17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 442

Introducing Operational Diagnosis Models for Ground Station Architectures using Behaviour Trees

Nikolena Christofi[a]*, Xavier Pucel[b], Claude Baron[c], Marc Pantel[d], David Canu[e], Jerome Golenzer[e], Christophe Ducamp [e]

[a] *Nikolena Christofi, IRT Saint Exupéry, LAAS-CNRS, INSA Toulouse, Airbus Defence and Space, University of Toulouse, B612 building, 3 Rue Tarfaya, CS 34436, 31405 Toulouse Cedex 4, France,* nikolena.christofi@laas.fr
[b] *Department of Information Processing and Systems, ONERA, Artificial and Natural Intelligence Toulouse Institute (ANITI), University of Toulouse, 2 Avenue Edouard Belin, 31055 Toulouse Cedex 4, France,* xavier.pucel@onera.fr
[c] *Electrical Engineering Department, INSA Toulouse, Systems Engineering and Integration Team, LAAS-CNRS, University of Toulouse, 7 Avenue du Colonel Roche, BP 54200, 31031 Toulouse Cedex 4, France* claude.baron@laas.fr
[d] *IRIT, Toulouse INP, ENSEEIHT, Université de Toulouse, 2 rue Charles Camichel – BP 7122, 31071, Toulouse Cedex 7, France,* marc.pantel@toulouse-inp.fr
[e] *Airbus Defence and Space, 31 Rue des Cosmonautes, 31400 Toulouse, France*
* Corresponding Author

Abstract

Although the use of models for the design of complex systems is nowadays prevalent, that is often not the case for their operations and maintenance. Complex systems are characterised by high criticality levels as regards to time and safety constraints. The lack of appropriate models for the monitoring of these systems, or the large volume and complexity of the existing ones e.g. system simulators, impose the need for the means and tools provision to the operators, to help them with their diagnostic tasks. The use of formal models for system monitoring and diagnostics can greatly augment the operators' comprehension of the system. To that end, we propose a methodology to create a new type of Operations-Dedicated Model from already existing system design models (functional and dysfunctional), using the Behaviour Trees formalism. With the assumption that Safety Analysis models describe dysfunctional aspects of the system – notably via Fault Tree Analyses, we us Fault Trees as a direct input for the ODM construction. We demonstrate our proposed approach with an Earth Observation Satellite Ground Station example, and discuss how ODMs can improve systems' operations.
Keywords: fault trees, behaviour trees, operational diagnosis, model-based systems engineering, model-based safety analysis, ground systems engineering

Acronyms/Abbreviations

Behaviour Tree (BT), Fault Detection and Diagnosis (FDD), Fault Detection, Isolation and Recovery (FDIR), Feared Event (FE), Finite State Machine (FSM), Fault Tree (FT), Fault Tree Analysis (FTA), Ground Station (GS), Model-Based Safety Assessment (MBSA), Model-Based Systems Engineering (MBSE), Operations-Dedicated Models (ODM), Telemetry (TM), Telemetry Image (TMI).

1. Introduction

Spacecraft operations are complex processes, requiring different levels of expertise by the operators in charge of the system monitoring, in combination with advanced embedded Fault Detection, Isolation and Recovery (FDIR) modules [1]. Major key operational elements include availability, autonomy, robustness and trustworthiness [2,3,4]. A fault impeding access to satellite data can cause a major loss of time – hence money, and a risk for the satellite itself. Consequently, the improvement of FDIR techniques on-board the satellite (satellite autonomy), and on the ground (*diagnosis* by operators) has been a constant challenge, since the beginning of the space age.

Operators' *troubleshooting* tasks vary depending on the health status of the system at each given moment, as indicated by the registered data. Per contra, if a symptom – or a combination of symptoms, is unknown, i.e. system designers did not model the symptom or foresee the specific failure occurrence, operators are expected to perform troubleshooting, in order to eliminate the error, and restore functions to their nominal state (or the least degraded possible state). Troubleshooting consists of the tasks of requesting for the system for additional information or/and performing manual tests/manual investigation, etc. The operators' priority is to avoid losing the services the system is expected to provide. Thus, unless the anomaly has an associated troubleshooting procedure e.g. in a symptoms'

database, the operators must take individual action i.e. carrying out a dedicated analysis and updating the system's knowledge database.

These actions are usually based on the operators' knowledge and experience. If this knowledge were to be supported by organized documentation of the system itself (architectural and behavioural, as well as functional and dysfunctional), *operational diagnosis* could be improved significantly. For this reason, we envisaged the creation of an Operations-Dedicated Model (ODM) to address Operations and Maintenance system tasks, in particular Fault Detection and Diagnosis (FDD).

As shown in Figure 1, system architecture and behaviour description models – created through Model-Based Systems Engineering (MBSE) activities, along with system dysfunctional behaviour models – produced during Model-Based Safety Analysis (MBSA) activities, are built in the beginning of the *system lifecycle*, during the system design phase. MBSE and MBSA models are used beyond the system design, particularly in the development and verification phases – in particular for traceability reasons. The first serves in meeting the requirements, and the latter in respecting safety objectives while complying with international standards and regulations, which are defined during the system design phase – mostly the case for the aeronautics domain, not applicable in the aerospace sector (except for satellite end-of-life management). As illustrated in the schema, the activities related to system design (prior to system launch), as much as operations and maintenance (post-system deployment), are, although related and interdependent, disconnected and detached.

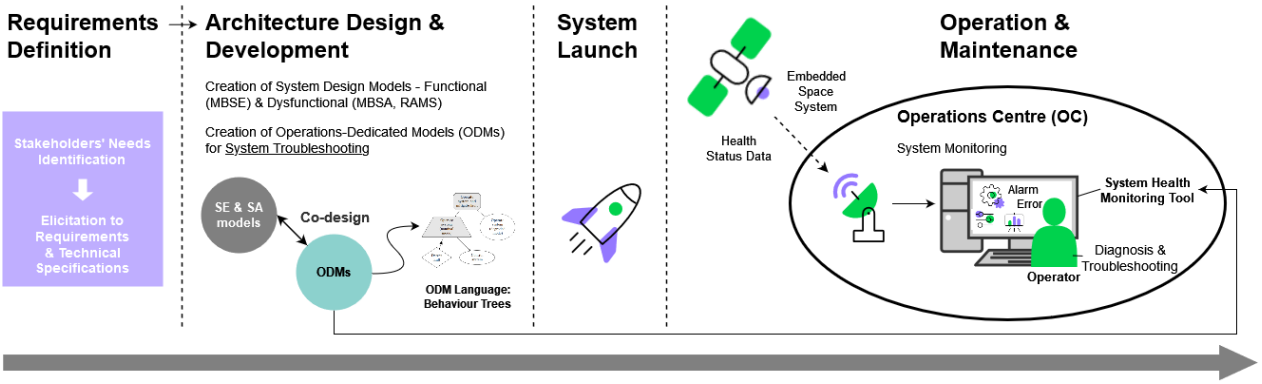

Figure 1 : ODM Approach Overview: created during design, and exploited in operations.

The ODM shall help the operators perform their diagnostics tasks more efficiently by *reducing their response time to failure*, but also facilitate access to documentation and dedicated procedures so as to optimise their troubleshooting actions. Hence, the ODM shall have the capability not only to provide the current system overview, but also to be exploited by a diagnosis tool. This way it shall provide possible fault candidates, in the case of a raised alarm, or erroneous received data.

Hence, the operators shall be able to exploit this new type of model, dedicated to system monitoring, during operations. One way to exploit the ODM is to drive a diagnosis tool, with which the operators could interact, through a dedicated Graphical User Interface (GUI). Moreover, the ODM can be built during the design phase and along the production of other design models and documents, in a **co-design** manner i.e. through a digital continuity between design, operational safety and ODM construction. Since our main research interest is aerospace, we illustrate the proposed approach with a space system use case; however, it can be extended to other domains, such as airplanes, drones, etc.

2. Methodological proposal for building ODMs

The ODM methodological approach consists in defining a model that describes the system's operational procedures, with particular *emphasis on fault diagnosis activities*. In this section we provide a step-by-step description of the proposed methodology. Moreover, we use an extension of the Behaviour Tree (BT) language, which we created for the needs of the method, with the aim to provide the language with the tools necessary to express ODMs using Fault Trees (FTs) as input artifacts, as well as to enrich the expressivity of its semantics – especially as regards to fault detection, avoidance and mitigation.

As illustrated in Figure 2, ODMs serve a double purpose. On the one hand, since they are **concurrently created** with the Systems Engineering (SE) & Safety Assessment (SA)/Reliability, Availability and Safety (RAMS) models, they can provide feedback to the designers, specific to the system's health monitoring and FDIR aspects. This allows the improvement of the system's structural and behavioural design. The left side of Figure 2 portrays these interactions, between system architects, safety experts and the ODM design team. The latter interacts with the operators as illustrated on the schema's right side.

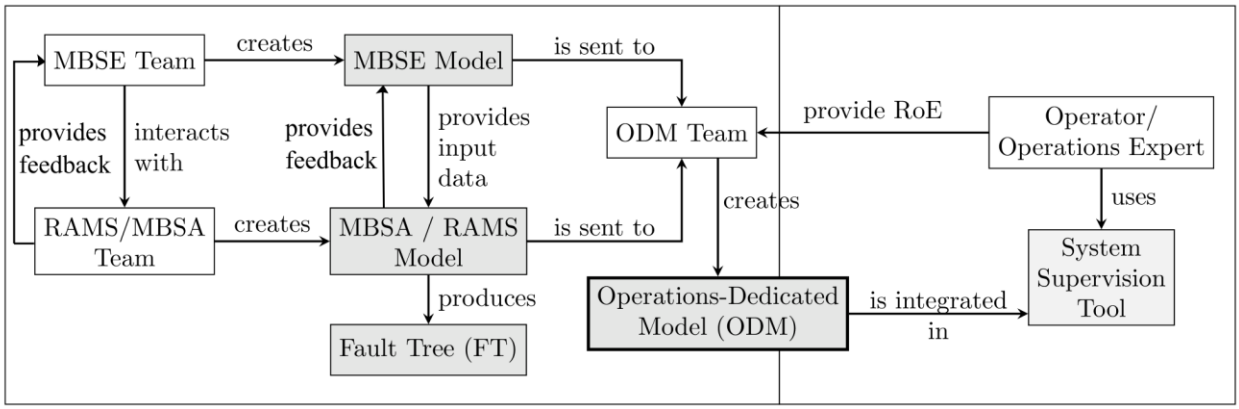

Figure 2 : Overview of the proposed methodology for the construction and exploitation of an ODM

On the other hand, following system deployment, the ODM can be used for system supervision and diagnosis. BTs are executable; hence, the ODM can facilitate the design of test scenarios, distribute diagnosis objectives across the various activities, and ensure that operators are given realistic tasks. This process is also illustrated at the right side of Figure 2. Note that the scope of this study solely tackles the ODM creation phase.

By not requiring particularly sophisticated skills prior to the initial training – other than system modelling and knowledge of operations, we can ensure that the monitoring model is easy to maintain throughout the years. In the following section we illustrate our proposed methodology through its application in a use case scenario.

As mentioned above, our methodology uses FTs as input. A FT addresses a single Feared Event (FE), and investigates its possible causes. The root of the tree is the FE itself. Each tree node is an event that contributes to its parent event. At the bottom are basic events, that are combined through either AND-events (occur when all their children have occurred) or OR-events (occur when any one of their children have occurred). FTs are produced during SA activities.

The *output* of our methodology is a *BT that describes the diagnosis activities* that take place in the system. It aims at covering **all diagnosis activities**, both *automated* and *manual*, both *on-board* and *on-ground*. The question of allocating each diagnosis activity to an FDIR module or to an operator should be addressed in an ulterior step, and will be tackled in future work.

A BT is a tree that represents a concurrent process. Each node of the BT represents a process, and the tree root is the global process represented by the BT. At each time instant, an active BT node can be in three states: RUNNING, SUCCESS or FAILED. The leaf nodes of a BT represent the elementary activities that take place in the process. The BT combines these activities using composite nodes to create more elaborate processes. Several composite node types exist, as listed below. For a complete description of BT nodes and their semantics, we refer to [5,6].

- The Fallback node provides mechanisms to recover from failures of its children;
- The Sequence node waits for its childrens' success to call the next child;
- The Parallel node launches all its children in parallel. There are two important types of Parallel nodes: the ParallelAll succeeds if and only if all its children have succeeded; the ParallelAny node succeeds as soon as one child has succeeded, and it interrupts the other children in this case.

Our methodology consists of a two-step transformation process, from a FT into a BT:

1. The first step is automated, and derives the diagnosis objectives from the FT. Each fault event of the FT is transformed into a diagnosis objective. Diagnosis objectives are organized in a BT with the same structure as the original FT. The interest of this step is to take into account all the fault events that were considered during the FT analysis by the system safety analysts.

2. The second step is manual, and consists in eliciting the diagnosis objectives into a BT that represents the diagnosis process. In this step, the ODM designers use information from SE models, but in a manner that is specific to the system and the objectives, which makes this step impossible to automate. In particular, the aim of this transformation is to account for fault tolerance and robust control mechanisms, fault mitigation activities, the fact that some activities occur in a predetermined sequence or the fact that some faults may have different observable effects depending on the system configuration.

One important aspect of our methodology is that every transformation is documented and justified. This guarantees that every fault event considered in the FT is either directly accounted for in the BT, or handled by one or several precisely identified behaviours. Regarding the expertise required for the building of the BT monitoring models, we propose that a dedicated team with operations background shall build the ODMs rather than the system architects or the safety analysts, so as to allow a distinct point of view. That would also ensure that the models would contain the necessary information *for supervision and diagnosis only*. Moreover, the construction of BTs with the PyTrees library is relatively easy and intuitive to code with. We hence believe that most system-modelling engineers with no specific coding background, provided with *sufficient training, guidance and documentation,* could create ODMs.

3. Case Study Illustration

In this section we illustrate our methodological approach proposal with the help of a satellite Ground Station (GS) use case. This Satellite Telemetry Image (TMI) receiving station includes the computing and control part of the GS.

Using the received plan information, the station's tasks include the following:

- acquisition/tracking phase initialisation;
- satellite tracking and data acquisition launching;
- metrics and log files generation, which reflect the TMI acquisition status;
- GS Antenna physical piloting, before and after the acquisition phase.

The GS comprises of its own Scheduler and Antenna, in order to manage all the programmed passes, and to communicate with the Satellite in order to receive the TMI. The information regarding the possible faults that can occur within the GS system was provided through a faults' propagation mind map from GS operators, which we reduced and modified for the needs of this illustration study. The FE of this faults' map is the loss of the Satellite Telemetry (TM) data.

In essence, the loss of TM data can be related to four main causes:

1. Data reception failure;
2. Image files processing failure;
3. Site power distribution failure;
4. Wrong activation of the emergency shutdown feature.

Each main failure of the GS can be traced back to one or several failure modes, through fault propagation sequences. Note that, although the notion of hierarchy between the faults' occurrence does exist, the analysis does not take into account the order in which each fault occurs and how that could impact differently the system as a whole. This means that the hierarchical relation is not the same as the classical FT parent/child relation. In an FT, a child "contributes to" its parent, and occurs before. Here, it is more that a child "is a more specific diagnosis" than its parent.

The reason is that this analysis is based on feedback from the operators, describing the faults often encountered while operating the GS, and not on formal safety analysis calculations. Moreover, the analysis does not take into account combinatory faults' induced failures. This insinuates that the occurrence of a sole fault leads to another fault, but never a combination of two or more faults at the same time (simultaneously or with time difference). This will reflect in the respective FT and BT.

We focus our study on the data receiving part of the GS, which includes the following three main subsystems:

- Base-band subsystem,
- Radio-Frequency (RF) subsystem,
- Antenna subsystem.

We used the information regarding the GS' functional and dysfunctional architecture provided by operators to build a SysML Activity Diagram. This diagram illustrates the GS' sequence of functions, from the moment the station receives the pass planning forecast for the day, until it successfully transmits the TMI it received from the passing Satellite. This process is divided into several operational phases, hence defining its overall Concept of Operations (ConOPS). The activity diagram depicting the functional scenario tackled in our case study is shown in Figure 3.

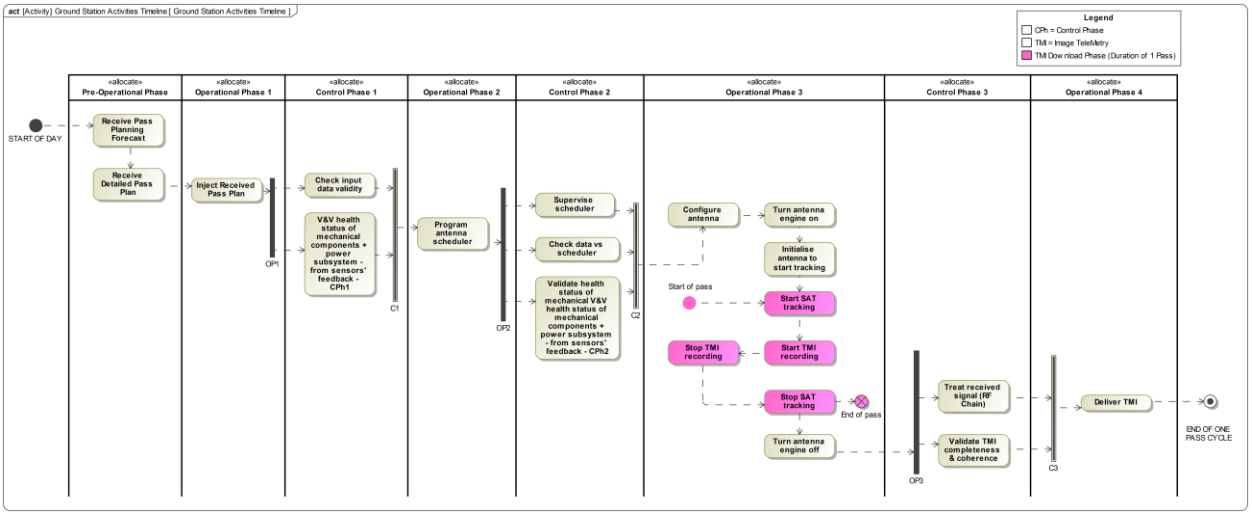

Figure 3 : SysML Activity Diagram representing the operational scenario (sequence of operational activities) of an image telemetry data receiving Ground Station within one satellite passage (single satellite pass cycle). We tackle this operational scenario in our case study.

Then, the FT of our case study targets the FE of losing the Satellite TMI data due to a malfunction of the Antenna pointing subsystem. For the needs of the study we make the assumption that the Antenna subsystem is isolated from the rest of the GS system. During a pass, there are a limited number of ways to mitigate faults due to the strict time constraints. To mitigate faults in the memory storage system, a redundant storage is present. Other failures, in the tracking of RF functions, lead to the loss of the pass data because there is no way to recover from the fault fast enough. Therefore, failures on the RF link and the tracking function cause the loss of the pass data. However, in order to mitigate this risk, equipment Verification & Validation (V&V) procedures are performed before the pass, to ensure the system is capable of receiving the Satellite TMI data. The FT that describes the possible causes for the loss of the Satellite TMI data is depicted in Figure 4, and described below.

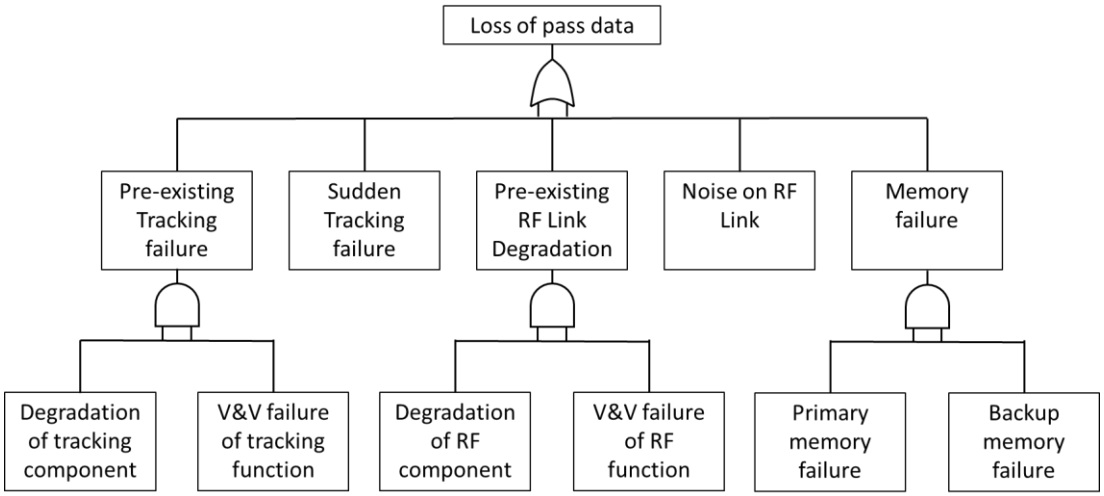

Figure 4 : Fault Tree for the Ground Station use case regarding the loss of Satellite TMI data.

The loss of pass data is the FE of the FT. It is an OR node, meaning that the occurrence of any of its children causes its occurrence. Its children include the tracking function failures, RF failures, and memory failures. In particular, the pre-existing tracking failure, pre-existing RF link degradation, and memory failures, are AND nodes. This means that all of their children must happen for them to happen as well. The other nodes are basic events, and happen randomly with different probabilities. For example, the probability of a sudden tracking failure occurring during a pass is very low, while the probability of a failure having happened in the past may be higher. However, the probability of this pre-existing failure being undetected by V&V activities is also very low.

The first version of the BT, automatically derived from the FT is depicted below. Each node of the FT, representing a fault event, is transformed into an equivalent operational objective. For example, the node "Noise on RF link" is transformed into the behaviour "Avoid noise on RF link". OR-nodes, namely the FE "Loss of pass data", are transformed into ParallelAll nodes, since, to avoid an OR-event, it is necessary to avoid all its possible causes. Conversely, AND nodes are transformed into ParallelAny nodes, since to avoid an AND event, it is sufficient to avoid at least one of its possible causes. This transformation applies the **De Morgan law** in a straightforward recursive manner. BT-v1 is represented in Figure 5.

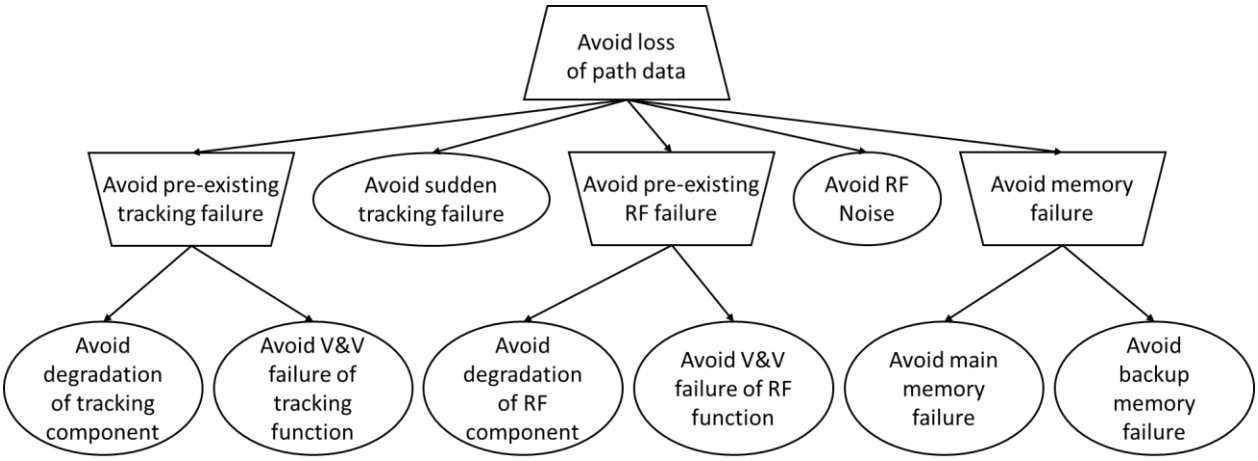

Figure 5 : First Behaviour Tree of the methodology, BT-v1. Result of a Fault Tree automated transform. Describes diagnosis objectives in a hierarchical manner.

The second version of the BT is obtained by manually eliciting the diagnosis objectives in the first version of the BT. For example, noise on the RF link is due to environmental perturbations, such as the weather, or poor equipment quality. However, there is nothing that the operators can do to avoid having excessive noise on the RF link. Thus, this objective is derived into a single operational activity "Operate RF", although it can also be derived into a robustness requirement for the RF function, which is out of scope for this paper. BT-v2 is represented in Figure 6.

The same consideration applies to each atomic behaviour of BT-v1. This is due to the fact that for readability purposes, we stopped the FT analysis at a depth that is appropriate for building our ODM. In practice, FTs are usually much deeper than required for building a BT. For example in a real FT analysis, the "Noise on RF Link" may be a fault event with hundreds of basic events underneath, that transforms into hundreds of diagnosis objectives in the first BT. However, from an operational point of view, all these fault events are undistinguishable, because their associated diagnosis objectives in the first BT are all derived into the same operational activities. Thus, the ODM designer can address the whole subtree at once by deriving the "Avoid RF Noise" objective directly.

The converse is also true: some events in the FT transform into complex diagnosis objectives that only an elaborate operational procedure can meet. This is the case for the events "Avoid pre-existing tracking failure", "Avoid pre-existing RF failure" and "Avoid memory failure". Let us focus on the first objective: "Avoid pre-existing tracking failure". It is composed of two sub-objectives "Avoid degradation of tracking component", and "Avoid V&V failure of tracking function", that are respectively derived into the operational activities "Track satellite" and "V&V tracking components", as indicated in the previous paragraph. While these two activities contribute to the same diagnosis objective "Avoid pre-existing tracking failure", they do not take place at the same time in the operations: the former takes place during the pass, while the latter occurs during the preparation for the pass. The existence of these two phases requires the use of a Sequence node as the root of the second BT, with the two children "Prepare GS for reception" and "Perform reception", each being the parent of its respective behaviour.

The same reasoning applies to the "Avoid pre-existing RF failure" that is also mitigated by a prior V&V of the RF link components. Consequently, the corresponding two activities "V&V RF components" and "Operate RF link" are children of respectively "Prepare GS for reception" and "Perform reception". On the contrary, the "Avoid memory failure" involves a "hot" redundancy mechanism that can be activated during the pass. Thus, it is implemented in a more local manner as a Fallback node: use the main memory in first intent, and fall back to using the backup memory if this fails.

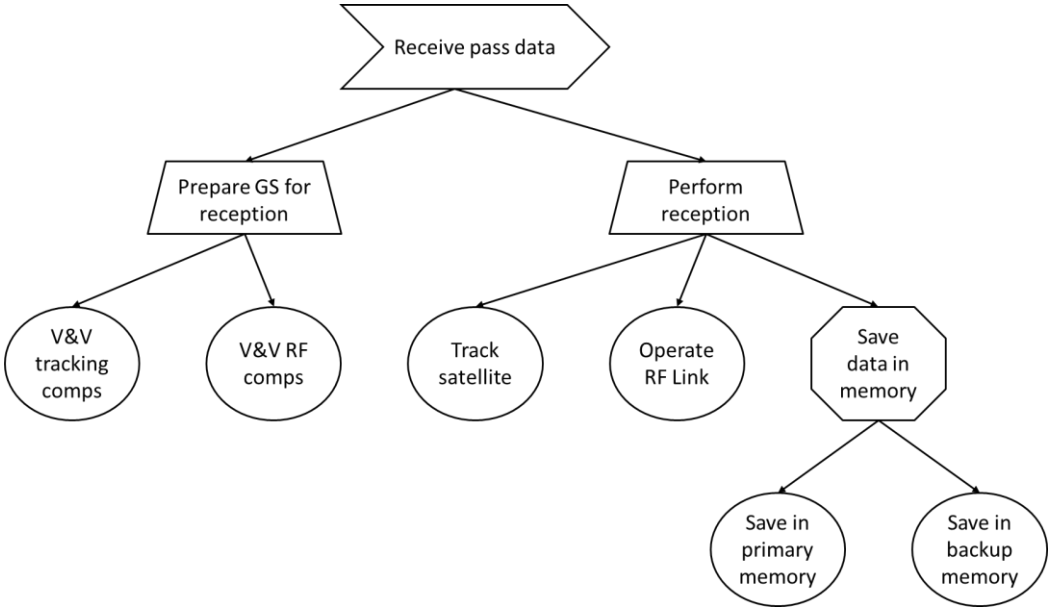

Figure 6 : Second Behaviour Tree, BT-v2. Obtained by manually eliciting the diagnosis objectives of BT-v1 and incorporating SE information. This is the result of our methodology.

The second BT is our ODM, as it describes the operations for operating the Ground Station, but our methodology offers several advantages. First, by starting from an FT Analysis, a very large set of faults is accounted for. In particular, most diagnosis approaches and tools do not take into account their own failure, but FT analyses systematically consider the case of false negatives in fault detection devices (and false positives as well). Second, the size of the BT is a good indicator of the complexity of the operational procedures. When the FT contains millions of events, and is too large for human manipulation (which is the norm in the industry), the first BT contains millions of diagnosis objectives. If we have some operational procedures that address high-level objectives, the second BT stays small and provides a succinct operational procedure. If not, the size of the second BT becomes unmanageable, which suggest that the system will be difficult to operate, and should be revised. Finally, our transformation helps tracing each fault in the FT to an activity that has the responsibility of handling it. In our example, this is illustrated in Table 1 below. Fault events that are handled by complex activity patterns are associated to the first node that is the parent of all associated activities. This is particularly useful to verify that the detection and mitigation means associated to low-level diagnosis objectives are correctly coordinated to achieve the high-level diagnosis objectives.

Fault event	Handling operation
Degradation of tracking component	Track satellite
V&V failure of tracking function	V&V tracking components
Pre-existing tracking failure	Receive pass data
Sudden Tracking failure	Track satellite
Degradation of RF component	Operate RF Link
V&V failure of RF function	V&V RF components
Pre-existing RF Link degradation	Receive pass data
Noise on RF Link	Operate RF Link
Primary memory failure	Save in primary memory
Backup memory failure	Save in backup memory
Memory failure	Save data in memory
Loss of pass data	Receive pass data

Table 1 : Table tracing each fault event of the Fault Tree to the operation that handles them.

5. Discussion

To our knowledge, this topic has received little attention in the literature. Diagnosis is often treated from the automation point of view, which carries several limitations. First, the diagnosis objectives must be defined at input, which is often quite difficult. Second, failures of the diagnosis functions themselves are rarely taken into account.

Finally, classic diagnosis approaches fail to take into consideration the actual benefit to the operators. Some exhaustive diagnosis approaches can return thousands of diagnosis candidates, and are rarely useful to operators in realistic time-critical systems.

Systems Engineering and Safety Assessment address the various activities that take place during system design, but without particular focus on operational diagnosis. Consequently, faults that are difficult to handle are often "left for the operator" to handle, and the workload and complexity of operations is often taken into account late in the design, if at all. This leads to the development of systems that are difficult to operate especially when they age and become subject to faults.

The choice of BTs to represent operations is debatable, as any Discrete Event System (DES) formalism would suffice. However, their history shows that the people who initially created and later on promoted them were system operators and users, in the video game industry and in the robotics domain. This distinguishes them from the rest of the DES models designed by academics, which are often more suited for modelling and analysis purposes. Still, BTs were initially intended for system control, while our approach uses them for system monitoring. This led us to define new standard node semantics, such as the ParallelAny node, of the Fault avoidance nodes, to express diagnosis objectives.

6. Conclusions

In this paper we addressed the problem of taking operational diagnosis into account at design stage. We proposed a design methodology that results in the production of an Operations-Dedicated Model (ODM). This methodology benefits from the completeness of Fault Tree (FT) analyses and accounts for a large set of faults. It allows experts with operational experience to take part in the system design and take advantage of their point of view. It produces a model of operations in which every fault event of the FT is associated to an operation that is in charge of handling it. It offers a way to detect early during design if the system is difficult to operate, because the functional objectives or because of the diagnosis objectives.

Acknowledgements

The authors would like to thank all individuals, companies and research institutes involved in the S2C project of IRT Saint Exupéry and in particular, Airbus Defence and Space (ADS) for proposing and funding this research topic. The results presented in this paper are part of the work of an interdisciplinary PhD undertaken by N. Christofi, supported by the French National Research Agency (ANR). Special thanks to Paul Albessard (ADS), for his contribution on the topic of Ground Segment Operations.

References

[1] Bob Ferrell, Mark Lewis, Rebecca Oostdyk, Jesse Goerz, Jose Perotti, and Barbara Brown. Lessons Learned on Implementing Fault Detection, Isolation, and Recovery (FDIR) in a Ground Launch Environment. In *AIAA Infotech@Aerospace 2010*, Atlanta, Georgia, April 2010. American Institute of Aeronautics and Astronautics.

[2] J.-P. Katoen, "Towards Trustworthy Aerospace Systems: An Experience Report," in *Formal Methods for Industrial Critical Systems*, Berlin, Heidelberg, 2011, pp. 1–4.

[3] Henry, David, Silvio Simani, and Ron J. Patton. "Fault Detection and Diagnosis for Aeronautic and Aerospace Missions." In *Fault Tolerant Flight Control: A Benchmark Challenge*, edited by Christopher Edwards, Thomas Lombaerts, and Hafid Smaili, 91–128. Lecture Notes in Control and Information Sciences. Berlin, Heidelberg: Springer, 2010.

[4] Katoen, J. P., and T. Noll. "Trustworthy aerospace systems. Public Service Rev. Eur." Sci. Technol 11 (2011): 204-205.

[5] Nikolena Christofi and Xavier Pucel. A Novel Methodology to Construct Digital Twin Models for Spacecraft Operations Using Fault and Behaviour Trees. *ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MODELS 22)*, Oct 2022, Montréal, Canada.

[6] Nikolena Christofi and Xavier Pucel. From Safety Assessment Models to Operational Diagnosis Models. *33rd International Workshop on Principle of Diagnosis – DX 2022*, LAAS-CNRS-ANITI, Sep 2022, Toulouse, France.