INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL.12, NO.2 JUNE 2007 158-166

An OSA-CBM Multi-Agent Vehicle Health Management Architecture for Self-Health Awareness

Cecilio ANGULO, Gustavo GONZÁLEZ, *Member, IEEE*, Cristóbal RAYA and Andreu CATALÀ

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 Open System Architecture - Condition-Based Maintenance (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.**

***Index Terms*— Multi-agent systems, OSA-CBM, Self-health awareness, Smart adaptive systems.**

1. INTRODUCTION

Vehicle health management [22], 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.

Current state-of-the-art in aircraft preventive maintenance includes, for instance, the Joint Strike Fighter (JSF) program which has incorporated prognostics health management (PHM) into its design, using sensors, advanced processing and reasoning, and a fully integrated system of information and supplies management [2]. The on-board JSF PHM system is hierarchical, dividing the aircraft into areas. Area data is generated by a mixture of dedicated, purpose-built sensors and analysis on existing control sensors to identify degradation and failures, compiled and correlated by area reasoners, and then correlated by system-level model-based reasoners.

For an in-deep review of the state-of-the art in vehicle health management, especially in aerospace, you can visit the webpage of the ISHEM First International Forum on Integrated System Health Engineering and Management in Aerospace [15].

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 [16]. 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 system standard should consist of publicly available descriptions of component interfaces, functions, and behavior. A brief introduction about OSA-CBM systems [3] 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-

Manuscript received October 15, 2006. This research was supported by the Spanish Science and Education Ministry project DPI2006-15630-C02-01 "Adaptive Systems for Human Mobility Aid"

C. Angulo, C. Raya and A. Català are with the Knowledge Engineering Research Group (GREC), Department of Automatic Control, Technical University of Catalonia (UPC), 08800 Vilanova i la Geltrú, Spain. (e-mail: {cecilio.angulo; cristobal.raya; andreu.catala@upc.edu})

G. González is with the Agents Research Lab (ARLab), Institute of Informatics and Applications (IIiA), University of Girona (UdG) and the EASY Innova Centre. Campus Montilivi, Building P4. E-17071, Girona, Spain. (e-mail: gustavog@ eia.udg.es).

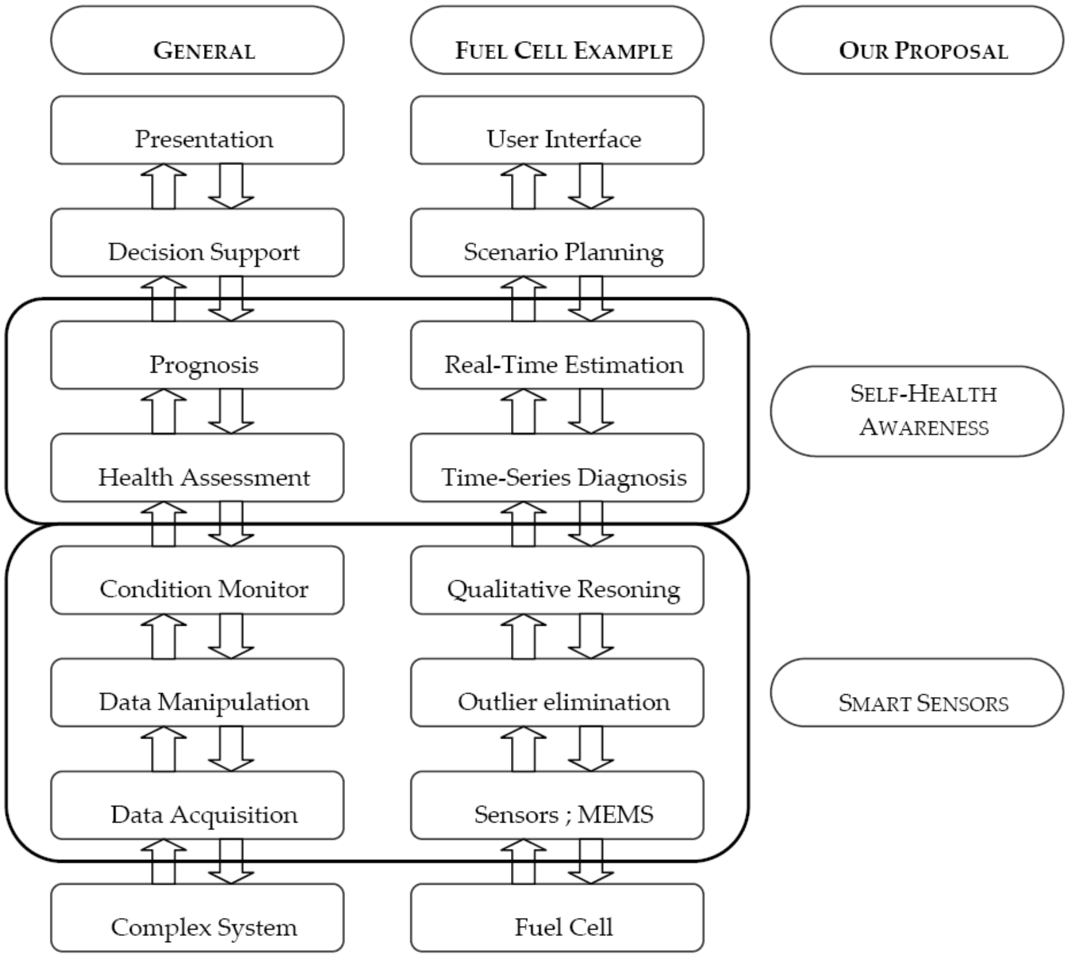

Fig. 1. Traditional CBM analysis flow path on both, a General and a Fuel Cell complex system example. It is also indicated the working zone of the proposed architecture in the general CBM flow path.

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 [24, 4] fits the novel approach. In section 3 is presented a general MAS architecture for vehicle self-health awareness introduced in [23] that constitutes the starting point of the general architecture proposed in this article. In section 4 the appropriated OSA-MAS CBM architecture based on smart device models 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 program is the development of an open standard for distributed CBM software components [18]. 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 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 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 (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 [17] 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. Hence, in [23] a generic architecture for the implementation of health monitoring within a complex system, the representation of system health information, and an approach for integrating health information with autonomous control is described. The core of the proposed architecture consists of a perception module and a response module. The perception module processes sensor information, commands, and messages to provide situational awareness for the controller. Depending on the source of the inputs to the perception module, the "situational awareness" may be internal, self-situational awareness, external situational awareness, or a combination of both. The perception module determines which behaviors should be activated to respond to the current situation. The response module accepts the behavior recommendations of the perception module and considers the necessary actions within the context of the current mission plan. Monitoring systems in this case must employ smart sensors dealing with high level information as opposed to typical sensor data.

3.1 Developing the Architecture on Distributed Units

An extreme centralization on a 'master' element of health management routines does not allow that functionality and capabilities of specific devices can be completely exploited. Many hardware devices or subsystems are not sufficiently smart as they could be. The main causes are the information and communication overload of the central processing unit. Architecture introduced in [23] 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.

Knowledge of interactions and dependencies between subsystems are key points in the architecture proposed in [20] jointly with a typical functional decomposition of the system. Here, 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 it. In this way, knowledge of the subsystems' dependencies will be captured by the nodes [7] and it will not be mandatory to design the subsystem monitors integrating interaction knowledge. In addition, their tasks will not be so critical [14].

Information processing in intelligent sensor nodes integrates sensor data fusion (perception) within nodes by using standard procedures, expert systems or some other technologies. Information is transmitted through messages without restriction to send only collected data, but also augmented information. (Fig. 2).

3.2 Smart Device Modeling

Smart User Models [11] have demonstrated their efficacy improving the quality of services personalization reducing the overload of processed information and€ capturing the behavior of the user in the next generation of open, distributed and networked environments. The aim of the proposed architecture is to move from 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 (i.e. self-health awareness) can be achieved through internal representations of the devices in the domain, i.e. device models. Such representations can be

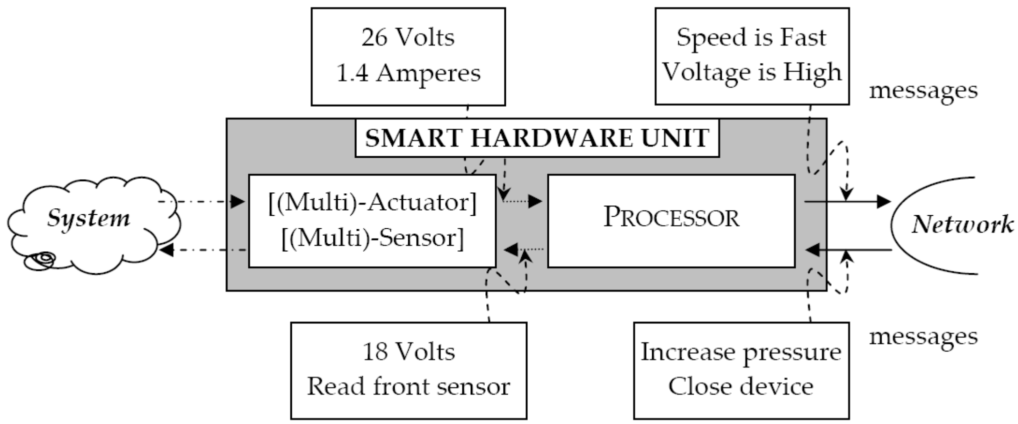

Fig. 2. Intelligent sensor node. Information processing integrates sensor data fusion (perception) within the node itself. Information is transmitted through messages, not being restricted to send only collected data, else augmented information.

analogously implemented in a similar way as user models are a key point in Human-Computer Interaction research.

However, the development of applications in open environments poses the challenge of modeling devices once and continuously. So, the real issue should be 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 the 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 [13] 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).

For a *j-th* device $j = 1,\dots,M$, a Smart Device Model (*SDM*) is a set of N_j attributes-value tuples $(a_{(i,j)}, v_{(i,j)})$, $i = 1,\dots,N_j$ defined as $SDM_{(j)} = \left[\left(a^F_{(i,j)}, v^F_{(i,j)}\right)\right]$, where F represents an objective feature ($F=O$) or a subjective feature ($F=S$) for the *i*-th tuple $(a_{(i,j)}, v_{(i,j)})$ of the *j-th* device. Objective features are directly obtained from sensors whereas subjective features come from pre-processing elements (i.e. soft-sensors) providing values 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 health-awareness. In order to enhance the *SDM*, it is defined the device model, *DM*, for a given existing *k–th* application domain DM_k capturing features of the device required in the specific application.

Then, it is established a relationship between the general internal smart device model, $SDM_{(j)}$, for the *j-th* device and the device model for a *k-th* specific application domain, $DM_{(j,k)}$ $k = 1, \dots, P$ by means of a weighted graph, $G(SDM_{(j)}, DM_{(j,k)})$. The weighted graph is used to shift device's information, i.e. attribute-value tuples, from existing applications to other ones by improving the *SDM*. Such graph connects device's internal features of the *SDM* with particular device features required at the application domain $DM_{(j,k)}$. In detail, the vertexes of the graph are the attributes, i.e. objective or subjective attributes, and arcs define pairs that describe a binary relationship between objective or subjective features and objective or subjective attributes of a device in a given domain. Weights are computed and normalized according to the value of each attribute in each particular domain. When the *SDM* is running its evolved features, it modifies the weights used on the graph according to the current device's state available in $DM_{(j,k)}$ [12]. For instance, let us consider one device which has interacted in a first domain D_1, then its representation $DM_{(1,1)}$ has the objective attributes $\left(a^O_{(1,1)}, a^O_{(2,1)}, a^O_{(3,1)}\right)$ and values $\left(v^O_{(1,1)}, v^O_{(2,1)}, v^O_{(3,1)}\right)$. So, the same device has its SDM_1 represented by a set of attributes $\left(a^O_{(1,1)}, a^O_{(2,1)}, a^O_{(3,1)}, a^O_{(4,1)}\right)$ with values $\left(v^O_{(1,1)}, v^O_{(2,1)}, v^O_{(3,1)}, v^O_{(4,1)}\right)$. The graph connects common and no-common attribute-value pairs of the same device according the interaction of the device in the domain D_1. In this example, the graph has updated the specific tuples for the SDM_1 according to $DM_{(1,!)}$. Note that the attribute-value pair $\left(a^O_{(4,1)}, v^O_{(4,1)}\right)$ already exists in the SDM_1, which has been captured from another $DM_{(1,k)}$. In our approach the weighted graph uses the SMARTS weighting method [8] to manage imprecise

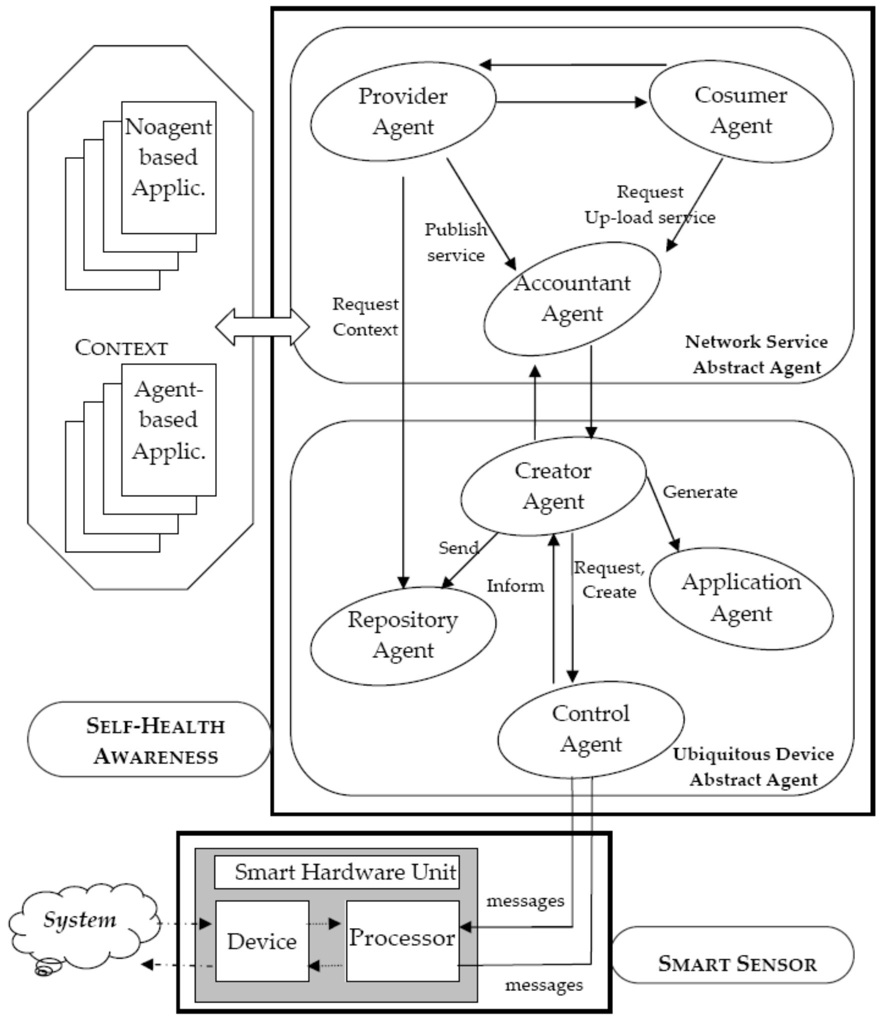

Fig. 3. Multi-agent System Architecture for the Smart Device Model. Two abstract agents exist: the *Network Service Abstract Agent* (*NSAA*) and the *Ubiquitous Device Abstract Agent* (*UDAA*). The *NSAA* provides capabilities of autonomy. The *UDAA* gives initialization, identification, interoperability,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

information derived from the unfeasible analysis that arises when are evaluated all the possible relevant alternatives to improve the *SDM*. We have choose this method to simplify our approach and by its practical implementation.

Therefore, $DM_{(j,k)}$ for each application are defined by shifting information from and to $DM_{(j,k)}$ of different existing domains according to the weighted graphs $G(SDM_{(j)},DM_{(j,k)})$ defined by each application where device runs.

Finally, we want to stress that there are several methods to deal with multi-attribute decision analysis, so, we encourage the reader to read [5] to obtain a more extended revision of these techniques.

4. Multi-agent Smart Management Architecture

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 [10]. 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 [6]. 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 of the 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* can be situated in a smart device. At the next abstract level, both abstract agents are implemented as multi-agent systems [13], 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 looking for 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*.

4.2 UDDA 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. 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) register the previous applications in the multi-agent system by means of the *Control Agent*.

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

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 [21] 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 [19]. 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 (see Fig. 4).

Three examples of the functional operation of the multi-agent architecture proposed will illustrate its performance. In the first example, a learned fuel cell *SDM* is simply used as a training device. In the second example, it is explained the adaptive shift of *SDM* in new application domains. In the third example, two learned similar devices share information about the health management in similar, but not the same, aircrafts. To support the development of our multi-agent system, a distributed communications environment is also required. In this sense, the Foundation for Intelligent Physical Agents (FIPA) provides a series of standards and specifications to build complex agent-based systems with a high degree of interoperability among agents. These standards facilitate communication, coordination and collaboration in autonomous systems using publicly available agent platform implementations [9].

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 learned fuel cell FC_1 is moved from an airplane to a newer similar one. Hence, it set ups its health management system and the *UDAA* starts. In a first step, the FC_1 device initializes its self-health aware system through the *UDAA* by registering its ID through the *Control Agent*. Next, the *Control Agent* requests the *Creator Agent* for registration

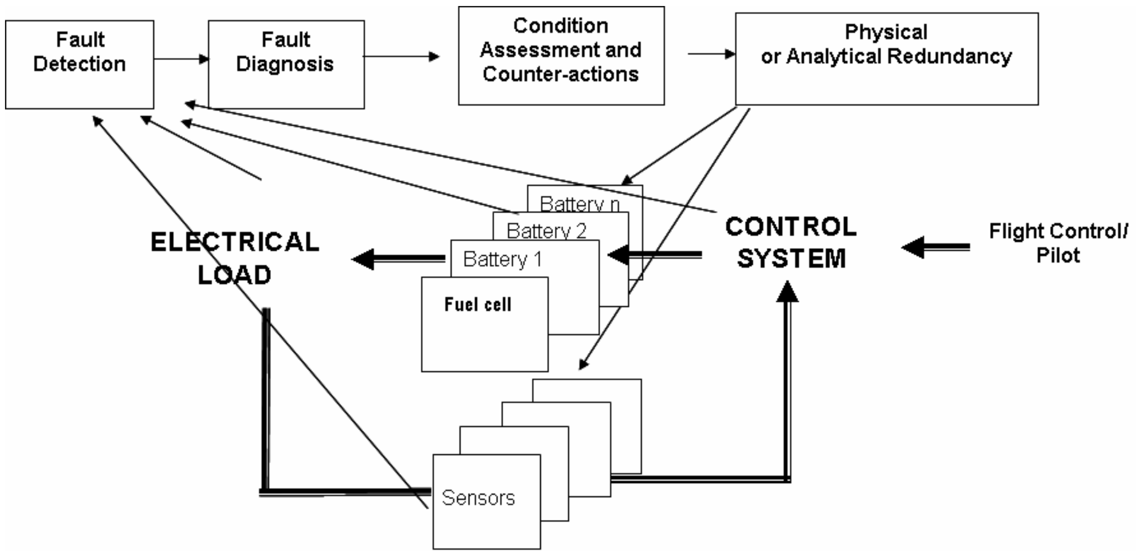

Fig. 4. An illustrative scheme about the fuel cell use in the case of study.

Such latter agent gathers the current information about the fuel cell FC_1 in the 'Airbeing 747 domain' (the domain for a specific type of aircraft), 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 aircraft Airbeing 747 health management system, and registers the aircraft 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_1 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 (contextual information), the *Provider Agent* believes that such information can be interesting 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_1 is moved to an Airbeing A-380 aircraft, 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 with the information regarding Airbeing A-380 aircrafts 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_2 has already interacted with the new application (for the 'Airbeing A-380 domain'). So, the corresponding *Application Agent* of FC_2 and FC_1 dialog. The *Application Agent* of FC_1 acquires the Device Model 'Airbeing A-380 domain' from the graph $G(SDM,DM_i)$ corresponding to the relationship of the *SDM* of FC_2 in the specified domain. Weighted graph *G* is adapted to the Smart Device Modeling of FC_1 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 airplane executing the health management of a certain mission. For instance, dialogue can be established between fuel cell node, undercarriage node and passengers' comfort node when aircraft is landing.

6. CONCLUSIONS

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.

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

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.

Further research will be directed to obtain accurate cost and performance models for the proposed architecture.

Acknowledgement

The authors gratefully acknowledge the support of the Spanish Science and Education Ministry.

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] Belcastro, M.C., Allen, C.L., "Aviation Safety Program. Integrated Vehicle Health Management", Technical Plan Summary, NASA Reports, 2006

[3] Bengtsson, M., "Condition based maintenance system technlogy-Where is development heading?" In: *Euromaintenance 2004. Proceedings of the 17th European Maintenance Congress*. May 2004. Spanish Maintenance Society, Barcelona, Spain.

[4] 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.

[5] Byeong Seok Ahn, "Multiattribute decision aid with extended ISMAUT," *Systems, Man and Cybernetics, Part A, IEEE Transactions on* , vol.36, no.3pp. 507- 520, May 2006

[6] 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.

[7] Duda, R., Hart, P., Stork D.G., *Pattern Classification*. Wiley-Interscience, 2nd edition, Oct. 2000.

[8] Edwards, W. and Barron, F. H., "Smarts and smarter: Improved simple methods for multiattribute utility measurement," Org. Behav. Human Decis. Process., vol. 60, no. 3, pp. 306–325, 1994.

[9] FIPA http://www.fipa.org/resources/livesystems.html

[10] Giret, A., Botti., V., "Towards an Abstract Recursive Agent". *Integrated Computer-Aided Engineering,* 2004. 11 (2).

[11] 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

[12] 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.

[13] 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.

[14] Guyon, I., Elisseeff, A., "Feature Selection and Causal Discovery". In: *NIPS 2006 Workshop on Causality and Feature Selection,* December 2006. Whistler, Canada.

[15] ISHEM Forum 2005. First International Forum on Integrated System Health Engineering and Management in Aerospace. November, 2005, Napa, CA. http://ic.arc.nasa.gov/projects/ishem/

[16] Kacprzynski, G. J., Roemer, M. J., "Health management strategies for 21st century condition-based maintenance systems". In: *International COMADEM Congress.* December 2000. Houston, TX.

[17] Kobsa, A., "Generic User Modelling Systems". *User Modelling and User-Adapted Interaction (UMUAI),* 2001.11, pp. 49–63.

[18] 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.

[19] 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.

[20] Naylor, A.W., "On Decomposition Theory: Generalized Dependence", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-11, No. 10, 699-713, October, 1981.

[21] 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,

[22] Paris, D.E., Trevino, L.C., Watson, M.D., "A Framework for integration of IVHM Technologies for intelligent integration for vehicle management", *2005 IEEE Aerospace Conference*, Vols. 1-4, 3843-3852, 2005.

[23] Reichard, K. M., Integrating self-health awareness in autonomous systems. Robotics and Autonomous Systems 49, 2004. pp. 105–112.

[24] Wooldridge, M., *An Introduction to Multiagent Systems.* 2002. John Wiley & Sons, Chichester, England.

Cecilio Angulo received his M.S. degree in Mathematics in 1993 from the University of Barcelone, Spain, and his Ph.D. degree in Sciences in 2001 from the Technical University of Catalonia (UPC), Spain.

He is a lecturer of the Department of Automatic Control, UPC, Spain. He was previously an assistant professor at University of Girona. He has published more than 20 papers, two books and one book chapter on learning and adaptive control.

He serves as a program committee member for several conferences on Artificial Intelligence (ESANN, CCIA, IWANN).

Gustavo González (M'04) is a PhD candidate in the Information Technologies PhD Program at the University of Girona, Spain. He obtained his Informatics Engineering degree and his M.Sc. in Telecommunications in 1999 and 2001 respectively, from the Industrial University of Santander (Colombia). In 2003, he obtained his M.Sc. on Information Technology from University of Girona.

Currently he is working in the Institute of Informatics and Applications (IIiA) of the University of Girona (UdG) as a member of the Agents Research Lab and the EASY Innova Center. He was working in 2006 as Guest Researcher in the Fraunhofer Institute for Applied Information Technology (FhG-FIT) (Germany) in the wearIT@work project: "Empowering the Mobile Worker by Wearable Computing". His current research interests include user modeling, personalization of services in ambient intelligence, multi-agent system and machine learning.

Mr. González is an IEEE Member (M) since 2004 and the founder of IEEE Student Branch at University of Girona.

Cristóbal Raya was born in Vilanona i la Geltrú, Spain, in 1972. He received the M.S. degree in Electronics Engineering in 1999 from the Technical University of Catalonia. At present he is a Ph.D. candidate from the Technical University of Catalonia. He is an assistant professor in the Automatic Control Department at the UPC. The subject of his research is electronics devices with applications in ubiquitous computing.

Andreu Català received his M.Sc. degree in Physics in 1980 and the Ph.D. degree in 1993 from the Technical University of Catalonia. He is currently an Associate Professor at the Technical University of Catalonia and Head Director of the Knowledge Engineering Research Group, GREC, which is a partner of the European Research Laboratory LEA-SICA. His main research interests include neural networks, uncertainty information treatment and sustainability analysis.