A Discussion on ISS Columbus Data Streams

Tino Noack[1], Enrico Noack[2], Ingo Schmitt[1], Sonja Sievi[2], Sita Mirzakhyl[2]

[1] Brandenburg University of Technology Cottbus, Germany
Institute of Computer Science, Information and Media Technology
{Tino.Noack, Ingo Schmitt}@tu-cottbus.de
[2] Astrium Space Transportation, 28199 Bremen, Germany
{Enrico.Noack, Sonja.Sievi, Sita.Mirzakhyl}@astrium.eads.net

Abstract. Monitoring space crafts is necessary to provide reliability and to avoid critical damage. Research on data stream processing provides a huge potential to facilitate, simplify, and enhance these monitoring tasks. We apply results from data stream research to our real-world scenario.

1 Introduction

Space crafts such as the ISS Columbus Module are complex systems with associated ground-stations. Space crafts are constantly monitored and controlled for keeping these systems reliable, stable, and healthy. Monitoring is applied onboard and remotely by telemetry and commands. To fulfil these tasks such systems are equipped with sensors that continuously produce sensor data streams. Data streams are used for online monitoring, control, and offline analysis. Online monitoring and offline analysis are prerequisites for anomaly detection, early detection of sensor degradation, and to identify the current system behaviour. These monitoring and control tasks are very time consuming and resource intensive. For remote systems any equipment failure must be avoided. The life-time of a space craft can be separated into the following four phases: *design*, *integration and test*, *launch and activation*, and *operation*.

The present paper aims to summarize and discuss our work on data stream processing in the context of space craft telemetry data. Please consider the following publications for further information and insights. Publication [1] demonstrates problems of the previously applied monitoring process. Publication [2] delineates the basis of our new monitoring approach under consideration of data stream processing. On this basis, publication [3] presents a real world case study with the use of complex event processing. The publications [4] and [5] give insights to frameworks and architectures used for our new monitoring approach.

In order to discuss our new monitoring approach, we provide a problem description in Section 2, explain system states and describe the ISS Columbus Failure Management System in Section 3, and delineate necessary gaps between data stream research and our lessons learned from monitoring the ISS Columbus Module in Section 4. Finally, Section 5 concludes our work.

2 Problem Description

Limit monitoring is a traditional monitoring approach that is broadly used over a variety of application domains. It is mostly applied by means of one-dimensional thresholds and can be used to detect gradual and sudden changes (concept drift or concept shift). However, limit monitoring considers only a few sensors while complex interrelations between system components, different data streams, and the credibility of the sensors itself (e.g. broken sensors) are ignored. Limit monitoring can be used to identify anomalies but without any indication of the root cause. Limit monitoring is robust enough to cover scenarios resulting in a complete loss of a function. Nonetheless, it is not robust enough in cases where a performance degradation has to be detected within noisy environments. This led to failures situations which were being neglected in the past [1].

Model-based monitoring is often used in the area of monitoring space crafts. It is utilized to build an early and static model of a space craft during design and test phases. Although, building such a preliminary model is very expensive and time consuming while the latter application environment (e.g. the space) is rather insufficiently addressed. Model-based monitoring approaches suffer from limited knowledge during design and test phases. Hence, the resulting model becomes very complex and inflexible during operational phases. Unpredictable changes of the system behaviour can occur at any time due to wear and tear or crashes. These changes might not been considered by the model; and therefore, maintenance becomes very intricately during operational phases [1, 5].

3 ISS Columbus Failure Management System

We distinguish between three possible categories of system states. The system works correctly if the system is in a *default state* and the system works incorrectly if the system reaches an *error state*. Both states assume that there exists knowledge about the system behaviour. The system does not work as expected if the system reaches an *anomaly state* and it exists no knowledge [3].

The implementation of the ISS Columbus Failure Management System [1,2] is distributed over different instances while each instance is responsible for different tasks and constrains. *On-board* instances should work automatically and in real-time while interaction with human experts is excluded. The on-board diagnosis entails the following functions: signal filtering, signal analysis (e.g. monitoring, classification or state detection, and trending), and data interpretation. *Ground* instances work semi-automatically and in long-term while human experts foster in-depth analysis (e.g. identification of failure causes). Offline data diagnosis covers the following tasks: data clustering, training of classifier models, knowledge base maintenance, system configuration, and validation. Moreover, user services are provided which involve the following capabilities: display of information (e.g. cockpits), user notification (e.g. email), telemetry and events logging, and automatic response to requests. The on-board instances are connected to ground instances via a downlink. The sensor data which is produced on the ISS Columbus Module is sent to the ground station (e.g. control centre) almost entirely (down-sampled). Down-sampling is mainly achieved by signal filtering and reducing the previous sampling frequency. A mission archive is used to store the transmitted data for long-term analysis and to plan long-term corrective actions.

4 Discussion

Based on the aforementioned system states, appropriate data stream processing algorithms must provide classification for identifying the current system state (default or error states) plus drift and anomaly detection for identifying gradual changes and sudden changes as well. The detection of the current system state can be a prerequisite for executing automatic actions. The application of data stream processing provides enormous advantages for recurring and long-duration missions. Furthermore, the aforementioned interpretation of system states provides a very flexible monitoring approach which can be used over all life-time phases and over all space craft instances (on-board and ground). Experiences gained during the integration and test phase are essential for the operational phase. Moreover, it is possible to compare different space craft missions and to provide historical data to avoid already occurred failures for future missions. Contradicting to model-based monitoring approaches, the described monitoring approach works empirically. Hence, it is an universal monitoring approach. Model-based monitoring approaches usually do not provide such comparisons due to specialized models for each mission. We performed experiments to underscore these statements [3, 5].

There exists a widely adopted assumption that data streams cannot be stored almost entirely. This assumption contradicts some real world applications. The presented example shows that data streams are stored in mission archives. Historical data are a prerequisite for long-term failure analysis and for planning long-term corrective actions. Such long-term corrective actions must be evaluated and assessed by human experts in order to avoid unwanted side-effects. Archives are necessary because human experts are responsible for related decisions and actions which have to be comprehensible.

Many data stream processing algorithms necessitate the existence of labelled training data during runtime for training a stream model. Such algorithms entail three drawbacks in the context of monitoring space crafts. First, the provision of labelled training data during runtime cannot be always guaranteed. Data analysis is a prerequisite in order to provide valid, assessed and adequate labelled training data. Second, model training or retraining during runtime expends computational resources which are actually envisaged for system monitoring. This is a problem while monitoring is applied under resource restrictions (e.g. processor speed or memory and power consumption). Third, training unvalued models during runtime can cause unforeseeable and critical side-effects for the system and the monitoring process. This can in turn cause critical system failures. The assessment of the resulting data stream models is often neglected by online training methods. While data analysis is performed on external information systems by human experts with the use of historical data it is preferable to train stream models offline. The trained models can be transmitted to the on-board instances after evaluation and assessment. However, predefined online adaptation of the data stream model could be an appropriate approach.

5 Conclusion and Future Work

We discussed ongoing work on data stream processing in the context of monitoring space crafts. Our discussion provides three main conclusions and hints for future work. First, data streams are stored almost entirely in some application domains. Second, data analysis is applied semi-automatically and human experts are obliged to validate their decisions and the resulting stream models. Third, lightweight data stream processing algorithms are necessary which consider restricted system resources while offline training is preferable.

Acknowledgements

This work was supported by the Brandenburg Ministry of Science, Research and Culture as part of the IGS at Brandenburg University of Technology Cottbus.

References

[1] E. Noack, W. Belau, R. Wohlgemuth, R. Müller, S. Palumberi, P. Parodi, and F. Burzagli. Efficiency of the Columbus Failure Management System. In *AIAA 40th International Conference on Environmental Systems*, 2010.

[2] T. Noack and I. Schmitt. Monitoring Mobile Cyber-Physical Systems by Means of a Knowledge Discovery Cycle. In *Seventh IEEE International Conference on Research Challenges in Information Science (RCIS)*, 2013.

[3] T. Noack and I. Schmitt. Monitoring Mobile Cyber-Physical Systems by Means of a Knowledge Discovery Cycle - A Case Study. In *Workshop on Knowledge Discovery, Data Mining and Machine Learning (KDML)*, 2012.

[4] E. Noack, T. Noack, V. Patel, I. Schmitt, M. Richters, J. Stamminger, and S. Sievi. Failure Management for Cost-Effective and Efficient Spacecraft Operation. In *Proceedings of the 2011 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE Computer Society, 2011.

[5] E. Noack, A. Luedtke, I. Schmitt, T. Noack, E. Schaumlöffel, E. Hauke, J. Stamminger, and E. Frisk. The Columbus Module as a Technology Demonstrator for Innovative Failure Management. In *Deutscher Luft- und Raumfahrtkongress*, Deutscher Luft- und Raumfahrtkongress, 2012.