Intelligent Control and Asset Management: A Retrospective

James H. Taylor, Prof. Emeritus*

Department of Electrical & Computer Engineering
University of New Brunswick
Fredericton, New Brunswick, E3B 5A3, Canada

Abstract

A research team comprised of six graduate students and the author participated in a five-year effort to produce an advanced supervisory control system called ICAM, the Intelligent Control and Asset Management system, with specific focus on petroleum industry applications. By design, however, it was devised to be widely applicable to a variety of automation and manufacturing arenas.

A multi-agent architecture was adopted to implement ICAM, and ICAM's functionality was divided among the various agents as follows: a supervisor ("master agent"), a fault detection, isolation and accommodation agent, a data reconciliation agent, a linear model identification agent, a steady-state detection agent, a wireless network control coordination agent, and an operator interface agent.

The largest part of this research and development effort has been documented in numerous publications; most notably, overviews can be found in [1, 2, 3]. The wireless network control coordination agent, which was added in the last stage of the project, is described in [4, 5].

The most critical aspect of this R & D program was devising and implementing ICAM's "intelligence" and deciding where and how to embed it. This retrospective will focus primarily on these issues. Two considerations will be (1) the types of knowledge involved and (2) how and where it can be embedded most efficiently and effectively. In essence, this presentation will be in the form of basic considerations and "lessons learned". First, however, the overall ICAM system will be described, to provide sufficient context.

1. Introduction

Comprehensive asset management and control of a modern process facility can involve many tasks with different time-scales and complexity, including but not limited to signal processing (gross error detection and correction, filtering or data reconciliation); fault detection, isolation, and accommodation; process model identification; process optimization; and supervisory control. The automation of these complementary and intertwined tasks within an information and control infrastructure promises to reduce maintenance expenses, improve utilization of equipment, enhance safety, and improve production and product quality. A comprehensive literature on this subject may be found in [2]; space for that is not available here.

As mentioned, the functionality of ICAM was compartmentalized by adopting a multi-agent architecture, and responsibility for the various agents was assigned to one or more graduate students: Atalla Sayda [11] developed the supervisory expert system to oversee and control all of ICAM's activities; Maira Omana [12], Liqiang Wang [13] and Jing He [14] worked on fault detection and isolation by various methods, Mazyar Laylabadi [15] and Pilar Moreno [16] developed an agent for gross error detection and data reconciliation; Maira Omana also created a linearized model identification agent; Pilar Moreno [16] also developed an agent to determine when process variables are in steady state; and Hazem Ibrahim [17] developed the wireless network control coordination agent, to allow a wireless sensor network (WSN) whose links are in control loops to manage it resources efficiently without compromising system stability. A high-level schematic of ICAM is provided in Fig. 1; as noted, all agents were implemented in MATLAB [18] except the supervisor, which was built using G2 [19]. ICAM, with all its agents, is

* Tel: +1.506.459.5499, Fax: +1.506.453.3589, email: jtaylor@ieee.org

portrayed as connected to a wireless sensor network via the Gateway, which interfaces with a process simulator's actuators Ai and sensors Si; the simulator models a three-phase crude oil separator.

The remainder of this paper will include a brief description of the petroleum industry application for ICAM, a three-phase crude oil separator; an overview of some agent's control-theoretic and logical function, i.e., those with significant embedded intelligence; a discussion of the distribution of "intelligence" in ICAM; and a summary of "lessons learned".

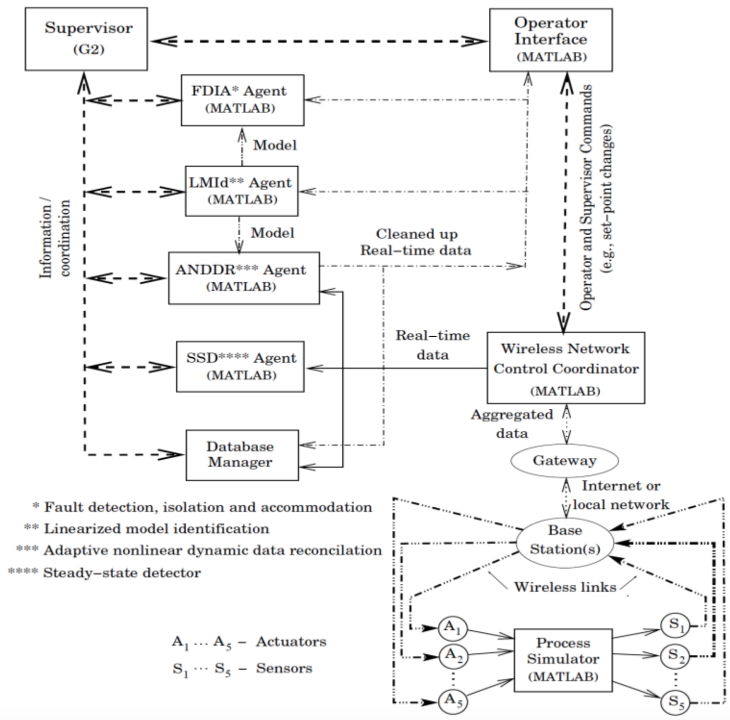

Figure 1: Schematic showing ICAM's architecture

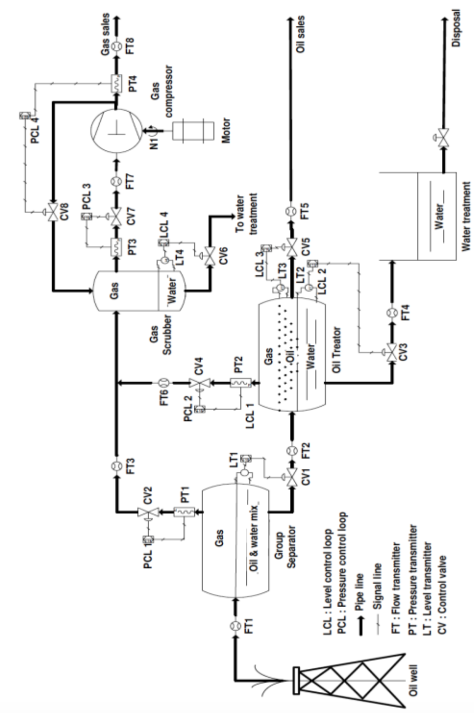

Figure 2: Crude oil separator schematic [21]

2. Crude Oil Separator Model

Oil production facilities often exhibit very complex and challenging dynamic behavior. The application treated in this project was a three-phase separator, consisting of two horizontal tanks, the first called a "group separator" in which most of the gas is separated from oil and water, followed by a "treater" where residual gas, oil and water are all separated to the extent possible (see Fig. 2). Each phase's dynamics are modeled; the hydrodynamics of oil-water separation was modeled based on the American Petroleum Institute design criteria [20], which involves solving an internal optimization problem, and the oil and gas phases' dynamic behaviors are modeled assuming gas-liquid phase equilibrium at the oil surface. The resulting very complex model [21] has three states that are quite slow (liquid levels) and two that are very fast (gas pressures); these states also comprise the model outputs, and they are controlled by five PID control loops. This oil production facility model was implemented in MATLAB to produce the ICAM simulator.

3. Linear Model Identification

Chemical processes are generally high-order and nonlinear in nature, making it difficult and impractical to derive or even use an accurate mathematical model for the system if one exists. This is one of the main reasons that the application of quantitative model-based techniques such as dynamic data reconciliation (DDR) has been limited. Also, many such methods are only applicable to linearized models, such as our fault detection, isolation and accommodation (FDIA) method (see Section 5). Thus, creating a Linear Model Identification (LMId) Agent was essential.

Linear model identification involves exciting and collecting input/output signals from the process and inferring the process dynamics and its environment (disturbances). Input/output signals were collected during closed-loop operation; performing tests in open loop may disrupt production or even cause instability. Closed-loop operation, on the other hand, allows LMId for a given desired setpoint which will be maintained with stability.

The model was estimated using generalized binary noise (GBN). GBN is generated according to a fixed sample interval T_0 and amplitude a; at each sample time T_k the GBN switches from a to $-a$ or *vice versa* with probability P [22]. Thus GBN signals are easy to design and of limited amplitude, and they give more consistent results overall. MATLAB's prediction error/maximum likelihood method (PEM) was used, which gave excellent results, in terms of fitting percentage; a sample LMId fit is depicted in Fig. 3 as a typical example. Obviously, there were a substantial number of additional decisions that had to be made in order for LMId to be successful.

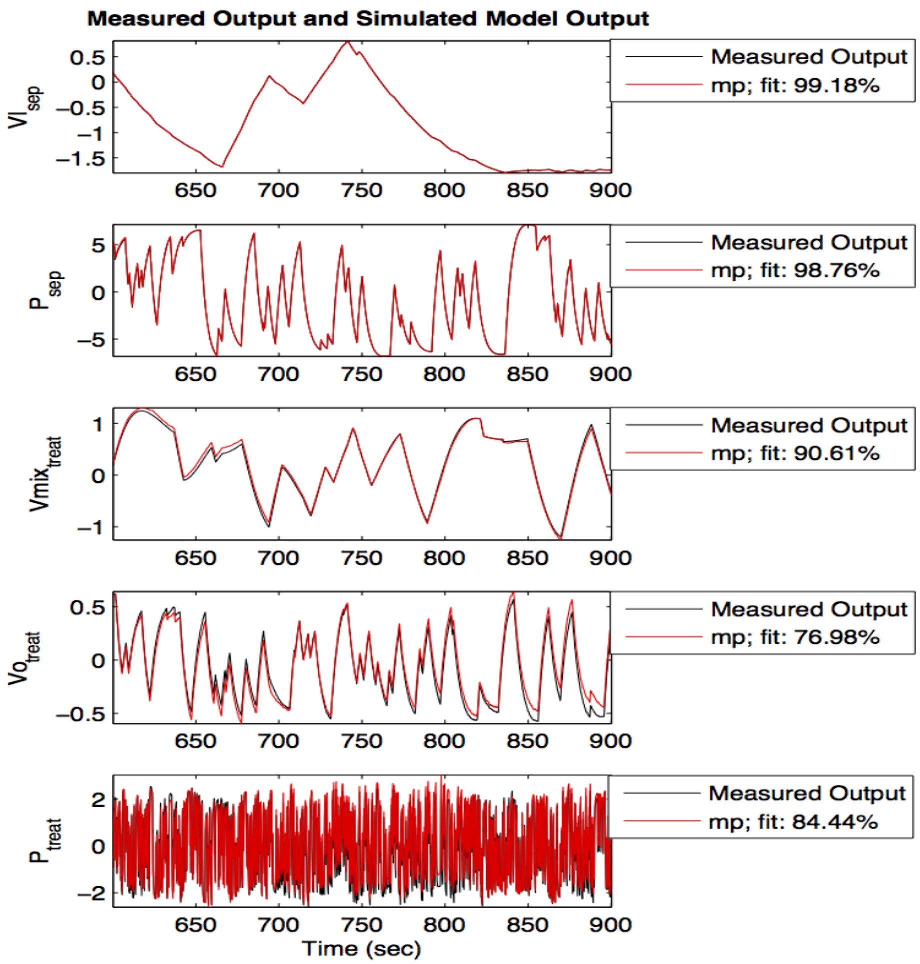

Figure 3: LMId results [12]

4. Signal Processing

Data from process sensors are typically corrupted by noise (usually high-frequency random signals, e.g., those modeled by first-order Markov processes) and gross errors (e.g., data drop-outs, spikes, analog-to-digital or digital-to-analog conversion errors) which are not discussed here. Low-pass filtering is often used for noise reduction, but this alters the spectrum of the desired process signals and thus introduces additional dynamics that may negatively effect LMId and other activities. An alternative technique that is often used in the chemical process control context is called dynamic data reconciliation (DDR), or, in the case of nonlinear processes, nonlinear dynamic data reconciliation (NDDR). Laylabadi and Taylor created and demonstrated a new heuristic approach for Gross Error Detection and Correction (GEDC) [23]; it is straightforward and not overviewed here.

An overview of the general NDDR formulation is given in [26]. A discretized version with weighted least squares (WLS) was used as the objective function, which can be expressed as follows [23]:

$$\min_{\hat{y}(t)} \sum_{i=0}^{ni+ns} \eta_i \sum_{j=c-H}^{c} \left(\frac{\tilde{y}_{ij} - \hat{y}_{ij}}{\sigma_i}\right)^2, \tag{1}$$

subject to:

$$\frac{d\hat{x}}{dt} - f(\hat{y}) = 0, \tag{2}$$

$$h[\hat{y}(t)] = 0, \tag{3}$$

$$g[\hat{y}(t)] \geq 0, \tag{4}$$

where

$\tilde{y}$ = corrupted measurements,
$\hat{y}$ = estimated (reconciled) measurements,
η = vector of weights,
c, H = current sample and window width,
σ = measurement noise standard deviations,
$f(\hat{y})$ = the system differential equation,
h = energy and/or material balance constraints,
g = process variable limits.

The lengths of $\hat{y}(t)$, $\tilde{y}$ and σ are equal to the total number of variables (states and inputs), i.e,. $y = [x|u]T$; ni is the number of inputs and ns is the number of states.. If the weights ηi are all equal then the estimation scheme is maximum likelihood. The dynamic constraint in Eq. (2) is discretized by solving the differential equation $\dot{x} = f(y)$ numerically over the window using a fixed step-size algorithm. There was a great deal of experimentation and many considerations considered in implementing NDDR [26]; however, once the algorithm was refined and validated there was little “intelligence” involved.

5. Fault Handling

In real world processes, such as oil and gas facilities, continuous production is required to achieve productivity and profitability requirements. As a result, stopping production suddenly in the middle of a run to fix or replace a sensor that has failed unexpectedly may result in significant economic losses. To minimize these interruptions in the plant operation, sensor fault detection and isolation (FDI) are essential and accommodation, if possible, would be even more beneficial. Similarly, actuator faults should also be detected and isolated; both should be integrated as part of an effective fault management strategy. These capabilities would provide a temporary solution to sustain safe operation while maintenance can be scheduled without significantly upsetting the process or production.

A powerful technique based on generating and interpreting directional residuals was investigated [12]. The directional residual method proved to be very effective and had the added benefit of supporting sensor fault accommodation for the class of failures considered. This is very desirable, as production can resume immediately or may not be interrupted at all. The directional residuals are generated using the generalized parity vector (GPV) technique; underlying mathematical basis is too lengthy to be presented here; refer to [28, 29, 30] for details. Suffice it to say that a directional residual has a very small magnitude in fault-free situations (zero magnitude, theoretically) that quickly assumes a large amplitude if a fault occurs. For a system with n states the residual lies in an n-space and points in a specific direction that is determined by the particular fault that happened.

Once the FDIA algorithm was developed, tested and refined there was a significant amount of decision-making and logic, as shown in Fig. 4. The complex FDIA logic is best illustrated by showing the interchanges among the Supervisor, FDIA Agent and the NDDR Agent during operation, for lack of space. Note that this involves invoking LMId (“SID AGENT”) when a linearized model must be obtained or updated, accepting reconciled process data from the NDDR agent, generating the FDIA residuals, and isolating and accommodating sensor faults.

6. Wireless Network Control Coordinator

A wireless network control system coordination agent (WNCSCA) was devised and built to solve a serious problem: if wireless paths are to be incorporated safely in process control loops one must not allow control signals to suffer slow data rates and variable time delays, both of which may degrade the performance of the control loop or even lead to instability. Those time delays, inherent to WSNs, stem from network delay and data latency, and depends on the network configuration and loading. In addition, the sampling rate also has a great impact on the performance and stability of the closed-loop control system networks, yet the data rate should be minimized to conserve WSN node battery life.

The WNCSCA (1) checks a proposed network configuration to determine the time delay which the control data packets will encounter, and rejects any proposed configuration that would lead to poor closed-loop system performance; and (2) determines the minimum acceptable sampling rate that does not degrade control loop performance excessively. The WNCSCA was demonstrated and validated by applying it to a jacketed continuous stirred-tank reactor model, a third-order model which has two control loops, three state variables (two of which are also outputs), and two controlled variables (two of which are also outputs), and two controlled variables. Specifically, the state variables are mixture height in the tank, *H*, temperature inside the tank, *T*, and the temperature inside the jacket, *Tj*.

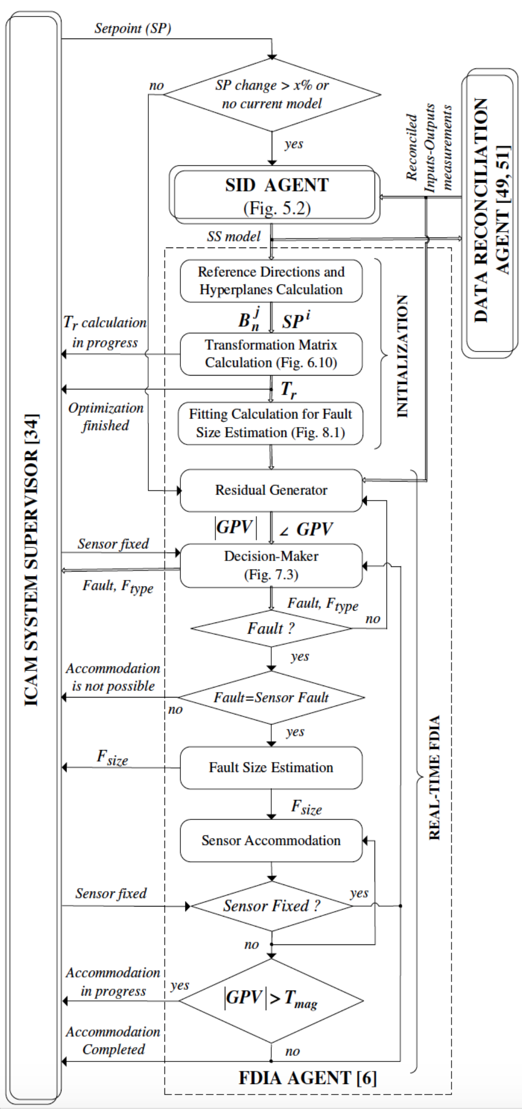

Figure 4: Fault handling logic [12]

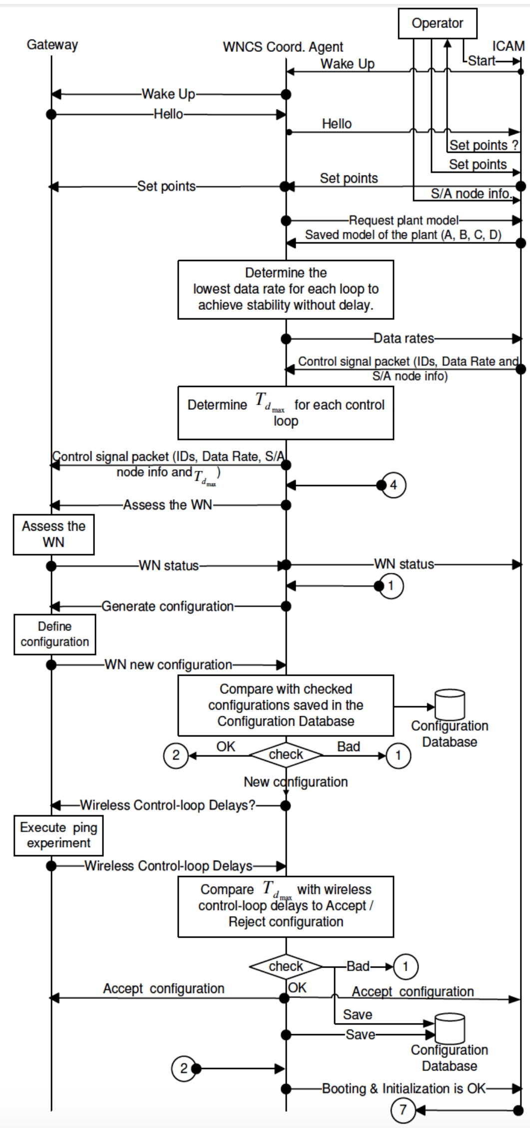

Figure 5: WNCSCA logic, booting phase [17]

Fig. 5 illustrates the complexity of the WNCSCA's logic, by showing the interchanges among the Supervisor (ICAM on the right-hand thread), WNCSCA (central thread) and the WSN Gateway during start-up (booting). The operator starts the system (upper right), and the three agents begin to collaborate. Note that the WNCSCA determines the lowest data rate acceptable for each loop, reports this to ICAM, which returns information about the configuration. The WNCSCA then evaluates the maximum delay tolerable for each loop, adds that that to the control signal packet and informs the Gateway along with a request to report the wireless network (WN) status. When the end of booting is reached the system should be ready to operate normally and the WNCSCA transitions to the "normal operations" logic (not shown).

7. Supervisory Control

A Supervisor prototype was developed to monitor and control the activities of the other agents overviewed previously and run the process effectively. It was designed to interact with the agents realistically, taking into account that the Supervisor was implemented in G2 [19], and it and the agents would be distributed over a number of platforms. This required a layered architecture, with middleware using the remote memory access (RMA) communication approach, which is part of the message passing interface (MPI) communication library, to address data communications among itself and the agents. The active target type RMA communication type was chosen to achieve high reliability, i.e., data are moved from the memory of one process to the memory of another, and both are explicitly involved in the communication [32].

Four RMA data communication channels were designed to transfer raw data, reconciled data, fault accommodation parameters and plant state space models to the consuming agents. When it comes to the communications between the reactive agents and the supervisory agent, the remote procedure call (RPC) paradigm is used to achieve looser connections. RPC is a client/server infrastructure that increases the interoperability, portability, and flexibility of an application by allowing it to be distributed over multiple heterogeneous platforms. The RPC communication part of the ICAM system prototype was designed so that the G2 supervisory agent acts as a client for the reactive agents (i.e., servers).

The supervisory agent was implemented in the G2 real time expert system shell, wherein the ICAM system internal and external behavior is codified in its knowledge base [19]. The rule-base codifies the desired system behavior in response to external environment dynamic changes and to process operator interactions. Therefore, it is crucial to carefully design the rule-base of the supervisory agent to achieve robust system performance. Fig. 6 illustrates the ICAM system prototype event sequence diagram, which is embedded in the supervisory agent rule-base.

The rule-base design process ceased in its preliminary stage after developing the extensive infrastructure mentioned; it may be further developed to address more complex situations in the future. It is, in fact, quite rudimentary – the Supervisor was able to manage simple scenarios, in order to test interactions between it and various ICAM agents; a sample scenario is shown in Fig. 6. Higher-level "reasoning" was not implemented, due to a lack of resources.

8. Summary and Conclusions: Distribution of Intelligence

One general principle emerged rather early in our work: knowledge about algorithms and numerical methods was incapsulated in our MATLAB agents, while knowledge about situation assessment and asset management was embedded in the Supervisor's expert system. As can be seen in Figs. 4 and 5, the degree of logic for some activities is quite substantial – it is probably justified to call some of them "smart agents".

One advantage to incorporating knowledge in smart agents is the reaction time savings – communications among the agents and the Supervisor are much slower than executing logic within the agents' MATLAB scripts and functions. On the other hand, a rule-based expert system would be preferable for handling higher-level decision-making for situation assessment and asset management. Such a rule base would structure this knowledge more systematically, making it easier to code, debug, maintain and extend this information.

Very briefly, much was accomplished in the ICAM project, and much learned. It was realized that developing "smart agents" in MATLAB is effective in many ways: testing, debugging and refinement was easy, and execution was much faster than could be accomplished with the Supervisor "in the loop". It also became apparent that the ICAM architecture and use of a rule-based expert system as the Supervisor is the best way forward.

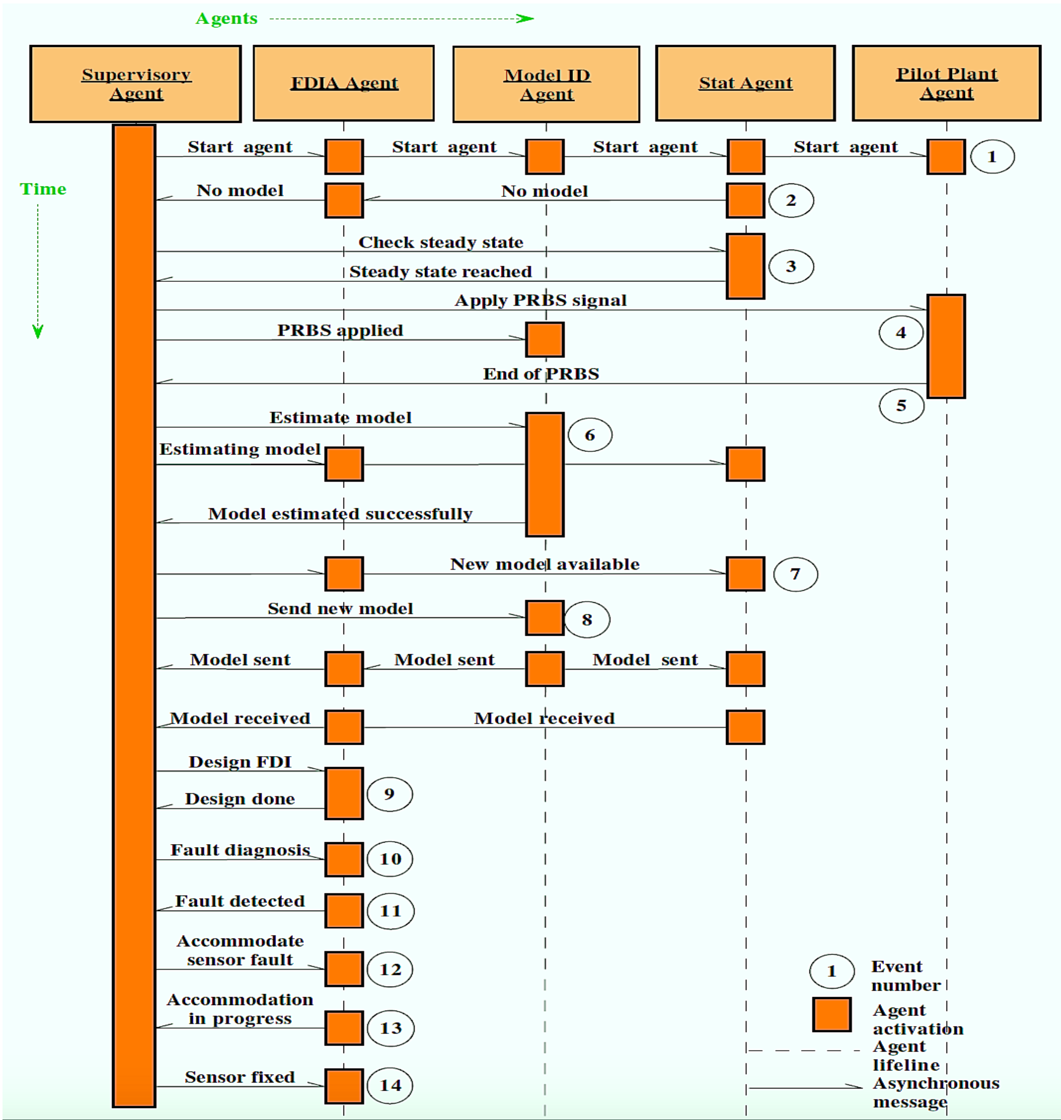

Figure 6: ICAM system start-up event sequence [11]

References

[1] *J. H. Taylor and A. F. Sayda, "An intelligent architecture for integrated control and asset management for industrial processes," in Proc. IEEE International Symposium on Intelligent Control, Limassol, Cyprus, June 2005.*

[2 *J. H. Taylor and A. F. Sayda, "Prototype design of a multi-agent system for integrated control and asset management of petroleum production facilities," in Proc. American Control Conference, Seattle, Washington, June 2008.*

[3] *A. F. Sayda and J. H. Taylor, "A multi-agent system for integrated control and asset management of petroleum production facilities - part 1, 2, 3," in Proc. IEEE International Symposium on Intelligent Control, San Antonio, Texas, September 2008.*

[4] *J. H. Taylor, H. S. Ibrahim, J. Slipp, and J. Nicholson, "A safe communication scheme for an intelligent wireless networked control system coordination agent," in Proc. IEEE International Conference on Systems, Man, and Cybernetics, Istanbul,Turkey, October 2010.*

[5] *J. H. Taylor, J. Akerberg, H. M. S. Ibrahim, and M. Gidlund, "Safe and secure wireless networked control systems," in Proc. IEEE Multiconference on Systems and Control, Dubrovnik, Croatia, October 2012.*

[6] *F. E. Ritter, N. R. Shadbolt, D. Elliman, R. Young, F. Gobet, and G. D. Baxter, "Techniques for modeling human performance in synthetic environments: A supplementary review," Human Systems Information Analysis Center (HSIAC),*

formerly known as the Crew System Ergonomics Information Analysis Center (CSERIAC), Wright-Patterson Air Force Base, OH, Tech. Rep., 2003.

[7] *E. Durfee and T. Montgomery, "MICE: A flexible test bed for intelligent coordination experiments," in Proc. 9th workshop on distributed AI, Rosario, Washington, September 1989.*

[8] *M. J. Wooldridge, An introduction to multi-agent systems. Chichester, England: Wiley, 2002.*

[9] *N. R. Jennings and E. M. Mamdani, "Using ARCHON to develop real–world DAI applications parts 1, 2, 3," IEEE Expert, vol. 11, no. 6, pp. 64–86, 1996.*

[10] *J. Gertler, "Survey of model based failure detection and isolation in complex plants," IEEE Control Systems Magazine, December 1988.*

[11] *A. F. Sayda, "Intelligent control and asset management of oil and gas production facilities," Ph.D. dissertation, University of New Brunswick, 2008.*

[12] *M. Omana, "Fault detection, isolation and accommodation using the generalized parity vector technique," Ph.D. dissertation, University of New Brunswick, 2009.*

[13] *L. Wang, "On-line fault diagnosis using signed digraphs," Master's thesis, University of New Brunswick, 2006.*

[14] *J. He, "Neuro-fuzzy based fault diagnosis for nonlinear processes," Master's thesis, University of New Brunswick, 2006.*

[15] *M. Laylabadi, "Adaptive nonlinear dynamic data reconciliation and gross error detection," Master's thesis, University of New Brunswick, 2006.*

[16] *R. P. Moreno, "Steady state detection, data reconciliation, and gross error detection," Master's thesis, University of New Brunswick, 2010.*

[17] *H. M. S. Ibrahim, "Wireless sensor network management agent," Master's thesis, University of New Brunswick, 2009.*

[18] *T. MathWorks, "Optimization toolbox for use with MATLAB," 1990-2004, natick, MA 01760.*

[19] *GenSym Corporation, "G2 for application developers reference manual, 8th ed." 2005, Burlington, MA 01760.*

[20] *Manual on Disposal of Refinery Waste, American Petroleum Institute, 1220 L Street, NW, Washington, DC 20005, 1969, chapter 5, Oil Water Separator Process Design.*

[21] *A. F. Sayda and J. H. Taylor, "Modeling and control of three-phase gravity separators in oil production facilities," in Proc. American Control Conference, New York, NY, 2007.*

[22] *J. A. F. Tulleken, "Generalized binary noise test-signal concept for improved identification-experiment design," Automatica, vol. 26, no. 1, 1990.*

[23] *M. Laylabadi and J. H. Taylor, "ANDDR with novel gross error detection and smart tracking system," in Proc. 12th IFAC Symposium on Information Control Problems in Manufacturing, Saint-Etienne, France, May 2006.*

[24] *M. J. Liebman, T. F. Edgar, and L. S. Lasdon, "Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques," Computers & Chemical Engineering, vol. 16, no. 10/11, 1992.*

[25] *K. F. McBrayer, T. A. Soderstrom, T. F. Edgar, and R. E. Young, "The application of nonlinear dynamic data reconciliation to plant data," Computers and Chemical Engineering, vol. 22, no. 12, pp. 1907–1911, 1998.*

[26] *J. H. Taylor and R. P. Moreno, "Nonlinear dynamic data reconciliation: In-depth case study," in Proc.IEEE Multiconference on Systems and Control, Hyderabad, India, August 2013.*

[27] *V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis part 1, 2, 3," Computer & Chemical Engineering, vol. 27, no. 3, pp. 293–346, 2003.*

[28] *M. Omana and J. H. Taylor, "Robust fault detection and isolation using a parity equation implementation of directional residuals," in Proc. IEEE Advanced Process Control Applications for Industry Workshop, Vancouver, Canada, May 2005.*

[29] *M. Omana and J. H. Taylor, "Enhanced sensor/actuator resolution and robustness analysis for FDI using the extended generalized parity vector technique," in Proc. American Control Conference, Minneapolis, Minnesota, June 2006.*

[30] *J. H. Taylor and M. Omana, "Fault detection, isolation and accommodation using the generalized parity vector technique," in Proc. IFAC World Congress, Seoul, Korea, July 2008.*

[31] *J. Loar, S. L. Alekman, G. Jubien, and G. Bihary, Control for the Process Industries. Chicago, IL: Putnam Publications, 1994, (Three articles by these authors.).*

[32] *A. F. Sayda and J. H. Taylor, "Toward a practical multi-agent system for integrated control and asset management of petroleum production facilities," in Proc. IEEE International Symposium on Intelligent Control (ISIC), Singapore, October 2007.*