A Bayesian paradigm for aircraft operational capability assessment and improved fault diagnostics

Borja Sanz López[1], Antonino Marco Siddiolo[2], Partha Pratim Adhikari[3], and Matthias Buderath[4]

[1,2,4] *Airbus Defence and Space, Rechliner Straße, 85077 – Manching, Germany*

borja.sanzlopez@airbus.com
antonino-marco.siddiolo@airbus.com
matthias.buderath@airbus.com

[3] *Airbus India, Bangalore, 560 048, India*

partha.p.adhikari@airbus.com

ABSTRACT

In recent years, Bayesian networks have been drawing attention of the industrial and research community especially in the field of diagnostics for the reasoning capabilities they offer under conditions of uncertainty.

Given the system of interest, a Bayesian network represents a graphical model of the system itself, in which the different players are linked to each other through probabilistic and causal relations. If the model is queried with appropriate statistical techniques, the whole approach can present several advantages over other data analysis methods. Among the others: 1) the approach can provide outputs even if some entries to the model are missing, due to the above mentioned dependencies between the players of the system; 2) the approach represents an ideal environment to include prior knowledge during the building up of the model, given the causal and probabilistic semantics; 3) a Bayesian network provides the possibility to learn causal relationships and gives therefore the possibility to improve the domain knowledge.

Airbus Defence and Space has been working on improving the aircraft diagnostics capabilities at component, sub-system and system level in terms of fault detection and isolation. The focus has been also to develop means for reasoning about the remaining operational and functional capabilities of the aircraft.

The initial outcomes have been tested on a simulation platform featuring a Data Acquisition Processing Unit, various computing nodes, on which the different aircraft systems (like the fuel system, the hydraulic system, the actuation systems, etc…) run. The data communication architecture of the platform is based on OSA-CBM (Open System Architecture for Condition-Based Maintenance).

Initial objectives of the project are: 1) to demonstrate the feasibility of integration of the concept within the above described simulation framework; 2) to develop means to allow an easy and structured translation of the system engineer knowledge in terms of a Bayesian network with associated conditional probabilities; 3) to provide a modular architecture for the concept facilitating effective coordination between the development-departments and efficient development and maintenance of the software and 4) to prove the scalability of the concept (i.e. applicability to systems of different sizes and reasoning on different levels from component to system level).

The candidate systems selected for the proof of concept are the fuel and the hydraulic systems of a generic aircraft. The results obtained so far look promising with respect to the above mentioned objectives of the project.

ABBREVIATIONS

ACE	Arithmetic Circuit Evaluation
ACFC	Air Cooled Fluid Cooler
BITE	Built-In Test Equipment
BN	Bayesian Network
CPT	Conditional Probability Table
DAG	Directed Acyclic Graph
EPGDS	Electrical Power Generation and Distribution System
FCOC	Fuel Cooled Oil Cooler
HLR	High Level Reasoning
HLS	High Level Specification
ISHM	Integrated System Health Management
LH	Left Hand
LRU	Line Replacement Unit
OFT	Operational Functional Tree

OSA-CBM	Open System Architecture for Condition-based Maintenance
OSA-EAI	Open System Architecture for Enterprise Application Integration
XFER	Transfer

1. INTRODUCTION

The sensory system of modern aircraft has been becoming more and more comprehensive and widespread. For example (Canaday, 2016), Airbus's A320 generates 15,000 parameters per flight, the A330 30,000, the A380 250,000 and the A350 will generate 400,000 parameters. This has opened up new opportunities in the field of aircraft diagnostics and awareness, and approaches based on the so called "Big Data Analytics" and "Internet of Things" are trying to make profit out of the immense availability of new data (Canaday, 2016). Per definition the conceived approaches are mostly data-driven and therefore are built upon the existence of such a big amount of data out of which trends and patterns are derived by means of the most advanced artificial intelligence techniques.

The present work starts from the very same observations and aims at the same results. However, it bases its foundations on an extensive usage of Bayesian Networks (BNs). The concepts and the mathematics behind the BNs will be briefly recalled within the next section. BNs represent in our vision an ideal environment in which different forms of knowledge can come together and collaboratively work. Physical knowledge of processes and models, inductively understood patterns and laws coming from in-service data, the expert advises of experienced and wise engineers, etc… represent all pieces of knowledge usually difficult to harmonize but that can find in the Bayesian paradigm an homogeneous domain to interactively support the reasoning about the aircraft health.

For the sake of completeness, it has to be recalled the *de-facto* existing methods for reasoning: case-based reasoning (Kolodner, 2014), rule-based reasoning (Davis & King, 1984), model-based reasoning (Davis & Hamscher, 1988) and data-based reasoning (Schawacher, 2005). Leaving to the reader the chance to go deeper into the other topics, the reasoning by means of BNs falls back into the model-based reasoning, in which one can distinguish the usage of explicit model to aid intelligent reasoning processes (in particular in this work it has been referred to static modelling).

Overall objectives of the present work are: 1) to improve the current fault isolation capabilities of existing diagnostics concepts and 2) to increase the operational and functional aircraft capability awareness. The above mentioned ones are long-term aims which have been guiding us throughout the project; this paper will however bring just limited evidences about the achievements of these objectives. As a matter of fact, being at an initial stage, the focus has been placed on the following topics, whose positive assessment represents in our perspective an essential prerequisite for a successfully implementation of the technology in future aircrafts: 1) to demonstrate the feasibility of the approach by integrating it in a relevant environment; 2) to establish a process to effectively and efficiently create a BN; 3) to allow a modular architecture of the whole framework; 4) to prove the scalability of the approach.

The work is structured as follows. Section 2 will provide a brief introduction to the mathematics behind BNs: the topic has been treated in many outstanding papers, which will be referenced and recommended for the ones who desire going deeper into the topic. After having provided the reader with the tools to understand the BNs' capabilities, section 3 will comprehensively unveil the details of the developed framework, i.e. the approach that has been followed to effectively use Bayesian reasoning for assessing aircraft functional/operational capabilities and for performing improved fault isolation. The general discussion will finally find an example in the test case which the section will be focused on: the capabilities of the developed reasoning tool will be shown, demonstrated and commented by means of a test-case. The paper will finish with a section that will summarize the outcomes of the presented work and – above all – trace the road-map of the future tasks that will be addressed, in order to increase the maturity level of the technology under investigation.

2. BAYESIAN NETWORKS

A BN is a compact, probabilistic graphical model of a probability distribution, which is used to represent: 1) a set of random variables and 2) the corresponding conditional dependencies via a Directed Acyclic Graph (DAG). A DAG is a graph with no closed chains (acyclic) with edges being oriented. BNs are particularly well-suited for modeling systems - under the presence of uncertainty - which need to be monitored, diagnosed and for which predictions have to be performed. For example, a BN can be utilized to model the probabilistic relationships between the mechanical failures of a pump and the readings of its sensors: once the values from the sensor have been read, the network can be queried to reason about the probability of occurrences of the different modeled failures.

From a mathematical point of view, BNs are DAGs whose nodes represent random variables in the Bayesian sense: therefore, they may be observable quantities (in this context usually called *evidences*), latent variables, unknown parameters or hypotheses. Edges between nodes denote conditional dependencies or typically causal relationship (although causality is not a requirement). Variables/nodes, that are not connected, are conditionally independent variables: more precisely it is usually stated that - given its parents - every variable is independent of its non-descendants. Each variable can assume values among a set of mutually exclusive states: it is often the case in which a variable has just two states: true or false. Each node is associated with the Conditional Probability Table (CPT), that quantifies the local relationships between a variable (and each one of its states) and its parents; in other words, in a tabular form, a local probability function is given to each node/variable. The CPT takes as inputs a particular combination of the values of the node's parents and gives as output the probability of each state that the variable can assume; this gives the conditional probability distribution of a variable X given its parents **U**. Through this local conditional distributions a global probability distribution over all network variables **X** is induced. If it is referred to **x** as a particular instantiation of these variables (in which each x component of the **x** vector represents the value associated to each variable X), then the probability associated to the happening of **x** is given by the product of the conditional probabilities $P(x|\mathbf{u})$ for each x that was set in **x**, in which **u** represent the sub-instantiation of **x** over the parent variable **U**:

$$P(\mathbf{x}) = \prod_{X \in \mathbf{X}} P(x|\mathbf{u}) \quad (1)$$

As can be probably thought by the above discussion, an important property of the BNs is their capability of compactly representing a joint probability distribution in term of local conditional distributions. This local representation, together with a small number of parents for all nodes (a CPT has a size that is exponential with the number of variables that are defined in it) is what makes the mathematical problem still tractable.

The computation of the probabilities of the states of selected nodes/variables of the network is called inferring, as information from some other nodes, acting as evidences, are used to derive information at other levels.

To let appreciate the process of reasoning on a system by means of BN is typically considered one of the multiple versions of the following example. This version is the one published in Wikipedia, and the origin of the "Sprinkler Network" can be traced to Darwiche (1993) and Russell and Norvig (1995): "Suppose that there are two events which could cause grass to be wet: either the sprinkler is on or it's raining. Also, suppose that the rain has a direct effect on the use of the sprinkler […]. Then the situation can be modeled with a BN (Figure 1). All three variables have two possible values, T (for true) and F (for false)". Given the above discussion on BNs, one can now distinguish the nodes/variables with the corresponding states, the causal edges linking them, and the CPTs, that are giving - in tabular fashion - the probability distributions of the nodes' states given the parents' states. The so structured BN can then be utilized to answer questions like: "What is the probability that it is raining, given that the grass is wet?", or - in other words – which is the value of $P(R{=}T|G{=}T)$?

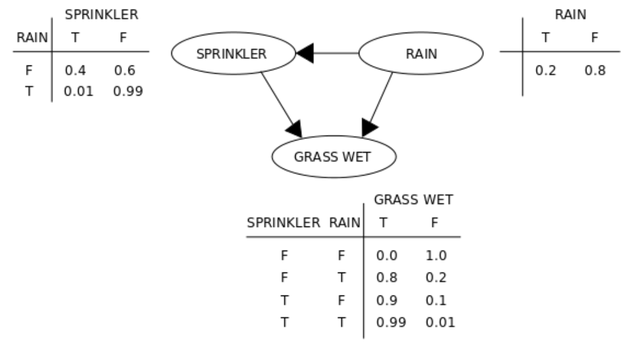

Figure 1. A simple BN with conditional probability tables (source: Wikipedia https://en.wikipedia.org/wiki/Bayesian_network).

In this section a short overview of the mathematics behind the usage of BNs has been provided to the reader. A short review of the publications in the last decades reveals how BNs have quite impressively established themselves as an indispensable tool in artificial intelligence, and are being used effectively more broadly in science and engineering. The domain of system health management is no exception to this trend. As a matter of fact, diagnostic applications have driven much of the developments in BNs over the past few decades, as shown in the recommended references.

The following section will discuss the details of the conceived implementation, by properly locating the BN approach within a reasoning paradigm characterized by well-defined requirements and objectives.

3. HLR BY MEANS OF BNS

As already stated, main aim of the project is to develop a framework in order to 1) improve the current fault isolation capabilities of existing diagnostics concepts and 2) to increase the operational and functional aircraft capability awareness. The framework in object has been called High Level Reasoning (HLR). In what follows, selected design requirements - to which HLR has to comply - are listed: 1) HLR shall have the provisions to accept, as evidences, outcomes of BITEs (Built-In Test Equipments), warnings, commands, sensors information, etc... 2) HLR shall provide as output the most probable failure modes, the remaining and lost operational/functional capabilities of selected Line Replacement Units (LRUs), sub-systems, systems or combination of systems, as well as the health grade of selected LRUs; 3) the HLR framework shall be composed by a module (System Model Definition) for defining the network structure (variables, dependencies and CPTs); a module (Model Creator and Validator) to create and validate the network and finally the HLR inferring module itself; 4) the HLR inferring module (or - in short - the HLR module) shall be the only module of the framework to run onboard; 5) the HLR module shall be real time capable; 6) the HLR framework code shall be open source, multi-platform and written in a certifiable programming language; 7) the HLR framework code shall be generic and valid for every system without software modifications; 8) the HLR Model Creator and Validator shall provide the means to easily create interfaces among systems; 9) the System Model Definition shall be readable as plain-text, in accordance to a specific standard to be defined; 10) the HLR framework shall use only deterministic algorithms; 11) the HLR Model Creator and Validator shall be able to generate out of the System Model Definition (High Level Specification) a graph that can be processed by the HLR inferring Module; 12) the HLR framework code shall meet the requirements of DO-178B and should meet the requirements of DO-178C.

After an extensive review of the literature, a BN approach has been chosen as a suitable candidate for representing the inferring engine of the HLR framework. BNs provide a traceable mathematical structure, in opposition to neural networks or fuzzy-logic algorithms. Although probabilistic outputs are calculated, the behavior of the network itself is deterministic, which enables the certification of systems based on this technology. The uncertain knowledge is moreover explicitly handled and the networks can be represented in a graphical and intuitive manner. This technology has been already satisfactorily applied to fault diagnosis of aerospace systems, as demonstrated in Mack, Biswas, Koutsoukos, Mylaraswamy and Hadden, (2011) and Mack, Biswas, Koutsoukos and Mylaraswamy (2011), for engine applications; Barua and Khorasani (2009) for satellites and Ricks and Mengshoel (2009) and Mengshoel, Chavira, Cascio, Poll, Darwiche and Uckun (2008) for electrical power systems of aircraft. Other common applications of this technology are health care, finances, artificial intelligence and data mining. Also, the use of system trees to improve the performance of the BNs has been studied in François and Leray (2006) and Cerquides and López de Màntaras (2003), both investigating the tree augmented naïve Bayes classifier, with positive results.

3.1. How to build the model

Considering the needed level of detail, a typical BN, which is modelling the probabilistic behavior of a system, may be composed by thousands of nodes. Moreover, given the relationships between child- and parents-nodes an even bigger challenge is represented by the need of filling out the corresponding CPTs. Therefore, manual construction of a BN in terms of structure (nodes and edges) and CPTs is usually almost impossible.

In the current work, a different approach has been followed. As a matter of fact, instead of manually building the network, the meaningful info to create a network is communicated and specified through plain text by using a set of rules that has been called: “High Level Specification” (HLS), similarly to the concept proposed in Mengshoel et al. (2008). By means of the HLSs it is possible to specify, among the other settings: 1) the type of node and is position, 2) by how many states the node’s behavior is described, 3) the probabilities of its states (in case dealing with leaf nodes), 4) how the node is linked to the surrounding nodes (which are its parents), and – above all – 5) by means of which procedure (as will be more discussed in the next section) the CPT (if any) associated to the node can be automatically derived. The last mentioned capability represents in particular one of the biggest motivations that pushed us to deal with the task by means of such an approach. The approach follows moreover the concepts developed and already presented by other researchers Mengshoel et al. (2008).

In order to provide a hint for the reader, in what follows an exemplary set of instructions given to define a generic health node is provided:

Table 1. Node definition through HLSs.

Node (Health_X).label	= 'Health_X';
Node (Health_X).state_no	= 2;
Node (Health_X).states	= {'Healthy', 'Fail'}
Node (Health_X).parents	= {Nd_1, Nd_2, Nd_3};
Node (Health_X).parents_weight	= {0.3, 0.3, 0.4};
Node (Health_X).operation	= OR;
Node (Health_X).position	= [2, 1];

The HLSs have to be filled out using Matlab syntax; the engineer responsible for the system in object should specify the HLSs with the support of the HLR team. As output, the Matlab script will generate a *net* file (compatible with .net format v5.7 of Hugin©) that fully and exhaustively describes the BN. This file can also be read by other tools available in the Internet, like the well-known software *SamIam* (UCLA Automated Reasoning Group, 2002), a comprehensive tool for modeling and reasoning with BNs, developed in Java by the Automated Reasoning Group of Professor A. Darwiche at UCLA. The mentioned tool was actually profitably used during the developing stage of the approach, since it provides an intuitively and efficient way for handling, editing and finally checking performances and results of BNs. At the current development stage, the SamIam tool is merely utilized for reading and validating the BN generated out of the *net* file; this procedure is carried out with the help of a system engineer, who provides means to properly stimulate the created network by means of selected scenarios (clamping of evidence) and evaluating the corresponding reaction of the network itself (answer to selected statistical queries).

3.2. Automatic CPT generation

Some nodes of a BN are associated with a CPT, which depicts the conditional probabilities of each state, depending on the states of the parent nodes. This information represents the core of a BN and defines the relation among nodes. A proper definition of these conditional probabilities is of crucial relevance for the use of this technology for aerospace applications and also to meet the certification requirements. The derivation of the conditional probability of an event (a certain state of a node), given the conditions of other events (certain states of the parent nodes) is not a trivial problem and a solution for the generic case has not yet been formulated. Therefore an expert or in-service data are usually needed to provide these values to the CPT. Despite the attempt of maintaining the network compact and the numbers of parents limited, the manual filling out of the CPTs is not always feasible and different solutions have been proposed over the years to make the challenge more treatable. As a matter of fact, looking at the literature, different papers propose various methods based on the use of probability distributions, weights and parameters; however, no algorithm is applicable in all cases. Das (2008) proposes a weighted sum of probability distributions; Kokkonen, Koisuvalo, Laine, Jolma and Varis (2005) propose the use of link strength parameters; Barua and Khorasani (2009) propose a weighted sum of probability distributions made with belief adjustment factors and Mengshoel et al. (2008) propose an H-E-C-P-R customized algorithm that is different for each type of node. The common idea to all the mentioned approaches is to compute the CPTs, based on parameters which are predefined by experts who are in charge of estimating how relevant each relation is. Another solution would be the use of statistical (in-service) data to feed the CPTs. As also discussed in Das (2008), learning the conditional probabilities out of databases is possible, but there is no standard method for such purpose and a proper database is usually not available.

Moving from the very same understanding but recognizing also the relevance and above all the opportunity of taking profit of existing documentations which could and shall further help defining the CPTs (e.g. existing FMECA, FTA, etc…), the developed HLSs allow from one side the easy translation in terms of CPT of the information contained in – for example – a FTA, similarly to Jong and Leu (2013), and from another side the flexibility of including expert knowledge by means of a relatively simple labelling operation of every edge by the corresponding weighting factor.

Therefore, having at hand a FTA of a system could be profitably used as starting point to derive an initial network for the Bayesian reasoning: the probabilities contained in the CPTs will in this case collapse in certain (instead of uncertain) values, having to replicate the behavior of logical ANDs or logical ORs. Furthermore, a more representative OR operation (called "Weighted OR") has been implemented, in which the different edges are labelled with weighting factors based on the contribution of parent nodes to the subsequent nodes, see Jong and Leu (2013). Furthermore, two additional methods of CPT automatic derivation have been implemented. The methods in object consider the expert judgment and namely are: the "Naïve method" and the "Hyperbolic tangent method". The first one is based on the naïve assumption that all events are independent and is based on the Bayes theorem. In what follows is an example of the conditional probability of an event A, given the independent events B and C:

$$P(\mathrm{A}|\mathrm{B},\mathrm{C}) = \frac{P(\mathrm{B}|\mathrm{A}) \cdot P(\mathrm{C}|\mathrm{A}) \cdot P(\mathrm{A})}{P(\mathrm{B}) \cdot P(\mathrm{C})} \qquad (2)$$

The second method is an empirical one that is following the idea of considering weighting factors and the expert opinion. More specifically the method has been developed following advices coming from the system engineers: being a child node depending from several parents, who all contribute to generate confidence on definite states of the child itself, giving the system engineers' opinion, a sort of saturation of the information has to be modelled, in the authors' opinion. As a matter of fact, if some evidences already support the happening of an event, additional evidences on the same direction could not increase remarkably the probability of the event (this statement is in contradiction with the hypothesis on the basis of the naïve method). The saturation effect has been implemented by conceiving a non-linear sum of the available evidences by means of the hyperbolic tangent as non-linear mapping.

3.3. The inference algorithm

Many different algorithms are nowadays available and some of them are also free and open-source: a complete comparison can be seen in Guo and Hsu (2002). Some of these algorithms allow an exact inference, leading therefore to exact results; in this class, one distinguishes the "elimination" algorithms (variable elimination, joint tree algorithm), the "conditioning" algorithms (cutset conditioning, recursive conditioning) and the "compilation" ones (arithmetic circuit evaluation). On the other hand, in the approximate inference class, one finds algorithms like: the "loopy" belief propagation, the stochastic sampling and search and the "vibrational algorithms".

After an extensive survey and based also on the above stated requirements for the HLR framework, the C++ library Dlib (King, 2009) has been chosen: it offers an implementation of the "exact inference" algorithm named "Join Tree Algorithm" (a.k.a. junction tree algorithm). The library in object is multi-platform, open source and free. Being written in C++ has allowed an almost straightforward integration into the overall framework. Since the primary objective is to prove the feasibility of the approach within a short time-window, it has been opted for the above mentioned C++ based Dlib library, against other toolboxes that would have been required code conversion into C or C++ language. It has shown however some disadvantages: for example it uses - as said - an implementation of the inferring engine "join-tree algorithm" and the mentioned implementation is such that, as long as new evidences are available, the computation starts again from the very beginning, losing therefore (or at least not taking profit of) the status-knowledge that was gained up to that time-instant. The performances look therefore far away from the "real-time capable" requirement stated before. Nevertheless, giving the initial stage of the technology project in object and other constraints, as remarked before, the solution looked a fair compromise. Moreover, having the framework a modular architecture, with fixed and defined interfaces between modules, the inferring module could be easily swapped in the future with a more performing one. In particular our interest has been focused on those algorithms which are moving off-line most of the computational and resource-consuming tasks. As a result, these algorithms produce secondary structures, known as arithmetic circuits (and the approach itself is called Arithmetic Circuit Evaluation - ACE), that can be more efficiently processed on-line, see Mengshoel et al. (2008).

3.4. Modularity of the HLR framework

Modularity has been defined as one of the primary requirements of the HLR framework.The "System Model Definition" is a module mainly utilized by the system engineer responsible for the system under study, who - by following predetermined rules (HLSs) – determines: 1) the main players (nodes/variables) of the network, 2) the states they can assume, 3) causal relations (edges) between the above mentioned players to be established, and - most importantly – 4) the kind of probability distribution (CPT) to be associated with the nodes. This module can be considered as the most critical step of the overall framework; it does correspond to the translation of the system knowledge in structured information that can be used within a Bayesian paradigm. A process has been established to effectively and efficiently allow the model creation. The process is iterative and tries to detail from a Bayesian perspective the system content progressively at each step. An extensive usage of questionnaires, interviews, meetings is pursued with the objective of gaining a representation of the system behavior with the desired granularity at the end of the definition process.

The HLSs - as output of the "System Model Definition" module - are taken as input by the "Model Creator and Validator" module. This is a Matlab-based module which will deliver, as output, the already mentioned *net* file. The *net* file contains the same information contained within the HLSs but differently organized and – for example –readable by tools like SamIam (this capability is actually used within the framework to allow the validation of the network).

Finally there is the Dlib-based module: it represents the only module that will run online and that will be responsible for interfacing with the other aircraft systems on which it is supposed to reason. Therefore it is located in the Data Acquisition Processing Unit (or DAPU) of the aircraft and adds capabilities to its Central Maintenance System (or CMS). As remarked in a previous section, this module still shows lots of margins for improvement. Currently, a C++ implementation of the ACE approach is under investigation: this will involve an additional module which will accept as input the *net* file and provide as output a precompiled version of the network. Therefore, instead of the *net* file, the precompiled network will be loaded in the DAPU and used for performing on-line inferring, gaining this way computational time and approaching more the real-time requirement. The modular architecture of the HLR framework is depicted in Figure 2.

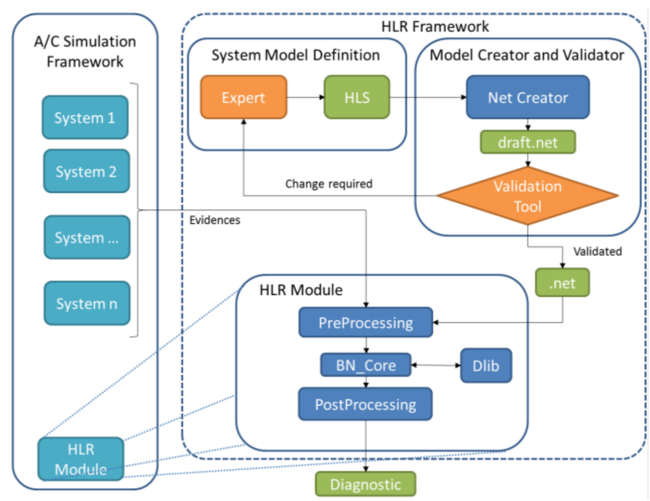

Figure 2. HLR Modular Architecture

This last section has provided the readers with an overview of the main actors playing a role within the developed HLR framework and the corresponding mutual relationships. The following section will help locating the described framework within a bigger system (an aircraft simulation).

3.5. The Simulation Framework

Airbus Defence and Space has been focusing along the years on the realization of a comprehensive simulation framework to be used as research and virtual V&V platform in the area of PHM: the framework includes airborne functions, hosted on embedded systems, as well as ground-based functions, hosted on PC-based systems. The primary objective is to interconnect both airborne and ground-based systems using a uniform data management philosophy and, as far as possible, uniform communication protocols. In this context, data management includes the entire data set life cycle: from initial instantiation of a sensor value, transportation to the data processor, downloading to the ground-based environment, to final storage and further processing. In recent technical papers, experiences from the task to define and implement the data management backbone for such a simulation framework (Löhr, Haines and Buderath, 2012 and Löhr and Buderath, 2014) have been reported. In order to have additional information on the actual implementation, the interested reader could refer to the mentioned papers, in which experiences while implementing a data management backbone based on OSA-CBM and OSA-EAI (Open System Architecture for Enterprise Application Integration) for a simulation environment supporting PHM systems in the aerospace domain are reported.

The simulation environment consists of an air segment and a ground segment, connected by the OSA-CBM and OSA-EAI based data management backbone. Leaving apart the ground segment, that would not be of interest for the current discussion, the air segment of the simulation framework models those systems and associated sensors for which ISHM (Integrated System Health Management) capabilities are intended to be developed. At the core of the framework is a central ISHM Data Processor (Figure 3) unit: sensors push their data to the ISHM Data Processor" via an OSA-CBM compliant implementation.

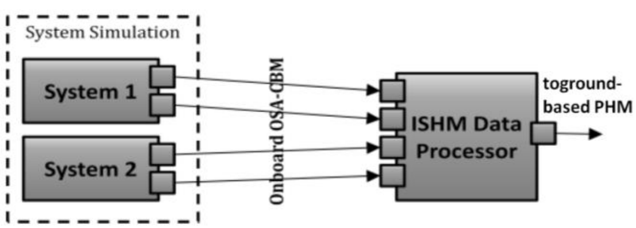

Figure 3. Air Segment of the Simulation Framework.

The ISHM Data Processor unit, that is located in the aircraft DAPU, is where the application responsible for the HLR on-line inferring is hosted: the environment is a real-time operating system (VxWorks). ISHM data instantiated by the corresponding aircraft systems and made available by the communication backbone will be then sent to specific inputs of the HLR inferring module.

General systems (e.g. hydraulic system, actuation system, fuel system, etc…) have been designed by means of Simulink and further deployed in the simulation framework, as C code: the C code has been automatically generated out of the Simulink models by means of the combined use of the Simulink Coder/Embedded Coder applications. An additional module to the simulation framework provides the capability of injecting selected failures into the system: the module in object has been developed in close cooperation with the system engineers in charge of issuing the FMECA/FTA documents for the system under consideration.

In the next section, a test-case for demonstrating the capabilities of the HLR framework will be presented and the corresponding results displayed and commented.

4. THE SELECTED TEST CASE

Aerospace systems usually share interfaces among each other: typical examples could be the Electrical Power Generation and Distribution System (EPGDS), sharing interfaces with all equipment that is connected to it, and the fuel system, sharing interfaces with the propulsion system. This makes the diagnostic of each system on its own more complicated and therefore it is critical that the "health" information generated in one system can "flow" to other systems.

The usage of fuel as a heat sink for the hydraulic oil among others, by means of a Fuel Cooled Oil Cooler (FCOC) as a heat exchanger, represents a current state of the art in the aerospace industry. This particular situation has been selected as the test case to demonstrate the capabilities of the proposed HLR framework, due to the following reasons: 1) it consists of relevant general systems found in many A/C, and 2) the interface among systems is a real example of how a failure in one system can affect the operation of another one. Therefore a BN model for the fuel system (FS) and the hydraulic and actuation system (HYD) used in the simulation framework has been developed and its interfaces defined, so that the diagnostic information can flow from one system to the other.

4.1. Generation and Validation of the Model

The model was generated and further validated as described in section 3: it was structured as an Operational-Functional Tree (OFT). An OFT is a graph representation of the system behavior, in this case using BNs. It considers the operational and functional characteristics of each node to define the logic of the distribution of edges. Typically this is completely different from a physical tree, which just considers the physical relations or from a safety tree, which just considers the safety relations among components. A tree is understood in graph theory as an undirected graph in which any two nodes are connected by exactly one path. However, in complex systems, nodes are typically connected to many other nodes at the same time. Therefore, the OFT will be in reality a network, in order to allow multiple linkages of the nodes, and the word "tree" is in this context used due to project-related historical reasons.

Although the generation of the model is part of the HLR framework, it is not under the scope of this work to report about the details of the model definition for this particular test case.

An extract of the FS/HYD network is represented in the following Figure 4. A simplified version of the network is displayed also for taking into consideration the readability of the information, beyond also other applicable company directives with regards to sensitive information. The amount of kept information has been considered enough for the purpose of the present paper.

The following sections will focus on the capabilities of the HLR module to perform diagnostic, within the simulation framework, using the below displayed FS/HYD BN model.

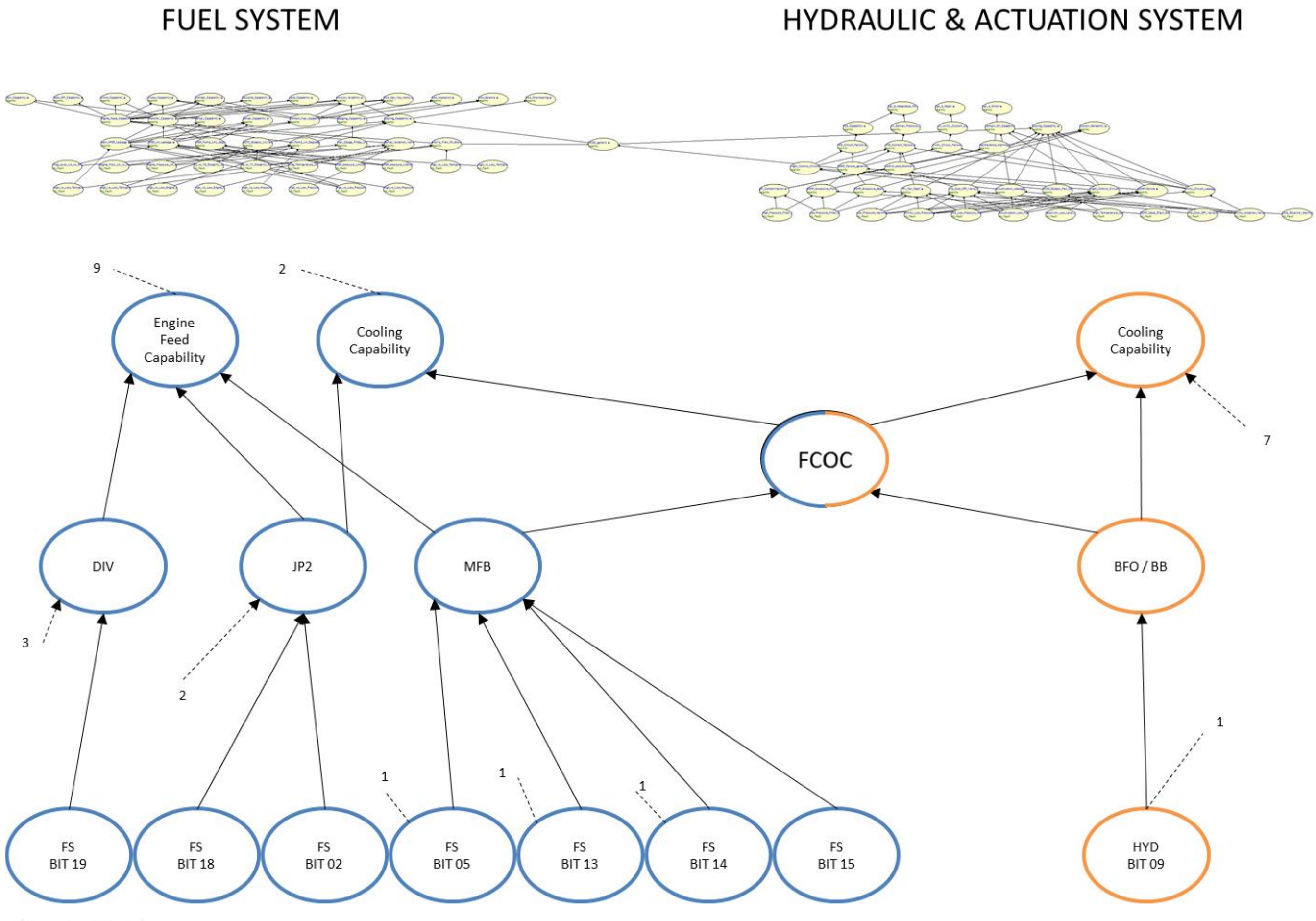

Figure 4. Extract of the FS-HYD/ACT network. Blue denote fuel system nodes, orange denote hydraulic and actuation system nodes and dotted lines denote edges not represented in this figure.

4.2. A coupled FS/HYD simulation

As previously described, the HLR on-line inferring module has been deployed into the ISHM Data Processor unit; this module together with the other modules responsible for the simulation of the FS and the HYD set up a virtual test bench by means of which the concept has been tested. The test bench has been equipped with two additional BIT modules belonging to the FS (FS_BIT) and to the HYD (HYD_BIT). The FS_BIT and HYD_BIT modules work as interfaces sending "evidences" to the HLR module. In this particular case, the data pushed to the HLR module within the ISHM Data Processor unit by the above mentioned two interfaces modules are all Boolean data. However, the flexibility of the Bayesian approach is such that the concept can in principle accept also other type of inputs, i.e. sensor data, enhanced health monitoring data, health grades of LRUs, etc. Depending on the input type and on the way in which the network's nodes are coded, the information should then be preprocessed as evidences triggering states or defining the percentage of a state of the root nodes.

This setup will prove that the on-line inferring HLR module is able to read the evidences of other modules inside the simulation framework and perform diagnostic in accordance with an Operational Functional Tree based on BNs.

4.3. Tests descriptions

The physical systems models have been coded with fault injection capabilities; as a matter of fact, a comprehensive set of injectable FS failure-modes has been set-up and used for the V&V activities on the current investigation. In order to test the performance of the HLR module, several test cases have been created; they are briefly summarized in Table 2. However, in agreement with what stated in section 4 related to the flowing of diagnostic information between systems and to the proper exploitation of such information, in what follows results related just to the injection into the coupled FS/HYD simulation of the test case coded as TC 08 will be presented.

Table 2. Test Cases description (look at the “Abbreviations table” for the corresponding meanings).

Test Case	Failure Mode Triggered
TC 01	Main XFER Leakage
TC 02	XFER LH Leakage
TC 03	XFER Pump LH #1 Failure
TC 04	ACFC Bypass LH Fail Open
TC 05	Jet Pump # 2 Failure
TC 06	Fuel Gauge Probe LH #1 Failure
TC 07	Defuel Isolation Valve Fail Open
TC 08	Motive Flow Blockage

Injecting such a failure within the FS will cause the firing of a subset of the monitored BIT values by the mentioned FS_BIT module, as showed in . No further information can be given regarding the meaning of the singular elements that are building up the pattern shown in the mentioned table. This Boolean information will be then pushed to the HLR Module, which will analyze the evidences and perform the inference.

The failure will be triggered on the FS side, as said, but the diagnostic task will be performed on the entire FS/HYD model, reaching operational level.

Before commenting in detail on the inference process of the HLR module, a few words will be spent in the next section regarding the scalability of the proposed diagnostic framework by defining for the term “scalability” in this context.

Table 3. BIT Definition. Depending on a “Limit Checking” approach and on the current values of the monitored physical parameters, some BITs are triggered (TC 08).

	TC 08 BIT Pattern
Disparity Checks	1
	2
	3
	4
	5
	6
	7
	8
Temperature Warnings	9
	10
	11
	12
	13
	14
	15
	16
	17
Engine	18
	19
	20
	21
XFER Pumps	22
	23

4.4. Initial Proof of scalability

Scalability is the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged in order to accommodate that growth. Two concepts are addressed in this work: 1) inference towards higher levels, 2) increase of complexity (nodes / edges / states).

The scalability of this concept for the inference towards higher levels, e.g. system level, operational level or fleet level, will be demonstrated by means of the test case under study for the FS and HYD (see section 4.5 for results).

The main advantage of the use of BNs for this concept is that they can be used at very different levels, as far as the logic of the model is properly translated into the BN structure and the CPT of each node is properly defined.

Regarding the scalability, as the complexity (nodes / edges / states) increases, it has to be highlighted that the used library and algorithms are not the one that are better performing, as already stated, and that a benchmarking that compares different software products is needed and already scheduled. Where Dlib takes several seconds to infer, other tools perform the same inference in a fraction of a second.

It is believed that the reason for such a time difference lies on how the inference is performed. In the case of Dlib, the processing of new evidences means a new complete inference. In the case of other tools, new evidences produce a propagation of such information through the network, resulting in less computational time.

Some research has also been performed with ACEs and other algorithms like Recursive Conditioning which promises a much better performance. Chavira and Darwiche (2005, 2007) demonstrated that a considerable performance improvement can be achieved with ACEs and Variable Elimination algorithms.

Other solution would be the parallelizing of the inference which also improves greatly the inferring performance, as Namasivayam, Pathak and Prasanna (2006) discuss in their work.

However, some tests have been performed by means of the utilized Dlib library in order to understand the scalability limitations of such approach. This is a preliminary benchmarking of the computational resources that will be later on performed.

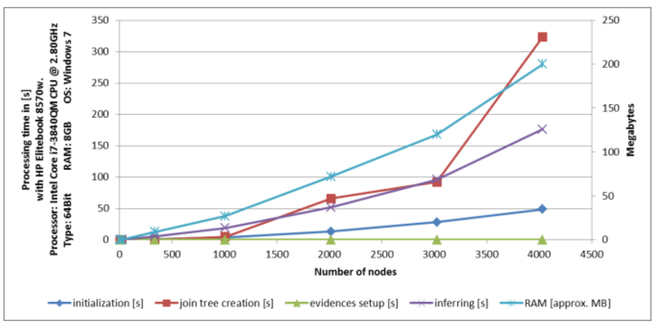

Figure 5. Example with binary nodes, using Dlib to perform inference.

Figure 5 shows that there will always be a limit in the size of the network (or complexity), due to the fact that the exact Bayesian inference is a NP-hard problem. If an A/C network is created by connecting different system trees, the number of nodes would raise easily to thousands.

Taking this into account plus the huge potential for improvement, e.g. the implementation of other algorithms and inferring approach (like ACE) and parallelizing, the diagnostic task for aircraft systems using BNs in a reasonable time can be achieved, if certain guidelines are followed:

- Minimum number of nodes,
- Minimum number of states,
- Minimum number of edges,
- Separate trees, when possible.

In other words, the network shall be as simple as possible without detriment to the inferring capabilities.

4.5. Results and comments

The initial state represented in Figure 4 is perturbed by the clamping of a set of evidences in accordance with the BIT pattern displayed in , as a consequence of the injection of the failure whose code is: TC 08 (see). The final state is analyzed below and the network is depicted in Figure 6.

This triggers on the other hand the failure of the node MFB which represents a “Motive Flow Blockage” in the FS. As a consequence, that also triggers the “cooling capability” failure and the node “FCOC”. Other capabilities are also affected; however they are not important to analyze the influence on the HYD/ACT system.

The node “FCOC” represents the interface between both systems, and due to its state’s change, the cooling capability in the HYD/ACT system is also lost. It is relevant to highlight that these two systems are modelled separately (two different HLSs definitions and therefore also two different system engineers’ teams have worked on it) and merged with the FCOC interface with the Model Creator and Validator.

This test case demonstrates that the transfer of information among systems is possible, enabling therefore a cross-system diagnostic, and diagnostic at system-level.

The test case shows high probabilities for several failure modes (DIV, JP2). This is due to several failure modes sharing the same evidences. In order to get a more accurate diagnostic, the CPTs should use also the evidences that allow this difference. In the end, the quality of the diagnostic relies directly on the quality of the CPTs.

In case a failure mode is properly modelled, it will show a high probability in comparison with the others, see Figure 7, considering of course the assumption that the failure modes are not happening together. Otherwise the probabilities would be similar, meaning that further differentiation nodes are necessary.

The System-Level diagnostic has also been tested with satisfactory results, although the inference time has to be improved with the use of other algorithm.

Finally, these results show that the HLR framework can be used to perform cross-system diagnostic, reaching operational level, and that the HLR module can be run within the simulation framework together with other already developed system models. What was discussed and presented so far represents a very promising result for the future stages of this investigation. In particular our next work will focus on the improvement of performance of the HLR Framework, that is required to reach real-time diagnosis, and on the certification aspects of this new technology.

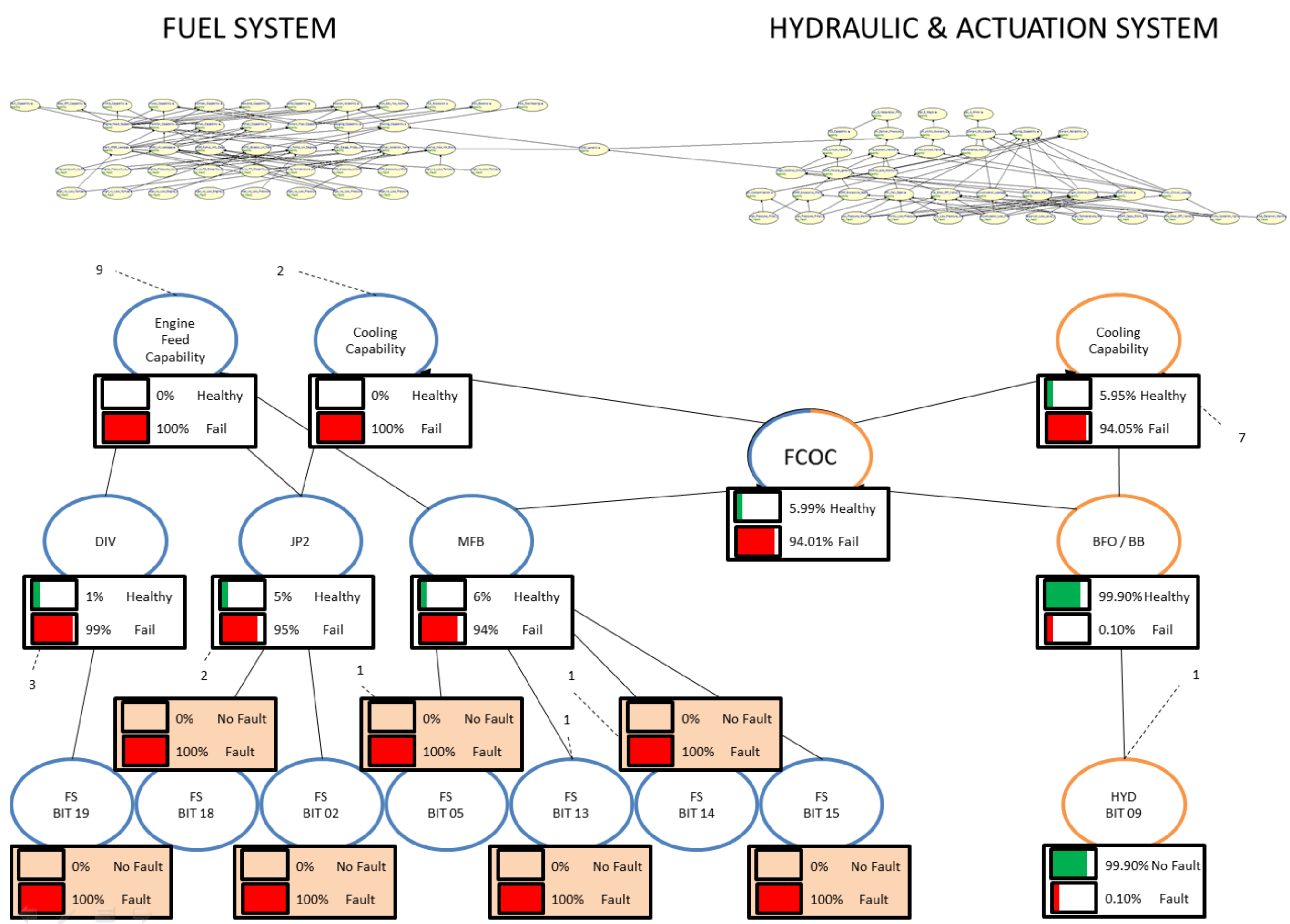

Figure 6. TC 08 Final State. Orange shadow represents evidences.

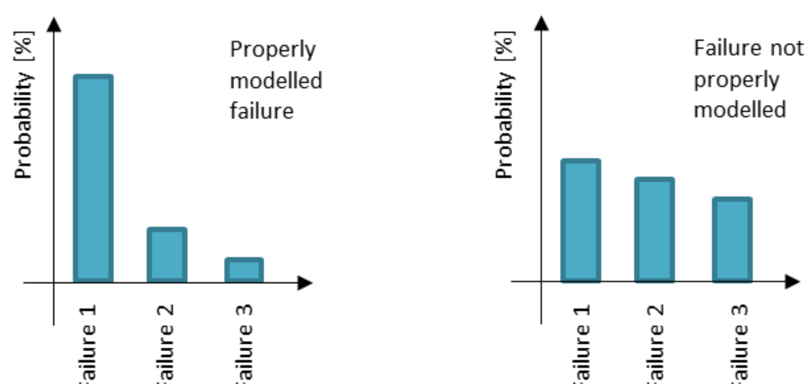

Figure 7. Examples for properly modelled and not properly modelled failures.

5. Conclusions and Way Ahead

In this paper, the initial results related to the usage of BNs to improve aircraft diagnostics are reported.

A BN represents a graphical model of a system, in which different players are linked to each other through probabilistic and causal relations. This approach could present several advantages over other data analysis methods: 1) it can provide outputs even if some entries to the model are missing; 2) it represents an ideal environment to include prior knowledge during the building up of the model; 3) it gives the possibility to learn causal relationships and therefore the possibility to improve the domain knowledge.

Given the above reasons, Airbus Defence and Space has been working on this topic with the main objectives of 1) increasing the aircraft diagnostics capabilities at component, sub-system and system level, and 2) providing means for reasoning about the remaining operational and functional capabilities of the aircraft.

The initial outcomes have been tested on a simulation platform featuring a Data Acquisition Processing Unit, various computing nodes, on which the different aircraft systems (like the fuel system, the hydraulic system, the actuation systems, etc…) run. The data communication architecture of the platform is based on OSA-CBM.

The major gained achievements have been: 1) the demonstration of the feasibility of the integration of the concept within the relevant environment represented by the simulation framework; 2) the development of means in order to allow an easy and structured translation of the system engineer knowledge in terms of a BN with associated conditional probabilities; 3) the provision of a modular architecture for the concept, in order to facilitate the effective coordination between different development-departments, as well as the efficient development and maintenance of the software and 4) the proof of the scalability of the concepts (i.e. applicability to systems of different sizes and reasoning on different levels from component to system level).

The investigations conducted have particularly regarded the fuel and the hydraulic/actuation systems: these systems, characterized by fault insertion capabilities, have been designed by means of Simulink and further deployed in the simulation framework, as C code. The corresponding designed Bayesian models - for reasoning about the systems under investigation - have been hosted into the ISHM Data Processor unit, located in the aircraft DAPU. A proper set of test-cases has been conceived and utilized to fully test the performance of the HLR framework, in terms of fault isolation and detection capability and ability to reason on the remaining operational and functional capabilities. The results related to a particular test-case have been here reported: in it, a failure injected into a system (the fuel system) has caused the trigger of a set of Boolean BIT. This set, used as evidence within the inferring logic of the HLR module, has allowed – on the other hand - assessing the reduced cooling capability of the hydraulic/actuation system. This has then proved that the transfer of information among systems is possible enabling a cross-system diagnostic, and diagnostic at system-level.

Obtained results are considered promising and Airbus Defence and Space will continue investing resources on this topic, in order to improve the performances of the so far developed HLR Framework, considering also the important certification aspects linked to this technology.

Acknowledgement

The authors are very thankful to Mr. Madhuraj Harisankaran (Airbus India, Bangalore) for stimulating discussions with regards to the architecture of the High Level Reasoning Framework and the definition of the High Level Specifications.

References

Barua, A. & Khorasani, K. (2009). *Hierarchical Fault Diagnosis in Satellites Formation Flight.* Annual Conference of the Prognostics and Health Management Society.

Canaday, H. (2016). *Predicting Failures – Using Big Data to solve the most expensive problems.* Aviation Week & Space Technology, January 4-17, 2016, p137.

Cerquides, J. & López de Màntaras, R. (2003). *Tractable Bayesian Learning of Tree Augmented Naïve Bayes Classifiers.* TR-2003-04 CSIC.

Chavira, M. & Darwiche, A. (2005). *Compiling Bayesian Networks with Local Structure.* Mark Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-2005).

Chavira, M. & Darwiche, A. (2007). *Compiling Bayesian Networks Using Variable Elimination.* Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007).

Darwiche, A. Y. (1993). *Argument Calculus and Networks.* Uncertainty in Artificial Intelligence: Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, The Catholic University of America, Washington, D.C., p420-427.

Das, B. (2008). *Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem.* Cornell University.

Davis, R. & Hamscher, W.C. (1988). *Model-Based Reasoning: Troubleshooting.* A.I. Memo No. 1059, Massachusetts Institute of Technology.

Davis, R. & King, J.J. (1984). *The Origin of Rule-Based Systems in AI,* reprinted as Ch. 2 of Rule-Based Expert

Systems: Edited by Bruce G. Buchanan and Edward H. Shortliffe, Addison Wesley Publishing Company.

François, O.C.H. & Leray, P. (2006). *Learning the Tree Augmented Naive Bayes Classifier from incomplete datasets.* Third European Workshop on Probabilistic Graphical Models, Prague, Czech Republic.

Guo, H. & Hsu, W. (2002). A Survey of Algorithms for Real-Time Bayesian Network Inference. AAAI Technical Report WS-02-15.

Jong, C.G. & Leu, S.S. (2013). *Bayesian-Network-Based Hydro-Power Fault Diagnosis System Development by Fault Tree Transformation.* Journal of Marine Science and Technology, Vol. 21, No. 4, pp. 367-379 367. doi: 10.6119/JMST-012-0508-3.

King, D.E. (2009). Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758. Dlib C++ library v18.16. http://dlib.net/

Kokkonen, T., Koivusalo, H., Laine, H., Jolma, A. & Varis, O. (2005). *A method for defining conditional probability tables with link strength parameters for a Bayesian network*. MODSIM 2005 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, pp. 428-434. ISBN: 0-9758400-2-9.

Kolodner, J. (2014). *Case-Based Reasoning*. Morgan Kaufmann.

Löhr A. & Buderath, M. (2014). *Evolving the Data Management Backbone: Binary OSA-CBM and Code Generation for OSA-EAI*, European PHM Conference Nantes, France

Löhr A., Haines, C. & Buderath, M. (2012). *Data Management Backbone for Embedded and PC-based Systems Using OSA-CBM and OSA-EAI*, European PHM Conference Dresden, Germany

Mack, D.L.C., Biswas, G., Koutsoukos, X.D. & Mylaraswamy, D. (2011). *Using Tree Augmented Naive Bayes Classifiers to Improve Engine Fault Models.* Uncertainty in Artificial Intelligence: Bayesian Modeling Applications Workshop.

Mack, D.L.C., Biswas, G., Koutsoukos, X.D., Mylaraswamy, D. & Hadden, G. (2011). *Deriving Bayesian Classifiers from Flight Data to Enhance Aircraft Diagnosis Models.* Annual Conference of the Prognostics and Health Management Society.

Mengshoel, O.J., Chavira, M., Cascio, K., Poll, S., Darwiche, A. & Uckun, S. (2008). *Efficient Probabilistic Diagnostics for Electrical Power Systems*, NASA/TM-2008-214589 NASA Ames Research Center.

Namasivayam, V.K., Pathak, A. & Prasanna, V.K. (2006). *Parallelizing Exact Inference in Bayesian Networks.* 10th Annual Workshop, HPEC 2006 Proceedings

Ricks, B.W. & Mengshoel, O.J. (2009). *Methods for Probabilistic Fault Diagnosis: An Electrical Power System Case Study.* Annual Conference of the Prognostics and Health Management Society.

Russell, S.J. & Norvig. P. (1995). *Artificial Intelligence – A Modern Approach.* Prentice-Hall Inc.

Schwabacher, M.A. (2005). *A Survey of Data-Driven Prognostics.* AIAA 2005-7002.

UCLA Automated Reasoning Group. 2002. Samiam: Sensitivity analysis, modeling, inference and more. SamIam 3.0, http://reasoning.cs.ucla.edu/samiam

BIOGRAPHIES

Borja Sanz López was born in Madrid, Spain, in 1987. He received his Aerospace Engineer degree from the Universidad Politécnica de Madrid, Spain in 2012. During 2011 he performed a research on applications of neural networks on aerodynamics at Airbus Operations S.L. in Madrid, Spain. In 2012 he collaborated on a research on numerical simulation of boundary layer interaction of crossflow jets at Institut für Aerodynamic und Gasdynamic in the University of Stuttgart, Germany. Currently, he works as Fuel System Engineer at the Airbus Defence & Space "Fuel, Inerting and Fire Protection Systems" Department, and he supports the Integrated System Health Management project performing research on High Level Reasoning Diagnostics.

Antonino Marco Siddiolo was born in Agrigento, Italy, in 1976. He received his M.S. and Ph.D. degrees in mechanical engineering from the University of Palermo, Italy, in 2000 and 2006, respectively. From 2004 to 2005 he was a Visiting Scholar at the Centre for Imaging Research and Advanced Materials Characterization, Department of Physics, University of Windsor, Ontario (Canada). Then, he worked as a researcher and Professor at the University of Palermo and as a Mechatronic Engineer for Sintesi SpA, Modugno (Bari), Italy. His research activities and publications mainly concern non-contact optical three-dimensional measurements of objects and non-destructive ultrasonic evaluation of art-works. His main contributions are in the field of signal processing to decode fringe patterns and enhance the contrast of air-coupled ultrasonic images. Currently, he works as Health Management System Engineer at the Airbus Defence & Space "Fuel, Inerting and Fire Protection Systems" Department, supporting the Integrated System Health Management project. His research activities are mainly focusing on the maturation of failure detection and prediction capabilities for electrical, mechanical and hydraulic aircraft equipment.

Partha Pratim Adhikari has more than 17 years of experience in the field of IVHM, Simulation of Aircraft Systems and Avionics. Partha has Bachelor's degrees in Physics (H) and B. Tech in Opto-electronics from Calcutta University and a Master's degree in Computer Science from Bengal Engineering and Science University. In his tenure across various aerospace organizations, Partha made significant contributions in the fields of IVHM, Navigation systems, Avionics and Simulation technologies. Partha published several papers in the fields of estimation, signal processing and IVHM in national as well as international conferences and journals. Partha, in his current role at Airbus Group India, Bangalore is working on devising ISHM technologies for aviation systems with focus on complete vehicle health, robust implementation and certification of the developed technologies.

Matthias Buderath - Aeronautical Engineer with more than 25 years of experience in structural design, system engineering and product- and service support. Main expertise and competence is related to system integrity management, service solution architecture and integrated system health monitoring and management. Today he is head of technology development in Airbus Defence & Space. He is member of international Working Groups covering Through Life Cycle Management, Integrated System Health Management and Structural Health Management. He has published more than 50 papers in the field of Structural Health Management, Integrated Health Monitoring and Management, Structural Integrity Programme Management and Maintenance and Fleet Information Management Systems.