierarchy, which offers intelligent behavior at the system level, as well as at the level of specialized task agents. At the lower level, agents generate goal-seeking reactive behavior. Three different fault detection and isolation agents (i.e., three complementary techniques) are embedded to generate three different assessments and to enhance the fault isolation process. Other utility agents are also incorporated to address topics such as data reconciliation, process model identification and optimization. At the middle layer, agents enable decision making, planning, and deliberative behavior. Two case-based reasoning agents are incorporated; the first manages the system in normal operation, while the other handles faulty process situations. A meta-management agent at the highest level monitors and coordinates other agents so as to make the whole system performance more robust and coherent.

1. INTRODUCTION

Abnormal event management (AEM) in large process plants has evolved as a higher and increasingly vital function of process control. When an abnormal event occurs it may take considerable time to diagnose its causal origin, and to take the appropriate actions to bring the process back to a normal, safe operating state or to a safe stop. This may have significant economic, safety, and environmental impact. Unfortunately, AEM is controlled manually in many manufacturing plants, which complicates the management and control of such plants. This can be attributed to several factors such as the size and complexity of modern manufacturing plants and increasingly massive information overload. The automation of AEM within an information and control infrastructure will reduce maintenance expenses, improve utilization and output of manufacturing equipment, enhance safety, and improve product quality. An integrated control and AEM system involves several sub-problem areas including data reconciliation and fusion, fault detection, isolation, and accommodation (FDIA), process model identification and optimization, and supervisory control. The integration of these complementary features into an intelligent fault-tolerant control framework will define a new arena for research in this area [6, 16, 23].

Many research studies, which proposed different combinations of systems theoretic and artificial intelligence techniques to tackle the AEM problem, have delineated a set of required features [23]:

- integrating different problem solving paradigms, knowledge representation schemes and search techniques,
- maintaining global databases of process data and knowledge,
- reasoning about process operations without requiring accurate models,
- coping with data explosion and the need for effective compression and interpretation, and
- understanding, and hence representing, process behavior at different levels of detail.

---

1. Email: jtaylor@unb.ca, Tel: +506.453.5101 ; FAX: +506.453.3589
2. Email: atalla.sayda@unb.ca, Tel: +506.449.0644 ; FAX: +506.453.3589

These requirements are similar to those proposed for intelligent supervisory control systems. For example, a proposed system for producing metal-matrix composite materials incorporated a central database of process data and knowledge, process planning via case-based reasoning, on-line learning, automated process optimization and model identification, robust control algorithms—all under the direction of an expert system coordinator [21]. This paper extends a proposed architecture for the integration of process automation and AEM in large process plants [22] by introducing a communication and behavioral framework for building an intelligent information management and control system. The paper is organized as follows: First, we review available conceptual models of complex intelligent systems followed by a detailed structural description of the proposed system architecture. Then, we discuss the behavioral model of the system. Finally, we conclude with future research and development steps.

2. CONCEPTUAL MODEL OF THE SYSTEM

We propose to use a combination of top-down and bottom-up approaches for modeling and developing an intelligent control and asset management system (ICAM system). The top-down approach deals with high level abstractions and conceptual tools, which facilitate capturing and modeling the structure and the behavior of the system being developed. Bottom-up modeling refers to developing scenarios that show in detail how the intelligent system should interact with users and complex external environments. Top-down modeling is the primary focus of this paper, so that the system architecture can be well explained and motivated.

Several conceptual frameworks have been suggested for modeling complex intelligent systems in the past two decades, such as expert systems, whose implementation results revealed several drawbacks, namely, lack of learning mechanisms and weak representation power [12]. Newell et al proposed another promising framework that is cognitive architectures, which model human cognition and problem solving behavior [14, 17]. Multi-agent systems (MAS), which can be considered as an instantiation of distributed artificial intelligence, are another conceptual framework for modeling complex systems. A MAS is defined as a loosely coupled network of problem solvers that work together to solve problems, that are beyond their individual capabilities [4, 25]. The MAS platform emphasizes distribution, autonomy, interaction (i.e., communication), coordination, and organization of individual agents.

Sloman [18, 19] introduced H-Cogaff, a human-like information processing architecture, which contains many components performing different functions all of which operate concurrently and asynchronously. The H-Cogaff architecture seems to represent a combination of the cognitive architecture and the MAS conceptual frameworks. It is worth mentioning that several projects are currently being developed to automate AEM using these conceptual frameworks, namely MAGIC which is developed by a joint venture of several European universities and companies [10], and AEGIS, which is developed by the Honeywell led Abnormal Situation Management (ASM) Consortium in the United States [1]. Having reviewed the different conceptual modeling frameworks, it is our opinion that Sloman's H-Cogaff scheme is the best candidate, which would meet most of the requirements of an ICAM system for complex process plants. The architecture of the system and its functional modules will be discussed in subsequent sections.

3. SYSTEM FUNCTIONAL DESCRIPTION AND ARCHITECTURE

Figure 1 illustrates the proposed architecture of the system, which consists of four information processing layers and three vertical subsystems, namely, perception, central processing, and action. The horizontal layers above the distributed control system (DCS) contain semi-autonomous agents that represent different levels of data abstraction and information processing mechanisms of the system. The middle two layers (i.e., the reactive and deliberative layers) interact with the external environment via the DCS and thus the industrial process by acquiring perceptual inputs and generating actions. The perceptual and action subsystems are divided into several layers of abstraction to function effectively. This can be achieved, for example, by categorizing observed events at several levels of abstraction, and allowing planning agents to generate behavior (actions) in a hierarchically organized manner.

The system layers interact with each other by means of bottom-up activation and top-down execution. Bottom-up activation occurs when a lower layer passes control to a higher layer because it is not competent to deal with the current situation. Top-down execution occurs when a higher-level agent makes use of the functionalities provided in a lower layer to achieve one of its goals. The basic flow of control in the system begins when perceptual input arrives at the lowest level in the architecture. If the reactive layer can deal with this input then it will do so, otherwise, bottom-up activation will occur and control will be passed to the deliberative layer. If the deliberative layer can handle the situation then it will do so, typically by making use of top-down execution. Otherwise, it will pass control to the meta-management layer to resolve any internal conflicts in the architecture or notify the operator for further intervention. In the remainder of this section, the functionalities of the agents in each layer will be discussed.

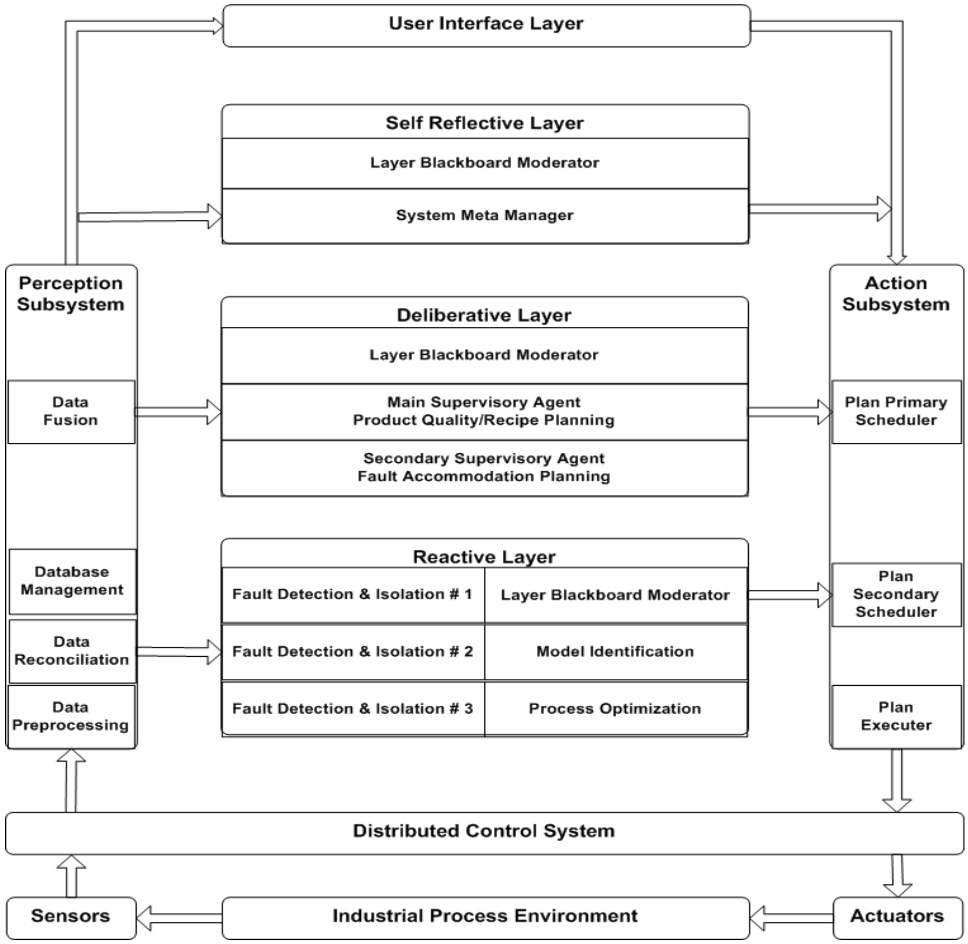

Fig. 1. ICAM system architecture

3.1. THE PERCEPTION SUBSYSTEM

In order to tackle the problem of data explosion in modern complex process plants, the perceptual subsystem will process data in a hierarchical manner, and categorize it into different levels of abstraction. The data stream is processed serially by different agents, where the first agent function is data acquisition and pre-processing. Gross discrepancies such as outliers and missing data are detected and removed by this agent. The data stream is then exposed to further statistical processing to estimate variances and detect changes in steady state. Such statistical information is communicated to the central processing subsystem to permit it to adapt to new situations. The next agent then reconciles process data in accordance with steady state conservation laws (e.g., material balance). The data is then archived in a database by the database management system. The last agent in the perceptual subsystem, the data fusion agent, aggregates the data to optimally determine operation critical variables. This will help the planning layer assess the situation of the external environment and to make appropriate decisions.

3.2. THE REACTIVE LAYER

Agents in this layer provide a direct response to events that occur in the environment. When an abnormal event occurs, several fault detection and isolation (FDI) agents work concurrently and complimentarily to generate different assessments. The integration of several FDI agents in the system will result in a better performance, as suggested by many FDI survey papers [23, 8, 7]. FDI basically consists of two tasks, namely fault detection, which indicates that something is going wrong in the plant. The determination of the exact location of the failure is the fault isolation task. Three different FDI techniques are being evaluated, namely, a directional parity vector model-based FDI technique, a fuzzy signed directed graph (SDG) model-based FDI technique, and a neuro-fuzzy data history-based FDI technique. These approaches are complementary in that they are based on entirely different world views, namely an analytic model, a cause/effect net and heuristic reasoning.

The first FDI approach exploits the concept of generalized parity space (GPS) to generate a set of directional residuals, from which process faults can be determined. When a fault occurs, it will result in an activity of the parity vector along certain directions or in certain subspaces. Therefore the fault isolation task involves determining which predefined direction the parity vector is most nearly aligned with [24]. A systematic approach to extend and to enhance the FDI properties of this technique in terms of the number of isolated faults, has been effectively developed [15]. A new fuzzy signed digraph (SDG) model-based FDI technique is another approach being evaluated. Signed digraphs, which have been widely used to model the cause/effect behavior of process plants, consist of nodes representing the process variables (and parameters) and signed directed arcs representing the cause/effect relationship between these variables. If a fault happens, process variables deviate resulting in a set of symptoms, which constitutes the pattern of this fault. The pattern is compared with a set of predefined fault patterns to isolate the fault [20]. The third FDI technique involves extending the adaptive neuro-fuzzy inference system (ANFIS) methodology. ANFIS is a data driven modeling approach that combines the reasoning capability of fuzzy logic and the learning capability of neural networks. The strength of this approach lies in its ability to use prior knowledge, and to update membership functions that provide a better model for the desired output. This makes the approach suitable for dealing with nonlinear processes [9].

A model identification agent is incorporated in the reactive layer, in order to improve the knowledge available to the FDI agents about the external environment (i.e., the plant), This agent will exploit an off-the-shelf model identification package to produce a multi-variable model, which will predict changes in process variables to estimate new process parameters (learning task), enhance the fault isolation task, and compensate for faulty sensor signals (estimation task). An optimization agent will be embedded in the reactive layer to make the best use of available equipment and raw materials. The agent will receive product quality plans and process operation constraints from the deliberative layer, and then the agent will formulate a new optimization problem to solve and generate the optimal raw material recipe to meet the new product quality. The optimizer may play the same role for faulty process situations, whenever possible.

3.3. THE DELIBERATIVE LAYER

Proactive behavior is achieved in the system in its deliberative layer, which is responsible for governing the system's actions in normal and faulty circumstances. Planning in this layer will not attempt to work in a vacuum. Rather, it will employ a library of pre-specified plans and a problem solving mechanism. There are several problem solving and inference paradigms that may be embedded in this layer of the architecture, such as rule-based reasoning, model-based reasoning, and case-based reasoning (CBR). Case-based reasoning provides a wide range of advantages over other paradigms. For instance, CBR can quickly propose solutions to problems that are not well defined, avoiding the time necessary to derive those answers from scratch, thus more easily meeting real-time requirements. CBR suggests a model of reasoning that incorporates problem solving, understanding, and learning, and integrates it all with episodic memory processes. It actually solves new problems by adapting previously successful solutions to similar problems [11].

The deliberative layer supervises the system through two CBR agents. The first agent is the main supervisor, which manages the system during normal operation circumstances. The agent's case library contains product quality profiles, their pre-specified raw material processing recipes, as well as the associated operation procedures. When a certain product specification is required, the main supervisory agent retrieves a set of cases that best match the required attributes and quality specifications. If the matching process is successful, the plan is sent to the action subsystem for execution. If not, the closest matching case is chosen and adapted by using model-based optimization, in which the main supervisory agent collaborates with modeling, simulation and optimization agents to generate the optimal recipe and operating conditions (e.g., pressure and temperature). The plan is sent to the user interface layer for further modifications by process operators if needed. Once the plan has been approved then it is sent to the action subsystem for execution. The actual quality specifications are monitored by the main supervisory agent, which will add the plan to its "good" case library should the actual and desired specifications match, or to the "bad" repository if they do not. This behavioral paradigm was central to the intelligent processing architecture proposed in [21].

The other CBR agent acts as a backup supervisory agent to manage the system in case of faulty situations. Pre-computed fault accommodation plans are stored in this agent's case library. These plans consist of schemes for sensor/actuator reconfiguration and controller tuning/restructuring, as well as fault propagation scenarios and recommended predictive maintenance procedures. When a process fault happens, the backup agent receives fault assessments for the different FDI agents in the reactive layer. Based on such assessments, the agent retrieves the most closely matching case from its library. Consequently, it alarms the user interface agent about the fault, its possible causes, and recommended mitigating actions for operator feedback and approval. The backup supervisory agent may interfere directly in critical situations to prevent the system performance from deteriorating excessively and to keep it in an acceptable state. Collaboration with the main supervisor may also occur to preserve the product quality at an acceptable level, if possible.

3.4. THE SELF REFLECTIVE LAYER

The self reflective layer provides the ability to monitor, evaluate, and control other agents in the architecture. For example, the deliberative layer is partly driven by decisions made by the reactive layer and perception subsystem, so it may unexpectedly acquire inconsistent information or goals. The same situation may occur in the action subsystem, which may not be able to meet the plan time frames sent by the deliberative layer. The meta-management agent can notice and categorize such situations, and perhaps through deliberation or observation over an extended time period develop a strategy to deal with these situations. Furthermore, the meta-management agent coordinates other agents so as to make the whole system performance more robust and coherent. It determines when other agents have completed their work, what agent to invoke next, and assesses credibility of each agent's behavior by monitoring their internal states. To be more specific and concrete, we refer to the system behavioral model in section 4.

The meta-management agent is basically a rule based expert system, which codifies all possible system behaviors and agent interactions as a behavior hierarchy is its rule base. an agent behavior is represented in the behavior hierarchy by a single structure, which has different dimensions for what is being achieved (i.e., goals), for how the results are being achieved (i.e., plans), for when and where activities are taking place, for who is involved in the activities, and for why the behavior has been adopted (meta-reasoning). Thus moving vertically through the behavior hierarchy leads to a more or less abstract representation of the agents' activities; emphasizing different aspects of their behaviors depending on the behavioral dimension traversed. Moving horizontally provides different views of behaviors at roughly equivalent levels of details, but emphasizing different behavioral dimension. An inference mechanism will help the meta-management agent resolve conflicts between the system agents by searching the behavior hierarchy and generating alternative behaviors so as to maximize (or at least improve) coordination. The agent may learn and generate new types of behaviors after a certain period of time.

3.5. THE USER INTERFACE LAYER

Process operators monitor and control the system through its user interface layer, which works concurrently at the top of the architecture. The user interface layer receives different types of information from the different layers and subsystems, namely:

- faulty components and their possible causes based on the different FDI agents' assessments,
- fault propagation scenarios based on the reasoning of the SDG based FDI agent,
- system recommendations in faulty situations such as instructions for control loop restructuring/tuning, predictive maintenance plans, and other reactive or proactive mitigating measures,
- product quality specifications and associated optimal raw material recipes, and
- internal system diagnostics and other data trend utility tasks.

Its most important obligations are to present process-critical information in a timely manner, and prevent data- and work-overload for the operator.

3.6. THE ACTION SUBSYSTEM

Plans which are sent by the deliberative layer are executed by the action subsystem. The action subsystem consists of hierarchically organized scheduling and execution agents. The main scheduling agent decomposes main plans into sub-plans that have shorter time frames. This results in better execution performance by alleviating the excessive computational burden on the main scheduling agent. The sub-plans are further decomposed by a secondary scheduling agent to simpler tasks in accord with the sub-processes in the plant. Finally, the subtasks are performed by their corresponding agents and the task outcomes are communicated to the DCS for final execution.

4. System behavioral formalism

Rigorous coordination of the behavior of the ICAM system layers and agents is crucial to success. A sound coordination scheme will allow us to assess its performance, and to evaluate how the internal agents of the system interact when a certain internal/external event occurs. Furthermore, it permits system behavior modeling to simulate the most critical design characteristics such as concurrency, autonomy, task distribution and parallelism, in order to guarantee robust and coherent performance. Due the complexity of modern manufacturing plants, intelligent systems (e.g., ICAM) have to be distributed, which makes the coordination of such systems very difficult and challenging. Durfee et al [5] proposed an informal theory that integrates organizational behavior, long term plans, and short term schedules into one coordination framework, and treats coordination as a distributed search process through the hierarchical space of the possible interacting behaviors of the individual agents to find a collection that satisfactorily achieves the agents' goals. The theory emphasizes several topics such as:

- hierarchical behavior representation to express different dimensions of behavior at different levels of detail,
- metrics for measuring the quality of coordination between agents,
- distributed search protocol for guiding the exchange of information between agents during the distributed search,
- local search algorithm for generating alternative behaviors at arbitrary levels of abstractions, and
- control knowledge and heuristics for guiding the overall search process to improve coordination.

Durfee also suggested that introducing a meta-level organization in the intelligent system to manage coordination between agents, and separating knowledge representation into domain-level and meta-level types would enhance coordination and make it more robust. Agents use domain-level knowledge to influence what goals they pursue, and use meta-level knowledge to decide how, when, and where to form and exchange behavioral models [3]. Durfee's informal theory and suggestions give the big picture of how agents should coordinate their activities within an intelligent system or even a society of intelligent agents. So far we have addressed the knowledge and organization separation issues by adopting the H-CogAff architecture proposed by Sloman. ICAM interacts with the external world through its reactive and deliberative agents, whereas the meta-level layer dictates the internal behavior of the system. Furthermore, domain-level knowledge is encoded in the deliberative agents and the meta-level knowledge is encoded in the self reflective layer.

When it comes to the internal behavioral model of the ICAM system, three approaches can be considered. One approach of coordination would be a direct interaction between the system agents according to their data flow requirements in a serial fashion. However, this approach is inflexible because it does not address the dynamic scalability of the system in terms of adding new agents or changing the internal architecture of any of the system agents. Another approach is to use an indirect and anonymous communication among agents via an intermediary such as a blackboard repository [2]. The blackboard technique consists of a global data repository and a control mechanism, which makes runtime decisions about posting, accessing, and removing the data in the repository and notifies other agents if useful information is available or not. However, this approach has its own problems such as communication bottlenecks, which occur when many agents try to gain access simultaneously. This would degrade the performance of the system and impose a sort of serial collaboration among agents instead of a concurrent one. A final approach would be to stipulate a direct and dedicated communication protocol between agents without resorting to a blackboard-like architecture. This approach permits a more distributed and parallel processing capabilities. Yet it adds more computational burden on individual agents to handle coordination, when they are supposed to direct most of their computations for problem solving. This would in turn affect the real-time design requirement of the ICAM system. It is our opinion that a balanced combination of the last two approaches would effectively address the trade off between computation and communication. Thus we will have multiple blackboard agents in the system to alleviate the communication bottleneck and a well established communication protocol to minimize the computational burden on agents. Two blackboard agents (moderators) in the reactive and deliberative layers will manage the systems interaction with the external world. Another blackboard agent in the self reflective layer will manage the internal interactions between agents.

In order to facilitate knowledge sharing, acquisition, and later revision or retraction, a communication language has to be developed to enable effective interaction between the different agents. This language uses ontologies to specify the major conceptual classes of the system such as plans, goals, and actions. Ontology identifies basic terms, relations among terms, and rules to combine them. The ICAM system will have two types of ontologies which match the knowledge representations in the system. The first type is domain-level ontology, which defines concepts for describing the external environment such as process instrumentations, variables, faults, and others. The second type is the task ontology (i.e., meta-level ontology) which provides vocabulary for describing terms involved in problem solving tasks such as goal, schedule, assign, and classify. Message between agents has a unified format, which is compatible with behavior representation structure of the system. This will result in a simple structure of the blackboard agents, where agents can post their conclusions and requests for information very effectively.

Petri nets constitute a graphical and mathematical modeling tool for describing and studying systems with critical characteristics such as those of the ICAM system. A Petri net is a particular kind of directed graph, together with an initial state called the initial marking. The underlying graph of a Petri net is a directed, weighted, bipartite graph consisting of two kinds of nodes, called places and transitions, where arcs are either from a place to a transition or from a transition to a place. A major strength of Petri nets is their support for analysis of critical behavioral properties and problems associated with concurrent systems [13]. Since we are designing a real-time dynamic system, the concept of time becomes crucial for performance evaluation and scheduling when modeling such systems. The timed Petri net introduces time delays associated with transitions and/or places in their net models to address real-time performance. The timed Petri net concept will be used to model the behavior of the architecture of the ICAM system.

5. PROJECT STATUS AND FUTURE WORK

A joint venture between several Atlantic Canadian universities, the National Research Council of Canada, and local and national companies was established in order to advance wireless sensor technology in the oil and gas industries and to assess the feasibility of an intelligent control and asset management system built on a wireless sensor network. As part of this joint venture, and as the leader in developing the ICAM system, we have formed a task force of five graduate students at the University of New Brunswick to address the integration of control and asset management for a large process industry application. Three team members were assigned the task of developing, testing and evaluating the different proposed FDI techniques. The FDI agents development task has successfully met several major goals, such as quick detection and isolation, isolability, robustness and disturbance decoupling. The task of evaluating the different data processing techniques which will be incorporated in the perception subsystem is assigned to another of the team members, who is presently focusing on data preconditioning and reconciliation. A rigorous review of the available system architectures and their characteristics has been done so as to match them with proposed system requirements. Starting in January 2004, the project has progressed well. Further steps are planned to implement a successful ICAM system prototype, namely:

- refinement of the FDI agents to address topics such as adaptability, explanation and reasoning capability, and to meet real-time requirements,
- development of appropriate data pre-processing, reconciliation, and aggregation techniques associated with the perception subsystem,
- consultation with industrial and automation partner companies to produce final specifications and documentation for the architectural level and execution platform in order to meet industry standards,
- design and modeling of the internal system coordinator using the Petri net approach,
- designing the two CBR agents and the pre-computed fault accommodation plans, and
- modeling the pilot plant which will be used to validate the system performance.

We believe that the successful design and development of the proposed system will lay a corner stone in the area of complex intelligent system development, and will open the doors for other applications such as distributed power plant management.

ACKNOWLEDGEMENT

This project is supported by the Atlantic Canada Opportunities Agency (ACOA) under the Atlantic Innovation Fund (AIF) program. The authors gratefully acknowledge that support and the collaboration of the University College of Cape Breton (UCCB), the National Research Council (NRC) of Canada, and the College of the North Atlantic (CNA). The authors also acknowledge the support of Natural Sciences and Engineering Research Council of Canada (NSERC) for funding the second author's research.

REFERENCES

[1] Cochran, T., Bullemer, P., and Nimmo, I. Managing abnormal situations in the process industries parts 1, 2, 3. 1997. Ann Arbor MI, NIST. Proceedings of the Motor Vehicle Manufacturing Technology (MVMT) Workshop.

[2] Corkill, D. D. Collaborating software: blackboard and multi-agent systems & the future. 2003. New York, New York. Proceedings of the International Lisp Conference.

[3] Durfee, E., Lesser, V. R., and Corkill, D. D., "Trends in cooperative distributed problem solving," IEEE Transactions on Knowledge and Data Engineering, vol. 1, no. 1, pp. 63-83, 1989.

[4] Durfee, E. and Montgomery, T. MICE: A flexible test bed for intelligent coordination experiments. 1989. Rosario, Washington. Proceedings of the 9th workshop on distributed AI.

[5] Durfee, E. and Montgomery, T., "Coordination as distributed search in a hierarchical behavior space," IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 6, pp. 1363-1378, 1991.

[6] Frank, P. M. and Köppen-Seliger, B., "New Developments Using AI in Fault Diagnosis," Engineering Applications of Artificial Intelligence, vol. 10, no. 1, pp. 3-14, 1997.

[7] Gertler, J., "Survey of model based failure detction and isolation in complex plants," IEEE Control Systems Magazine, 1988.

[8] Isermann, R. and Balle, P., "Trends in applications of model based fault detection and isolation and diagnosis of technical processes," Control Engineering Practice, vol. 5, no. 5, pp. 709-719, 1997.

[9] Jang, S. R., "ANFIS adaptive network based fuzzy inference system," IEEE transaction on systems, man, and cybernetics, vol. 23, no. 3, pp. 665-685, 1993.

[10] Köppen-Seliger, B., Marcu, T., Capobianco, M., Gentil, S., Albert, M., and Latzel, S. MAGIC: An integrated approach for diagnostic data management and operator support. 2003. Washington D.C. Proceedings of the 5th IFAC Symposium Fault Detection, Supervision and Safety of Technical Processes - SAFEPROCESS05.

[11] Leake, D. B., Case-based reasoning: experiences, lessons and future directions AAAI Press/MIT Press, 1996.

[12] Moore, R. L. and Kramer, M. Expert systems in online process control. 1986. Asilomar, California. Proceedings of the 3rd international conference on chemical process control.

[13] Murata, T., "Petri Nets: properties, analysis, and applications," Proceedings of the IEEE, vol. 77, no. 4, pp. 541-580, 1989.

[14] Newell, A., Unified theories of cognition Harvard University Press, 1990.

[15] Omana, M. and Taylor, J. H. Robust Fault Detection and Isolation Using a Parity Equation Implementation of Directional Residuals. IEEE Advanced Process Control Applications for Industry Workshop (APC2005). May 9-11 2005.

[16] Patton, R. J. Fault-Tolerant Control Systems: The 1997 Situation. Patton, R. J. and Chen, J. 3, 1033-1054. 1997. Kingston Upon Hull, UK, IFAC. IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes.

[17] Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R., Gobet, F., and Baxter, G. D. Techniques for modeling human performance in synthetic environ-ments: A supplementary review. 2003. Human Systems Information Analysis Center (HSIAC), formerly known as the Crew System Ergonomics Information Analysis Center (CSERIAC).

[18] Sloman, A. Varieties of affect and the CogAff architecture schema. 2001. York, UK. proceedings of symposium on Emotions, Cognition, and Affective Computing at the AISB'01 convention.

[19] Sloman, A. and Scheutz, M. Framework for comparing agent architectures. 2002. Birmingham, UK. Proceedings of the UK Workshop on Computational Intelligence.

[20] Tarifa, E. and Scenna, N., "Fault diagnosis, directed graphs, and fuzzy logic," Computer and Chemical Engineering, vol. 21 pp. S649-S654, 1977.

[21] Taylor, J. H., Harris, L. P., Houpt, P. K., Wang, H. P., and Russell, E. S. Intelligent processing of materials: Control of induction-coupled plasma deposition. Matthys, E. F. and Kushner, B. 1991. Warrendale, PA, TMS Publications. Advanced Sensing, Modelling, and Control of Materials Processing.

[22] Taylor, J. H. and Sayda, A. F. An intelligent architecture for integrated control and asset management for industrial processes. Joint 2005 International Symposium on Intelligent Control & 13th Mediterranean Conference on Control and Automation (2005 ISIC-MED). June 27-29 2005.

[23] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K., "A review of process fault detection and diagnosis part 1, 2, 3," Computer & Chemical Engineering, vol. 27, no. 3, pp. 293-346, 2005.

[24] Viswanadham, N., Taylor, J. H., and Luce, E. C., "A frequency domain approach to failure detection and isolation with application to GE21 turbine engine control system," Control Theory and Advanced Technology, vol. 3, no. 1, pp. 45-72, 1987.

[25] Wooldridge, M. J., An introduction to multiagent systems Wiley, 2002.