THE COLUMBUS MODULE AS A TECHNOLOGY DEMONSTRATOR FOR INNOVATIVE FAILURE MANAGEMENT

Enrico Noack (1), Andreas Luedtke (2), Ingo Schmitt (3), Tino Noack (3)
Eric Schaumlöffel (4), Ernst Hauke (1), Johannes Stamminger (1), Erik Frisk (5)
(1) Astrium GmbH, 28199 Bremen, Germany
(2) OFFIS, 26121 Oldenburg, Germany
(3) Brandenburg University of Technology, 03013 Cottbus, Germany
(4) RedHat, 85630 Grasbrunn, Germany
(5) University of Linköping, Sweden

Abstract

Over the past decades new Failure Management technologies have been investigated and were presented. With the experience of more than four years of successful operation of the Columbus Module, which is an integrated element of the International Space Station, we are now able to identify technologies which are of interest for the operators.

Efficient employment of the resources is of high interest for all stakeholders. This fact becomes more important especially for long term missions. Resource management during operation is also a topic for the European Air Traffic (Euro Control) and Airport Control. Just recently the Collaborative Decision Making framework was implemented here to optimise the resource management. We found Collaborative Decision Making to be appropriate also for our purpose. Our paper describes the steps of analysing the tasks and the distribution of tasks, which need to be performed at the Control Centres. It also describes the necessary steps on investigating the overall efficiency, safety, and resilience of the Columbus Ground System. This includes all elements such as tasks, agents (human and machine), resources, and their dynamic interaction.

Within this framework we are now able to deploy existing technologies that support the resource management. The technologies selected so far are the Data Mining for Anomaly Detection, Model Based Diagnosis, and Complex Event Processing. Deploying complementary techniques based on either predictive or field knowledge, we aim to compensate the drawbacks of one technique with the assets of the other. We are using the example of the Columbus air loop, and we are able to demonstrate that Failure Management, besides the fulfilment of the safety and reliability requirements, can also contribute to an efficient use of the resources. The technologies will be deployed into the existing Columbus Ground System using commercial and open source software such as Enterprise Service Bus (Asteria based on the Apache ServiceMix) and JBoss Enterprise Business Rule Management System (based on Drools/Fusion).

1. INTRODUCTION

1.1. Motivation

Over the past decades new Failure Management technologies have been investigated and were presented. For manifold reasons these technologies have not been used in space applications such as the Columbus Module. We intend to use the Columbus Module as a technology demonstrator. That should open space technology to new ideas and if possible make space technologies to be a driver for research and development activities.

Shrinking budgets reduce the effort which is spent during development phases. Only the need from the operators, to carefully use the available resources, and to safely operate the spacecraft, can require innovations and investments for a better Failure Management. Columbus is now in the operational phase, thus operation is in the focus of all involved teams. The major challenge, the teams are currently facing, is the demand to reduce the operational costs. Automatic flight system monitoring is surely a candidate which supports an efficient employment of resources.

Failure Management is spread over the different mission segments and mission phases. Therefore, different stakeholders like flight segment engineers, ground system engineers, operators, research facilities are involved. An efficient Failure Management can only be achieved, when all the different stakeholders come together. As Columbus is operational until 2020, the different teams/disciplines are available and can attend the introduction of new technologies.

The International Space Station (ISS) is in the low earth orbit. There are long and frequent contact times allowing the deployment of a ground based Failure Management. Compared to the flight system, the ground system provides easy access and allows the application of commercial and/or open source software. Thus, quick implementation with rather low cost is possible. Using the ground segment is a first step. Later the results can be mitigated to the flight segment (§6.3).

1.2. Teams and Tasks

This project intends to bring different disciplines together. The idea behind is to make use of existing competences, to support their further development, and by this to maintain them for long-term. For that reason, we have several teams with specific tasks.

Columbus operation is a very complex task. Several groups (e.g. Columbus Control Centre, Engineering Support Centre, ISS Control Centre, Payload Control Centers, …) have to work individually and collectively to operate Columbus. Each group performs special functions (e.g. preparation, planning, execution, …) to fulfill the different types of operations (e.g. experiment, maintenance, trouble shooting, ..). The activities are prepared by groups that might not be available during execution (shift plan, time shift between control centers, …). We have a specific team, namely the OFFIS institute, that is taking care about the integration of the new technologies into the user's world. Section 2 gives more insight to that.

ASTRIUM GmbH is focusing on their competences. These competences are twofold. The first competence is the integration of the new services into the existing ground system. This is discussed in Section 3. The second competence encompasses the engineering knowledge of the flight system.

Failure Management technologies are considered by two universities. The BTU Cottbus is dealing with Data Mining, specifically with data classification and clustering [1], [2]. Refer to section 4.2 for more details. In addition, the University of Linköping is engaged with model based failure detection (section 4.4).

The Complex Event Processing tool is very powerful. In order to deploy this tool with the full capabilities, the RedHat team will be responsible for the integration of the Business Rule Management System (BRMS), they will mentor the data stream management (e.g. data dispatcher), and mentor the rule selection/creation. Read more in section 4.3.

2. FRAMEWORK

As proposed by „Cost-Effective Space Mission Operations“ [3] we considered, for our purpose, spacecraft operation as an act of Resource Management. Power from the spacecraft solar arrays, network services for ground data transmission, they are all resources. These resources are managed and used by the spacecraft operators. In the preparation and planning phase the operators establish a mission plan. They insure that the resource budgets have a positive margin. In the execution phase, the operators perform the activities from mission plan and they make use of the procured resources. As the things never go as planned, the mission plans have to be adapted to the actual circumstances. The resource consumption has to be adapted to the actual situation.

The resources are provided to perform certain activities such as experiments, system maintenance, public affairs events and so on. Efficient operation can be understood as an optimization problem, where we have on one side the activities and on the other the resources. While the outcome of the activities is our earnings, the provision of the resources is causing costs. To achieve better efficiency the procurement of resources not used at space craft operation must be avoided. This is a well known problem. For instance the air traffic and airport control is facing a similar situation. Just recently a process, called Collaborative Decision Making (CDM) has been introduced at Euro Control (European Air Traffic Control) and the European airports. This process has two major goals. The first goal is to reduce the unknowns/uncertainties in the planning phase, which aims to avoid the provision of resources which are not used in the operational phase. This is achieved by replacing assumption with empirical knowledge from the operational phase. The second goal is to provide, in the operational phase, to all operators (Euro Control, Airports, Airlines, ..) useful and consistent information. This helps them to maximize the number of take off and landings or to say it in other words to minimise the process down times.

We see the Columbus Ground System (CGS) as a collaborative system. A collaborative system is a system of agents cooperating to the achievement of some common, super-ordinate goals. The cooperative CGS (cf. FIGURE 1) is defined as a set of agents, which can be either human ①a (e.g. operators) or machine ①b (e.g. the Columbus onboard Failure Management) agents. These agents basically interact and communicate with each other, thus creating a collaborative network. Tasks ③ are assigned to the collaborative system and allocated to the agents, who achieve them by using their own as well as distributed resources ⑤. Each agent has access ⑥ to specific usually limited resources. The cooperative system operates on the Columbus space module ⑦.

Moreover, each of the components of the cooperation framework can be static or dynamic. In addition, the agents in the system may change (some of them leaving, others incoming) or the capabilities of agents may change (e.g. performance of human agents deteriorates under stress). Also, tasks may change (e.g. a failure occurs and a recovery has to be initiated) as well as the available resources (e.g., a data transmission equipment may fail).

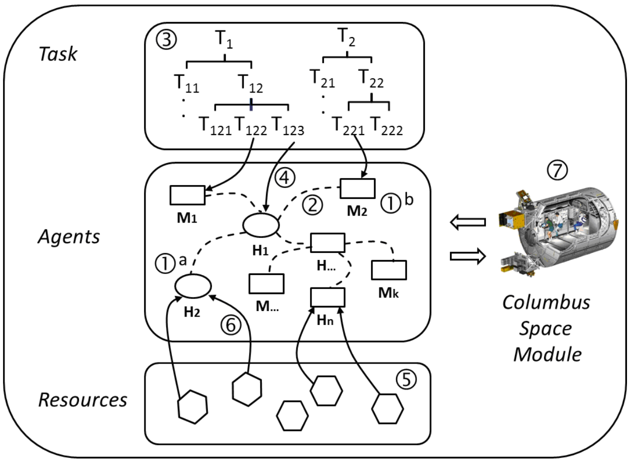


FIGURE 1. The cooperative Columbus Ground System

We will investigate as well as redesign the way in which the agents share the performance of tasks using shared resources. This holistic perspective allows to investigate the overall efficiency, safety and resilience of the CGS by considering all elements (tasks, agents, resources) that are involved in maintaining situation awareness of the CGS health and mission status, in making decisions on how to be-have and react as a whole, in distributing associated tasks to currently available agents (human and machine), who will perform them based on currently available resources. The design goal is to optimize the cooperative system based on well-defined performance measures. We will use three measurement categories:

- measurements for task performance
- measurements for communication performance
- measurement for cognitive performance

During the project the individual measurements will be integrated into a compound optimization metric.

We foresee to specify, design and implement the following improvements of the cooperative CGS:

- adding a knowledge base as a new resource containing information on past incidents and successful solutions
- improving existing resources for CGS monitoring
- redesign of task distribution and task procedures

Further improvements will be derived from the detailed investigation of the current design of the co-operative CGS.

3. SYSTEM ARCHITECTURE

As stated in the previous section the knowledge base is one of the improvements. The knowledge base:

- brings the teams into the position to make their own decisions, and
- contains the empirical knowledge.

The central role of the knowledge base is shown in FIGURE 2. The incoming Columbus path telemetry is continuously analysed by the online diagnosis. In case the nominal or off-nominal system state is known, the knowledge base provides relevant information for the manual and automatic processes. In case the state is unknown, the online process can not support real time operations. In this situation the offline process has to start. Now, the signature and the relevant information of the new state have to be identified and a knowledge base update is required.

The system (FIGURE 2) consists of different parts, which are the Backbone (grey), the services needed to integrate it into the existing Columbus ground system (red/yellow), the online/offline diagnose services (blue), and the user services (purple). The different parts are shortly presented in the next sections.

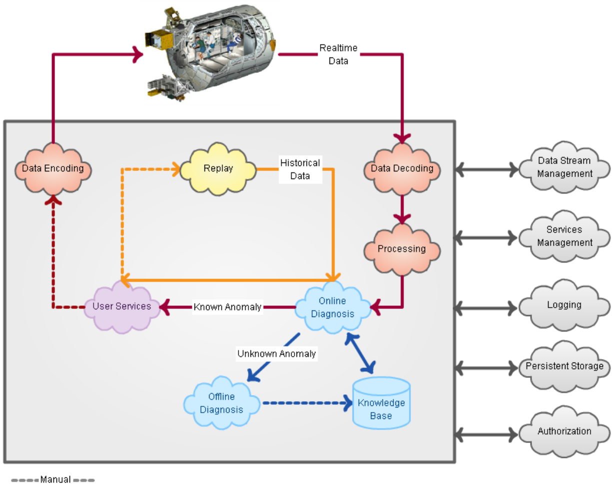


FIGURE 2. Overview

3.1. Backbone - Asteria

Already the Online Diagnosis requires different services (e.g. Complex Event Processing, Signal Processing, ..). There are also the data pre-processing services (e.g. decoding, archive, …) which may even change for the different applications. Finally user services exists (visualisation, notification, ..) which in similarity to the data pre-processing, will change with the applications. All these services must be able to operate together.

The solution is the Enterprise Service Bus (ESB), which allows the deployment of the different services into so called OSGi container (Open Service Gateway initiative). The OSGi container transforms the specific service from the components (BRMS, archive, display, ..) into a service which is available on the ESB. Furthermore, the OSGi container provides an open architecture which allows the exchange and addition of further services in the future.

The ESB will also provide services which are common for network related applications. Amongst others, this includes network security, testability, logging or, routing.

We will use the Asteria - ESB. It is characterised by:

- Based on Apache Karaf OSGi Runtime (light-weight container)
- Provisioning through Maven/Nexus
- Management through JMX, SSH or Web Console
- Hot Deployment
- Logging System
- Configuration System
- Inversion of Control (Spring/Blueprint)

FIGURE 3 shows as an example the implementation of the online data diagnosis using the Asteria ESB.

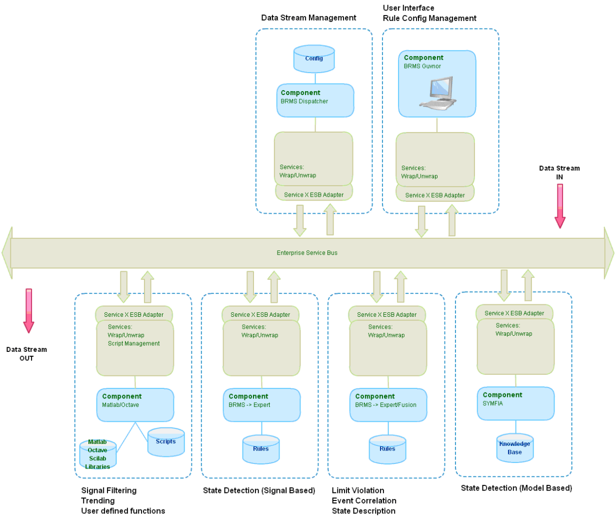


FIGURE 3. Architecture for the online data diagnosis

3.2. Embedding into Columbus Ground System

Before the Columbus Data Stream can be analysed it must be:

- collected,
- pre-processed, and
- controlled.

There are three different data sources which are considered. These data streams are the path telemetry, the commands and the events (messages) generated by the onboard system. All these data are collected. There exist independent services for that purpose. The services will take care, that the data streams are decoded into a user readable format (engineering values). In case of the path telemetry a further pre-processing is done. After decoding and pre-processing the data stream is made available for the other services (e.g. online data stream monitoring).

3.3. Online Data Diagnosis

The Online Data Diagnosis is split into the following functions:

1) Signal Filtering
2) Signal Analysis
    a) Monitoring
    b) Mode/State Detection
    c) Trending
    d) User defined functions
3) Data Interpretation

The sequence, in which each of these functions is used, is not predefined. Services may be used more than once during the data analysis process.

3.4. Offline Data Diagnosis

The Offline Data Analysis covers the following task:

- Data Classification/Clustering
- Knowledge Base Maintenance
- System configuration
- Validation process

For the data classification and clustering process, specific tools will be used (see §4.2).

3.5. User Services

The user services provide the capabilities to

- visualise the information,
- to notify the user (pop-up window, eMail, Call-In),
- to log telemetry and events, and
- to request an automatic response.

All these services are based on the services of the existing ground system. For instance, the visualisation service provides its information to the crew displays (Unified Synoptic System). To be able to display the new information just new displays have to be prepared.

4. TECHNOLOGY DEMONSTRATION & COLUMBUS UTILISATION

This section is addressing the new technologies which are applied in the project. In order to enable the reader to follow this excurse a brief introduction to the Columbus air loop is given.

4.1. Columbus Air Loop

The equipments which are used in this paper (FIGURE 4) are the fan assemblies and the HEPA filter. The four fan assemblies are:

- Inter Module Ventilation Supply Fan (ISFA),
- Inter Module Ventilation Return Fan (IRFA),
- Cabin Fan Assembly 1 (CFA1) and,
- Cabin Fan Assembly 2 (CFA2).

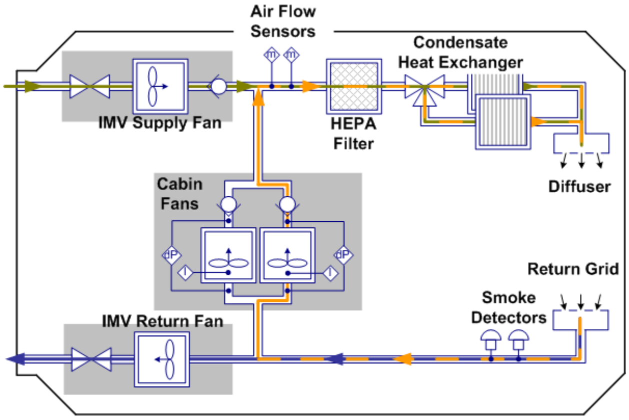


FIGURE 4. Columbus air loop

The Columbus air loop can be operated in 8 stable modes and 43 interims modes. The interim modes are used for air loop reconfiguration.

The COL air loop equipments (CFA1, CFA2, ISFA, and IRFA) provide a forced air flow to avoid dead air pockets (safety critical for the crew), to enable the fire detection, and to remove the heat from air cooled equipment. As the Columbus Module has no O2 supply and no CO2 removal, the inter module ventilation has to provide the fresh air.

4.2. Mode & State Detection using Data Classification

In the context of failure management, clustering plays the role of offline data processing. Time-based sensor data are analyzed and evaluated with respect to the occurrence of failures. In other word, it is required to cluster signal data into a set of different failure classes. The obtained classes can be used to train classifiers. A set of classifiers refers to a trained model. After a classifier has been trained its generalization power needs to be tested. A standard way to evaluate the trained model is the so-called cross-validation.

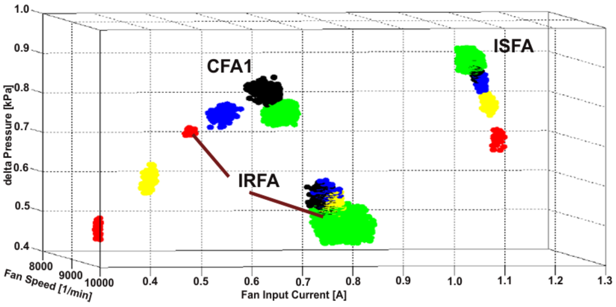


FIGURE 5. Data cluster for failure scenario

The data clusters of FIGURE 5 are described in TAB 1.

Data cluster	Color
Nominal	Green
Angular point	Black
81% clogging	Blue
81% … 99% clogging	Yellow
100% clogging	Red

TAB 1. Colour coding of FIGURE 5

FIGURE 5 depicts a failure scenario related to return grid clogging. This includes the assemblies CFA1, ISFA, and IRFA. In this failure scenario, the return grid is obstructed either gradually by pollution (e.g. dust, cf. FIGURE 6) or instantly by a larger object (e.g. paper sheet). Different clogging ratios have been applied (cf. TAB 1). For the data classification the three dimension fan speed, input current, and fan delta pressure are used.

FIGURE 6. Dust Collection in the air loop

Very important in our context is the time dimension in two aspects:

- Failures like wear-out effects are long-term failures. Thus, we need for a correct classification not only data from one instant of time but from long-term. A wear-out like effect is shown in FIGURE 7. For CFA1 two data clusters are shown, once for the beginning and once for the end of year 2008. The effect is caused by the dust collection (cf. FIGURE 6) of the air loop and the respective change of the characteristics.

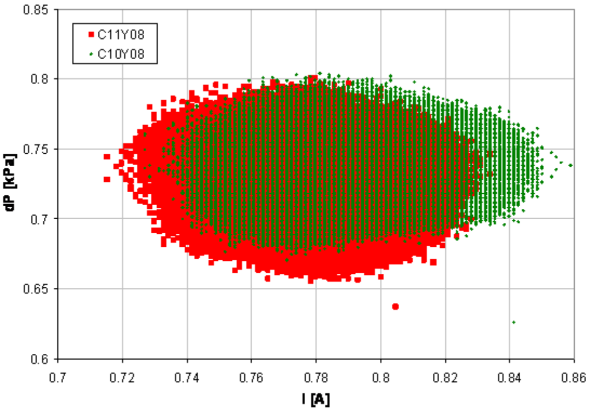


FIGURE 7. Data cluster showing wear-out like effect

- At the time of training a classifier we do not know all failure classes [2]. We must be prepared to learn new classes since unpredictable situations can occur. Therefore, we need a steady training and classification cycle which repeats the learning phase again and again. A situation, where new training is required is presented in FIGURE 8. After removing and replacing the failed IRFA, the signature of this equipment had changed. Shown here is the probability of the speed measurement for both fans with the same speed setup. It can be seen that the speed measurement is shifted by 50 to 90 [1/min].

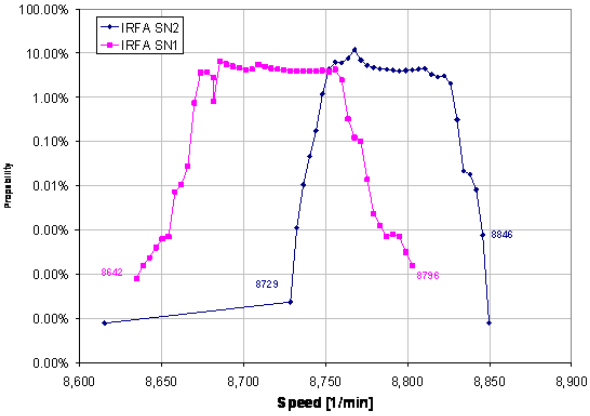


FIGURE 8. Speed signature before and after IRFA replacement

When data is pre-processed into classes then we know how to deal with the current state. Thus, to every failure class we assign action rules and further meta data in order to define the consequence on a given state classification. This information is than available for the online data processing.

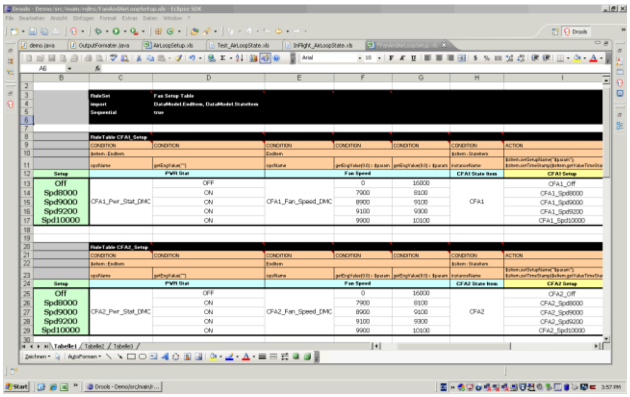
4.3. Complex Event Processing

A new trend can be observed in industry which is the application of Complex Event Processing (CEP) for the monitoring of the production processes. Complex Event Processing deals with streams of events. The streams can be provided via an Enterprise Service Bus, flat text files, database tables, raw sockets, or even web service calls. Regular facts (facts with no temporal constraints) are processed independent of time and in no particular order. Events, which have strong temporal constraints, must be processed in real-time or near real-time. CEP processes events chronologically as they are inserted into the rules engine. It uses a session clock that enables the rules engine to process events as they occur in time and based on age, synchronize streams of events (so events in different streams can be processed in chronological order), implement sliding windows of interest, and enable automatic life-cycle management.

The advantages of those systems are the real time capability and the flexibility/adaptability. The flight and ground segments already provide the real time capability. But the enormous flexibility/adaptability, when being in operation, makes the CEP Tools also interesting for space applications. Spacecrafts are built to explore the unknown. During the mission the experience will grow and the spacecraft must be adopted to fulfill its mission goals. For a continuous operation (no service interruption) it is important that the monitoring system

- can also be adopted the new spacecraft configurations and setups in near real time by the operators, and
- complex relations can be formulated to cope with not anticipated scenarios.

In the frame of our application the CEP Tool is used for the implementation of the data cluster which have been derived in §4.2. Learnt classifiers can be easily transformed into rules for a Complex Event Processing within the context of a stream data management system. Streaming is adequate since sensor data about systems and equipments are time-based and are produced by the sensors as a stream of data. Rules can be implemented per decision tables. The decision tables have the advantage of presenting the rules in a very user friendly way. Find an example for the cluster definition in FIGURE 9. The Power Status and the Fan Speed of the CFAs is evaluated to gain the current fan mode. The known modes are on the right hand of the table.



	B	C	D	E	F	G	H	I
2								
3		RuleSet	Fan Setup Table					
4		import	DataModel.EndItem, DataModel.StateItem					
5		Sequential	true					
6								
7								
8		RuleTable CFA1_Setup						
9		CONDITION	CONDITION	CONDITION	CONDITION	CONDITION	CONDITION	ACTION
10		$eItem : EndItem		EndItem			$sItem : StateItem	
11		opsName	getEngValue("")	opsName	getEngValue(0.0) > $param	getEngValue(0.0) < $param	instanceName	$sItem.setSetupName("$param"); $sItem.setTimeStamp($eItem.getValueTimeSta
12	Setup	PWR Stat		Fan Speed			CFA1 State Item	CFA1 Setup
13	Off	CFA1_Pwr_Stat_DMC	OFF	CFA1_Fan_Speed_DMC	0	16000	CFA1	CFA1_Off
14	Spd8000		ON		7900	8100		CFA1_Spd8000
15	Spd9000		ON		8900	9100		CFA1_Spd9000
16	Spd9200		ON		9100	9300		CFA1_Spd9200
17	Spd10000		ON		9900	10100		CFA1_Spd10000
18								
19								
20		RuleTable CFA2_Setup						
21		CONDITION	CONDITION	CONDITION	CONDITION	CONDITION	CONDITION	ACTION
22		$eItem : EndItem		EndItem			$sItem : StateItem	
23		opsName	getEngValue("")	opsName	getEngValue(0.0) > $param	getEngValue(0.0) < $param	instanceName	$sItem.setSetupName("$param"); $sItem.setTimeStamp($eItem.getValueTimeSta
24	Setup	PWR Stat		Fan Speed			CFA2 State Item	CFA2 Setup
25	Off	CFA2_Pwr_Stat_DMC	OFF	CFA2_Fan_Speed_DMC	0	16000	CFA2	CFA2_Off
26	Spd8000		ON		7900	8100		CFA2_Spd8000
27	Spd9000		ON		8900	9100		CFA2_Spd9000
28	Spd9200		ON		9100	9300		CFA2_Spd9200
29	Spd10000		ON		9900	10100		CFA2_Spd10000
30								

FIGURE 9. Implementation of data classification using JBoss Enterprise BRMS

FIGURE 9 shows the JBoss Enterprise Business Rule Management System (based on Drools/Fusion). Combining events with rules makes decision-making active. To ease the implementation of active decision-making for analysts and developers, JBoss Enterprise BRMS brings events and rules together so that they can be defined using a common set of authoring tools. Where some vendors provide separate tools that specialize in events or specialize in rules, each with its own idiom and audience, JBoss has incorporated Complex Event Processing (with temporal rules capability) within the BRMS itself.

This CEP tool has also been selected as it supports an enterprise and a community version. The enterprise version delivers high-quality software and maintenance along with information and support services that span the entire application infrastructure lifecycle. It gives continuous access to all supported versions of the software in both binary and source form, including all security updates and bug fixes. Therefore the enterprise version will be used for the real application. The community version enables universities/institutes to join and share without the need to purchase the enterprise version.

4.4. Model Based Diagnosis

The basic idea of model based diagnosis is to check consistency between a model and observed data. Measurements are used, together with the model, to detect changes in behavior and also try to determine the location of the component that caused the change in behavior. Methods for model based diagnosis has been developed in many different fields of research, for example automatic control [4], artificial intelligence [5], sensor fusion, estimation, and signal processing [6], [7], and [8] to name a few.

The key object in a model based diagnosis system is the model which may come in many forms, for example black-box models derived directly from data or physics based equation oriented models. A model based technique have the potential advantage that it is typically easy to adapt to changes in the process or if a new sensor is added etc. A physics based model also has advantages with respect to fault isolation, since typically it is not needed to have data from a faulty condition when developing the diagnosis algorithm. However, the main disadvantage is of course the need to develop a reliable model which may require significant engineering effort.

In this work, first principle physics based models in the form of a set of equations is used to describe the air-loop system. Such an approach has been successful in other, similar, applications, e.g., diagnosis of the air intake system in automotive engines [9]. It is also interesting to discuss and compare properties with for example a machine learning based approach to diagnosing the air loop.

The objective of the modeling work is to describe the measured quantities of the air loop, such as pressure drops, temperatures, air flows, currents, and fan speeds etc. Initial experiments indicate that with limited effort, using standard first principles models together with parameter identification techniques, it is possible to obtain models of high enough accuracy to detect the faults of interest. FIGURE 10 illustrates modeled and measured pressure drop over the HEPA filter which indicates that the model has a relative error of about 10%. This also indicates that faults, for example clogging, that changes the pressure drop more than this should be detectable using the simple model. Measured data indicate increases of pressure drops between 50% and 180% before replacement.

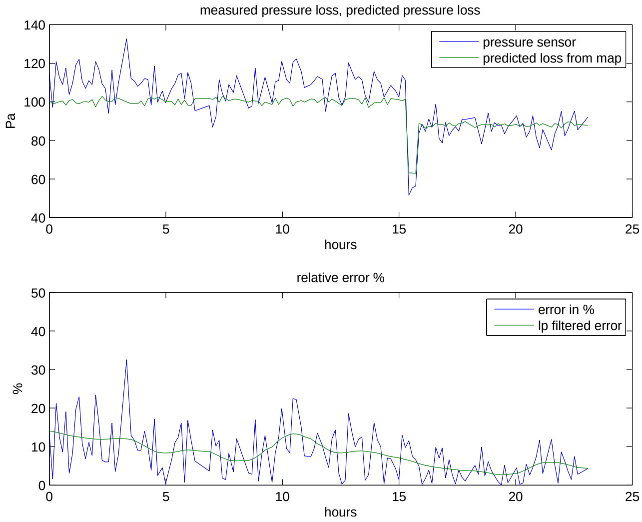


FIGURE 10. Predicted vs. estimated pressure drop over the HEPA filter.

Other measured quantities, for example the air-flow sensors have proven more difficult to model accurately, mainly due to turbulent air-flow in different operation modes of the system. Model based techniques also allow for prognostics, i.e., where it is possible to in advance predict when a component may fail. This can be used to avoid costly unplanned stops in operation. For example, when looking at data from a scenario of a fault in the IRFA, it is clear that by observing the power consumption of the fan that the impending failure of the fan is visible in observational data several hours before actual breakdown.

5. STATUS

5.1. Status

FIGURE 11 shows the current configuration. Today the data diagnosis is done manually. The data stream is emulated by sequential file access.

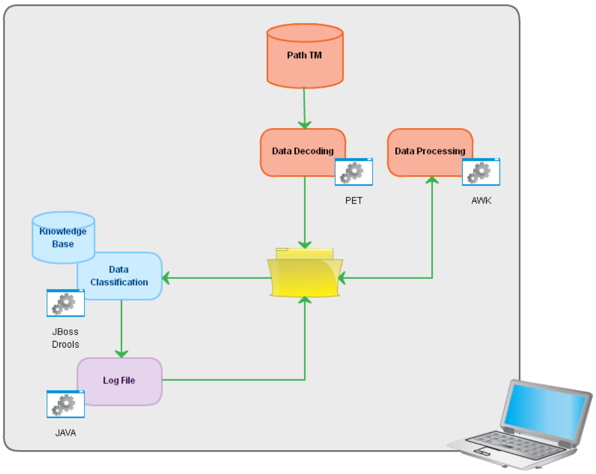


FIGURE 11. Current configuration

This configuration enabled the implementation and testing of the 421 rules for the diagnosis of the Columbus air loop. The Columbus air loop represents one out of six subsystems of the module. Performance tests have shown that a standard laptop is able to process 123000 measurements per seconds. The maximum number of measurements to be processed for the Columbus Module is less than 1000. Thus we are confident, that when extending the data diagnosis to the other subsystems of the Columbus Module:

- online data diagnosis can be supported, where all the incoming measurements must be evaluated in real-time, and
- also offline diagnosis can be performed, where data from long time periods must be processed, preferably within hours.

6. OUTLOOK

6.1. Offline Data Diagnosis

The next step is the development and deployment of a system until 2014, which can be used for offline purposes (cf. FIGURE 12). The system will have the following capabilities:

- tracking of system performance & trend analysis,
- tracking of resource consumption,
- tracking of COL configuration,
- tracking of duty cycle assessment, and
- support of near real time problem resolution process.

The system will cover all Columbus subsystems. It will have a complete knowledge base, which is called 'OPS Engineering Catalogue'.

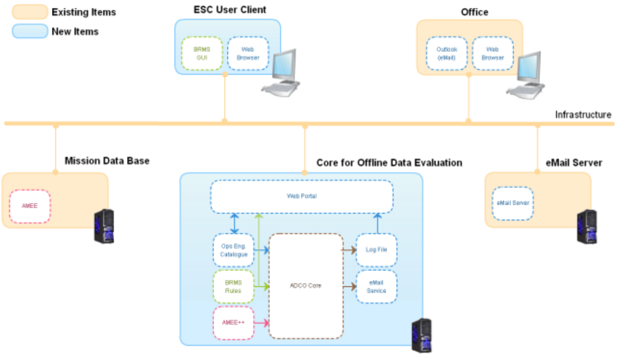


FIGURE 12. Offline version

6.2. Online Data Diagnosis

The development and deployment of the final version (cf. FIGURE 2) is planned for the time period of 2014 to 2016. Beginning with 2016 the system will have the capability to automatically monitor and log the onboard system.

6.3. Onboard Monitoring Experiment

An experiment has been proposed [10] which aims to mitigate the rule based mode and state detection into the Columbus Flight System. In this experiment the data diagnosis runs in parallel to the nominal Data Management System (DMS). The experiment will have no means to interact with the flight system. Instead the reaction will be logged, linked down to ground, and compared on ground with the DMS data.

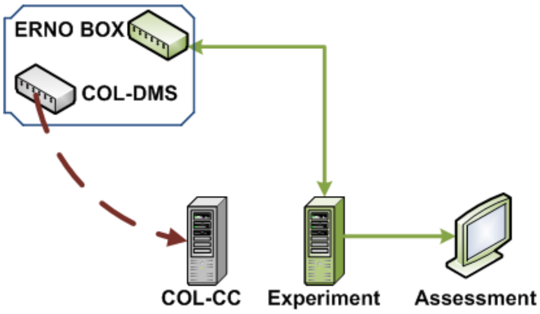


FIGURE 13. Mitigation of the ground monitoring into the flight segment

It is envisaged that the data diagnosis is implemented on the ERNO BOX (cf. FIGURE 14). The box is already space qualified and in orbit.

FIGURE 14. ERNO Box

7. ACKNOWLEDGMENTS

The authors wish to thank and acknowledge the ASTRIUM Space Transportation management for their support, with special thanks to Detlef Wilde, Stefan Wübker, Gerd Hajen, Dirk Schulze-Varnholt, Rüdiger Kledzik and Uwe Brauer.

References

[1] Martin Kaufmann, *Datenbankentwurf und Datenauswertung von Telemetriedaten des Columbus-Moduls der ISS,* BTU Cottbus, 2012, Master Degree

[2] T. Noack and Ingo Schmitt, *Monitoring Mobile Cyber-Physical Systems by Means of a Knowledge Discovery Cycle - A Case Study*, KDML 2012

[3] D.G. Boden and W.J. Larson, *Cost-Effective Space Mission Operations*, McGraw-Hill, 1996.

[4] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and fault-tolerant control*, Springer, 2003.

[5] W. Hamscher, L. Console, and J. de Kleer, *Readings in Model-Based Diagnosis*, Morgan Kaufmann, 1992.

[6] F. Gustafsson, *Adaptive Filtering and Change Detection*, Wiley, 2000.

[7] T. Kailath, A.H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, 2000.

[8] M. Basseville and I.V. Nikiforov, *Detection of Abrupt Changes*, Prentice Hall, 1993.

[9] M. Nyberg and L. Nielsen, *Model based diagnosis for the air intake system of the SI engine*, SAE Transactions, Journal of Commercial Vehicles, vol. 106, pp. 9—20, 1998.

[10] ASTRIUM-ST/ BTU Cottbus / ITP Engines UK Limited, *Failure Management for cost-effective Spacecraft Operation*, Response to ESA Call for Ideas (CFI) “ISS for Exploration”, November 2010