Athens Institute for Education and Research
ATINER

ATINER's Conference Paper Series
COM2012-0167

Distributed On-line Safety Monitor Based on Assessment Model and Multi-agent System

Amer Dheedan
Computer Science Department
University of Hull
UK

Yiannis Papadopoulos
Computer Science Department
University of Hull
UK

Darryl Davis
Computer Science Department
University of Hull
UK

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209
Email: info@atiner.gr URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm

Printed in Athens, Greece by the Athens Institute for Education and Research.

ISSN **2241-2891**
12/09/2012

An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. The papers published in the series have not been refereed and are published as they were submitted by the author. The series serves two purposes. First, we want to disseminate the information as fast as possible. Second, by doing so, the authors can receive comments useful to revise their papers before they are considered for publication in one of ATINER's books, following our standard procedures of a blind review.

Dr. Gregory T. Papanikos
President
Athens Institute for Education and Research

This paper should be cited as follows:

Dheedan, A., Papadopoulos, Y.and Davis, D. (2012) **"Distributed On-line Safety Monitor Based on Assessment Model and Multi-agent System"** Athens: ATINER'S Conference Paper Series, No: COM2012-0167.

Distributed On-line Safety Monitor Based on Assessment Model and Multi-agent System

Amer Dheedan
Computer Science Department
University of Hull
UK

Yiannis Papadopoulos
Computer Science Department
University of Hull
UK

Darryl Davis
Computer Science Department
University of Hull
UK

Abstract

On-line safety monitoring, i.e. tasks of fault detection and diagnosis, alarm annunciation, and fault controlling, is an essential task in safety critical systems. Recently, monitors that use knowledge derived in off-line safety assessments have been developed. The motivation of this work was to bring the wealth of knowledge derived during such assessments in the service of operators. Although the concept has demonstrated value, the centralised architecture of these monitors has prevented the application of this approach in larger scale systems of a distributed nature. On the other hand, recent work on multi-agent systems shows that the distributed reasoning paradigm could cope with the nature of such systems.
This paper brings together these two strands of work and develops a distributed on-line safety monitor. The monitor consists of a multi-agent system – effectively a set of collaborating belief-desire-intention (BDI) agents – which are informed by a distributed monitoring model derived from a model-based safety assessment. Guided by the knowledge of the monitoring model, agents are hierarchically deployed and work collaboratively to integrate and deliver safety tasks, both locally at the sub-system levels and globally at the higher level of the monitored system. The benefits of the proposed monitoring scheme include increasing the flexibility, composability and extensibility of monitoring.

Keywords: Fault Detection and Diagnosis, Alarm Annunciation, Fault Controlling.

Contact Information of Corresponding author:

1. INTRODUCTION

Research on safety monitoring has recently focused on model-based approaches, in which knowledge about the normal behaviour and failures of a system contained in system models is combined with real-time observations of the system and used to drive the reasoning on the safety of the system while in operation. Models such as state-machines (Papadopoulos, 2003; Eo et al, 2001), goal trees (Larsson 1994), signed direct graph (Dong et al, 2010) and fault trees (Papadopoulos, 2003; Peng et al, 2007) have been used for the purposes of real-time monitoring.

Papadopoulos (2003) has developed a monitor that exploits monitoring knowledge derived from the application of a semi-automated off-line safety assessment method called Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS). In this approach, the monitoring model is composed of: (a) A hierarchy of state-machines that describes the behaviour of the system capturing normal and abnormal transitions of the system and its sub-systems; (b) A number of fault trees which relate the symptoms of failure to their underlying causes in the architecture of the system. A variant of this monitor which maintains fault trees but replaces state-machines with a control chart has been proposed by Peng et al (2007).

This work has been motivated by the observation that, in current industrial practice, vast amounts of knowledge derived in off-line safety assessments ceases to be useful following the certification and deployment of a system. The aim is to utilise this knowledge in the service of operators for the purposes of on-line safety monitoring. Although success has been demonstrated, one of the difficulties with these monitors is their centralised nature in which all components of a plant are delegated to a single reasoning agent. This does not align well with the distributed nature of most modern systems. Systems, such as nuclear power plants and aircrafts, are typically large, complex and show dynamic behaviour that includes complex mode and state transitions. As a result, such systems are distributed and ideally would also need distributed and collaborative reasoning agents to monitor their safety. Recent work shows that such distribution of reasoning is possible (Mendes et al, 2009; Ng and Srinivan, 2010).

This paper explores the synthesis of these two strands of work on model-based safety assessment models and multi-agent systems to develop a distributed on-line safety monitor. The monitor delivers a range of monitoring tasks extending from fault detection and diagnosis to alarm annunciation and fault controlling through a set of collaborating monitoring agents.

The rest of this paper is organised as follows: section two presents a generic abstraction of a complex system as a hierarchy of subsystems and components and discusses how dynamic behaviour and failure could be represented within this general structure for the purposes of monitoring. Section three presents the position, role, and constituents of a monitor, represented within the scheme of this general abstraction. Section four discusses the application to an aircraft fuel system, while section five draws a conclusion and proposes further work.

2. THE MONITORED SYSTEM

To manage large scale and complexity, modern engineering systems are typically organised as hierarchies of subsystems and components that perform observations and control exploiting the information and control capabilities of lower level subsystems and components. For the purposes of this work, this hierarchical organisation is arranged in a number of levels, across which elements appear as parents, children and siblings. As shown in Figure 1, we classify those levels into three types: the lowest level (level*0*) is classified as the basic components level - this is the physical layer where sensors and actuators exist and where monitoring and control interfaces to the environment. The upper levels extending from level*1* to level*n-1*, are classified as Sub-system levels – they are layers were the function of increasingly higher level subsystems is defined and implemented via affecting the behaviour of lower level elements. The top level (level*n*) is classified as the system level.

To model dynamic behaviour in this scheme, it would be useful to examine the way in which behavioural transitions are initiated. Typically, transitions are outcomes of, firstly, normal conditions in which the system engages its components in different structures and behaviours, so it delivers different functionalities. The conditions which trigger such transitions always come from basic components at level*0* since this is where the system interacts with users and the environment. For example, during the cruising of an aircraft, navigation sensors may convey signals to the navigator sub-system (NS) which in turn calculates positional information using those signals and notifies the flight control computer (FCC). Assuming that it is time for launching the approach to an airport, FCC accordingly instructs the power plant sub-systems (PPS) to achieve the required thrust and in turn the surface hydraulic controller (SHC) to achieve the required body motions at the low level. Abstract states at system level like taxiing, cruising, and approaching are typically called operational *modes*.

Secondly, dynamic behaviour could be an outcome of a fault or the response to a fault in a fault tolerant architecture. Fault tolerance is typically implemented as (a) active fault-tolerance, in which the system takes action to limit or reverse the effects of the fault, e.g. the fault of one engine of a two-engine aircraft can be compensated by adjusting the operation of the other engine; (b) passive fault-tolerance, in which the system has the ability to tolerate the fault for a while, e.g. faults that are caused by software error, ionisation radiation, electromagnetic interference, or hardware failure that can be corrected within a short interval by restarting the relevant component or by isolating the faulty component and starting up a redundant one.

It could, therefore, be said that during a mode, a system may appear in different health *states* which can be classified into two types. The first is the Error-Free State (EFS) in which the system or a sub-system functions correctly and delivers the intended function. The second type is the Error State (ES), which in turn is classified into three different states: (a) Temporary Degraded or Failure State (TDFS), in which there is one or more functional failures, but corrective measures can be taken to transit to another state where functionality has been recovered; (b) Permanent Degraded State (PDS), in which an

uncorrectable permanent fault occurs, but the safe part of the functionality can still be delivered; (c) the Failure State (FS) in which the intended function is totally lost.

Given that they provide the impetus for state transitions, events that are initiated at the lowest level (*level0)* by basic components play a key role in tracking the behaviour of a system. To track behaviour, such events should be continuously monitored. The best hierarchical level at which events could be monitored effectively is level*1*. This can easily be justified; at level*1* the occurrence of events could be identified as either normal or abnormal, e.g. the decreasing of velocity and altitude seem normal when the FCC has already launched the approach phase of the aircraft. Excluding knowledge about the modes and focusing only on the measurements provided by the relevant sensors would certainly result in misinterpreting system behaviour, i.e. decreasing velocity and altitude would appear as a malfunction and a misleading alarm would accordingly be released. Level*1* is also preferable since at this level malfunctions are detected while in their early stages. Finally, due to the potentially large number of the basic components, monitoring events at the level*0* may be computationally expensive or even unworkable, whereas level*1* offers a space where a degree of simplifying abstraction already exists.

3. DISTRIBUTED ON-LINE SAFETY MONITOR

As shown in Figure 1, the monitor takes a position between the system and the operator interface(s). During normal conditions, the monitor provides simple feedback about those conditions. In abnormal conditions it delivers fault detection and diagnosis, alarm annunciation and fault controlling.

The monitor consists of a distributed monitoring model, which holds the reference knowledge; and, secondly, a multi-agent system, effectively a set of Belief-Desire-Intention (BDI) agents. Agents are deployed and provided with their portions of the monitoring models to reason locally about failures that occur at the level of the subsystem that they monitor. They are also integrated globally to deliver system level safety monitoring functions.

3.1 Distributed Monitoring Model

As the state of the system and its subsystems evolves with time, agents should be able to track the behaviour of the monitored components over different states, both error free and error states. State-machines provide a widely used tool to describe dynamic behaviour. In this approach the first component of the monitoring model is a hierarchy of state-machines which propagates transitions of the system and its subsystems from normal to temporary degraded and failed states – see Figure 1.

Practically, there are always relationships among sub-systems, at the same level of abstraction, and with other subsystems including parent and child elements and such relationships can be implemented in the hierarchical state-machines that compose the frame of the monitoring model. For instance, the flight control computer (FCC) and power plant sub-system (PPS) are siblings and have the same parent; the aircraft control sub-system (ACS). During the cruising mode, an event may trigger a transition to the approaching mode in the

state-machine of the FCC. That event in turn triggers a state transition in the state-machine of ACS, i.e. the parent, that also leads ACS to the approaching mode, while this new event now triggers a state transition in the state-machine of the PPS, i.e. a child, leading PPS also to approaching mode.

Similarly, error states of the children could also trigger transitions in the state-machines of the parents and vice versa. For example, the failure state of an engine of a two-engine aircraft triggers a transition to the permanent degraded state in the state-machine of the PPS. This event, in turn, triggers a transition to a new state of the operating engine in which the lost functionality of the faulty engine is compensated.

In the state-machine of the sub-systems of level*1*, events represent (a) monitored conditions that indicate normal, failure or corrective events in the (level*0*) process; (b) transitions of the parent subsystem that have a significant effect at low level subsystems at level*1*. In the state-machine of a sub-system of the levels from level*2* to level*n-1*, events appear as error-free and error states of the parent and children. Finally, in the state-machine of the system, i.e. level*n*, events appear as error-free and error states only of the children.

Knowledge about the normal behaviour of the system and its sub-systems can be obtained from design models, such as Data Flow Diagrams (DFD), Functional Flow Block Diagram (FFBD) or Unified Model Language (UML). Knowledge about abnormal behaviour, i.e. error stares, abnormal events, assessment, guidance, and corrective measures, can be obtained by applying the Functional Failure Analysis (FFA) or HAZard and OPerability study (HAZOP) techniques on those models.

During the monitoring time, agents cyclically monitor only those events whose occurrence triggers transitions from the current state; every such cycle is called monitoring cycle. As such, the computational load of the agents is kept to a minimum and prompt responses to the occurrence of state relevant events are established. In the state-machines of the sub-systems of level*1*, every failure event would be associated with (a) an alarm statement that would be quoted and provided to the operators upon the occurrence of the failure event; (b) corrective measures that can applied to control the failure; (c) diagnosis, if the failure and the underlying cause are in one-to-one relationship the cause would be associated, otherwise a further diagnostic process should take place by exploring possible causes. Note, that some corrective measures might be applied only after diagnosing the underlying causes.

For the purposes for diagnosisfault trees can be used as they logically record the propagation paths that connect symptoms to underlying causes. A fault tree could automatically be constructed from a variant of Failure Mode and Effect Analysis (FMEA) applied to the basic components in reference to each failure event that appears in the state-machines of level*1*'s sub-systems. Corrective measures could also be incorporated in the failure mode nodes of the fault tree.

Across the hierarchical state-machines, every state encloses two fields: (a) assessment of the associated conditions; (b) guidance on the preferable directing instructions. Knowledge of those fields can be obtained from a HAZOP study.

Overall, the monitoring model can be constructed using a range of design models and results from a variety of classical safety analysis techniques including FFA, and FMEA. However, state-of-the-art techniques, such as HiP-HOPS, already produce an electronic safety assessment model that incorporates a hierarchy of state-machines and fault trees. This assessment model could be be adopted as a distributed monitoring model after certain extensions:

a) formalising the events of the state-machine and the symptoms as monitoring expressions that could be mechanically evaluated by the monitor in real time.
b) associating failure events with alarms, and corrective measures
c) augmenting state-machines with assessment and guidance fields and the diagnostic model with the required corrective measures

In the rest of the paper we assume a monitoring model derived from HiP-HOPS.

3.2 Monitoring Expressions

Events in state-machines and fault trees are formalised with executable monitoring expressions. The computational evaluation of the expressions, verifies the occurrence of the events in real time. An event is expressed as a constraint. In its simple form, a constraint consists of three main parts: (a) observation which is either a state of a child or the parent or sensory measurement defined by the identifier of the relevant sensor; (b) a relational operator – equality or inequality; (c) a threshold whose violation means that the event has occurred. Thresholds might appear as a numerical or Boolean value.

The formalisations of events in the state-machine of level*1*'s sub-systems and the symptoms of the diagnostic model might require more complicated forms of constraints that incorporate (a) observation that should be calculated over a number of sensory measurements; (b) two operational operators, when the threshold is a range of values rather than a single value; (c) a threshold that represents a sensory measurement or a calculation of more than one measurement. Moreover, observations and the threshold might be calculated to find the average change of a quantity over an interval (Δt), i.e. differentiation, or the cumulative effect of integration of parameters over time. Such calculations necessitate holding sensory measurements over the time, i.e. keeping historical measurements. Updatable buffers that are updated systematically are used for this purpose.

Typically, sensors may deliver temporary spurious measurements because of (a) their own faults; (b) mode changes, which might be followed by an interval of unsteady behaviour before the monitored parameters stabilise in new control set-points. Perhaps the best way to filter out such measurements is by evaluating the expressions successively over a filtering interval and based on a number of measurements. The final result of that evaluation is obtained by making cumulative conjunctions across successive evaluations. If the final result is true, that means the delivered measurements remain relatively constant over the filtering interval. Hence, the occurrence of an event like a permanent failure can be distinguished from transient anomalies. The filtering interval of

every expression depends on the conditions that may result in spurious measurements.

A simple three-value logic that incorporates a third 'Unknown' truth value is also employed to save evaluation time and produce prompt results while filtering spurious measurements and in the context of incomplete sensory data without violating the evaluation logic. Consider for example the following expressions:

Expression1 OR T(Expression2, Δt) (1)
Expression1 AND T(Expression2, Δt) (2)

In the above the expression T(Expression2, Δt) evaluates to true only when Expression2 holds true for an interval of time that extends time Δt in the past. The truth value of T(Expression2, Δt) is unknown until Δt has elapsed. However, in the approach followed here (1) or (2) can be evaluated before Δt has elapsed, taking advantage of the fact that the disjunction of True with Unknown is True and the conjunction of False with Unknown is False.

3.3 Multi-agent System

As shown in Figure 1, agents are deployed over the sub-systems and the system, and appear as a number of sub-system monitoring agents and one system monitoring agent. Figure 2 shows a general illustration of a monitoring agent. By perceiving the operational conditions and exchanging messages with each other, each agent obtains up-to-date beliefs, deliberates among its desires to commit to an intention and performs a means-ends process to select a course of action, i.e. plan. The selected plan is implemented as actions towards achieving the monitoring tasks locally and as messages sent to other agents towards achieving global integration.

Each subsystem agent of level*1* updates its belief base by perceiving (a) its own portion of the monitoring model which consists of a state-machine and set of fault trees; (b) sensory measurements that are taken to instantiate and evaluate monitoring expressions; (c) messages that are received from the parent to inform the agent about the new states and the siblings, in which they either ask for or tell the given agent about global measurements, as there might be a need to share measurements globally. The main desires of a subsystem agent of level*1* are to monitor the local conditions of the assigned sub-system and to collaborate globally with its parent and siblings. On the achievement of the local desire, the intentions are to track the behaviour of the sub-system and to provide the operators with alarms, assessment, guidance, and diagnoses and control faults. On the achievement of the global desire, the intentions are to exchange messages to (a) inform the parent about the new states; (b) tell or ask the siblings about global measurements.

Each subsystem agent of the intermediate level (levels extending from level*2* to level*n-1*) updates its belief by perceiving (a) its own portion of the monitoring model, which consists of a state-machine of the assigned sub-system, and (b) messages received from the parent and the children to inform them about their new states. The main desires of each of these agents are to monitor the local conditions of the assigned sub-system and to collaborate

globally with its parent and child agents. On the local desire, the intentions are to track the behaviour of the sub-system and to provide the operators with assessment and guidance. On the global desire, the intention is to exchange messages with the parent and child agents to inform each other about new states.

The perceptions, desires and intentions of the system agent are similar to those of the subsystem agents of the intermediate levels. The only difference is that system agent has no parent to exchange messages with.

According to the Prometheus approach and notation for developing multi-agent system (Padgham and Winikoff, 2004), Figure 3 shows the collaboration protocols among agents to track the behaviour of the monitored system. Figure 4, shows the collaboration protocol among the Subsystem agents of level*1* in which they share their sensory measurements.

4. CASE STUDY: AN AIRCRAFT FUEL SYSTEM (AFS)

Figure 5 gives an illustration of an example aircraft fuel system. The system functions to maintain safe storage and even distribution of fuel in two modes. The first is the consuming mode in which the system provides fuel to the port and starboard engines of a two-engine aircraft. The second is the refuelling mode. During the consuming mode and to maintain the central gravity and stability, a control scheme applies a feedback-control algorithm to ensure even fuel consumption across the tanks. Another algorithm is applied similarly to control the even distribution of fuel injected from the refuelling point to the tanks during the refuel mode. The system is arranged in four sub-systems: a central deposit (CD), left and right wing (LW, RW) deposits and an engine feed (EF) deposit which connects fuel resources to the two engines. An active fault-tolerant control strategy is implemented; specifically, in the presence of faults there are alternative flow paths to connect the two engines into the available fuel resources.

As shown in Figure 6, five monitoring agents are deployed: four agents to monitor the four sub-systems; EF_MAG, CD_MAG, LW_MAG, and RW_MAG. The fifth agent is AFS-MAG which monitors the entire fuel system. The monitor is implemented using the Jason interpreter which is an extended version of AgentSpeak programming language (Bordini et al, 2007).

Among the faults that have been injected to test the monitor is a structural leak in the inner tank of the left wing deposit. The fault is detected by the agent LW_MAG after verifying the occurrence of the following expression:

$$\big(LL1(T-5) - LL1(T)\big) > \int_{T-5}^{T} \big(FL1(t) + FL2(t)\big)\, dt + 0.06$$

Where:

$\big(LL1(T-5) - LL1(T)\big)$: is the reduction in the inner tank over an interval that always extends from T-5 seconds in the past to current time T.

$\int_{T-5}^{T} \big(FL1(t) + FL2(t)\big)\, dt$: is the fuel volume that has been (a) drawn from the inner tank by pump PL1 over the same interval; (b) drawn or added

by pump PL2 over the same interval. 5 seconds is the time required to verify a structural leak.

0.06: is the maximum allowable discrepancy.

When the expression is evaluated to true, LW_MAG responds as follows: (a) releasing an alarm warning about a leak; (b) controlling the event locally by isolating the LW deposit (switching pump PL1 off and closing valves VL1 and VL2) and jettisoning the fuel into the atmosphere by opening valve VL3 and starting pump PL3; (c) as this fault is in one-to-one relationship with its underlying causes, structural leak is firmly diagnosed as the cause; (d) executing the occurred event on the state machine moving to a new temporary degraded state; (e) providing the pilots with the corresponding assessment and guidance; (f) sending a message to the parent agent (AFS_MAG) to be informed about the local transition to the new state.

The AF system agent receives the new state, executes it on the behavioural model, transits to a new state (PDFS) and sends a message to inform the child agents accordingly. Each of the three other child agents (CD_MAG, EF_MAG and RW_MAG) applies new flow rates from its deposit to compensate for the isolated flow of the LW deposit and transits to a new state from which assessment and guidance can also be provided to the pilot. After jettisoning all the fuel, LW_MAG achieves (a) switching pump PL3 off and closing valve VL3; (b) transiting to failure state in which the LW is confirmed to be shut down safely.

An inadvertent closure of valve VF2 has also been injected to test the monitor. The fault is detected by agent EF_MAG after verifying the following timed expression:

$$|FF1| < 0.03 \text{ for } 4 \text{ sec}$$

where: 0.03 is the possible bias of flow meter FF1and 4 sec is the filtration interval of the expression.

After detecting the fault, EF_MAG react with (a) announcement of an alarm of “Port Engine is not Fed”; (b) verification that diagnosis process is needed, as this fault is in one-to-many relationship with causes, application of diagnosis and fault controlling according to the diagnosed cause; (c) transition to a new state (TDFS) from which the pilot is provided with assessment of “Currently port engine is not fed while starboard engine is fed normally” and guidance of “fault controlling is in progress”; (b) communication of the new state to the parent agent (AF system agent). As this state does not trigger state transition, AF takes no action.

Before, launching a monitoring cycle for the new state, FE_MAG updates the relevant fault tree, whose top node encloses the verified expression, to be traversed from top to bottom. For this traversal, FE_MAG combines blind-depth-first and heuristic search strategies. Assuming that an inadvertent closure of valve VF2 is diagnosed as the underlying cause, the associated corrective measure is to reopen the valve. Accordingly, EF_MAG takes the measure and launches a monitoring cycle for the new state. If the measure succeeded in rectifying the conditions and resuming the flow to the port engine, then

EF_MAG transits back to the error-free state and confirms that to the pilot. Failure of the measure leads EF_MAG to (a) open valve VF3 to cross feed both engines from the rear tank; (b) transit to a new state (PDS) from which the pilot is provided with assessment of "both engines are fed from the rear tank" and guidance "none"; (c) communicate the new PDS to the parent AF system agent. The parent, in turn, transits to the corresponding state (PDS) and communicates it to the rest of the child agents. Each of the three child agents CD_MAG, LW_MAG and RW_MAG, applies a new flow rate to its sub-system to cope with cross feeding both engines from the rear tank.

5. CONCLUSION AND FUTURE WORK

This paper proposed a distributed monitor based on a multi-agent system and knowledge derived from safety assessment. Agents exploit this knowledge to deliver fault detection and diagnosis, alarm annunciation and fault controlling. An aircraft fuel system has been used to demonstrate and test the monitor.

This work addresses limitations in earlier work on safety monitoring by proposing a distributed and more flexible monitoring scheme. The architecture proposed is generic and applicable in a variety of safety-critical systems and contexts, such as those used in numerous transport industries or industrial processes.

Two open research issues remain. Firstly, the quality of the deliverable tasks depends mainly on the integrity and consistency of the monitoring model. The validation of the model, therefore, is an area for further research. Secondly, more work is needed on dealing with uncertainty in diagnosis. The incorporation of Bayesian probabilities and Bayesian network modelling is suggested for future investigation.

REFERENCES

Bordini R., Hubner J., & Woorldridge M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason.* UK, Chichester: Wiley.

Dong G., Chongguang W., Beike Z., & Xin M. (2010). 'Signed directed graph and qualitative trend analysis based fault diagnosis in chemical industry.' *Chinese Journal of Chemical Engineering* 18(2): 265-276.

Eo S. Y. , Chang T. S., Lee B. , Shin D., & Yoon E. S. (2001). 'Function-behavior modeling and multi-agent approach for fault diagnosis of chemical processes.' *Computer Aided Chemical Engineering* 9 (2001): 645-650.

Larsson J. E. (1994). **'**Diagnostic reasoning strategies for means-end models.' *Automatica* 30(5): 775-787.

Mendes, M., Santos B. & Costa J. (2009). 'A matlab/Simulink multi-agent toolkit for distributed networked fault tolerant control systems.' Paper presented at the 7th IFAC symposium on Fault Detection, Supervision and Safety of Technical Processes, 30 June - 3 July, in Barcelona, Spain.

Ng, Y.S. & Srinivan, R. (2010). 'Multi-agent based collaborative fault detection and identification in chemical processes.' *Engineering Applications of Artificial Intelligence* 23(6): 934-949.

Padgham L & Winikoff M 2004. *Developing Intelligent Agent Systems: a Practical Guide*. UK, Chichester:Wiley.

Papadopoulos Y. (2003). 'Model-based system monitoring and diagnosis of failures using state-charts and fault trees.' *Reliability Engineering and System Safety* 8(3): 325-341.

Peng H., Shang W., Shi H., & Peng W. (2007). 'On-Line Monitoring and Diagnosis of Failures Using Control Charts and Fault Tree Analysis (FTA) Based on Digital Production Model.' Paper presented in the 2nd international conference on Knowledge science, engineering and management (KSEM'07). Lecture Notes in Computer Science (4798/2007). Berlin, Heidelberg: Springer, 544-549.

Figure 1. *Monitored system and the position and constituents of the monitor*

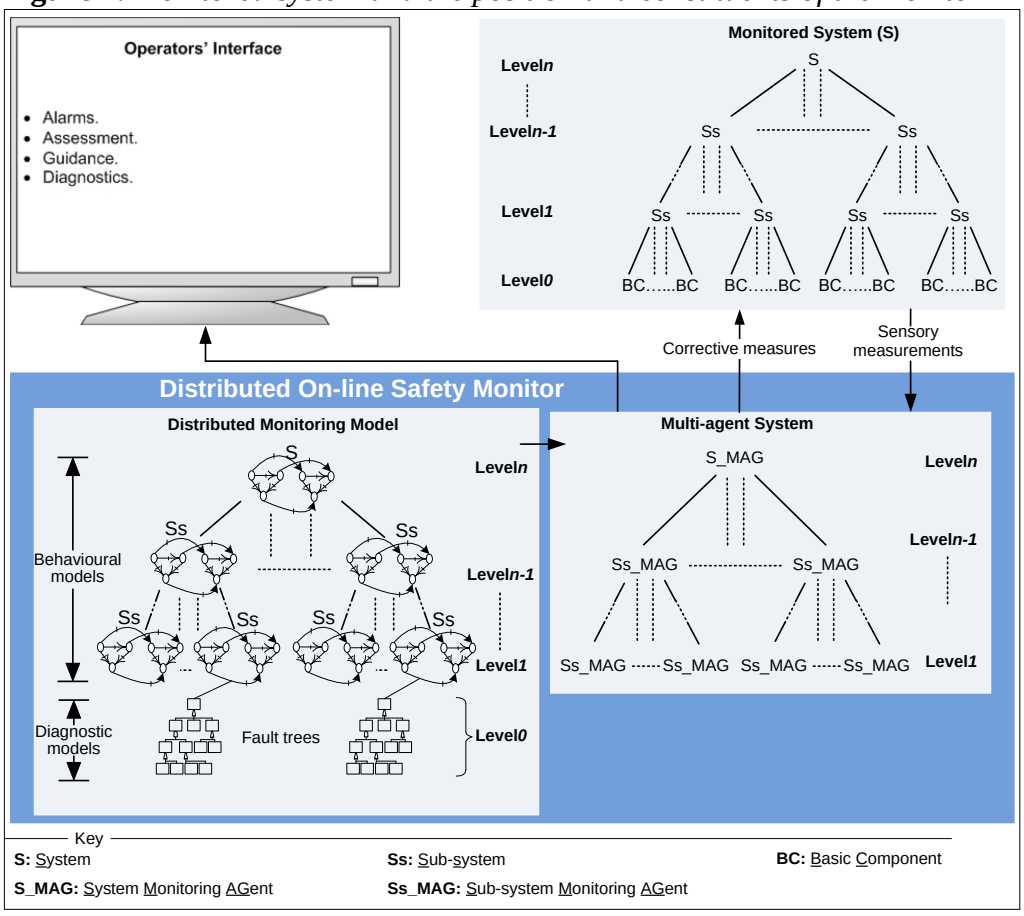

Figure 2. *a general illustration of the monitoring agent.*

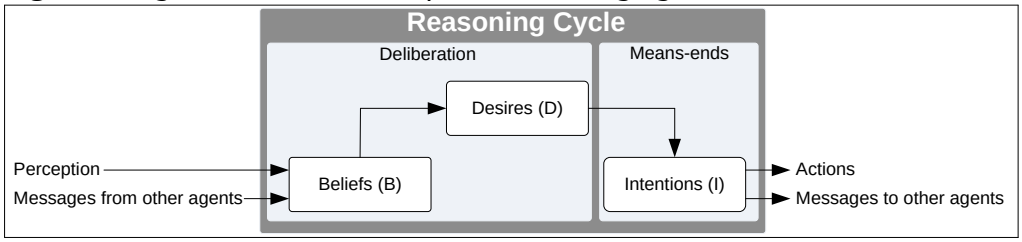

Figure 3 a *collaboration protocol to track the operational behaviour of the monitored system.*

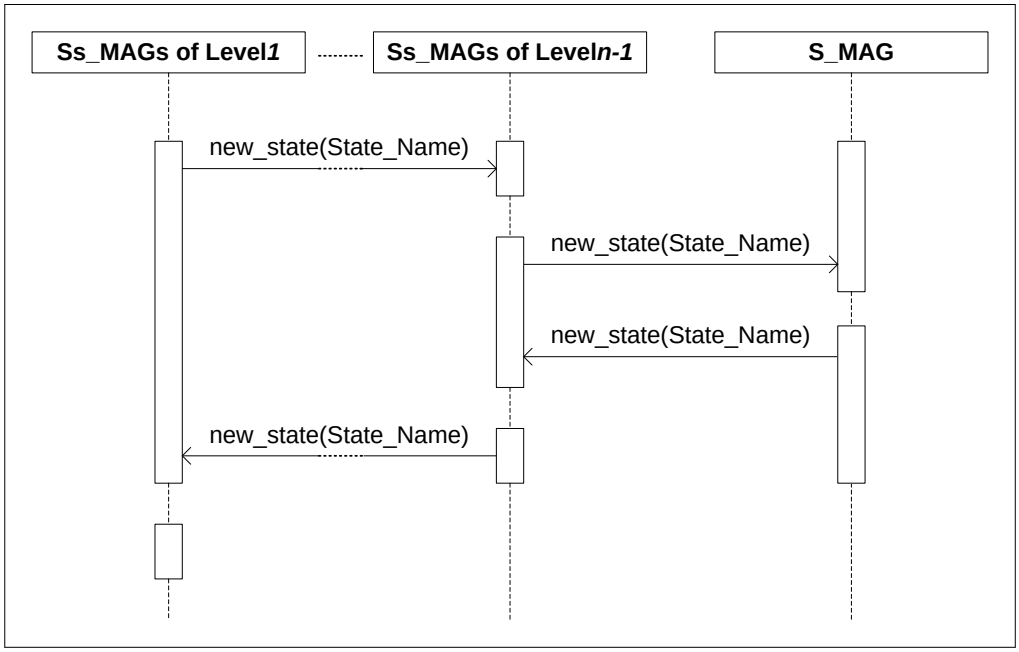

Figure 4 a *collaboration protocol to share sensory measurements among subsystem agents of level1.*

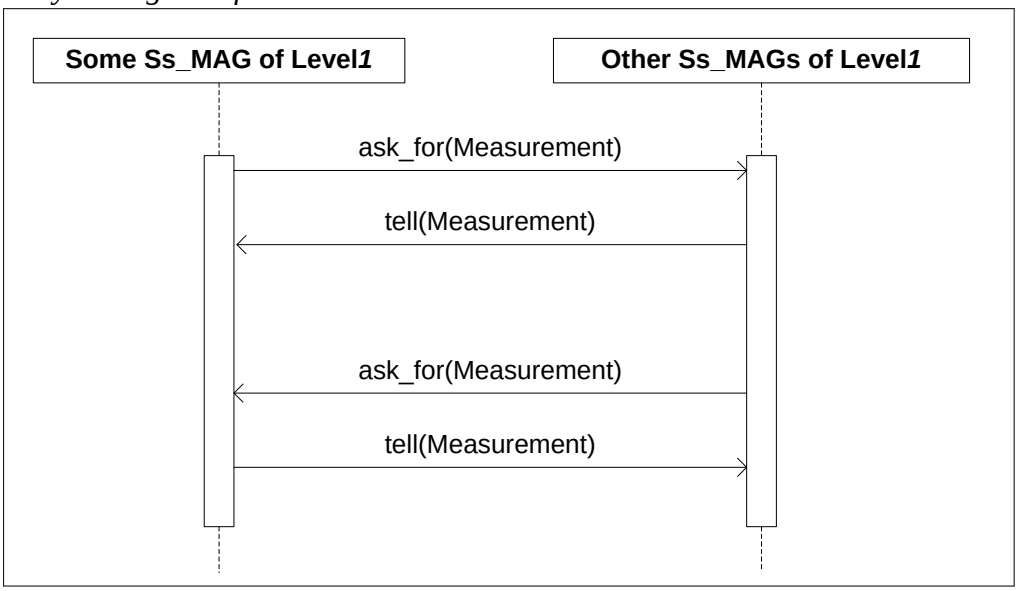

Figure 5 *physical illustration of the aircraft fuel system.*

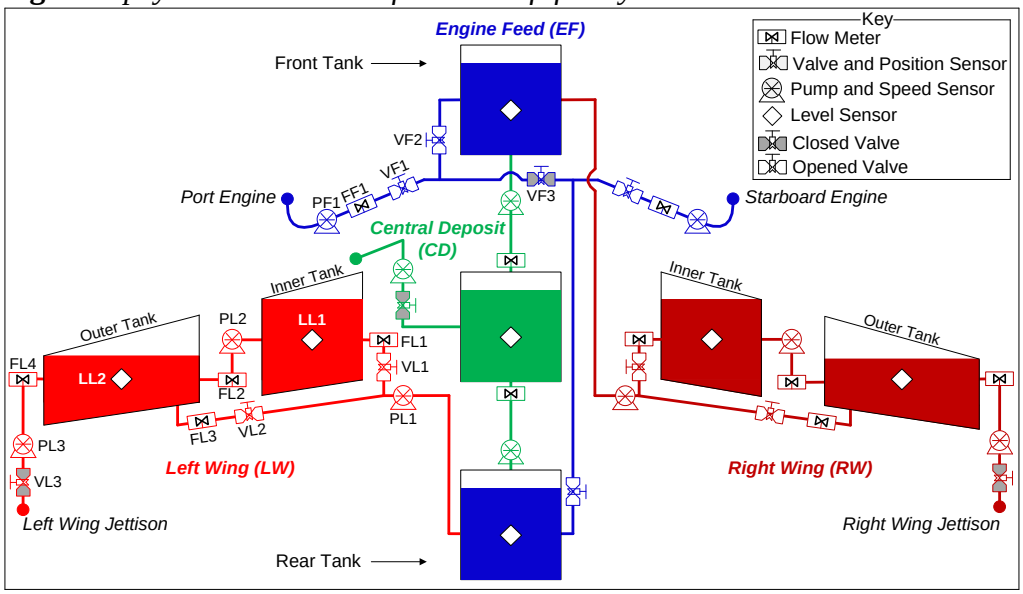

Figure 6 *architectural view of the deployment of agents to monitor the aircraft fuel system.*