

A Multi-Agent-Based Management Approach for Self-Health Awareness in Autonomous Systems

Gustavo González
University of Girona
Agents Research Lab
17071, Girona, Spain
gustavog@eia.udg.es

Cecilio Angulo
T. University of Catalonia
GREC
08800, Barcelona, Spain.
cecilio.angulo@upc.edu

Cristóbal Raya
T. University of Catalonia
GREC
08800, Barcelona, Spain.
cristobal.raya@upc.edu

Abstract

Integrated Vehicle Health Management (IVHM) systems on modern aircraft or autonomous unmanned vehicles should provide diagnostic and prognostic capabilities with lower support costs and amount of data traffic. When mission objectives cannot be reached for the control system since unanticipated operating conditions exists, namely a failure, the mission plan must be revised or altered according to the health monitoring system assessment. Representation of the system health knowledge must facilitate interaction with the control system to compensate for subsystem degradation. Several generic architectures have been described for the implementation of health monitoring systems and their integration with the control system. In particular, the OSA-CBM approach is considered in this work as initial point, and it is evolved in the sense of self-health awareness, by defining an appropriated multi-agent smart health management architecture based on smart device models, communication agents and a distributed control system. A case study about its application on fuel-cells as auxiliary power generator will demonstrate the integration.

1. Introduction

Vehicle health management, machinery monitoring and diagnostic systems are tasks related with maintenance. Main objectives in maintenance of systems and equipment are ensuring safety, equipment reliability, and reduction of costs. Traditionally, there have been two common maintenance philosophies employed: preventive maintenance (sometimes called schedule-based or event based) that uses statistics to estimate the machine behavior, leading to very conservative estimates of the probability of failure; and corrective (sometimes called restorative) maintenance philosophy, with the machine running until it fails and

being then restored to good health; hence maintenance costs are reduced, but unexpected failures may result in longer than expect system down times.

The objective in Condition-Based Maintenance (CBM) is to accurately detect the current state of engineered systems and their operating environments and use that information for maintenance and prognosis activities [11]. A CBM system is intended to monitor the operation of a complex system and provide the operator or the autonomous control system with an accurate assessment of the system's current health. In the traditional CBM analysis flow path information is directed from a layer to the nearest one. A conceptual view of a general condition- based monitoring system is shown in Fig. 1.

There are numerous differences between collaboration in a computer-supported environment versus a face-to-face environment. Even though scientists are working hard to simplify collaboration by producing systems that provide virtual environments, and collaborators work hard to be cooperative, there are still unknown problems for collaborators to find and overcome when using a collaborative system. In building a collaborative system, conflict resolution is one of the most important problems to overcome.

Implied software in a traditional CBM system is application or hardware specific, no standardized, so it is difficult to upgrade it. Moreover, control and monitoring machinery is hard coded and wired, no distributed software is possible.

Standardization of an interface specification within the CBM community is the first step driving CBM suppliers to produce interchangeable hardware and software components, reducing cost and development time. In this sense, an Open System Architecture (OSA) is defined as a systems engineering approach that facilitates the integration and interchangeability of components from a variety of sources. An open systems standard should consist of publicly available

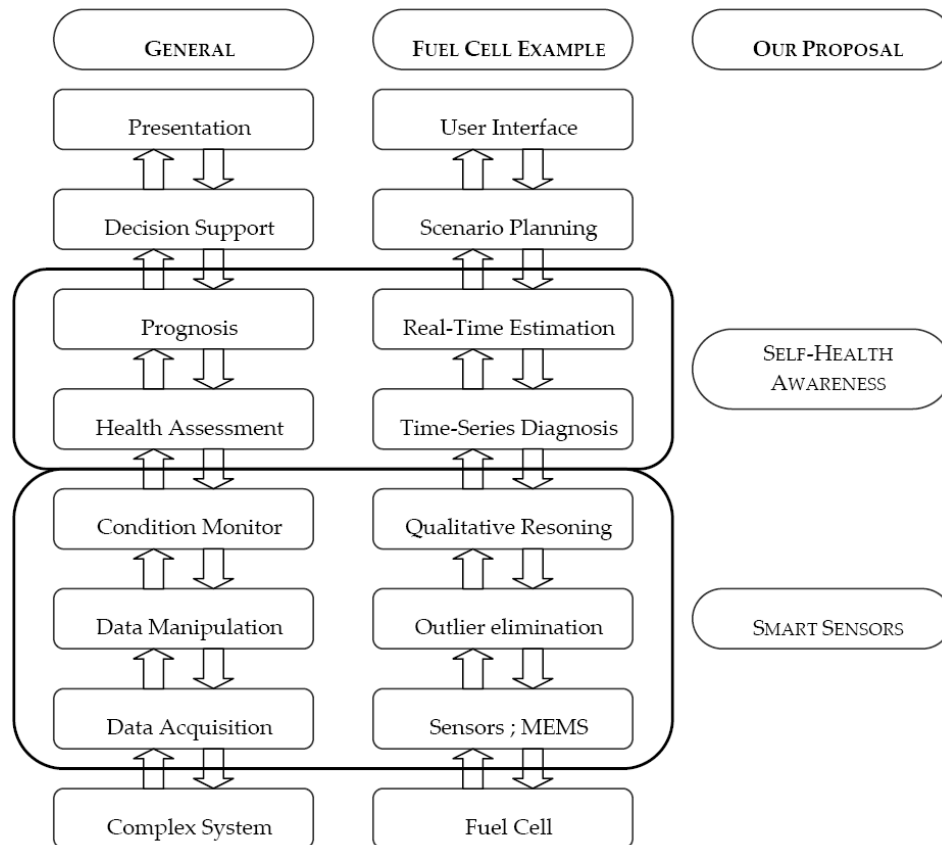


Fig. 1. Traditional CBM analysis flow path on both, a General and a Fuel Cell complex system example. It is also indicated the working zone of the proposed architecture in the general CBM flow path.

descriptions of component interfaces, functions, and behaviors. A brief introduction about OSA-CBM systems [2] is introduced in section 2.

Openness is a general interesting concept improving the CBM system design for health monitoring. However an OSA-CBM monitoring system is not enough when self-health awareness is the goal. Next step towards awareness is to move the focus of the research from the classical diagnosis techniques to determining loads and demands on critical components and subsystems. Devices result the focus of the health monitoring system in order to create the integrated vehicle health management system. Since components are considered in a distributed form, multi-agent systems (MAS) paradigm [19, 3] fits the novel approach. In section 3 is presented a general MAS architecture for vehicle self-health awareness introduced in [16] that constitutes the starting point of the general architecture proposed in this article. In section 4 the appropriated Multi-agent-based OSA-CBM architecture based on Smart Device Models

(SDM) for IVHM systems is displayed through the definition of the different communication agents and a distributed control system. A case study about its application on fuel cells as auxiliary power generator demonstrating the integration is presented in Section 5. Finally, some conclusions and further research are outlined.

2. OSA-CBM architecture

The focus of the OSA-CBM architecture is the development of an open standard for distributed CBM software components [13]. The architecture should not be exclusive to any specific hardware implementations, operating systems, or software technology. It is a specification for transactions between components within a Condition-Based Maintenance system. The core of the OSA-CBM standard is the Object Oriented data model, defined using UML (Unified Modeling Language) syntax. In fact, the OSA-CBM data model is a mapping of key concepts from the MIMOSA CRIS

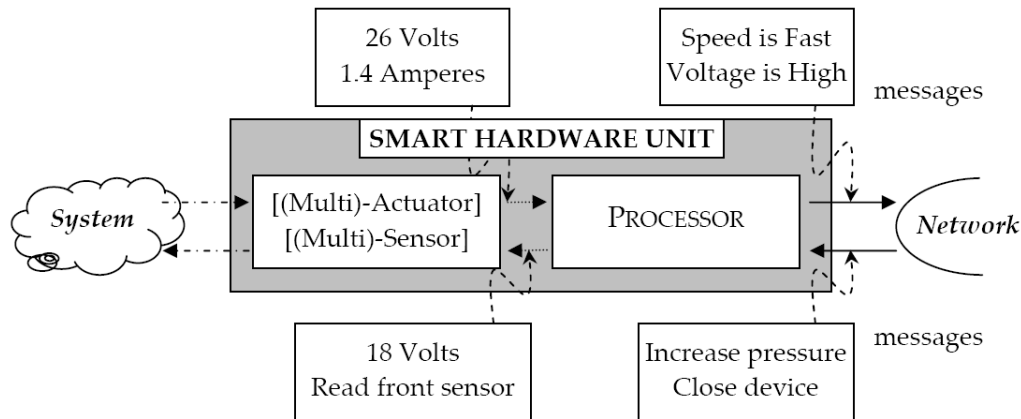


Fig. 2. Intelligent sensor node. Information processing integrates sensor data fusion (perception) within the node itself. Information is transmitted through messages, not being restricted to send only collected data, else augmented information.

with extensions for diagnostics, prognosis and data transactions. MIMOSA is a standard for data exchange between Asset Management systems (plant and machinery maintenance information systems). The CRIS defines a relational database schema for machinery maintenance information. The schema provides broad coverage of the types of data that need to be managed within the CBM domain.

Fundamentally, middleware allows applications to communicate with other remote programs as if the two were located on the same computer. The two major competing standards are CORBA, developed by OMG, and DCOM, developed by Microsoft. Main use of the OSA-CBM layered architecture is to split the functionality of CBM application to layers and to define the application programmer interfaces (APIs) of the layers. Main improvement of the OSA-CBM architecture, in front to the standard one, is that, in general, a component can access data directly from any layer. This fact favors distributed computing and, in particular, the use of multi-agent system paradigm.

The OSA-CBM development has defined three different module interfaces that may be used to query, configure and communicate data:

- Data interface. All the data relating to a particular event is defined by the data interface. This interface has a different structure for each layer (Dynamic information)
- Configuration interface. It contains the configuration information about the certain module. Semi-dynamic interface, it changes only as often as the configuration of the device changes.
- Explanation interface. Information about the device and its setup (i.e. static information).

Configuration and Explanation interfaces, containing information about the module, its configuration and setup, will be used for the communication agents in the proposed architecture to define the Smart Device Model (SDM). This concept, adopted from the user modeling [10] community, attempts to capture the behavior of every device monitored by the IVHM system. So, the focus on the health management system is moving from the diagnosis techniques to the devices behaviors and their interrelation.

3. Towards self-health awareness

Usually, research in health management has been focused on diagnosis techniques and tools; however it remains unsolved the problem about how to determine the loads and demands on critical components and subsystems, in order to prevent failures or adapt to device degradation. This determination for each planned mission is on the base of the definition of a 'failure', whether a failure is occurring in then system, and what it signifies for the mission. This new perspective allows to evolve health management from fault-tolerant or diagnostics systems towards self-health awareness [16]. Monitoring systems in this case must employ smart sensors dealing with high level information as opposed to typical sensor data.

3.1 Developing an architecture on distributed units

An extreme centralization on a 'master' element of health management routines do not allows that

functionality and capabilities of specific devices can be completely exploited. Many hardware devices or subsystems are not so smart how it could be possible because the information will overcharge the central processing unit. Architecture introduced in [16] is very useful to obtain self-health awareness by reducing centralization, however ‘subsystem monitors’, a hierarchically superior controller managing several smart devices, completely determines the actuation on the ‘intelligent sensor nodes’; no interaction exists peer-to-peer between devices in the whole complex system. As is described by the author, the knowledge of the interaction and dependencies between subsystems is a key point in the proposed architecture and not only a functional decomposition of the system. Therefore, the effort of the designer is directly proportional to the hierarchical dependencies between subsystems in the designed architecture. This work proposes a mechanism to allow interaction between intelligent sensor nodes and to reduce the charge in the subsystem monitor, even suppress them. In this form, knowledge of the subsystems dependencies will be captured by the nodes. It will not be mandatory to design the subsystem monitors integrating interaction knowledge, or their tasks will not be so critical.

Information processing in the intelligent sensor nodes integrates sensor data fusion (perception) within the node itself by using standard procedures, expert systems or some other technologies. Information is transmitted through messages, not being restricted to send only collected data, but also augmented information (Fig. 2).

3.2 Smart device modeling

Smart User Models [10] have demonstrated their efficacy improving the quality of services personalization reducing the overload of processed information and capturing the user’s behavior in the next generation of open, distributed and networked environments. The aim of the proposed architecture is to replace the user-centered to a device-centered objective using the flexibility of intelligent agents in order to capture the most relevant functionalities and behaviors of a device through an adaptive process. Mainly, a multi-agent architecture must be developed in order to manage services and device preferences in several complex systems (i.e. domains).

Context is a multi-dimensional parameter that includes time, place, states among other variables. Context-aware health management (self-health awareness) can be achieved through internal representations of devices in particular domains (i.e. device models) in similar way that user models are a

cornerstone in human-computer and human-robotic interaction [12, 17]. However, the development of applications in open environments poses the challenge for modeling devices once and continuously. So, the real issue is the use of a unique model for all the applications with which the device interacts. In order to contribute to such kind of future device models, we will develop a Multi-agent Smart Device Model (*SDM*). The proposed *SDM* is able to deal with features of the device in several domains and it continuously increases the knowledge of device requirements and functionalities in an unobtrusive way. The flexibility (re-activity, pro-activity and social-abilities) of agents are a cornerstone to implement the *SDM* in this framework.

Architecture is concerned with the ongoing work of an unobtrusive adaptive device model. Since the proposed approach to device modeling encompasses the communication (inter operability) and coordination (coherent actions) with health assessments, agent technology provides the appropriate flexibility to achieve all such issues. Consistently, in [10] has been developed a prototype of user model as a multi-agent system, which is able to provide services through its proactive, reactive and social capabilities to other agents, applications and users.

Similarly, the Smart Device Model will be able to provide information about the device when a new application in the environment requires it (reactivity); it is able to search new applications in which the device can be interested (pro-activity) and it can interact with other device models to obtain health assessments in a collaborative way (social-ability).

Broadly speaking, a Smart Device Model (*SDM*) is a collection of attribute-value pairs (a, v) , in the following form: $SDM^F = \left[\left(a_p^F, v_p^F \right)_p \right]$ where F can be Objective (O) and Subjective (S) features of the device. So, *SDM* could take attributes and values from n Objective ($F=O$) and m Subjective ($F=S$) features of the device. Objective features are directly obtained from sensors whereas subjective features come from pre-processing elements (soft-sensors) for not available on-line information. In this form, each device behavior is captured by a Smart Device Model, *SDM*, defining its internal representation in the environment, to achieve context-aware personalization. In order to extend the use of the *SDM* in several complex systems, it is defined the device model, *DM*, for a given existing application domain i as DM_i capturing features of the device required by the specific application.

Then, it is established a relationship between the general internal smart device model, *SDM*, and the device model for a specific application domain, DM_i ,

by means of a weighted graph, $G(SDM, DM_i)$. Such a graph connects device's internal features of the SDM with particular device features required at the application domain DM_i . When it is running, evolved features of the SDM modifies the weights used on the graph according to the state of the device [9].

Therefore, device's DM for each application are defined by shifting information from and to DM 's of different existing domains according to the weighted graphs $G(SDM, DM_i)$ defined by each application where device runs.

4. Multi-agent smart management architecture

Multi-agent systems (MAS) are based on the idea that a cooperative working environment comprising synergistic software components can cope with problems which are hard to solve using the traditional centralized approach to computation. Smaller software entities – software agents – with special capabilities (autonomous, reactive, pro-active and social) are used instead to interact in a flexible and dynamic way to solve problems more efficiently in real environments. The Foundation for Intelligent Physical Agents (FIPA) [6] is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies. It provides specifications to allow the construction of complex agent-based systems with a high degree of interoperability among autonomous systems.

To support our Self-Health Awareness theoretical approach on a networked (even a web-based) application, it is proposed a multi-agent architecture defined at two main levels of abstraction, following the design defined in [7]. There, a dynamic and recursive definition of agents as high-order multi-agent systems has been introduced in order to describe complex systems. At the highest level, two abstract agents exist: the *Network Service Abstract Agent (NSAA)* and the *Ubiquitous Device Abstract Agent (UDAA)*. The *NSAA* provides capabilities of autonomy regarding the automatic discovering of services in the network for the device [4]. It communicates with the applications in a specific domain. When applications are not agent-based, a *wrapper agent* operates like a middleware between the *NSAA* and the application. The *UDAA* gives initialization, identification, interoperability, control, coordination and management of the device preferences allowing a flexible and autonomous device-agent interaction. Coordination between *NSAA* and *UDAA* is established mainly by two mechanisms:

- (i) *NSAA* requests to *UDAA* personalized information to deal with the applications in the environment (health management)
- (ii) *UDAA* receives information from *NSAA* regarding the success or failure of the application interaction.

Such relevance feedback is used by the *UDAA* to learn about the device state, so the corresponding *SDM* and the weighted graph $G(SDM, DM_i)$ of the application is updated. Both, the *NSAA* and the *UDAA* are designed to be implemented in a distributed platform. The *NSAA* can be stored in a central unit (even a server) while the *UDAA* in a smart device. At the next abstract level, both abstract agents are implemented as multi-agent systems [8], as we explain in the remaining of this section.

4.1 NSAA architecture

Three types of agents (see Fig. 3) compose the *NSAA*, namely:

Accountant agent. It maintains a register of device- interacted applications and domains. It also requests to the *UDAA Application Agents* the establishment of new applications (see subsection 4.2).

Provider agent. Using contextual information and interacting with the *UDAA Repository Agent* it captures the pro-active behavior of the device by finding new applications in not registered domains in which it can be interested.

Consumer agent. It finds a device requested service by communicating with the *Provider Agent*, up-loading the service and creating an *Application Agent*.

To improve the performance of the *UDAA*, once the *Creator Agent* has realized its functions, it is removed.

4.2 UDAA architecture

The *UDAA* has four types of agents, namely (see Fig. 3): *Control Agent*, *Creator Agent*, *Application Agents* and *Repository Agent*.

Control agent. Its tasks are: (i) device login service; (ii) to dialogue with the device regarding its interaction with an application (suggested by the *NSAA* or requested for the device); (iii) to request to the *Creator Agent* for the generation of an *Application Agent* to manage the application confirmed by the device.

Creator agent. It is a temporal agent managing the device information in previous-applications the first time that it is registered in the system.

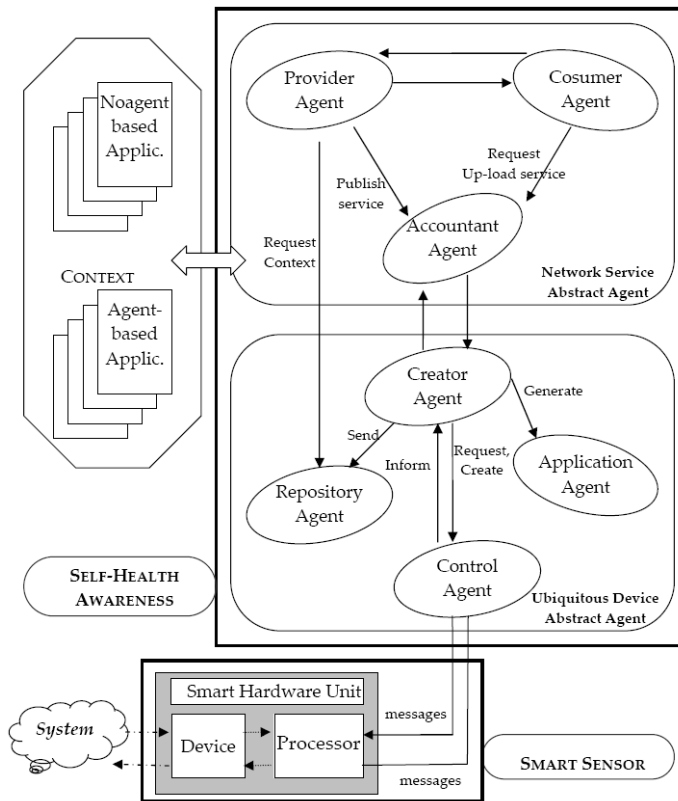


Fig. 3. Multi-agent System Architecture for the Smart Device Model. Two abstract agents exist: the *Network Service Abstract Agent (NSAA)* and the *Ubiquitous Device Abstract Agent (UDAA)*. The *NSAA* provides capabilities of autonomy. The *UDAA* gives initialization, identification, interoperability, control, coordination and management of the specific device preferences.

It has three goals:

- (i) to acquire the device profile by capturing all the information spread in its interaction with integrated vehicle health monitoring systems, and communicate it to the *Repository Agent*; objective features of the device are on-line learned via the methodology described in [1];
- (ii) to generate *Application Agents* from past device interactions that will be in charge of the interaction of the device and the application from now on;
- (iii) to register the previous applications in the multi-agent system by means of the *Control Agent*.

Application agent. Dynamically created when interaction with an application exists, the number of *Application Agents* varies from device to device. They provide all the information about the device that an application requires (reactivity), by acquiring and saving the relationship graph between the *SDM* and the

device model of the application DM_i . Endowed with networking abilities, *Application Agents*, are connected with other multi-agent *SDM*'s, establishing a social network [15] of Smart Device Models. When more than one agent has interacted with a certain application, so, more than one possible graph in the network can be found, the *Application Agent* composes the graphs by means, for instance, of trust measures [14]. Else, if no agent of the network has interacted with the application, default parameters are established for the graph.

Repository agent. It provides database storage procedures to save the knowledge of the device represented at the *SDM*. Individual device information is kept in a non-redundant, complete and consistent way in order to share it when and where necessary.

5. A case study

In this section, the set of functionalities and capabilities of each agent in the multi-agent Smart Device Model is described and illustrated with a case study: an application on fuel cells as auxiliary power generator of an aircraft.

Three examples of the functional operation of the architecture proposed will illustrate its performance.

Let FC_{UPC} be a fuel cell that has interacted in the past with the integrated health management system of an Airbeing aircraft in the 'Airbeing 747 domain'.

First, let's suppose that in a certain moment, the fuel cell FC_{UPC} is moved from an airplane to a newer similar one. Hence, it set up its health management system and the $UDAA$ starts. In a first step, the FC_{UPC} device initializes its self-health aware system through the $UDAA$ by registering its ID through the *Control Agent*. Immediately the *Control Agent* requests to the *Creator Agent* for registration. Such latter agent, gathers the current information about the fuel cell FC_{UPC} in the 'Airbeing 747 domain', when the device was in a different airplane of the same series, and sends it to the *Repository Agent* of the IHVM of this airplane, in order to increase knowledge. Then, the *Control Agent* creates an *Application Agent* for the Airbeing 747 health management system, and registers the Airbeing 747 application to the *Control Agent*.

In fact, it is not necessary to physically move the fuel cell. It can be used as a training device for new airplanes flying on certain routes (i.e. context-awareness).

As a second example, let's suppose that the *NSAA Provider Agent* gathers information about new IVHM systems in the 'Airbeing 747 domain'. That is the case, the FC_{UPC} fuel cell has used until now a health management system of a certain airplane, and the *Provider Agent* has discovered an IVHM about an airplane managing well when a fault in the hydrogen supply occurs. Since the fuel cell is observing that some degradation exists in the hydrogen supply system (i.e. contextual information), the *Provider Agent* believes that such information can be interest for the device. Hence, the *NSAA* requests to the $UDAA$ about the possibility of generating a new application on this new health management system.

Finally, after a while the fuel cell FC_{UPC} is moved to an Airbeing A-380, a different Device Model DM . Newly, the *Control Agent* of the IHVM system log in the device. In this case, at the $UDAA$, the *Control Agent* prompts the fuel cell the information regarding Airbeing A-380 and the fuel cell selects one application. Then, the *Control Agent* creates an

Application Agent to deal with the new application. The *Application Agent* looks in the network for a fuel cell that has deal with the new application. It is the case that the fuel cell FC_{UDG} has already interacted with the new application (for the 'Airbeing A-380 domain'). So, the corresponding *Application Agent* of FC_{UDG} and FC_{UPC} dialog. The *Application Agent* of FC_{UPC} acquires the Device Model 'Airbeing A-380 domain' from the graph $G(SDM, DM_i)$ corresponding to the relationship of the SDM of FC_{UDG} in the specified domain. Weighted graph G is adapted to the Smart Device Modeling of FC_{UPC} and its *Application Agent* is ready to deal with the health management process.

In this latter example, communication is directly established between two similar devices. However, this idea can be extended to deliberation between different devices in a same aircraft executing the health management of a certain mission. For instance, in landings, a dialogue can be established between fuel cell node, undercarriage node and passengers comfort node.

5. Conclusions and discussion

Openness is a general interesting concept improving the CBM system design for integrated vehicle health monitoring. However, an OSA-CBM monitoring system is not enough to obtain self-health awareness. Moving the focus of the research from the classical diagnosis techniques to devices, components can be considered in a distributed form, so Multi-Agent Systems (MAS) paradigm fits the novel approach. By the use of multi-agent technology Open Systems Architecture (OSA) for CBM can be improved two fold: firstly, the system can become self-aware, and secondly, it can be easily distributed among different communicating agents, and thus more flexible.

Defining Smart Devices Models, a concept imported from the user modeling research, it has been possible to design an appropriated Multi-agent OSA-CBM architecture for IVHM systems through the definition of the different communication agents and a distributed control system. This system avoids to exactly define the functional characteristics of the smart devices and their integration within the whole system by allowing direct communication between smart devices.

By the use of multi-agent technology Open Systems Architecture (OSA) for CBM can be improved two fold: firstly, the system can become self-aware, and secondly, it can be easily distributed among different

communicating agents, and thus more flexible. Although other approaches for IVHM systems have been studied in literature [5], we argue in this work the flexibility of agent technology to make more feasible its application in real environments.

The question however remains open for complex domains and critical missions where practical communication limitations exist (i.e. technological limitations) between agents in the whole system and their collaboration is not so simple. In this sense, acquiring relevant information to keep Smart Device Model up-to-date is crucial in order to maintain the autonomicity of this kind of complex systems [18].

A case study with three possible situations about its application on fuel-cells as auxiliary power generator demonstrating the integration between the main control module and the IVHM system has been presented.

In future work, we will evaluate the performance of our approach on simulated unmanned vehicles for non-critical missions.

Acknowledgment

This research was supported by the Spanish Science and Education Ministry project DPI2006-15630-C02-01 "Adaptive Systems for Human Mobility Aid"

References

- [1] Angulo, C., Parra, X., Català, A., "K-SVCR. A Multi-class Support Vector Machine". *Neurocomputing. Special issue: Support Vector Machines* 55 (1/2), 2003. pp. 57–77.
- [2] Bengtsson, M., "Condition based maintenance system technology- Where is development heading?" In: *Euromaintenance 2004. Proceedings of the 17th European Maintenance Congress*. May 2004. Spanish Maintenance Society, Barcelona, Spain.
- [3] Bussmann, S., Jennings, N., Wooldridge, M., "Multiagent Systems for Manufacturing Control: A Design Methodology". In *Springer Series on Agent Technology*. 2004. Springer-Verlag Berlin and Heidelberg GmbH & Co. K.
- [4] Dale, J., Lyell, M., "Towards an Abstract Service Architecture for Multi-Agent Systems". In: *Challenges in Open Agent Systems '03 Workshop*. July 2003. Melbourne, Australia.
- [5] Fijany, A. Baroth, E. Gharib, M. Mackey, R., "Novel model-based diagnosis approaches for advanced IVHM systems". In proceedings of 2002 CS/APS/PSHS/MSS Meeting Destin, Florida, USA. Available on-line at: <http://hdl.handle.net/2014/13625>
- [6] FIPA <http://www.fipa.org/>
- [7] Giret, A., Botti, V., "Towards an Abstract Recursive Agent". *Integrated Computer-Aided Engineering*. 2004. 11 (2).
- [8] González, G., "Towards Smart User Models for Open Environments". Master's thesis, Institute of Informatics and Applications. Universitat de Girona, October 2003. Girona, Spain. Available on-line at: <http://eia.udg.es/~gustavog/publications/tr-iiia-03-10-rr.pdf>
- [9] González, G., López, B., de la Rosa, J., "Smart User Models for Tourism: An Holistic Approach for Personalized Tourism Services". *ITT Information Technology & Tourism Journal*, March 2004. 6 (4), pp. 273–286.
- [10] González, G., López, B., de la Rosa, J., "A Multi-agent Smart User Model for Cross-domain Recommender Systems". In: *Proceedings of Beyond Personalization 2005: The Next Stage of Recommender Systems Research. International Conference on Intelligent User Interfaces, IUI'05*. January 2005. San Diego, CA, USA.
- [11] Kacprzyński, G. J., Roemer, M. J., "Health management strategies for 21st century condition-based maintenance systems". In: *International COMADEM Congress*. December 2000. Houston, TX.
- [12] Kobsa, A., "Generic User Modelling Systems". *User Modelling and User-Adapted Interaction (UMUAI)*, 2001.11, pp. 49–63.
- [13] Lebold, M., Reichard, K., Boylan, D., "Utilizing DCOM in an open system architecture framework for machinery monitoring and diagnostics". In: *Proceedings of the IEEE Aerospace Conference*. Vol. 3. March 2003. pp. 1227 – 1236.
- [14] Montaner, M., López, B., de la Rosa, J. L., "Opinion-based Filtering Through Trust". In: Matthias Klusch, S. O., Shehory, O. (Eds.), *Proceedings of the 6th International Workshop on Cooperative Information Agents (CIA'02)*. LNAI. Springer-Verlag Berlin Heidelberg, Madrid (Spain), 2002. pp. 164–178.
- [15] Palau, J., et. al., Collaboration Analysis in Recommender Systems Using Social Networks. In: Klusch, et. al (Eds.), *Cooperative Information Agents VIII: 8th International Workshop, CIA 2004*. Vol. 3191 of LNCS. September 27-29 2004. pp. 137–151. Springer-Verlag Heidelberg, Erfurt, Germany,
- [16] Reichard, K. M., Integrating self-health awareness in autonomous systems. *Robotics and Autonomous Systems* 49, 2004. pp. 105–112.
- [17] Ritter, F.E.; Van Rooy, D.; Amant, R.St.; Simpson, K., "Providing user models direct access to interfaces: an exploratory study of a simple interface with implications for HRI and HCI". *Systems, Man and Cybernetics, Part A, IEEE Transactions on*. Volume 36, Issue 3, May 2006 Page(s):592 – 601.
- [18] Truskowski, W.F.; Hinchey, M.G.; Rash, J.L.; Rouff, C.A., "Autonomous and autonomic systems: a paradigm for future space exploration missions". *Systems, Man and Cybernetics, Part C, IEEE Transactions on*. Volume 36, Issue 3, May 2006 Page(s):279 - 291
- [19] Wooldridge, M., *An Introduction to Multiagent Systems*. 2002. John Wiley & Sons, Chichester, England.