DOCUMENT

Advanced Alert Management System (AAMS)

Executive Summary

Prepared by	**SOFTCOM International**
Reference	**EGOS-GEN-AAMS-ESR**
Issue	**1.0**
Revision	**00**
Date of Issue	**2020-05-19**
Status	**Final**
Document Type	**ESR**

European Space Agency
Agence spatiale européenne

APPROVAL

Title AAMS Executive Summary	
Issue 1.0	**Revision** 00
Author SOFTCOM International	**Date** 2020-05-19
Approved by Petros Pissias	**Date** 2020-09-09

CHANGE LOG

Reason for change	Issue	Revision	Date

CHANGE RECORD

Issue 1.0	**Revision** 00		
Reason for change	**Date**	**Pages**	**Paragraph(s)**

Table of contents:

1 INTRODUCTION

1.1 Purpose

The purpose of the document is to summarize the findings of the AAMS derisking activity.

1.2 Scope

The present document covers the whole scope of the activity.

1.3 Document Overview

Section 1- Introduction (this section) provides the purpose, scope and this document's overview.

Section 2 - Applicable and Reference Documents provides the list of reference documents.

Section 3 - Terms, Definitions and Abbreviated Terms provides a list of acronyms and terms used throughout this document.

Section 4 - Executive summary provides the summary.

2 APPLICABLE AND REFERENCE DOCUMENTS

2.1 Applicable Documents

Ref.	Document Title	Issue and Revision, Date
[AD-1]	ESA Contract No. 4000127714/19/NL/BJ/va	2019.07.12

2.2 Reference Documents

Ref.	Document Title	Issue and Revision, Date
[RD-1]		

3 TERMS, DEFINITIONS AND ABBREVIATED TERMS

3.1 Acronyms

Acronyms	Description
CEP	Complex Event Processing

3.2 Definition of Terms

Terms	Description

4 EXECUTIVE SUMMARY

Currently, space system operators at ESOC get alerted when certain system parameters exceed pre-set threshold values. These out-of-limit (OOL) alarms are simple in nature, each referring to a single parameter, and do not necessarily offer operators deep insight into a system's condition just by themselves. Operators can be repeatedly alerted of the same condition, as currently only basic filtering and prioritisation of alarms are available.

Remote (standby) operators get alerted to possible problems when the Mission Control System detects such system conditions, using mobile phone messages (SMS). Due to the limited nature of SMS and the lack of contextual information, it can be difficult for operators to quickly determine the relative importance of these notifications. They then have to remotely connect to the mission system using their laptop to investigate the issue.

The main idea behind the AAMS is to improve on this situation and enable advanced remote monitoring of operations by performing a technology push with two approaches:

1. Introduce intelligence to the alarm generation and notification processes by performing advanced processing of the alarm data using industrial-grade *Complex Event* Processing technologies.
2. Exploit the powerful capabilities offered by modern smartphone devices to present alarms together with contextual information about the monitored system.

The main objective of this activity was to reduce the technical risks associated with introducing these technologies to operational space systems by implementing a proof of concept prototype. This will ease the implementation of the fully-fledged AAMS product in a future activity.

To create a successful product, a deep understanding of end user needs, intended use cases, and necessary functionality was essential. To ensure relevance, the first step was the identification of relevant stakeholders inside the Agency for the requirements elicitation process. In addition to eliciting requirements from the Technical Officer, potential stakeholders were identified from missions (spacecraft operators, flight control engineers, system engineers), Ground Station engineers, as well as non-operations personnel. These included Space Situational Awareness personnel, such as for Near Earth Object tracking, and users outside the Agency, such as academia making use of space data products e.g. Fireball tracking networks.

Following best practices for requirements engineering, a workshop was organised at ESOC to perform stakeholder interviews. The main findings of the workshop were that introducing prioritisation to the space system alarms, managing the repetition of notifications, and adding the ability to correlate multiple events occurring at different times would all be valuable features to implement. This confirmed the need for flexible event processing of multiple data sources and highlighted the importance of enabling remote notification functionality for non-operations users. Furthermore, there was confirmed interest in using smartphone apps as user interface, modernising the process.

Thus, the main functions of the AAMS product were consolidated to include:

1. Retrieval of data from ESOC operations systems, ESA SSA systems, and providers outside the Agency (e.g. CNES data products), or the general internet (e.g. Twitter).
2. Process the above data using Complex Event Processing to correlate, filter, aggregate, etc. events and generate notifications for humans, customised for different users.
3. Deliver the notifications to users using a smartphone app. The system should provide customised contextual information to the users, explaining the reasons for the notification raised and aiding troubleshooting.

The simultaneous need for a system that is easily extensible to processing data from multiple sources and obeying strict ESOC network security rules led to a modular design with only minimal presence near operational systems.

A software prototype implementing the main identified use cases was built as part of the activity. This connected to two data sources: an EGS-CC instance to receive simulated spacecraft telemetry, and Twitter. The prototype integrated a Complex Event Processing engine, allowing custom rules to be flexibly defined by end users to perform a variety of operations on the data received. This has turned notification setup changes into a configuration process as opposed to a software development or scripting process.

Example rules created generate alarms when detecting: a complex event, inferred when multiple EGS-CC parameters have specific values within a time window (a capability not currently available at ESOC); lack of data received (e.g. to actively monitor for communication dropouts); ordinary mission control system alarms (demonstrating forwarding of existing alarms); more than 3 tweets with the hashtag "#fireball" within 1 minute (e.g. to monitor fireball sightings by the general public).

A mobile app software prototype was built for the Android platform to receive the generated alarms. The major functionality developed included: receiving EGS-CC OOL and custom CEP alarms in realtime using push notifications (i.e. capable of notifying the user even when the app is running in the background); displaying alarm details including the CEP rule that raised it and the involved parameter(s); plotting the time series of the involved parameters and other parameters to provide context for the alarm.

The vision to integrate a mobile app with production space operations systems posed the challenge of adequately testing it to the same quality level as the rest of the ground systems. To achieve this, the ESA automation testing systems (ART) could be extended integrate with a mobile app testing framework. This way ART could then be used for the extensive verification of the entire AAMS product while integrated in the ground segment. This would in turn ensure that ESA mobile applications will be as thoroughly tested as current operations software and be able to reach TRL8.

To that end, a survey of 11 mobile application test automation frameworks was first conducted. The survey considered several analysis criteria, such as Android/iOS support, support for hardware-based and emulator-based testing, licensing, and envisioned compatibility with ART. Using these criteria (and some more specific sub-criteria) the Appium testing framework was selected. It is open-source, supports real and emulated devices, has a compatible architecture to the ART, and supports both major mobile OSs as well as web-based mobile applications.

A first iteration for the design of an Appium-ART extension was then created. The ART framework architecture was first revisited as groundwork for this task. In the ART context, the Appium extension can be considered an additional instrumented test tool in the same fashion as the existing Java SWT GUI test tool. This way the integration of Appium will be in line with the current design of the ART.

The design document produced addressed the identified trade-offs related to the type of interfacing and integration of Appium components with ART. This first iteration of the extension design identified which ART components would be most affected. It also showed that to incorporate mobile-specific testing actions during test specification with existing ESA tools (TEMMPO designer), new testing libraries will have to be created.

The Appium framework was then used to create confidence, through verification and validation, that the AAMS mobile application was correctly implemented and behaved as expected. This way the selection of Appium was validated through its actual use. To get one step closer to the ART testing logic, basic test actions like clicks, taps, text input, swipe, etc, were implemented using wrapper methods that follow the ART layered, reusable, and decoupled approach.

A set of test cases to cover each testable requirement (through the mobile application) was written using the Java language and run using Appium. Each set of test cases was executed on a real device and a set of Android emulators with different OS versions and physical settings to widen the scope of the tested devices and increase the robustness and confidence of the automated tests.

The AAMS derisking activity aimed at performing the necessary technical work to reduce the risks associated with developing a full AAMS product in a follow-on activity.

All 4 objectives defined for the activity have been achieved:

1. The requirements engineering process has collected use cases and functional requirements from multiple stakeholders, leading to a Software Requirement Specification (SRS) document that was accepted by ESOC at the Requirements Review.
2. Complex Event Processing has been validated as a suitable technology to implement the core AAMS use cases in an elegant and flexible manner. It was also shown suitable for easily implementing a multitude of additional potentially interesting use cases outside the strict ESOC operations scenario.
3. The Android mobile application prototype built for the smartphone form factor demonstrated the use of push notifications for CEP alarms and presented alarm data in a user friendly and intuitive manner.
4. The ART framework extensions were designed to integrate with the popular Appium mobile testing framework. Appium was used to test the AAMS mobile app.

Furthermore, the technical risks identified during the activity have also been closed. These were:

1. *Potential incompatibility of mobile testing frameworks with the ART*. The "Appium" framework that was selected has been identified as compatible.
2. *Delay-prone mobile application push notification systems*. The Firebase Cloud Messaging (FCM) notification system has been shown to perform adequately for the ESOC use cases.
3. *Graphing libraries on the mobile platform could be inadequate for ESOC use cases*. Libraries available for drawing charts in Android have been shown to provide sufficient functionality for the identified ESOC use cases.
4. *Use cases requiring very high throughput for the Complex Event Processing system that may not be achievable on a single machine*. Forwarding of all parameters (~230/minute) received from an EGS-CC instance to AAMS did not require significant resources on a modest machine (i5 laptop).

With the above in mind, follow up activities can focus on:

- Implementing the remaining components defined in the architecture and completing the functionality defined in the requirement specification, including the ART extensions for testing mobile apps.
- Fully integrating the Appium framework with ART to create to a complete solution for testing ESA mobile applications.
- Maturing the AAMS implementation through extensive end-to-end verification using ART. This will test the entire AAMS processing chain from input data to data presentation in the mobile app.
- Definition of the required CEP rules for the identified use cases and end-users.
- Deploying the prototype implementations in an operationally representative environment, such as ESOC's Ground Segment Reference Facility (GSRF).
- Complete documentation.