

Modeling Space Operations Systems Using SysML as to Enable Anomaly Detection

Luis Rabelo
University of Central Florida

Tom Clark
ERC Inc.

ABSTRACT

Although a multitude of anomaly detection and fault isolation programs can be found in the research, there does not appear to be any work published on architectural templates that could take advantage of multiple programs and integrate them into the desired systems. More specifically, there is an absence of a methodological process for generating anomaly detection and fault isolation designs to either embed within new system concepts, or supplement existing schemes.

This paper introduces a new approach based on systems engineering and the System Modeling Language (SysML). Preliminary concepts of the proposed approach are explained. In addition, a case study is also mentioned.

CITATION: Rabelo, L. and Clark, T., "Modeling Space Operations Systems Using SysML as to Enable Anomaly Detection," *SAE Int. J. Aerosp.* 8(2):2015, doi:10.4271/2015-01-2388.

INTRODUCTION

Spacecrafts are comprised of numerous complex systems. This includes the equipment that makeup the spacecraft used for delivery, and the various payloads and science instruments used to meet the research objectives. Accompanying this complexity will be the assorted complications. As individual components do not always operate as projected, some failures are to be expected. This is further compounded as the system's complexity increases. With payload costs just recently starting to approach \$1000 per pound (a tenth of that experienced by the space shuttle) ([SpaceX, 2011](#)), designers must still strike a balance between system redundancy, sensor allocation and hardware weight. An absence of redundancy minimizes the options an operator has should a failure occur. A lack of instrumentation limits visibility into system performance.

For many systems, timely recognition of an anomalous situation means the issue gets evaluated and a course of action is decided on quickly. Taking corrective action sooner can help minimize damage generated from off-nominal conditions, or avoid serious outcomes for those problems that can escalate rapidly. The ability to quickly detect and identify anomalies that arise is vital for critical space operations.

Anomaly Detection

The predominant method for developing an anomaly detection system is to select one or more applications that may fit one's needs. The selection process may be limited to what is readily available, and a scrub against the system requirements may end up diminished.

After these modules are selected, development then starts on a system-level architecture that can integrate the various elements into the current design. As this is a niche application, a certain amount of tailoring will be required to render the forthcoming architecture functional. An application that is not a good fit can amass excessive development hours in an effort to make the system operational.

Detecting an anomaly is the first half of providing the operator with the vital data necessary to develop a course of action. After an anomaly has been detected, the other half of the process (and equally important) is to isolate the fault within the system. Isolation to a component or subassembly provides the domain experts with the essential information necessary to respond and possibly remedy the situation. Fault isolation will optimally be able to point to a specific source. However, a lack of sensor information often leads to uncertainty which can result in an inadequate diagnosis. It is often the case that the initial data related to the anomaly is not sufficient to pinpoint the original problem, and involve additional troubleshooting to find the cause. This may result in an initial isolation of the problem to an upper subsystem level. In this limited-visibility scenario, it would not be unusual to have multiple suspect subassemblies and/or components identified.

In addition to issues related to limited sensor data, system complexity can make the fault isolation process arduous. The task of isolating to a fault entails assessing all possible contributors to the problem, each with varying degrees of sensor coverage. For a complete system, these contributors can number into the hundreds and possibly thousands. A

fault that yields additional damage compounds the problem as one must differentiate if that damage is collateral to a single fault or originating from an entirely different fault source.

LITERATURE REVIEW

A survey of the literature was performed on the topics of anomaly detection and fault isolation, as well as system engineering tools that could be used to develop a detection/isolation framework. Numerous models are available in both model-based and data-driven classes. The algorithms involved tend to be ‘specialists’ in that they are most effective for selective failure modes and/or component types. Therefore, it is anticipated that a detector/isolator system will be comprised of multiple applications so that it is effective on the complex system for which it is being designed

The primary method of anomaly detection is an exception-based method. This method notifies an operator if design or operational limits are exceeded. Data-driven models have found a role in complex, dynamical systems, and function by detecting outliers in the data which have not yet exceeded predetermined limits. Such models can disregard the physics behind the system allowing for distribution across multiple systems, though detection accuracy is dependent on the quality of training data and effectiveness of the scoring-algorithm. Fault isolation techniques tend to be model-based as system knowledge is required to isolate the fault to its source. For example, expert systems that are intended to mimic human-reasoning (the predominant method employed to identify fault sources), have been widely used for fault isolation applications (See Table 1).

Table 1. Expert system techniques for fault detection and diagnosis (Angeli, 2010).

ADVANTAGES	DISADVANTAGES
Rule based diagnostic expert systems	
Rules can be added or removed easily	Lack of generality
Explanation of the reasoning process	Poor handling of novel situations
Induction and Deduction process is easy	Inability to represent time-varying and spatially varying phenomena
A process model is not required	Inability to learn from their errors
Efficiency and effectiveness in fault detection	Difficulties in acquiring knowledge from experts reliably
	Development and maintenance is costly
Model based diagnostic expert systems	
Device independent diagnosis	Domain dependent
Knowledge acquisition is not needed	Difficulties in isolation of faults
Ability of diagnosing incipient faults	Knowledge bases very demanding
Deal with unexpected cases	
Flexibility in the cases of design changes	
Dynamic fault detection	
On-line diagnostic expert systems	
Real time fault diagnosis	Domain dependent
Ability to handle noise	Good models are required
Generalization	Require considerable data
Fast computation	Inability to explain the reasoning process
Ability to handle with dynamics	Computationally expensive

Data driven applications have been used for fault isolation. A data-driven application called Inductive Monitoring System (IMS) is a distance-based anomaly detection tool that uses a clustering technique. The data structure used for distance-based analysis is a vector of concurrent values from related system parameters. IMS reads real-time (or archived) data and formats it into a vector structure. It then searches the knowledge base of nominal data (training data) and returns the distance between real-time and the nearest nominal data vectors (Matthews, Srivastava, Iverson, Beil, &

Lane, 2011) (Martin, Schwabacher, & Matthews, 2010). When the real-time data is consistent with nominal, this difference is close to zero. If the data vectors start to diverge, an increase in the vector differences is noted and the real-time data is then deemed ‘out-of-family.’ This can be an indication of an anomaly that is occurring.

It should be noted that the real-time data is being compared to previous collected empirical data. Thus, an out-of-family indication can also reflect a normal condition that was not fully characterized within the nominal data sets used to ‘train’ the model. The IMS application works well with unsupervised data which is likely the only type of data available for most large complex systems. Unsupervised means there is an assumption of normalcy, but a potential exists that undetected anomalies are embedded within such data sets. In these cases, IMS could treat some anomaly precursors as nominal, requiring even greater vector disparity before getting flagged as anomalous.

IMS has been used for anomaly detection testing in the Space Shuttle (wings, main engines) and ground launch systems. In addition, it is currently used to monitor Space Station subsystems (Matthews et al., 2011). Reference figure 1 for a process flow example using a data-driven distance-based anomaly detection model.

On the other hand, isolation is a difficult task as systems often lack the requisite sensor data, hence lacking the necessary insight for accurate identification. This difficulty is further compounded by large numbers of potential sources to evaluate within complex systems. An architectural framework that combines these methods is desirable. A model based system engineering tool, SysML, shall be used to evaluate the premise that such a framework is possible. A descriptive model that can assist with analysis, specification, design and verification of this concept is the desired outcome.

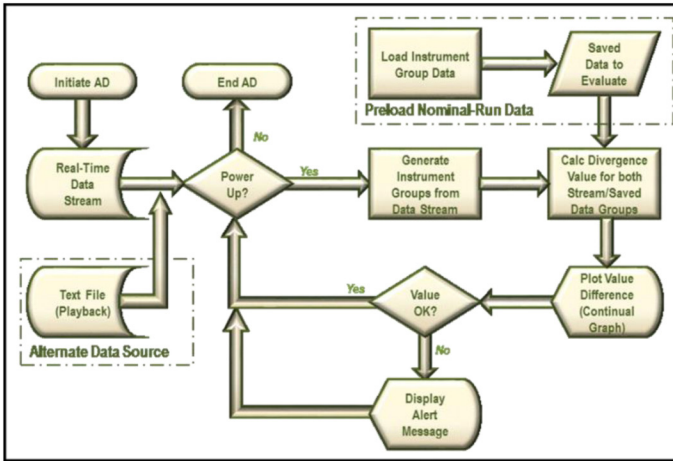


Figure 1. Anomaly Detection Process Flow Example

PRELIMINARY FRAMEWORK

This section presents the preliminary framework that will describe the principles and procedures used to pair anomaly detection and fault isolation (AD/FI) applications to new or existing complex systems. This framework involves a multi-stage process.