



**HAL**  
open science

## Investigation for defining a first functional architecture of an intelligent embedded diagnosis system

Walid Mechri, Christophe Simon, Benoît Iung, Didier Theilliol, Flavien  
Peysson

► **To cite this version:**

Walid Mechri, Christophe Simon, Benoît Iung, Didier Theilliol, Flavien Peysson. Investigation for defining a first functional architecture of an intelligent embedded diagnosis system. 14th International Workshop on Advanced Control and Diagnosis, ACD 2017, Nov 2017, Bucharest, Romania. hal-01628908

**HAL Id: hal-01628908**

**<https://hal.science/hal-01628908>**

Submitted on 5 Nov 2017

**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.

# Investigation for defining a first Functional Architecture of an Intelligent Embedded Diagnosis System

MECHRI Walid

CRAN UMR CNRS 7039

Université de Lorraine

Nancy, France

walid.mechri@univ-lorraine.fr

SIMON Christophe

CRAN UMR CNRS 7039

Université de Lorraine

Nancy, France

christophe.simon@univ-lorraine.fr

IUNG Benoit

CRAN UMR CNRS 7039

Université de Lorraine

Nancy, France

benoit.iung@univ-lorraine.fr

THEILLIOL Didier

CRAN UMR CNRS 7039

Université de Lorraine

Nancy, France

didier.theilliol@univ-lorraine.fr

PEYSSON Flavien

PREDICT,

Vandoeuvre, France

flavien.peysson@predict.fr

**Abstract**—This paper aims to investigate a frame for defining a first functional architecture of IEDS (Intelligent Embedded Diagnosis System) in the context of FDIR (Fault Detection, Isolation and Reconfiguration) and predictive maintenance. The IEDS has to be implemented on a system/component to be monitored/maintained (on board part) and should possess a lot of capabilities such as data collection, data processing, health status generation, degradation/failure identification ... allocated either on-board or off-board in accordance to the technological constraints. Its objective is therefore not only to diagnose faults, but also to deliver relevant information/knowledge for aiding decision-making both with regards to FDIR objective or predictive maintenance. The engineering of such IEDS is not so easy to be developed because it is concerned with functional, technological and physical requirements. A frame for supporting the consistency between all these different types of requirements is not really existing today. So, we propose in this paper to investigate a modeling frame based on system engineering principles and leading to a first generic functional architecture of IEDS.

**Index Terms**—Fault diagnostic, health monitoring, embedded diagnostic, functional architecture, intelligent system, maintenance

## I. INTRODUCTION

The needs expressed by various industrial domains such as aeronautics, space, energy, transport, automotive, in terms of maintenance requirements is the identification of operational drifts, and the monitoring of failures [1]. To meet these industrial needs, it is necessary to have a function allowing the diagnosis of failures which can affect the system functioning [2].

A diagnosis system executes the tasks of detection of malfunctions, identification and isolation of the probable root

causes of these anomalies [3], [4]. A diagnostic function generates symptoms from observed data and compares them to the thresholds which define the frontier between a normal and an abnormal operational situation of the monitored system [2].

This diagnostic function can be applied as well to the system, sub-system or component abstraction levels. In relation to this vision at component level, the concept of embedded diagnostic is developed. The embedded diagnostic unit is integrated into the monitored system i.e. is implemented on-board, and aims to execute a diagnostic during operation of the monitored system. Thus, it offers more autonomy and reactivity [4], [5]. The embedded diagnostic allows to use the contextual information available on-board of the monitored system. This complementary information contributes to increase the embedded diagnostic performance.

Some embedded capacities are considered smart or intelligent [6], [7]. Indeed, it goes toward an autonomous intelligent diagnosis which possesses the ability to observe continuously the monitored system during its operation and to elaborate a reasoning allowing to establish a diagnosis in an autonomous manner from these observations; to communicate and to interact with other units in order to better improve the diagnosis system performance [6]. In that way, the results of the conventional diagnosis function could be extended towards *direct* decision-making such as proposed by FDIR concept [5] or towards *indirect* decision-making, for example by using the results as inputs of prognostics process in the frame of predictive maintenance as advocated by PHM (Prognostics and Health Management community).

In link to all the previous features expected for the IEDS, a lot of issues can be isolated such as: What type of capacities have to be associated to the diagnostic system to confer it Intelligence property (I of IEDS)? How to allocate these capacities in a right way between on-board and off-board

This work was supported by the ANR LabCOM PHM-FACTORY, grant ANR-15-LCV1-0005-01 of the French Research Agency. We would like to thank all the LabCOM actors through the entities involved: ANR, PREDICT company, MATERIALIA cluster, CRAN, CNRS and Lorraine University.

possibilities to confer Embedded property (E of IEDS); What are the hardware and software technologies to be used to support concretely the on-board and off-board parts? How to validate the resulting diagnosis architecture? What are the compatibility and interoperability constraints within the architecture? Some works address already some aspects of these problems [5], [7] but it is really missing a modeling framework to support all the engineering of IEDS in a generic manner [4].

Thus, the main contribution presented in this paper, is related to a frame based on system engineering principles and leading to provide a first generic functional architecture for an IEDS (Intelligent Embedded Diagnosis System). In that way, Section 2 presents a state of the art on the existing functional architectures of embedded diagnosis. It allows to identify the limits of these architectures. To overcome some of the limitations, Section 3 is devoted to present a first generic functional architecture for an IEDS in consistence with system engineering practices. Section 4 concludes the paper work and draws some future works.

## II. STATE OF THE ART ON FUNCTIONAL ARCHITECTURE OF EMBEDDED DIAGNOSTIC

A functional architecture is generally investigated during the requirement analysis process as advocated by system engineering practices [8]. It is defined as a set of functions (and their sub-functions) identifying the transformations (from inputs to outputs involving data, material, energies ...) to be performed by the system to complete its mission. The inputs and outputs are the flows exchanged between the functions or with other entities such as the stakeholders.

According to this *functional architecture definition*, several implementations of diagnosis functional architectures exist in many application domains. These architectures depend of the diagnosis function perimeter (e.g. its mission), the nature of the available data on the monitored system and the type of applications addressed [4], [5], [9], [10].

From a very conventional manner, the mission of diagnosis function is generally defined by the FDI term (Fault Detection and Isolation, or Fault Detection and Diagnosis). It leads to a conventional functional architecture where are appearing the sub-functions of fault detection and fault isolation supporting both the fault location and cause identification [11].

By extending this mission with decision-making concerns, the functional architecture can be considered more *smart* [7] but always centered on initial diagnosis purpose. Indeed, relevant example of this extended architecture is obtained through the implementation of new FDIR strategy as proposed by [5] describing an architecture of embedded diagnosis for improved maintenance in the avionic domain. Reconfiguration is added as action to improve the system operation. This architecture allows to identify on-board information sources and automated reasoning techniques which build upon existing built-in-test results to improve fault isolation precision.

Additional functional architectures have been also developed by considering the diagnostic function not only in link

with its own objective but also in link with other needs such as maintenance, prognostics, monitoring ones. For example, other works present a diagnosis module integrated in a health monitoring architecture of a system, where the diagnosis takes a decisive role in the monitoring architecture performance. In this framework, [4] defines an *active* monitoring architecture based on intelligent agents for the maintenance assistance of mobile systems, in the case of railway application. In the same way, [12] proposed a diagnosis function within a health monitoring architecture of system exposed to multiples faults. This architecture is interesting because the presence of a fault has an impact on the next occurrence of faults.

With regards to prognostic process, [13] defines an architecture in which is combined the diagnostic and prognostics functions to form a unique system. A generalization of this coupling, in terms of functional architecture, is represented by the OSA/CBM architecture, for which ISO-13374 is existing. This coupling is also implemented in the domain of Health and Usage Monitoring System (HUMS) [14].

Finally, some diagnosis architectures have been also developed in the context of Integrated Vehicle Health Management (IVHM) [9]. This type of architecture is used to increase the reliability and safety of transport systems and to reduce maintenance costs by implementing on-board advanced diagnosis and prognostic functions [1].

From all the architectures previously underlined, it can be mainly synthesized that:

- It does not exist an approved view on diagnostic functional architecture because depending on the function perimeter, its sub-functions, its interactions with the other functions etc.
- It is difficult to develop a consistent engineering to guarantee the genericity of such architectures.

For finding some solution to these previous issues, we proposed to follow a system engineering approach to construct a generic frame for defining IEDS.

## III. PROPOSAL OF A FRAME TO DEFINE IEDS FUNCTIONAL ARCHITECTURE

The frame proposed to support all the IEDS engineering is constructed from the system engineering practices [8]. More precisely, this frame is referred to three views:

- *Needs and requirements* view: defines what the system (IEDS) must do (functional requirements) and how well it must perform in order to satisfy systems stakeholders needs. It should be representative mainly of the system purpose, mission, objectives.
- *Functional/Logical architecture & scenario* view: aims at transforming the functional system requirements into a coherent **black-box** description of the system use cases (scenario). It means to propose a functional architecture, structuring all the functions and sub-functions that allow the system to perform all the identified operational scenarios throughout its life-cycle. Within this structure, it is also included the exchanged flows between functions and external world (interface).

- *Physical architecture* view: defining the **white-box** architecture capable of performing the required use cases. It leads to select a set of concrete components (on-board, off-board) that support both functions and interactions between the components including physical connexions.

These three phases are addressing more globally two main steps of system engineering as formalized in MBSE (Model-Based System Engineering) [15]: the specification of the system and the the design of the system (the difference between what the system should do, and how it should be done).

With regards to these steps, the contribution developed in this paper is mainly focused on the specification of the IEDS (Needs and Requirements view; Functional architecture and Scenario) by illustrating each item with textual content and schemes because the models based on SysML diagrams are still in progress. It was decided to use SysML diagrams to support MBSE to offer consistency between the required System Engineering processes and not to use **specialised models** such as SADT which are supporting only one SE process (e.g. functional analysis) without considering really the link with the other SE processes in the processes chain. Thus, the main SE processes addressed for IEDS are: IEDS purpose/mission/objective definition, then the IEDS stakeholders in link to the mission, the scenarios/uses cases for which IEDS is utilized, and the functions/sub-functions and flows describing these scenarios (Figure 1). Finally, the link from specification to design and more precisely to physical architecture is investigated to show the potential on-board and off-board functions allocations.

All the contribution proposed is defended as generic for defining an IEDS whatever the application domain is, whatever the target system/component is etc. This generic material should be used for defining a specific IEDSx functional architecture through an instantiation procedure.

#### A. Needs and Requirements view

The proposition of an architecture for IEDS in a functional manner should start from the characterization of the IEDS purpose (or perimeter), mission and objectives (Figure 1) ... by considering some aspects of the stakeholders. It is an iterative process and not only a sequential one.

1) *Purposes, Missions and Objectives of the IEDS:* The IEDS purpose should define why the IEDS does exist. In that way, it is necessary to consider a first interaction with the main stakeholder which is the *target system* on which this IEDS is implemented. Thus the major purposes are:

- To contribute to the improvement of the global properties of target system dependability and mainly availability one,
- To contribute to the improvement of the safety performance,
- To contribute to the reduction of the operation and maintenance costs ...

Then IEDS missions (What the IEDS does?) have to be described to achieve concretely the purposes. They can be summarized as:

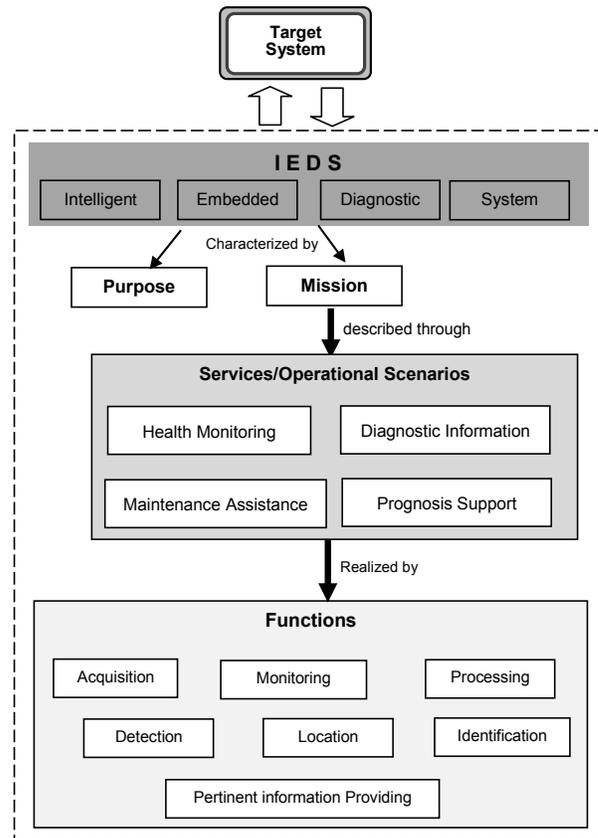


Fig. 1. Way of working for IEDS specification

- To support diagnosis of the target system in terms of FDI for finding fault and its cause,
- To elaborate from FDI results, a health state, a situation on the target system (pertinent information) to be provided in order to aid decision-making in operation and maintenance phases,
- To support *local* action decisions for realizing a first reconfiguration (FDIR) of the target system in the case of failed situation,
- To alert on the potential risks occurring on the target system ...

For each of the purposes and missions, objectives have to be defined representing all the qualitative and quantitative aspects of them. In this frame, it can be the percentage of faults to be detected by the IEDS, the size of ambiguity group to be provided by the IEDS, the degree of confidence to be conferred to the FDI results, the maximum time to provide the cause from the fault detection, ...

The purposes and the missions underlined a part of stakeholders (e.g. target system, maintenance operator) but they have to be completed in the way to take into account all the interactions of the IEDS with its environment.

2) *IEDS Stakeholders:* It is focused on IEDS operational stakeholders (and not engineering ones) as represented by Figure 2. These stakeholders are:

- Target system/component: the IEDS is implemented on

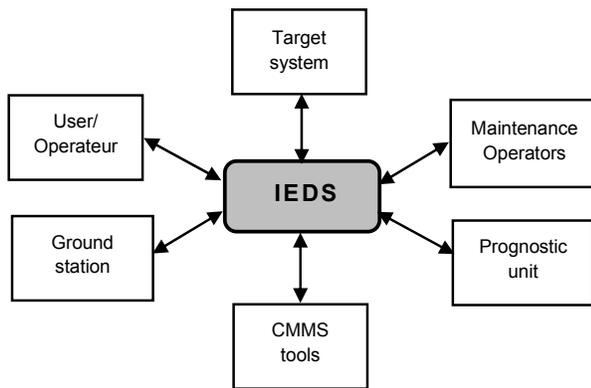


Fig. 2. IEDS Stakeholders

the monitored system/component considered as relevant in regards to the missions. Thus IEDS is connected to target system/component both for acquiring data but also potentially to act on it (reconfiguration case).

- Ground station: The IEDS provides, to ground station, data/information processed by itself from raw data of target system in order to perform off board capacities (in complementary of those done on board). The link between off-board and off-board is necessary to support all the chain of the processes (capacities).
- User/Operator: The IEDS provide to the user relevant information (results of FDI, health state, situation ...) allowing it to follow the degraded/failed situation of the target system/component in the way to take the right decision if needed in relation to operation requirement. It could be to decrease some value of control parameters.
- Maintenance operators: The IDES provide to the maintenance operator relevant information (results of FDI, health state, situation ...) allowing it to follow the degraded/failed situation of the target system/component in the way to take the right decision if needed in relation to maintenance requirement. It could be to anticipate a maintenance action.
- CMMS tool (Computerized Maintenance Management System): The IEDS can transfer to the CMMS tools some values of important indicators (part of health state) in the way to achieve Condition-Based Maintenance (CBM) by the CMMS. In the opposite way, the IEDS can receive information from the CMMS about the last maintenance action done on the target system/component.
- Prognostic unit: The IEDS exchange, with this unit, information on fault cause, degraded situation ... in order to aid the prognosis for calculating Remaining Useful Life of the target system. This RUL can be used also by the IEDS to plan some actions in the reconfiguration phase.

The stakeholders previously mentioned underlined a lot of interactions (in bilateral way) with the IEDS. Indeed, the IEDS is able, in accordance with its purposes/missions/objectives, to receive and provide data/information to or from stakeholders.

It should be done by following a use case, an operational scenario.

### B. Functional architecture and scenarios view

1) *IEDS Uses Case / Scenarios*: An operational scenario aims to describe a *story* based on the expected utilization of the future IEDS in terms of actions required to achieve a mission. In that way, it was isolated 4 main scenarios: FDI, Health Monitoring, Maintenance, and Prognostics.

- Diagnosis: From the data acquired on the target system, IEDS is able to detect fault, to localize fault, to identify the cause (FDI capabilities) and to alert different stakeholders.
- Health Monitoring: From the results of FDI but also CMMS information, the IEDS is able to create health indicators of the target system, to merge these indicators to create a situation state (health state) well representative of the target system performances and to transfer this state to the stakeholders.
- Maintenance: From the results of FDI, the health state and CMMS information, the IEDS is able to calculate some pertinent dysfunctional indicators representative of the degradation/failure mode detected and their impacts on target system performances. All this pertinent information has to be provided to maintenance operator mainly for aiding his decision-making process.
- Prognostics: The IEDS is able to provide to Prognostics unit all the information/indicators required as input for the prognostics in the way to calculate the RUL.

These operational scenarios are implemented through a set of functions.

2) *IEDS functions and sub-functions*: The IEDS functions aim to support the scenarios by means of a chain of value based on the transformation of input flows (e.g. raw data) to output flows (e.g. health state). Thus the flows are the links between the functions but also with the stakeholders. The main IEDS functions are:

- To acquire data,
- To format and normalize data,
- To store the normalized data,
- To detect and isolate faults from normalized data (FDI),
- To generate functional indicators from normalized data,
- To generate dysfunctional indicators from normalized data,
- To generate a health state indicator of the target system/component,
- To store all the generation results,
- To decide reconfiguration actions from normalized data and indicators,
- To perform reconfiguration action on the target system/component (FDIR),
- To communicate generation results, normalized data, reconfiguration report ...

All these functions should be detailed in sub-functions to go towards the *white-box* vision (design step). For example, the function called *To acquire data* can be detailed

with sub-functions as *To acquire raw data from target system/component*, *To acquire maintenance data from CMMS tools*, *To acquire data from Ground station*, *To acquire data from Prognostic Unit*. On the same principles of decomposition, we can illustrate the list of sub-functions with the following items:

- To filter data,
- To format data,
- To validate the data,
- To generate information from validated data,
- To detect fault from validated data,
- To localize the fault,
- To identify the fault cause or the group of ambiguity,
- To define the degradation mode, the most credible attached to this fault cause,
- To define health indicators (functional, dysfunctional) from the validated data,
- To aggregate these health indicators to form health state,
- To provide the health state to maintenance operator and user,
- To analyze the validated data, the fault, the indicators to propose reconfiguration actions (software vision),
- To select reconfiguration action, the most suitable to re-establish target system/component performances (e.g. dependability parameters),
- To operate the reconfiguration action . . .

From the identification of these sub-functions and the flows they are processing (inputs, outputs), algorithms can be selected to achieve the behavior of these sub-functions. In the modeling frame for IEDS, the main idea is to be able to select the right type of algorithms with regards to the type of inputs flows to be processed. For example, in link to the *knowledge* available on the target system/component (and also provided by other stakeholders such as CMMS tools), the digital technologies for supporting the FDI algorithms will be different [6]. This knowledge can be represented either by data, a history, an experiment, either by a known or estimated model of the target system behavior [16]. Therefore, a generally accepted categorization for FDI algorithm is built on four typologies of digital technologies:

- *Model-based numerical technologies*: They rely on precise physical knowledge of the monitored system and are based on a mathematical representation of the behavior of the target system. In this cas, the IEDS makes it possible to determine a residual (difference between actual measurements and values estimated by the mathematical model used) [16]
- *Data-driven numerical technologies*: These are used when the sensor data (several available data) constitute the main source of information. The IEDS allows identifying and understanding the degradation phenomena of the system in order to characterize these actual states.
- *Experience-based numerical technologies*: They do not require a thorough physical knowledge of the monitored system, neither the disposition of sensor data. This type

of technology simply requires the provision of failure history or component design recommendations under similar operational conditions. The IEDS is used to determine and to identify failure possible cause of the target system [6].

- *Hybrid numerical technologies*: They combine several types of technologies (data-driven, experience-based and model-based) to take advantage, for example, of knowledge about the physical behavior of the target system and information obtained from a pertinent analysis of the data [16].

The algorithms selection is the last step of the functional architecture development. The next step is now to allocate the different functions/sub-functions on an organizational/physical architecture.

### C. Way from functional to physical architecture

The organizational/physical architecture is a structure of functions and sub-functions. It allows to identify a set of concrete components that supports both the functions and the interactions among the components, including the physical connections. Sub-Functions to be detailed as operations are projected on a physical architecture (organic) consisting of a set of components both hardware and software (Figure 3). By this way, the functions/sub-functions are performed by components, these components are connected with the others through interface, port, link, network . . . which are associated to flows exchange (depending of the type of flows), and finally all the components, connections etc. form the physical architecture required to perform the operation scenarios. In the IEDS context, the architecture is mainly decomposed into an embedded part (on-board) and a base ground part (off board). The particularity of an IEDS architecture is the ability to have both on-board and off-board diagnostics. The communication of an IEDS with its stakeholders is realized according to the available connections.

For example, in the case of a car, it can be proposed different IEDS-based physical architectures to observe car degradations in the way to diagnose/prognose what is/will happened and then to take the right decision from the degradation level (ex. stop the car, make a maintenance action, reduce the speed). Indeed, for a car, a generic IEDS can be developed in the form of a raspberry pi supporting all the functional capacities (functions, sub-functions) previously proposed in terms of data processing, communication and storing. Thus a specific IEDS-based physical architecture (distributed one) could be developed from implementing different copies of this raspberry. Each copy can be **attached** to a specific car component (acquiring the data of it) considered as critical one, can be parametrised to this specific component such as brake, engine block, active suspension (instantiation step from the generic capacities) and can communicate with the others (ex. zigbee protocol) both with those implemented at the components level and those implemented at higher levels (e.g. moving function, or car itself) to synchronise the information for taking a global decision.

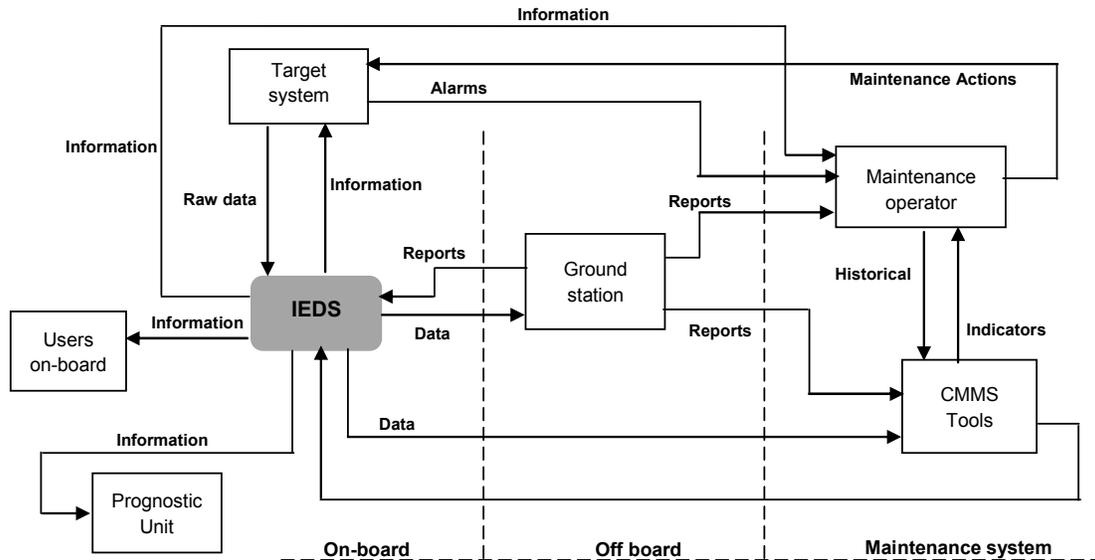


Fig. 3. IEDS organizational architecture in link to monitored system

This organizational/physical architecture is the last step of the design phase of the IEDS. It has to lead to validation phases to assess if this resulting architecture is supporting all the needs/requirements expressed (see section 3.1.) both from a qualitative and quantitative aspects as defined in the objectives.

#### IV. CONCLUSION

The work presented in this paper is focused on a methodological investigation for proposing a frame supporting the definition of an IEDS functional architecture. This investigation is based on system engineering practices by considering mainly the specification step (*needs and requirements view; functional and scenarios view*) allowing to establish a generic content usable for different specific IEDS. A way towards physical/organizational architecture is also presented to illustrate how the functions/sub-functions can be allocated to on-board and off-board components connected together. This contribution has to lead now to a formalized step as already started by the MBSE approach in progress. A lot of additional issues have to be considered in the future as: the validation procedures to be implemented; the integration of system constraints, the instantiation procedure needed for constructing a specific IEDS from the generic functional architecture etc.

#### REFERENCES

- [1] A. Muller, A.C. Marquez, and B. Iung. On the concept of e-maintenance: Review and current research. *Reliability Engineering and System Safety*, 93(8):1165–1187, 2008.
- [2] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & chemical engineering*, 27(3):293–311, 2003.
- [3] A.K.S. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510, 2006.
- [4] A. Le Mortellec, J. Clarhaut, Y. Sallez, T. Berger, and D. Trentesaux. Embedded holonic fault diagnosis of complex transportation systems. *Engineering Applications of Artificial Intelligence*, 26:227–240, 2013.
- [5] C.S. Byington, P.W. Kalgren, R. Johns, and R.J. Beers. Embedded diagnostic/prognostic reasoning and information continuity for improved avionics maintenance. In *AUTOTESTCON. IEEE Systems Readiness Technology Conference*, 2003.
- [6] G.J. Vachtsevanos and F.L. Lewis. *Intelligent fault diagnosis and prognosis for engineering systems*. John Wiley & Sons, Hoboken, New Jersey, 2006.
- [7] X. Olive, S. Clerc, and D. Losa. Smart architecture for highly available, robust and autonomous satellite. In *18th IFAC World Congress*, 2011.
- [8] *Systems Engineering Handbook - A Guide for System Life Cycle Processes and Activities*. John Wiley and Sons, Inc., Hoboken, NJ, 4.0 edition, 2015.
- [9] O. Benedettini, T.S. Baines, H.W. Lightfoot, and R.M. Greenough. State-of-the art in integrated vehicle health management. *Proc. IMechE, Part G: J. Aerosp. Eng.*, 223(2):157–170, 2009.
- [10] G. Medina-Oliva, A. Voisin, M. Monnin, F. Peysson, and J-B. Leger. Prognostics assessment using fleet-wide ontology. In *Annual Conference of the Prognostics and Health Management Society*, Minneapolis, Minnesota: United States., 2012.
- [11] I. Hwang, S. Kim, Y. Kim, and C. E. Seah. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18(3):636–653, May 2010.
- [12] B. Zhang, C. Sconyers, R. Patrick, and G. Vachtsevanos. A multi-fault modeling approach for fault diagnosis and failure prognosis of engineering systems. In *Annual Conference of the Prognostics and Health Management Society*, 2009.
- [13] S. Narasimhan, E. Balaban, M. Daigle, A. Roychoudhury, I. Sweet, J. Celaya, and K. Goebel. Autonomous decision making for planetary rovers using diagnostic and prognostic information. In *8th IFAC Symposium on Fault Detection*, 2012.
- [14] J.E. Land. Hums-the benefits-past, present and future. In *Aerospace Conference, IEEE Proceedings*, volume 6, pages 3083–3094, 2001.
- [15] J.P. Hoffmann. *Systems Engineering Best Practices with the Rational Solution for System and Software Engineering*. Somers, NY, USA, 2011.
- [16] K. Tidriri, N. Chatti, S. Verron, and T. Tiplica. Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual review in Control*, 42:63–81, 09 2016.