2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Prototype Design of A Multi-agent System for Integrated Control and Asset Management of Petroleum Production Facilities

James H. Taylor and Atalla F. Sayda

Abstract— **This paper addresses a practical intelligent multi-agent system for asset management for the petroleum industry, which is crucial for profitable oil and gas facilities operations and maintenance. A research project was initiated to study the feasibility of an intelligent asset management system. Having proposed a conceptual model, architecture, and implementation plan for such a system in previous work [1], [2], [3] and defined its autonomy, communications, and artificial intelligence (AI) requirements [4], [5], we are proceeding to build a system prototype and simulate it in real time to validate its logical behavior in normal and abnormal process situations. We also conducted a thorough system performance analysis to detect any computational bottlenecks. Although the preliminary system prototype design has limitations, simulation results have demonstrated an effective system logical behavior and performance.**

I. INTRODUCTION

Asset management and control of modern process plants involves many tasks of different time-scales and complexity including data reconciliation and fusion, fault detection, isolation, and accommodation (FDIA), process model identification and optimization, and supervisory control. The automation of these complementary tasks within an information and control infrastructure will reduce maintenance expenses, improve utilization and output of manufacturing equipment, enhance safety, and improve product quality. Many research studies proposed different combinations of systems theoretic and artificial intelligence techniques to tackle the asset management problem, and delineated the requirements of such system [6], [7], [8].

Several research programs addressed the automation of asset management in large complex systems, namely the Pilots Associate (PA) program sponsored by the Defense Advanced Research Projects Agency (DARPA) [9], [10], the Rotorcraft Pilots Associate (RPA) program funded by the US army [11], MAGIC (Multi-Agent-Based Diagnostic Data Acquisition and Management in Complex Systems) developed by a joint venture of several European universities and companies [12], ISHM (Integrated System Health Management) project developed by NASA for space applications [13], AEGIS (Abnormal Event Guidance and Information System) developed by the Honeywell led Abnormal Situation Management (ASM) Consortium in the United States [14], and CHEM-CSS (Advanced Decision Support System for Chemical/Petrochemical Manufacturing Processes) developed by the European Community (EC) Intelligent Manufacturing Systems (IMS) consortium [15].

Among all projects, AEGIS is the most relevant. It proposes a comprehensive asset management framework from an industrial view point. AEGIS built on the experience of military aviation research projects, especially the Pilots Associate (PA) and the Rotorcraft Pilots Associate (RPA) [16]. Although the 12 year old research program has achieved several goals and developed a well established abnormal situation management awareness and culture, it did not address the automation of massive process data interpretation and process fault diagnosis and accommodation, which would be aimed to minimize the workload on process operators [17].

A new asset management research project, PAWS (Petroleum Applications of Wireless Systems), was initiated by a joint venture of Atlantic Canadian universities and the National Research Council of Canada (NRC) for oil and gas applications [1], [18], [2], [19], [20], [3], [21], [22], [23], [24], [4], [25], [5], [26]. The PAWS project scope is to develop a control and information management system which consists of two subsystems. The first subsystem is a wireless sensor network which will alleviate the need for data cables in offshore oil rigs and onshore refineries, and improve flexibility for adding and reconfiguring sensors. Wireless sensors will be used where permitted by safety. The second subsystem intelligently manages the massive data flow from oil rigs and interprets it so as to help operators take more appropriate decisions during abnormal events and, through intelligent control, improve process economics. This effort is building now on the AEGIS project experiences.

As part of the PAWS project, our team is developing an *intelligent control and asset management system* (ICAM system) in which several milestones have been achieved. The conceptual model of an automated asset management system, its architecture, and its behavioral model have been defined [1], [2]. An implementation plan for such system has been prepared, and the appropriate development platforms have been chosen [3]. Furthermore, a general ICAM system agent-based structure and its communication and the artificial intelligence requirements were defined [4], [5]. This paper builds on the previous requirements analysis work and describes a real-time ICAM system prototype. Furthermore, a real time simulation experiment is conducted to verify the system design and validate its performance.

The paper is organized as follows: First, we describe the structure of the ICAM system prototype and discuss its supervisory agent design in section 2. Then, we describe a real-time simulation-based implementation of the ICAM system prototype, discuss the simulations results, and analyze system performance in sections 3. We discuss the ICAM

James H. Taylor is with the Department of Electrical & Computer Engineering, University of New Brunswick, PO Box 4400, Fredericton, NB CANADA E3B 5A3 jtaylor@unb.ca

Atalla F. Sayda is a PhD candidate with the Department of Electrical & Computer Engineering, University of New Brunswick, PO Box 4400, Fredericton, NB CANADA E3B 5A3 atalla.sayda@unb.ca

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

system limitations and design challenges in section 4. Finally, we summarize our research findings in section 5.

II. The ICAM system prototype

In order to have the ICAM system requirements deployed in a real-world system, a prototype has to be developed. Figure 1 illustrates the simplified ICAM system prototype. Data from an external plant (a pilot plant at the College of North Atlantic) or a simulation model is received by the statistical data monitoring agent, which preprocesses the data by removing undesired discrepancies such as outliers and missing data. Processed data is stored in a real-time database for logging and other purposes, and is then sent to the model ID and FDI agents for further processing. When the data statistical preprocessor detects a change in the operating point or an abnormal change in data, it alerts the model ID and FDI agents to further identify the nature of the data change. If a significant change in the process operating point occurs, the system supervisory agent asks the model ID agent to update the process model parameters. If the change is a process fault (i.e., a sensor or actuator failure), the FDI agent detects the nature of the fault and notifies the ICAM system supervisor for further processing. For every event that occurs, the supervisor is notified, which in turn monitors, directs, and assesses the logical behavior of the system. Processed data at every agent is sent to an operator interface, which allows operators make the appropriate decision depending on the plant situation.

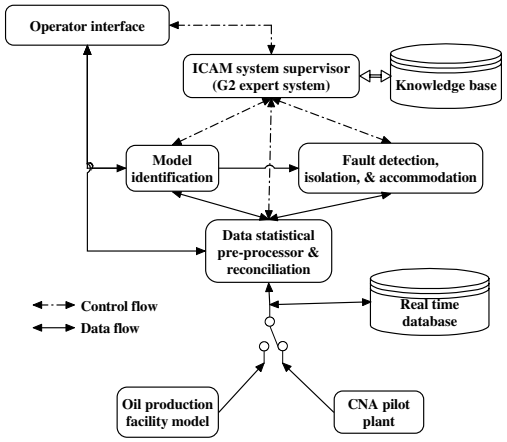

Fig. 1. ICAM system prototype

The FDI agent exploits the generalized parity space (GPS) to generate a set of directional residuals, from which process faults can be determined [27], [18], [22], [25], [28], [26]. The statistical preprocessor agent removes missing data and outliers by exploiting the median absolute deviation algorithm [29]. The model ID agent estimates the multivariable plant model by using the subspace method, which uses the canonical variable algorithm (CVA) in its singular value decomposition stage [30], [31], [32]. The supervisory agent is a G2 real time expert system [33], which codifies the ICAM system internal and external behavior in its knowledge base. The external plant model represents an oil production facility, which separates oil well fluids into crude oil, sales gas, and water.

The simulation model basically consists of two processes, as illustrated in figure 2. The first is a two-phase separator in which hydrocarbon fluids from oil wells are separated into two phases to remove as much light hydrocarbon gases as possible. The produced liquid is then pumped to the three-phase separator (i.e., the second process), where water and solids are separated from oil. The produced oil is then pumped out and sold to refineries and petrochemical plants if it meets the required specifications. Gas is processed further and sent as sales gas.

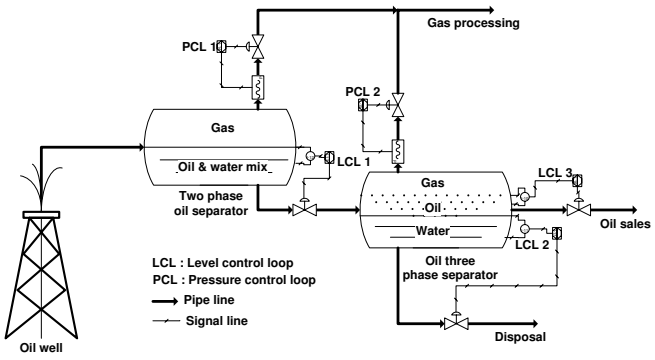

Fig. 2. Oil production facility P&ID

The two separation processes of the simulation model are controlled to maintain the operating point at its nominal value, and to minimize the effect of disturbances on the produced oil quality. As shown in figure 2, the first separation process is controlled by two PI controller loops. In the first loop, the liquid level is maintained by manipulating the liquid outflow valve. The second loop controls the pressure inside the two-phase separator by manipulating the amount of the gas discharge. The second separation process has three PI controller loops: An interface level PI controller maintains the height of the oil/water interface by manipulating the water dump valve, the oil level is controlled by a second PI controller through the oil discharge valve, and the vessel pressure is maintained constant by a third PI loop [34].

A. The supervisory agent rule-base design

Since the supervisory agent of the ICAM system coordinates its internal and external behavior, it is crucial to carefully design the rule-base of the supervisory agent to achieve robust system performance. The rule-base codifies the desired system behavior in response to external environment dynamic changes and to process operator interactions. Figure 3 illustrates the ICAM system prototype event sequence diagram, which is embedded in the supervisory agent rule-base. The rule-base design process is in its preliminary stage; and it will be further developed to address more complex situations in future work. The ICAM system supervisory agent starts up the other reactive agents, which are implemented as MATLAB functions and scripts for ease of development and debugging [5].

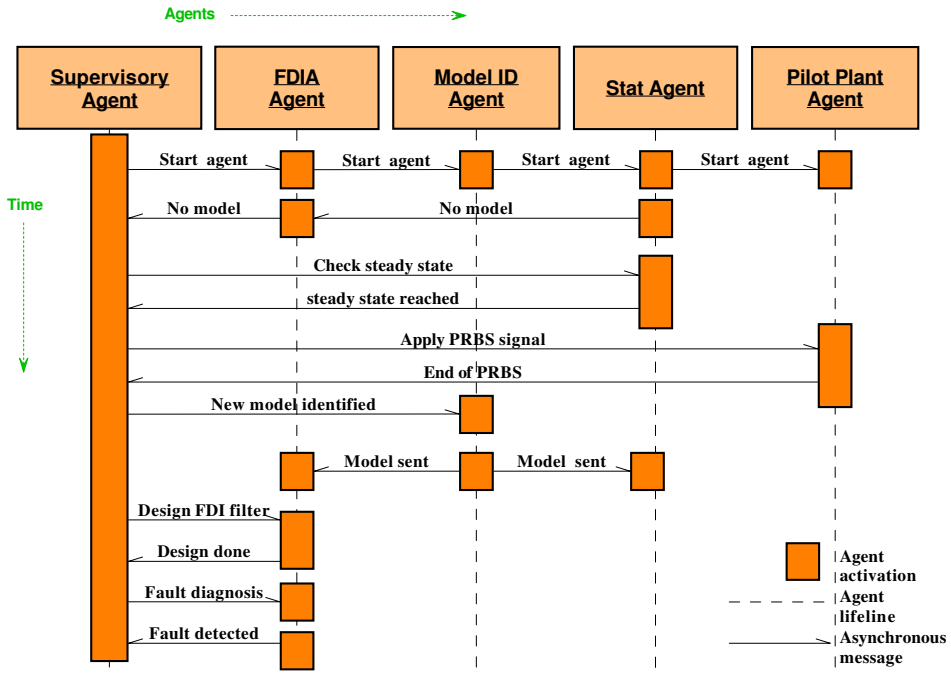

Fig. 3. ICAM system prototype event sequence

If the FDI agent or the statistical pre-processing agents do not have any process model, they report their status to the supervisory agent, which, in turn, commands the statistical pre-processor to check if the external plant is in steady state. If the external plant is in a steady state, the supervisory agent asks the low level control system to apply a small pseudo random binary signal (PRBS) for a specified period of time $\Delta t = 300sec$. The model ID agent collects process data during the application of the PRBS signal. Once the low level control flags back the end of PRBS signal application to the supervisory agent, the supervisor flags to the model ID agent to estimate a new process model. The estimated model is then sent to the appropriate agents. The supervisor then requests from the FDI agent to design the FDI filter based on the received process model. The FDI agent starts monitoring the external process for sensor and actuator faults. If the FDI agent detects a fault in the plant, the fault location, type, time, size and other information are reported back to the supervisor for further processing. In the case of a sensor fault, the FDI agent will also recommend the appropriate accommodation (correction) [26].

III. ICAM SYSTEM PROTOTYPE SIMULATION SCENARIO

To analyze the performance of the ICAM system prototype in terms of its logical behavior and its response to the external environment dynamics, a real-time simulation experiment was conducted for a time span of 30.92 minutes, and a sampling period of 150 milli-second. The pilot plant simulation model consists of 10 states, 5 manipulated variables, 5 controlled variables, and 17 auxiliary measured inputs and outputs (e.g., disturbances, product quality variables, etc...). The time-stamped raw data is sent to the statistical preprocessor for removing any discrepancies. The processed data is then sent to the model ID agent to identify new process models, and to the FDI agent for process fault diagnosis. After the systems starts up and updates its knowledge about the pilot plant state, a bias fault is applied to the liquid level sensor in the two-phase separator (refer to control loop LCL1 in figure 2). The size of the fault is $+15\%$ of the liquid level nominal value. The system behavior during startup and fault occurrence is discussed in the next sections, where we discuss the results of two process variables namely, the liquid level of the two-phase separator and the oil phase level in the three-phase separator. .

A. Simulation results

After the ICAM system supervisory agent starts executing its rule-base, other reactive agents are started and initialized. The pilot plant agent starts its simulation around a nominal value of $V = 146ft^3$, $P = 625psi$ for the two-phase separation sub-process and $V_{wat} = 77.5ft^3$, $V_{oil} = 46.5ft^3$, $P = 200psi$ for the three-phase separation sub-process. Since the ICAM system has no knowledge about the pilot plant agent (i.e., no dynamic model), it sends a message to the statistical pre-processor to check if the pilot plant is in steady state. Once it is in steady state, the supervisor then commands the control system of the pilot plant to apply a sufficiently exciting PRBS signal with an amplitude of 2% about the nominal operating point. This allows the model ID agent to collect enough data to identify the pilot plant model, which is then used by the FDI agent to design its parity vector filter. Having gained new knowledge about the current dynamic state of the pilot plant, the ICAM system now can start monitoring the pilot plant for any failures. Figure 4 illustrates the liquid volume and the associated outflow in the two-phase separator during this simulation experiment. Outliers

and missing data were applied to the two-phase separator measurements to simulate real-world data. The oil volume of the three-phase separator and the associated outflow is also shown in figure 5, where the PRBS signal is also applied soon after the beginning of the simulation experiment. The effect of the faulty volume sensor at t_{fault} = 13:36:42 on the oil volume in the three-phase separator is also shown.

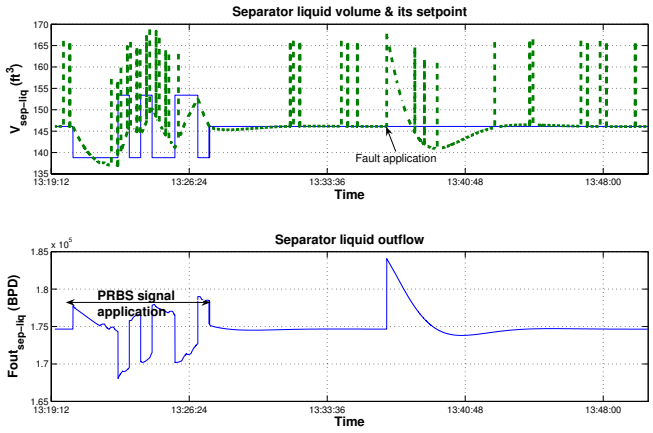

Fig. 4. Two-phase separator liquid volume logged at the pilot plant agent

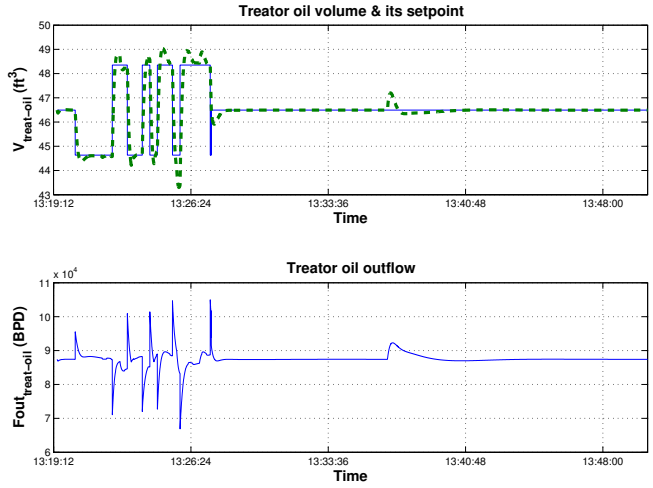

Fig. 5. Three-phase oil volume logged at the pilot plant agent

Raw data is received by the statistical pre-processing agent, which removes any outliers and corrects missing data by replacing it with the previous data value, as demonstrated by the clean liquid volume data record in figure 6 (compare with the top traces in figure 4). The statistical agent first checks if the pilot plant is in steady state to prevent applying the PRBS signal is a transient state. Apparently the pilot plant takes a time period of $t = 54.73sec$ to reach it steady state due to the plant small initial conditions, as shown in figure 6. Processed data is sent to the model ID agent during the PRBS signal application, after which a new process model can be estimated. Figures 7,8 show the two-phase liquid volume and the three-phase oil volume process variables collected during the PRBS signal application. Each process variable data record has a length of 300 seconds, which is the pre-specified PRBS signal application time.

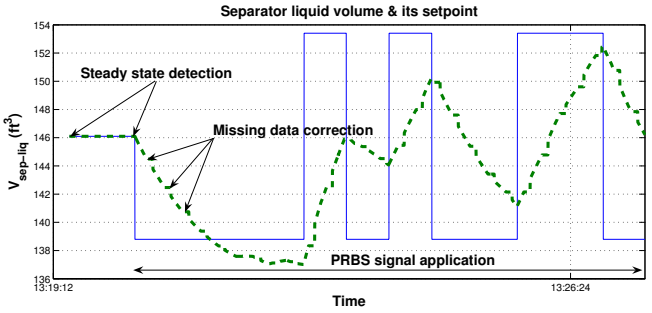

Fig. 6. Two-phase separator liquid volume logged at the statistical pre-processing agent

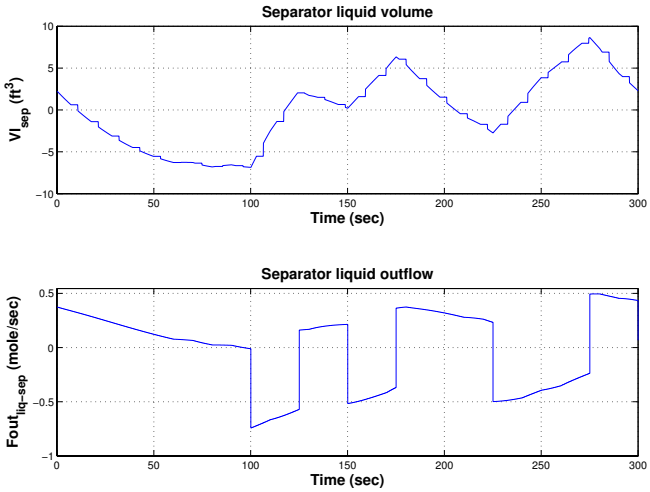

Fig. 7. Two-phase separator liquid volume logged at the model ID agent

Once the new process model is received by the FDI agent, then it can design its FDI filter and deploy it to diagnose plant faults. Figures 9,10 show the two-phase liquid volume and the three-phase oil volume process variables collected after the FDI filter is deployed. When a fault occurs, its effect can be noticed not only in the local control loop, but also downstream, which is seen as a disturbance in the three-phase oil volume control loop, as shown in figure 10. The FDI agent generates a generalized parity vector whose abnormal magnitude detects faults, and whose angle with respect to certain reference directions isolates them. When there is a fault, then the smallest angle indicates the approximate alignment of the parity vector with the reference direction of the particular fault. Hence the fault can be isolated based on the smallest angle after the fault detection. It is clear from the top plot in figure 11 that the general parity vector (GPV) magnitude increased significantly, which indicates that a fault occurred. Furthermore, the smallest angle after the fault detection instant is the one that corresponds to liquid volume sensor in the two-phase separator, as indicated by the dotted trace in the middle plot of figure 11(i.e., fault F1). The other GPV angles are larger than the faulty volume sensor

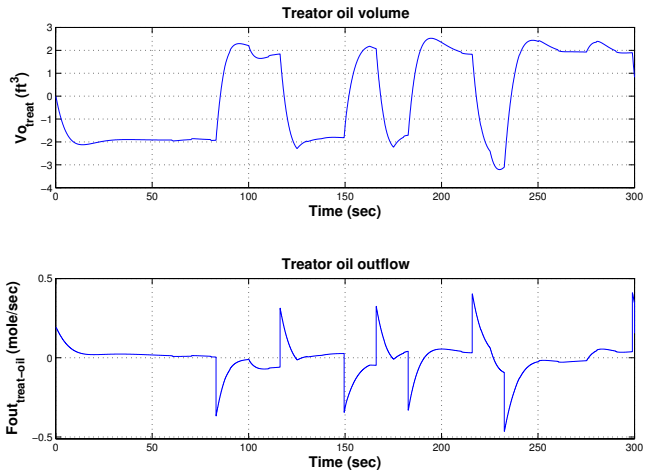

Fig. 8. Three-phase oil volume logged at the model ID agent

GPV angle, as indicated by the other traces in the middle and bottom plots of figure 11.

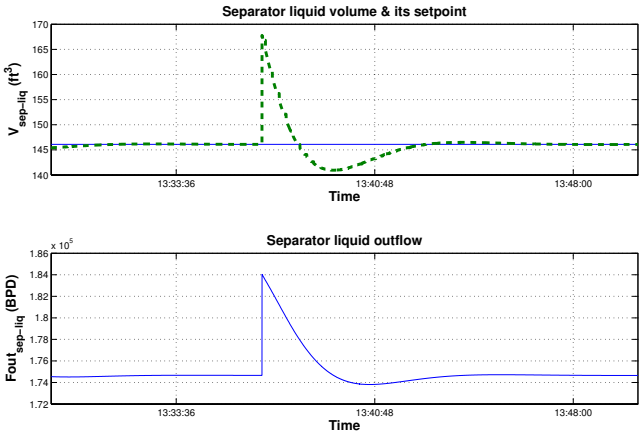

Fig. 9. Two-phase separator liquid volume logged at the FDI agent

The parity vector-based FDI technique is highly sensitive to process variable changes when there is no fault. This is because of the small size of the GPV vector in no-fault situations, so it can change its angle widely even in case of very small process variable changes, as indicated by the large variation of the GPV angles before fault occurrence in figure 11. The local decision-making logic of the FDI agent isolates the fault after its occurrence is signaled by the large GPV magnitude, as demonstrated by figure 12. It interesting to notice a fault # of -1 is indicated at the beginning of fault isolation task; -1 denotes an unknown fault. The FDI agent isolates faults when the process variables have reached an acceptable steady state level, so it is ineffective during the transient part of the fault dynamics. The FDI decision-making mechanism fails to isolate the faulty instrumentation between the times $t_1 = 13{:}37{:}55$ and $t_2 = 13{:}40{:}48$ due to the very small GPV magnitude as discussed earlier.

The FDI agent also estimates the fault size and sign and reports the fault information back to the supervisor

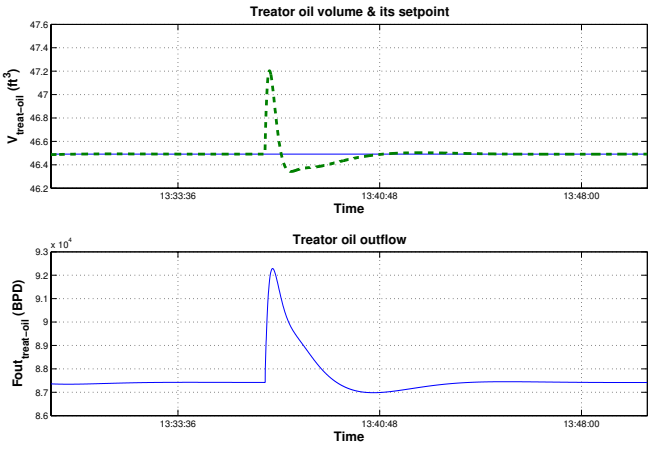

Fig. 10. Three-phase oil volume logged at the FDI agent

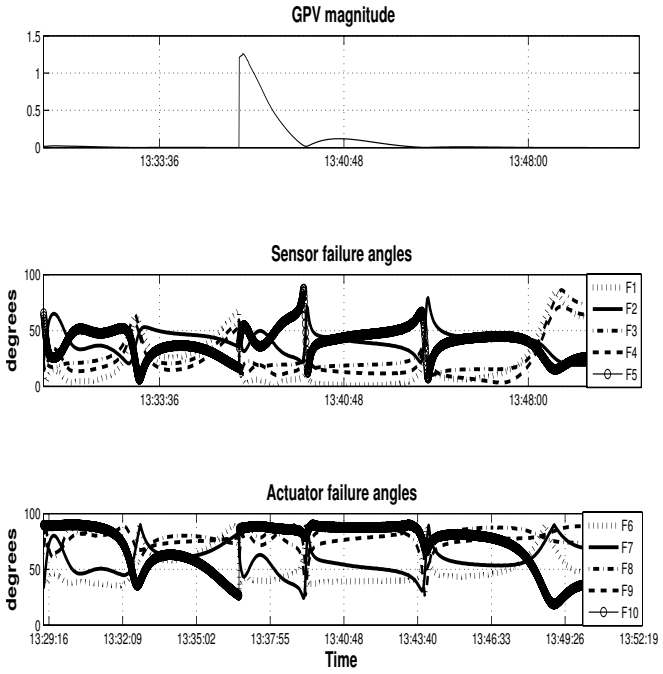

Fig. 11. FDI agent diagnostic signals

for further processing and actions. The corresponding fault information and recommended actions by the supervisor are shown by the attributes table of the FDI object in the G2 supervisory expert system in table I. The model status and the FDI design status attributes indicate that the FDI agent has received the process model and has deployed the designed FDI filter. The accommodation status attribute is the system recommendation, which is in this case to repair the faulty sensor. Since the fault type is bias and not of a ramp type, then recursive fault size estimation is not required as indicated by the last attribute of the table. Work is proceeding to accommodate sensor faults in the FDI agent [26]. Some of the attributes are related to the internal operation of the ICAM system, which will not be discussed.

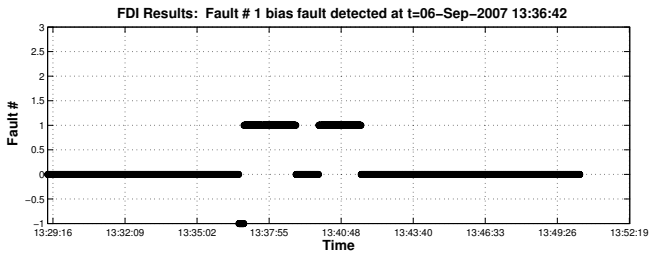

Fig. 12. FDI agent fault display

	FD AO BE C
R	3
Sta e	e
Mpi comm	
Mpi com size	4
G2 I nk status	e d
Mpi link status	e d
	a s
Simulation status	
Model status	ok
Fdi design status	ok
F a	f1
Fa lt sig	
Fa lt size	4 8
Fa lt typ	a
F a tion	"06-Sep-2007 13:36:42"
Accommodation status	repair liquid level sensor # 1
Fault size e commendation	not required

TABLE I
SUPERVISOR FDI FRAME

B. Performance analysis

Having verified and validated the ICAM system prototype functionality during the simulation scenario, it is crucial to analyze its real-time performance to pinpoint any computational bottlenecks and to verify the computation/communication coordination in each agent. We embedded a time-delay function of 100 milli-seconds in the pilot plant ordinary differential equation (ODE) solver to emulate a real-time clock functionality. The time-delay is less than the sampling period because the pilot plant ODE model evaluation takes about 50 milli-seconds every sampling period. Table II illustrates the profile of the oil separation plant agent which was simulated in real time. It is evident that the real-time clock functionality took the biggest execution time slot (i.e., about 61.36%). The evaluation of the oil separation ODE model took 29.19% of the total execution time, due to the nonlinear problem being solved every sampling period [34]. Communicating data to other agents and messages to the supervisory agent did not have a significant effect of the agent's performance, which indicates efficient communication/computation coordination. The approximate one execution cycle for this agent is around $150msec$. This validates our real-time simulation design decision to take the computational cost of the ODE model evaluation into account.

While computational functionalities dominated the separation plant agent, data communications with other agents in the statistical processing agent took the largest execution time slot, as shown in table III. This is due to synchronization with other agents during data reception. However, this is less significant on the agent performance when sending data to other agents (i.e., about 4.8% of the execution time), as was specified by the system design requirements [4], [3]. It is evident that there is a performance bottleneck in this agent due to data storage (around 12%). This can be rectified by adding a database agent to the system which stores the different data types across the ICAM system. Again the communication part with the supervisor had a minimum effect on performance.

Functionality name	Calls	Total Time	% Time
Real time simulation clock	11453	1253.053 s	61.26%
Separator ODE model evaluation	11453	597.10 s	29.19%
Raw data sending	11453	4.28 s	0.2%
Communication with supervisor	11453	1.3s	0.06%
Other functionalities		188.404 s	9.21%
Totals		2045.137 s	100%

TABLE II
THE PILOT PLANT AGENT PERFORMANCE PROFILE

Functionality name	Calls	Total Time	% Time
Raw data reception from pilot plant	11453	1476.829 s	72.3%
Processed data storage	11453	152.843 s	7.5%
Processed data sending	10802	98.559 s	4.8%
Raw data storage	11453	99.88 s	4.89%
Communication with supervisor	11453	6.33 s	0.3%
Other functionalities		216.322 s	10.59%
Totals		2041.763 s	100%

TABLE III
THE STATISTICAL AGENT PERFORMANCE PROFILE

When it comes to the model ID agent, the reception of processed data from the statistical processing agent had the biggest effect on performance (i.e., about 86.3% of the agent's execution time). While the process model estimation functionality took 4.8% of the agents' total execution time, communications with the supervisory agent had the least effect on performance, as shown in table IV. The FDI agent

had a similar profile of the model ID agent, in which data reception had 77.5% of execution time. We do notice here that data storage has a fairly undesirable effect of 7.4% on the FDI agent performance, as illustrated in table V. Table VI demonstrate the performance of the supervisory agent during the real-time system simulation. The G2 supervisory agent was in an idle state for almost 99.0% of the total simulation time, whereas communications with other agents had almost no impact of the agent's performance, as was specified for in the design requirements. The total performance of the ICAM system prototype during the real-time simulation was satisfactory and will be improved in future design stages.

Functionality name	Calls	Total Time	% Time
Processed data reception	10802	1756.774 s	86.3%
Process model estimation	1	88.092 s	4.3%
Communication with supervisor	11317	2.36 s	0.11%
Other functionalities		190.156s	9.34%
Totals		2035.021 s	100%

TABLE IV
THE MODEL ID AGENT PERFORMANCE PROFILE

Functionality name	Calls	Total Time	% Time
Processed data reception	10802	1584.122 s	77.5%
Communication with supervisor	10909	3.06s	0.14%
FDI variables storage	7410	76.03 s	3.71%
Processed data storage	10802	75.63 s	3.69%
Other functionalities		305.785	14.95%
Totals		2044.627 s	100%

TABLE V
THE FDI AGENT PERFORMANCE PROFILE

IV. SYSTEM DESIGN CHALLENGES AND LIMITATIONS

Designing an intelligent multi-agent system is very challenging task, as all agents are distributed and semi-autonomous. Although we proposed the colored petri nets approach to design the internal logic of the ICAM system agents in our development plan [3], we did a preliminary design the agents' internal logic in an *ad hoc* manner. We have faced some difficulties during the design stage of the ICAM system prototype. For example, the ICAM system crashed during early simulation runs due to communication deadlocks, in which two agents were trying to send messages to each other simultaneously. The problem was solved by imposing conditions on communicating agents to prevent such deadlocks.

Computation/communication coordination was another design problem, in which computation and communication

Functionality name	Total Time	% Time
Idle time	1899.00 s	99.11%
Scheduling time	3.637 s	0.19%
Communication with agents	4.677 s	0.24%
Other functionalities	8.686 s	0.45%
Totals	1916.00 s	100%

TABLE VI
THE SUPERVISORY AGENT PERFORMANCE PROFILE

code chunks were not ordered correctly in the agent code. For example, we combined the process model estimation (computation task) and sending the estimated model to other agents (communication task) into one functionality in the model ID agent, which proved to be a design flaw. Model estimation took a long time (i.e., over one minute), during which other agents were locked waiting for the estimated model due to synchronization failure. The problem was solved by separating the one functionality into two separate computation and communication functionalities (i.e., separate agent states) and modifying other agents accordingly. Although some design flaws had to be corrected, the ICAM system prototype acted as a set of distributed stochastic colored petri nets during real-time simulation. This implies that a careful agent design should be done along with a thorough system logical behavior analysis.

The plant data characteristics also had a major impact on the ICAM system performance. For example, the ICAM system prototype is not robust against noisy data due to the design of the data differentiation-based steady state detection algorithm. Likewise, the general parity vector (GPV) based FDI algorithm is not robust to noise, which significantly affects the fault isolation task in moderate to high noisy data situation. Detection and isolation of fast dynamics faults (e.g., faulty gas pressure sensor) is another limitation of the ICAM system prototype. The outlier removal algorithm in the statistical processing agent treats fast dynamics faults as outliers, which changes the nature of processed data sent to the FDI agent. Data filtering also may change the data dynamics, which may have an impact of the system performance.

The multi-threaded stochastic execution nature of the supervisory agent's rule-base adds another complexity level to the ICAM system design process. The ICAM system prototype design is still at early stages and as the ICAM system design matures, most of these limitations can be eliminated or minimized.

V. CONCLUSIONS

We have demonstrated good progress and described lessons learned in the design and development of the ICAM

system. A system prototype was built and simulated in real-time. These results verified the system's logical behavior in normal and faulty process situations. A detailed system performance analysis was done, which revealed some computation bottlenecks to be resolved in future design stages. Although the preliminary ICAM system prototype design has limitations, the experimental results supported our requirements analysis done in previous work. Moreover, our system design approach can be exploited to develop and rapidly prototype real time distributed multi-agent systems. We believe that the ICAM system will pave the way to real intelligent multi-agent systems for many applications.

VI. Acknowledgement

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

References

[1] J. H. Taylor and A. F. Sayda, "Intelligent information, monitoring, and control technology for industrial process applications," in *The 15th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, Bilbao, Spain, July 2005.

[2] ——, "An intelligent architecture for integrated control and asset management for industrial processes," in *Proc. IEEE International Symposium on Intelligent Control (ISIC05)*, Limassol, Cyprus, June 2005, pp. 1397–1404.

[3] A. F. Sayda and J. H. Taylor, "An implementation plan for integrated control and asset management of petroleum production facilities," in *IEEE International Symposium on Intelligent Control ISIC06*. Munich, Germany: IEEE, October 4-6 2006, pp. 1212–1219.

[4] ——, "An intelligent multi agent system for integrated control and asset management of petroleum production facilities," in *In Proc. of The 17th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, Philadelphia, USA, 18-20 June 2007, pp. 851–858.

[5] ——, "Toward a practical multi-agent system for integrated control and asset management of petroleum production facilities," in *IEEE International Symposium on Intelligent Control (ISIC)*, Singapore, 1–3 October 2007.

[6] 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.

[7] R. J. Patton, "Fault-tolerant control systems: The 1997 situation," in *IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes*, R. J. Patton and J. Chen, Eds., vol. 3. Kingston Upon Hull, UK: IFAC, August 1997, pp. 1033–1054.

[8] P. M. Frank and B. Köppen-Seliger, "New developments using AI in fault diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 10, no. 1, pp. 3–14, 1997.

[9] R. L. Small and C. W. Howard, "A real-time approach to information management in a pilot's associate," in *Proceedings of Digital Avionics Systems Conference*. IEEE/AIAA, 14–17 Oct 1991, pp. 440–445.

[10] S. B. Banks and C. S. Lizza, "Pilot's associate: a cooperative, knowledge-based system application," *IEEE Expert*, vol. 6, no. 3, pp. 18–29, June 1991.

[11] C. A. Miller and M. D. Hannen, "Rotorcraft pilot's associate: Design and evaluation of an intelligent user interface for cockpit information management," *Knowledge-Based Systems*, vol. 12, no. 8, pp. 443–456, Dec 1999.

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

[13] J. Schmalzel, F. Figueroa, J. Morris, S. Mandayam, and R. Polikar, "An architecture for intelligent systems based on smart sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 4, pp. 1612–1616, August 2005.

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

[15] S. Cauvin, "Chem: Advanced decision support system for chemical/petrochemical manufacturing processes," in *CHEM Project Annual Meeting*. Lille, France: http://www.chem-dss.org/, 25-26 March 2004.

[16] E. L. Cochran, C. Miller, and P. Bullemer, "Abnormal situation management in petrochemical plants: can a pilot's associate crack crude," in *Proceedings of the 1996 IEEE National Aerospace and Electronics Conference, NAECON*, vol. v2. Dayton, KY, USA: IEEE, Piscataway, NJ, USA, May 20-23 1996, pp. 806–813.

[17] A. Ogden-Swift, "Reducing the costs of abnormal situations ...the next profit opportunity," in *IEEE Advanced Process Control Applications for Industry Workshop (APC2005)*, Vancouver, Canada, May 2005.

[18] M. Omana and J. H. Taylor, "Robust fault detection and isolation using a parity equation implementation of directional residuals," in *IEEE Advanced Process Control Applications for Industry Workshop (APC2005)*, Vancouver, Canada, May 2005.

[19] W. Larimore, in *Multivariable System Identification Workshop*. Fredericton, New Brunswick: University of New Brunswick, 31 October – 2 November 2005.

[20] C. Smith, C. Gauthier, and J. H. Taylor, in *Petroluem Applications of Wireless Sensors (PAWS) Workshop*. Sydney, Nova Scotia: Cape Breton University, 22–23 August 2005.

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

[22] M. Omana and J. H. Taylor, "Enhanced sensor/actuator resolution and robustness analysis for fdi using the extended generalized parity vector technique," in *Proc. of American Control Conference*. Minneapolis, Minn.: IEEE, 14-16 June 2006, pp. 2560–2566.

[23] J. H. Taylor and M. Laylabadi, "A novel adaptive nonlinear dynamic data reconciliation and gross error detection method," in *Proc. of IEEE Conference on Control Applications*. Munich, Germany: IEEE, October 4-6 2006, pp. 1783–1788.

[24] M. Laylabadi and J. H. Taylor, "Anddr with novel gross error detection and smart tracking system," in *12th IFAC Symposium on Information Control Problems in Manufacturing*. Saint-Etienne, France: IFAC, May 17-19 2006.

[25] M. Omana and J. H. Taylor, "Fault detection and isolation using the generalized parity vector technique in the absence of a mathematical model," in *IEEE Conference on Control Applications (CCA)*, Singapore, 1-3 October 2007.

[26] J. H. Taylor and M. Omana, "Fault detection, isolation and accommodation using the generalized parity vector technique," *submitted to the IFAC World Congress*, Seoul, Korea, July 6–11 2008.

[27] N. Viswanadham, J. H. Taylor, and E. C. Luce, "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.

[28] M. Omana, "Robust fault detection and isolation using a parity equation implementation of directional residuals," Master's thesis, University of New Brunswick, 2005.

[29] P. H. Menold, R. K. Pearson, and F. Allgower, "Online outlier detection and removal," in *Proc. of the 7th Mediterranean Conference on Control and Automation (MED99)*, Haifa, Israel, June 28–30 1999.

[30] P. vanOverschee and B. DeMoor, *Subspace Identification of Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996.

[31] W. E. Larimore, "Canonical variate analysis in identification, filtering and adaptive control," in *In Proc. 29th IEEE Conference on Decision and Control*, Honolulu, 1990, pp. 596–604.

[32] L. Ljung, *System Identification - Theory For the User*, 2nd ed. Upper Saddle River, N.J.: PTR Prentice Hall, 1999.

[33] *G2 for Application Developers Reference Manual*, 8th ed., GenSym Corporation, Burlington, Massachusetts, December 2005.

[34] A. F. Sayda and J. H. Taylor, "Modeling and control of three-phase gravity separators in oil production facilities," in *the American Control Conference (ACC)*, New York, NY, 11-13 July 2007.