



Distributed embedded health assessment of networked instruments

Mickaël Diévert, Philippe Charbonnaud, Xavier Desforges

► To cite this version:

Mickaël Diévert, Philippe Charbonnaud, Xavier Desforges. Distributed embedded health assessment of networked instruments. 12th IFAC Symposium on Large Scale Systems : Theory and Applications (LSS 2010), Jul 2010, Villeneuve d'Ascq, France. pp.0. <hal-04087754>

HAL Id: hal-04087754

<https://hal.science/hal-04087754v1>

Submitted on 3 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 4816

To link to this article : DOI: [10.3182/20100712-3-FR-2020.00038](https://doi.org/10.3182/20100712-3-FR-2020.00038)
URL : <http://dx.doi.org/10.3182/20100712-3-FR-2020.00038>

To cite this version :

Diévar, Mickaël and Charbonnaud, Philippe and Desforbes, Xavier (2010) *Distributed embedded health assessment of networked instruments*. In : 12th IFAC Symposium on Large Scale Systems : Theory and Applications (LSS 2010), 12-14 juillet 2010, Villeneuve d'Ascq, France.

Distributed embedded health assessment of networked instruments

Mickaël DIEVART, Philippe CHARBONNAUD and Xavier DESFORGES

Université de Toulouse, INPT, Ecole Nationale d'Ingénieurs de Tarbes, Laboratoire Génie de Production,
47 avenue d'Azereix, 65016 TARBES France;
(Tel: +33 5 62 44 29 34, e-mail: {Mickael.Dievert, Philippe.Charbonnaud, Xavier.Desforges}@enit.fr).

Abstract: This article presents an applicative architecture based on a solving method for embedded technical health assessment of complex systems. This architecture is defined in order to provide services enabling the evaluation of the health status of complex systems. Diagnostic and prognostic services provide information to the maintenance decision support system that leads to reduce the periods of unavailability and determine if future missions of complex systems can be carried out. The architecture presented in this paper implements distributed diagnostic and prognostic functions using multi-agent techniques. An example shows the effectiveness of the applicative architecture.

Keywords: Distributed diagnosis and prognosis, health status, multi-agent system, complex systems, embedded systems, solving method, system architecture, health assessment

1. INTRODUCTION

New regulations in terms of environment, goods and people protection, and needs of new services have consequences on the complexity of systems that produce goods or services. To face this increasing complexity, multiple functionalities of the resources are embedded and deployed into networks of functions achieved by Line Replaceable Units (LRU). In transportation systems, faulty LRUs are replaced when the vehicle is at its base and repaired in the maintenance workshops, while the repaired system carries on with its mission. The increasing number of functionalities of the embedded systems contributes to raise the possession and acquisition costs leading the resources customers to optimize their availability rate (Hitt, 1997).

The Condition-Based Maintenance (CBM) recommendations usually contribute to improve the equipment availability (Jardine *et al.*, 2006 ; Scarf, 2007). Indeed, the CBM depends on the effectiveness of the system state provided by monitoring and diagnostic functions. They are carried out in particular from on line data generally processed by an embedded centralized diagnosis function. However, in the case of system of systems also called complex system, the identification of the faulty components is difficult using centralized architectures. After the mission, the maintenance operators must collect information by interactions with the embedded diagnostic system, in order to isolate possible faulty LRUs, and to apply troubleshooting procedures (Villareal & Cicek, 2004). The drawbacks of such architectures are related to the numerous pieces of information to process, which might be wrong. The automated diagnostic processes combine these errors and lead to useless removals of LRUs. Those removals are costly and increase the risk of damaging the system.

Alternatively, a decentralized/distributed diagnosis can be proposed to reduce the number of useless removals of LRUs. For applications to system of systems, monitoring and diagnostic functions can be implemented closer to the LRUs thanks to agents that carry out them. In the case of a distributed approach, a collaborative mechanism between diagnostic agents have to enable the convergence of the local diagnoses towards a set of accused LRUs which should ideally correspond to the true faulty ones.

This article presents an applicative architecture for implementing a distributed diagnostic function. In section 2, the problem statement is defined. In section 3, the difficulties of implementing this diagnostic function in such systems due to the various kinds of the subsystems and to the necessary knowledge and models for its achievement is discussed. In section 4, an embedded diagnosis function is proposed. In section 5, an applicative architecture and its cooperation protocol is presented. Its objective is to carry out the identification of a set of faulty LRUs from LRUs declared faulty by the local diagnoses.

2. PROBLEM STATEMENT

The technical diagnosis of complex transportation systems provides to the maintenance operators a list of LRUs that should be replaced. Online diagnosis enables the maintenance operators to prepare the intervention sooner. This reduces the duration and the costs of maintenance actions. The main task of a diagnostic function is to define a set of faulty components and the severity of the fault. Diagnosing such systems is difficult because of their numerous kinds of functions integrating different technologies. Thus, the implemented diagnostic techniques must be adapted to those technologies and require knowledge about the system. During its use, various faults may impact the resource. Those faults degrade its operating modes.

The diagnosis classifies the failure and determines the operating mode (nominal, degraded, exception...) of the system to the various types of faults occurring in the system. A FMEA (Failure Modes, Effects and Analysis) can address this problem by the construction of causal trees.

Monitoring and diagnosis functions associated to the LRUs are implemented different ways. Indeed, when the know-how of a supplier is highly involved in monitoring and diagnostic tasks design of the LRU, the supplier may propose to host them in the LRU or to provide the software that set them into operation in order to be hosted by third part platforms. Nevertheless those tasks shall communicate thanks to digital buses. In the case of aircrafts, the Centralized Maintenance System (CMS), provide a list of likely faulty LRUs for the maintenance operator. This list is established according to information from the built-in test equipments that collect information from the LRUs and generate tests if needed. The CMS correlates data to provide a “pre-diagnosis” of the LRUs. The flight warning system provides to the cockpit crew information on aircraft failed functions (Ramohalli, 1992 ; Byington *et al.*, 2003).

Complex systems can be considered as sets of systems that depend more or less on each other. A system implements one or several functions. For safety purposes, functions can be redundant as well as the LRUs that implement them. That is why several models are necessary to classify the different operating modes of the LRUs and their health status.

3. SYSTEM OF SYSTEMS MODELING

Generally, the system is analyzed from different models in order to obtain a satisfying representation for diagnosis purpose. This analysis enables to collect knowledge on the complex systems. Most of them are used to design diagnostic functions which generally less involve the know-how of the LRU supplier than the monitoring functions that often require behavioral models. Those models can be functional (Abu-Hanna *et al.*, 1991), structural and behavioral (Chittaro & Ranon, 2003 ; Keuneke, 1991). They enable to model the behavior of components, of functions and of their interactions according to normal or degraded modes. A Complex System (CS) can be defined by a finite set of m system \sum_i . $CS = \{\sum_1, \sum_2, \dots, \sum_m\}$. A system \sum_i can be defined as a set of n function $F_{i,j}$. $\sum_i = \{F_{i,1}, F_{i,2}, \dots, F_{i,n}\}$. A function $F_{i,j}$ can be defined as a set of k LRUs implementing this function. $F_{i,j} = \{LRU_{i,j,1}, \dots, LRU_{i,j,k}\}$. If a LRU contribute to the implementation of more than one function, a decision has to be taken when defining the system. A LRU should be part of one and only one function. After defining the hierarchical decomposition of the system, a system has to be modeled to design its diagnostic function. The necessary knowledge to diagnose the complex system is collected in the set SK made of three types of knowledge:

- the Functional Description (FD) which is the set of functions ensured by every system. FD represents links between LRUs, functions and system;
- the Structural Description (SD) is dedicated to the identification of the set of LRUs and of physical

connections between them. SD introduce predicate $CONNECT(X,Y)$ that means that X is connected to Y . $S : CONNECT(LRU_{i,j,q}, LRU_{p,r,s})$ with q and s respectively one of the LRUs implementing $F_{i,j}$ and $F_{p,r}$;

- the Behavioral Models (BM) that are used in order to identify the relevant indicators that are used to generate symptoms for the various faults that may affect the LRUs. This task consists of the monitoring layer. It enables to classify the faults of the LRUs from their symptoms. The LRU supplier should achieve this part of the design because it involves models of the LRU that are part of its know-how. For diagnostic purpose, BM provides the relationships between the symptoms, the LRUs and their faults.

Finally, $SK = FD \cup SD \cup BM$

4. EMBEDDED HEALTH ASSESSMENT

The OSA-CBM project (Mathew *et al.*, 2006) defines a Health Assessment (HA) layer whose primary function determines the health status of a monitored system, subsystem, equipment or component in terms of fault, failure, availability. The health assessment module should take into account diagnoses, trends in the health history, operational status and loading, and the maintenance history. The HA layer provides the Health Status (HS) of a monitored entity. This leads to Prognostics and Health Management (PHM) that is defined in (Vachtsevanos *et al.*, 2006) as the phase involved with predicting future behavior, including the Remaining Useful Lifetime (RUL), in terms of current operating state and the scheduling of required maintenance actions. According to Mathur *et al.* (2001), diagnosis and prognosis are processes of assessment of a system's health. Diagnosis is an assessment about the current (and past) health of a system based on observed symptoms, and prognosis is an assessment of the future health.

The different results given by diagnostic and prognostic functions must be considered as a decision support to operate appropriately the system. In the proposed approach, the HA is introduced as a combination of the results of the diagnostic and prognostic functions.

The diagnostic function consists of several activities: to condition, to detect and to identify failures and their causes. Relevant and significant indicators are generated by the function “to condition” and are based on measurements of the system. These indicators can be statistics, signals values, parameters, state observers, residuals, errors.... The function “to detect symptoms” uses these indicators to generate values, called “detected symptoms”, which are provided when a fault occurs. The decision can be made thanks to a decision-making support technique. The outputs can be digital and define which fault occurred. Monitoring methods of LRUs depend on the subsystem supplier practices. “To identify failures” is done by two sub-functions: “To identify

failed function(s)” and “To identify faulty component(s)”. The user (pilot, driver...) cares about the failed functions of the system whereas the maintenance operator cares about the faulty components that must be replaced. The proposed approach is consistency based diagnosis as defined in (Hamscher *et al.*, 1992).

Ideally, the diagnosis identifies a set Δ_2 of failed function and locates their causes, *i.e.* a set Δ_1 of faulty LRUs from a set of symptoms S and a set of tests T . This leads to the next relationship where *Diag* is the diagnostic function:

$$(\Delta_1, \Delta_2) = \text{Diag}(SK, S, T)$$

The function *Diag* can be implemented thanks to two sub-functions as presented in figure 4. The function *Diag1* allows, starting from a set of symptoms and a set of tests, to identify the set of faulty LRUs of the system $\Delta_1 = \text{Diag1}(SK, S, T)$, where $\Delta_1 = \{AB(LRU_{i,j,q}), \dots, AB(LRU_{p,r,s})\}$ and the function *Diag2* enables to locate the set of failed function from tests and the set of failed LRUs: $\Delta_2 = \text{Diag2}(SK, \Delta_1, T)$ where $\Delta_2 = \{AB(F_{i,j}), \dots, AB(F_{p,r})\}$. $AB(\cdot)$ enables to denote either a faulty LRU or a failed function.

The prognostic function, that completes the health assessment, can be implemented thanks to methods proposed in (Roemer *et al.*, 2007). The different techniques used are based on the same domains from the diagnostic function. Prognosis like diagnosis can be based on models, data or experiences. The RUL can either be determined by extrapolating indicators or from statistic values like Mean Time To Failure (MTTF) or Mean Time Between Failures (MTBF). The RUL is the remaining operational time until the system becomes unable to successfully complete its mission according to the conditions of operation. This means that the prognostic function requires a new kind of knowledge: A historic of the current and future missions with their constraints. This knowledge enables to determine the threshold beyond which the LRU cannot operate in nominal mode. This also corresponds to the moment when the RUL is null. The RUL of an LRU is calculated by the monitoring layer. Let us note $RUL(LRU_{i,j,k})$ the RUL of the $LRU_{i,j,k}$, and $RUL(F_{i,j})$ the RUL of the function $F_{i,j}$. Therefore, for a given function $F_{i,j}$ without redundant LRUs, $RUL(F_{i,j}) = \text{Min}(RUL(LRU_{i,j,1}), \dots, RUL(LRU_{i,j,k}))$. Let us considered now the case of a function $F_{i,j}$ implemented by A redundant LRUs. And this function is not considered as failed if at least B of A LRUs are not failed. In this case, $RUL(F_{i,j})$ is equal to the B^{th} maximum RUL among the A RUL of each LRUs implementing $F_{i,j}$. For example, if a function $F_{i,j}$ is ensured by 3 LRUs ($LRU_{i,j,1}$, $LRU_{i,j,2}$ and $LRU_{i,j,3}$) and the function is not considered as failed if at least 2 of this 3 LRUs are not failed. Given $RUL(LRU_{i,j,1}) = 5UT$, $RUL(LRU_{i,j,2}) = 10UT$ and $RUL(LRU_{i,j,3}) = 15UT$, $RUL(F_{i,j}) = 10 UT$.

Ideally, the prognosis identifies the set \prod_2 of the RUL of each function implementing the system of systems and the set

\prod_1 of RUL of each LRU from a set of Extrapolated Data (*ED*) and a set of threshold *Th*. This leads to the next relationship where *Prog* is the prognostic function:

$$(\prod_1, \prod_2) = \text{Prog}(SK, ED, Th)$$

The function *Prog* can be implemented thanks to two sub-functions. The function *Prog1* allows, starting from a set of extrapolated data and a set of threshold, to identify the set of RUL of the LRUs of a function $\prod_1 = \text{Prog1}(SK, ED, Th)$, where $\prod_1 = \{RUL(LRU_{i,j,k}), \dots, RUL(LRU_{p,r,s})\}$ and the function *Prog2* enables to identify the set of RUL of the function from tests and the set of RUL of LRUs: $\prod_2 = \text{Prog2}(SK, \prod_1, Th)$ where $\prod_2 = \{RUL(F_{i,j}), \dots, RUL(F_{p,r})\}$. $RUL(\cdot)$ denotes either the RUL of a LRU or the RUL of a function.

In this article, the HA of the system is ensured by a diagnostic function and a prognostic function. Data produced by these functions, list of RULs and failed components and functions, are differently interpreted according to the current user of the system.

Considering $HS(X)$ the HS of X , the HS of a LRU can be defined by:

$$HS(LRU_{i,j,k}) = (RUL(LRU_{i,j,k}) \wedge \neg AB(LRU_{i,j,k})) \vee (RUL(LRU_{i,j,k}) \wedge AB(LRU_{i,j,k}))$$

Considering $\Delta_1^{F_{i,j}}$ the list of failed LRUs implementing the function $F_{i,j}$ and $\prod_1^{F_{i,j}}$ the list of the RUL of LRUs implementing $F_{i,j}$ the health status of $F_{i,j}$ can be defined as:

$$HS(F_{i,j}) = (\Delta_1^{F_{i,j}} \wedge \prod_1^{F_{i,j}})$$

The HS of the complex system is also defined as:

$$HS(SC) = (\Delta_2 \wedge \prod_2)$$

5. APPLICATIVE ARCHITECTURE

Distributed approaches of Information and Communication Technologies often provide satisfying solutions to face complexity. The diagnostic function was implemented in a distributed structure according to the multi-agent system concept. The agents of the structure cooperate and exchange data whatever the language used to model the information they contain is. This implementation requires data and models that have been collected and organized. In the case of the complex systems some works show the feasibility to implement an embedded distributed diagnostic function with or without cooperation between its elements (Biteus, 2005 ; Heck *et al.*, 1998 ; Wörn *et al.*, 1998).

The architecture presented herein is based on a distributed implementation. Local diagnosis agents cooperate to make the diagnosis of the system. A middleware makes it possible to implement the services provided by the agents. The architecture consists of several LRUs gathered into several functions denoted LRU layer. Each LRU is observed by a monitoring function. The monitoring layer represents the monitoring functions designed by the suppliers. The monitoring functions send their symptoms to Diagnostic Agents (DAs). Using the cause-symptom relations, each DA determines the set of faulty LRUs among the LRUs that implement a function. One or more databases (KB) contain the set SK that supports the activities of the DAs. A Human/Machine Interface (HMI) displays the failed functions of the system for the production operators and the LRUs to replace or fix for the maintenance operators. If the collaboration is correct the global diagnosis of the system in terms of faulty LRUs is the union of local diagnoses.

Several types of symptom are considered: a “failure symptom” is generated by the monitoring layer, a “propagation symptom” is generated by the function “To propagate symptom” of a DA. A “known symptom” is a symptom with a cause identified by studies made at the system design stage and stored in KB and an “unknown symptom” is the symptom for which the cause of its failure is unknown in KB. The distributed diagnostic function is described by the activity diagram shown in figure 1.

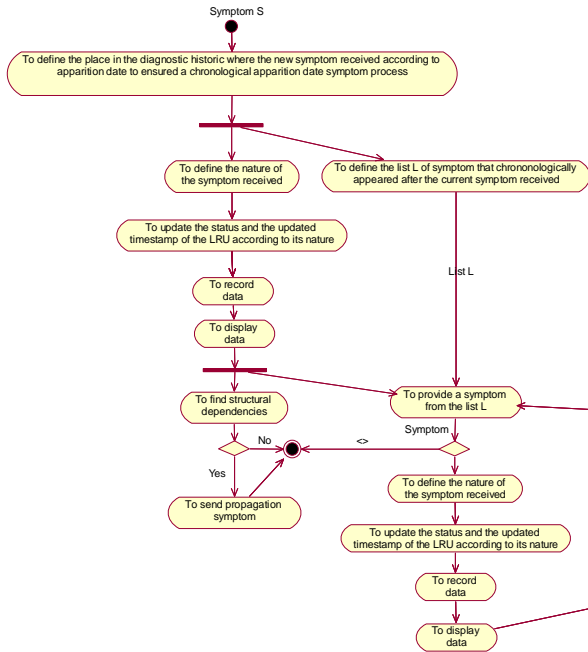


Fig. 1. Activity diagram of DA.

The activities of the diagram are carried out by cooperation between the DAs. The reception of a symptom by a DA starts the process. The corresponding DA begins by inserting the received symptom in the chronologic list according to the

date the monitoring layer. Then, the DA checks the type of symptom received. If the symptom is known, the database returns the cause of failure of the LRU and the DA declares it as known, otherwise, the cause of failure of the LRU is declared as unknown. It also manages the list of symptoms, chronologically. Δ_1 and Δ_2 are updated according to the type of the symptom and the answers of the database dealing with a status of the symptom (known or unknown) in the case of a failure symptom. Then, the DA searches structural dependencies with the failed LRU thanks to KB. If there is at least one dependent LRU, “propagation symptoms” are sent to the DAs that diagnose those LRUs which may not operate correctly and are declared “out of order”. During this stage, the DA updates Δ_1 and Δ_2 according to the chronological list of symptoms. The status and the timestamp of the LRU are updated and data are recorded and displayed. Supervision of the DAs is ensured during “the fault propagation” function. If a DA did not confirm that it receives the message during the propagation task, the diagnostic process declares the agent as failed.

Aiming at prognosing the LRUs and the functions, the monitoring layer sends RUL of each LRU periodically to its Prognostic Agent (PA). Each PA prognoses one function of the system and uses KB. Figure 2 shows the activity diagram of a PA. The reception of a RUL by a PA starts its process. The involved PA updates the new RUL of this LRU in the list it manages and defines the new RUL of the corresponding function by finding the minimum RUL of the RULs of the LRUs that implement this function. Then, it sends the new RUL of this LRU to the PAs prognosing functions carried out by LRUs depending on the LRU for which the RUL has been received. These PAs may then update the RULs of their prognosed functions.

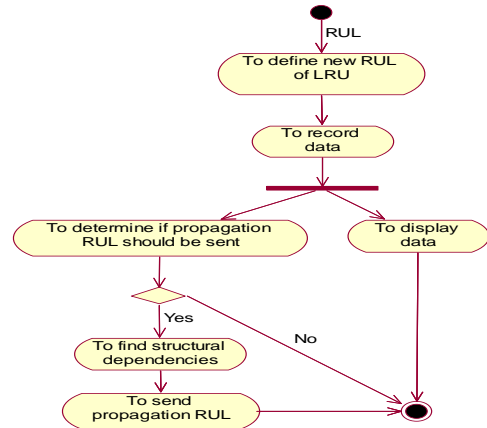


Fig. 2. Activity diagram of PA.

Figure 3 illustrates an example of implementation of the distributed health assessment function onto a system made of 2 functions both ensured by 3 LRUs. This structure leads to the health assessment function to be implemented by 2 DAs and 2 PAs. DAs and PAs receive data generated by the

monitoring layer respectively symptoms (full arrows between monitoring and health assessment layers) and RULs (dotted arrows between monitoring and health assessment layers). DAs communicate between them and KB (bold dotted arrows inside the health assessment layer) as well as PAs do it between them and KB (bold arrows inside the health assessment layer). In fact, the communication between Agent of the same nature (DA or PA) is processed according to the structural dependencies between LRUs and the description of diagnostic and prognostic functions.

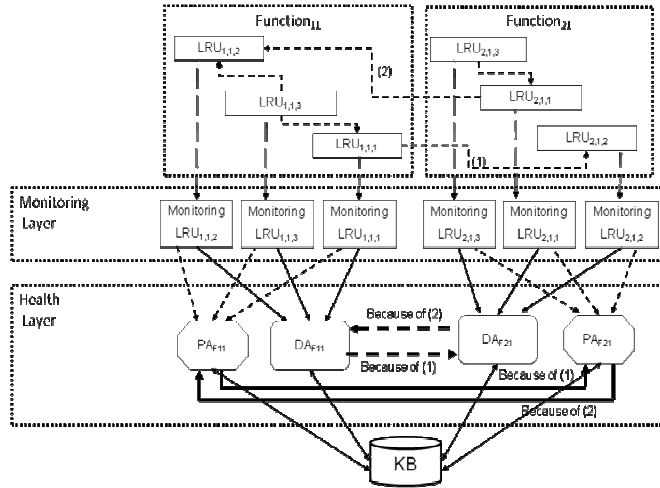


Fig. 3. Example of implementation of the distributed health assessment function

Each time a DA receives a symptom, it copies Δ_1 and Δ_2 and updates them with the result computed from the new symptom. This carries out the tracking of the evolution of the diagnostic process and this enables the diagnostic process to be non-monotone. If the DA receives a “propagation symptom”, this one is used for to refine the diagnosis. To explain this activity, let us considered the an example with 3 LRUs ($LRU_{1,1,1}$, $LRU_{1,1,2}$ and $LRU_{1,1,3}$) structurally dependant. SD (the only considered knowledge for this example) contains knowledge about the system: $CONNECT(LRU_{1,1,1}, LRU_{1,1,2})$, $CONNECT(LRU_{1,1,2}, LRU_{1,1,3})$. Each LRU has its proper monitoring periods for its symptoms. For example, if we consider one symptom per LRU, we suppose that the symptom emitted from the monitoring of $LRU_{1,1,1}$ can be send to the DA every 5 minutes, the one of $LRU_{1,1,2}$ every minute and the one of $LRU_{1,1,3}$ every second. Let us now consider that only $LRU_{1,1,1}$ failed, $LRU_{1,1,2}$ and $LRU_{1,1,3}$ should be defined as “out of order”. But the symptoms are not received in the order they occur because of the different monitoring periods. Considering symptoms $S_{i,j,k,t}^t$ of the $LRU_{i,j,q}$ that occurs at time t and $t_1 < t_2 < t_3$ it can be inferred that $S_{1,1,1,t_1}^{t_1} < S_{1,1,2,t_2}^{t_2} < S_{1,1,3,t_3}^{t_3}$. A difference exists between the time the symptom occurs and the time the symptom is sent. Firstly, the DA

receives $S_{1,1,3,t_3}^{t_3}$. So, $LRU_{1,1,3}$ is declared has failed. Then, the DA receives $S_{1,1,2,t_2}^{t_2}$. As $t_2 < t_3$ and because of the structural dependencies, $LRU_{1,1,2}$ is declared failed and $LRU_{1,1,3}$ is updated to an “out of order” status. Finally, the DA receives $S_{1,1,1,t_1}^{t_1}$. Because of the timestamp ($t_1 < t_2 < t_3$) and the structural dependencies, $LRU_{1,1,1}$ is declared “failed” and $LRU_{1,1,2}$ and $LRU_{1,1,3}$ are declared as “out of order”. That is why four different values describe the state of a LRU:

- “OK” when the LRU is not faulty,
- “UF” when the LRU is faulty but the cause is unknown (Unknown Failure),
- “KF” when the cause of its failure is known (Known Failure),
- “OO” when the LRU does not work in nominal mode or is failed because of the failure of another LRU (Out of Order).

The state of the LRU changes and the timestamps of its changes are updated. At the end of the HA process, a list of faulty LRUs with their causes, their dated changes of status and a list of failed functions are available. The causes of faulty LRUs are described in terms of sentences from FMEA studies or of faulty LRUs for “OO” LRUs. The causes of failed functions are described in terms of faulty LRUs.

For example, if $LRU_{2,1,1}$ and $LRU_{3,1,1}$ failed and the cause of the failure of $LRU_{2,1,1}$ is known to be a power failure and the cause of $LRU_{3,1,1}$ is unknown. After symptom generation, the diagnosis result is given by:

$\Delta_1 = LRU_{2,1,1}(\text{status: KF, cause: power failure, timestamp: 2009/07/05 15h26min56s}) \& LRU_{3,1,1}(\text{status: UF, cause: unknown, timestamp: 2009/07/05 15h27min05s}) \& LRU_{1,1,2}(\text{status: OO, cause LRU}_{2,1,1} \text{ failure, timestamp: 2009/07/05 15h26min57s}) \& LRU_{3,1,2}(\text{status: OO, cause: LRU}_{3,1,1} \text{ failure, timestamp: 2009/07/05 15h27min06s})$

$\Delta_2 = \text{Function}_{11}(\text{status: OO, cause: function}_{12} \text{ failure, timestamp: 2009/07/05 15h26min58s}) \& \text{Function}_{21}(\text{status: KF, cause: LRU}_{2,1,1} \text{ failure, timestamp: 2009/07/05 15h26min57s}) \& \text{Function}_{31}(\text{status: KF, cause: LRU}_{3,1,1} \text{ failure, timestamp: 2009/07/05 15h27min06s})$

In each set Δ_1 and Δ_2 the LRUs or functions that do not operate in a nominal mode are listed with their current states, the causes of their failures and the timestamps of their changes of state.

Considering the same example as described in figure 3 with these RUL values at the beginning of the prognosis process: $RUL(LRU_{1,1,1}) = 10$ TU (Time Unit), $RUL(LRU_{1,1,2}) = 90$ TU, $RUL(LRU_{1,1,3}) = 30$ TU, $RUL(LRU_{2,1,1}) = 20$ TU,

$RUL(LRU_{2,1,2}) = 40$ TU, $RUL(LRU_{2,1,3}) = 50$ TU. After prognosis process, the prognosis is given by:

$\Pi_1 = RUL(LRU_{1,1,1}) = 10$ TU (Time Unit), $RUL(LRU_{1,1,2}) = 90$ TU, $RUL(LRU_{1,1,3}) = 30$ TU, $RUL(LRU_{2,1,1}) = 20$ TU, $RUL(LRU_{2,1,2}) = 40$ TU, $RUL(LRU_{2,1,3}) = 50$ TU.

$\Pi_2 = RUL(F_{1,1}) = 10$ TU, $RUL(F_{2,1}) = 20$ TU.

6. CONCLUSION

The distributed diagnostic and prognostic functions, presented in this paper, implement the health assessment for systems of systems. The health assessment is here based on a monitoring layer that provides symptoms and estimations of the remaining useful lives of the components of the system.

The proposed diagnosis is a non monotonic process that makes it possible to take into account the events detected by the monitoring layer and time stamped before their asynchronous transmissions.

All the pieces of diagnostic and prognostic data and their timestamps are recorded to enable performance evaluation at the end of the health assessment session. Performance indicators (speed of convergence, data flow, and computational load...) may therefore be evaluated from the timestamps of the different data that are exchanged between the agents or stored. Further works will so deal with performance evaluation of the proposed structure and with their comparison with centralized structures.

Acknowledgement: The project was supported by Airbus France

References

- Abu-Hanna, A., Benjamins, R., and Jansweijer, W. (1991). Device understanding and modeling for diagnosis, *IEEE Intelligent Systems and Their Applications* Vol. 6(2), pp. 26-32.
- Biteus, J. (2005). Distributed Diagnosis and Simulation Based Residual Generators, *Dept. of Electrical Engineering*, LiU-TEK-LIC-2005:31, Thesis No. 1176.
- Byington, C., Kalgren, P., Johns, R. and Beers, R. (2003). Embedded diagnostic/prognostic reasoning and information continuity for improved avionics maintenance, *AUTOTESTCON. IEEE Systems Readiness Technology Conference. Proceedings*, pp. 320-329.
- Chittaro, L. and Ranon, R. (2003). Hierarchical model-based diagnosis based on structural abstraction, *Artif. Intell.*, Vol. 155(1-2), pp. 147-182.
- Hamscher, W., Console, L., and De Kleer, J. (1992). Readings in model-based diagnosis, *Morgan Kaufmann Publishers Inc*, San Francisco, CA, ISBN:1-55860-249-6.
- Heck, Florentin, Thomas, Längle and Heinz, Woern (1998). A Multi-Agent Based Monitoring and Diagnosis System for Industrial Components, *Proceedings of the DX '98*, pp. 63-69.
- Hitt, EF (1997). Avionics cost of ownership, *Aerospace and Electronic Systems Magazine, IEEE*, Vol. 12, Issue 11, pp. 3-7.
- Jardine, A., Lin, D. and Banjevic, D. (2006) A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical Systems and Signal Processing*, vol 20, pp. 1483-1510.
- Keuneke, A. (1991). Device representation-the significance of functional knowledge, *IEEE Intelligent Systems and Their Applications*, Vol. 6(2), pp. 22-25.
- Mathew, A. D., Zhang, S., Ma, L., Earle, T. and Hargreaves, D. J. (2006) Reducing maintenance cost through effective prediction analysis and process integration. *Advances in Vibration Engineering*, vol. 5(2), pp. 87-96.
- Mathur, A., Cavanaugh, K., Pattipati, K., Willett, P. and Galie, T. (2001). Reasoning and modeling systems in diagnosis and prognosis, *Component and Systems Diagnostics, Prognosis, and Health Management*, Proc. SPIE Vol. 4389 pp. 194-203.
- Ramohalli, G. (1992). The honeywell on-board diagnostic and maintenance system for the boeing 777, *Digital avionics systems conference, 11th IEEE/AIAADigital Avionics Systems Conference, Proceedings.*, IEEE/AIAA 11th, pp. 485-490.
- Roemer, M., Nwadiogbu, E. and Bloor, G. (2001). Development of diagnostic and prognostic technologies for aerospace health management applications, *Aerospace Conference, IEEE Proceedings*, Vol. 6, pp. 3139-3147.
- Scarf, P. (2007). A Framework for Condition Monitoring and Condition Based Maintenance, *Quality Technology & Quantitative Management*, vol 4, pp. 301-312.
- Vachtsevanos, G., Lewis, F.L., Roemer, M., Hess, A. and Wu, B. (2006). Intelligent fault diagnosis and prognosis for engineering systems, *John Wiley & Sons Inc*, ISBN: 978-0-471-72999-0.
- Villareal, C. and Cicek, I. (2004). Automated troubleshooting tools for minimizing downtime and reducing the labor and material costs of C-5 aircraft, *AUTOTESTCON 2004, Proceedings*, pp. 504-510.
- Wörn, H., Längle, T., and Albert, M. (1998). Multi-Agent Architecture for Monitoring and Diagnosing Complex Systems, *The Fourth International Workshop on Computer Science and Information Technologies*, Patras, Greece.