Software Integrity Analysis Applied to IRIO EPICS Device Support Based on FPGA Real-Time DAQ Systems

D. Sanz, A. Bustos, J. Autrán, M. Fernandez, S. Urueña, M. Ruiz, S. Esquembri

***Abstract*–Nuclear fusion environments require dependability and safety analysis to ensure a reliable design and a deterministic behaviour. Failure modes identification, risk assessment and mitigation, guarantee that quality control procedures at different architectural levels comply with all the well-defined prerequisites at all the commissioning stages. Therefore, exhaustive analysis based on Reliability, Availability, Maintainability and Safety (RAMS) must be an unavoidable activity in such a kind of undertaking. The results of this analysis impacts on the hardware and software development: invalidating inadequate software architectures and hardware components, and forcing a given development assurance level depending on its criticality (and thus its costs). This paper applies RAM analysis methodology for an advanced Data Acquisition System (DAQ) based on FPGA, using standards and techniques commonly used for critical systems developments.**

The proposed DAQ system interfaces with signals coming from different sensors, acquiring data at high sampling rates (up to hundreds of MS/s), and in some cases performing Real-Time pre-processing. In turn, it must provide acquired data to control system, where control loopback will be applied. This fact implies that the DAQ system shall guarantee integrity, continuity, availability and accuracy, providing with the necessary integrity level.

This paper presents the analysis for the IRIO software tools as part of an EPICS IOC running under a hardware architecture compliant with the ITER catalogue for fast controllers. This analysis focuses on: RAM analysis to ensure the technical risk control and mitigation; criticality analysis and assessment of mixed-critical systems for failure propagation mitigation by the usage of segregation strategies, such as virtualization techniques by using hypervisors; the IRIO Software Integrity Level (SWIL) analysis, according to nuclear critical system requirements; and software verification methodology based on source code static analysis to reduce errors present in the final product. This analysis will provide confident methodologies to be considered in future software implementations for minimizing costs and risks in such kind of nuclear environments.

I. Introduction

UNDER nuclear fusion environments development activities, dependability and safety analysis are mandatory to ensure the compliance with all the established prerequisites for such a kind of systems. Reliability, Availability, Maintainability and Safety (RAMS), well-known terms in critical systems, comprise the main attributes for describing the system dependability. Independently of the environment where the system is housed, like aerospace, transportation or nuclear; systems have to operate under RAMS attributes, according to the specific requisites of each area. Failure modes identification, risk assessment and mitigation, guarantee that quality control procedures at different architectural levels complying with all the defined prerequisites at all the commissioning stages. The instrumentation and control architecture that can be found at fusion experiments is usually separated in different parts [1]; the safety tier, in charge of mitigating and providing radiation risks; interlocks machine protection tier for investment protection; and conventional instrumentation and control tier for retrieving data and communication. The interoperability of these tiers is crucial for the functioning of the machine, therefore depending on the criticality of the applications hosted in the different instrumentation and control plants, the systems dependability requirements are set from determined levels of quality to more restricted ones.

The system that we propose to carry out this analysis, is an FPGA based DAQ system supported by an EPICS Device Support that uses IRIO tools. This software (IRIO EPICS Device Support) is an EPICS driver that has two differentiate software layers: the IRIO API that addresses all the FlexRIO I/O registers and DMA channels to Linux threads, and the asynDriver that interfaces FlexRIO resources with EPICS Process Variables (PV) [2].

In nuclear fusion plants, where this type of DAQ systems [3] performance instrumentation and control, usually acquire data at high rates (in this case up to MS/s), and send it to the Plasma Control System (PCS) where data received is processed and control loopback is applied [4]. PCS will determine which actions shall be applied to the Tokamak device for achieving long time plasma pulse without disruptions, and maximum efficiency. This fact implies that data must be acquired with the necessary level of integrity, continuity, availability and accuracy. Besides sending data to the PCS, DAQ systems are in charge of sending data to archiving systems. All acquired data from experiments must be storage for off-line analysis. All these tasks must be absolutely synchronized using synchronization mechanisms as IEEE1588-2008 that will provide Real-Time clock for timestamping [4].

Manuscript received May 28, 2016.

D. Sanz, A. Bustos, J, Autrán, M. Fernandez and S. Urueña are with GMV, P.T.M. Tres Cantos, 28701, Madrid, Spain (telephone: +34918072100/2137, e-mail: diegos@gmv.com).

M. Ruiz and S. Esquembri are with the Instrumentation and Applied Acoustic Research Group, Technical University of Madrid (UPM), Spain (telephone: +34913364696 Ext: 29403, e-mail: mariano.ruiz@upm.es).

II. ANALYSIS METHODOLOGY

Dependability analysis it is an unavoidable task for critical systems from conception to disposal stages. RAM analysis implies the use of analysis strategies like Hazard analysis, Fault Tree Analysis (FTA), Failure Mode and Effects Analysis (FMEA) and Failure Modes Effects and Criticality Analysis (FMECA) etc., being all they common but not exclusive between them.

The entry point to RAM analysis consists in a functional analysis (see fig. 1) of the system presented (DAQ system supported by IRIO EPICS Device Support), that together with the whole of the analysis proposed: FTA, FMECA, etc., an architectural system solution is obtained. After criticality analysis, system could require different critical levels of implementation, in such a case, virtualization techniques are evaluated for avoiding costs. This analysis and results will be extensively assessed in the paper.

IRIO driver provides full control access to the FlexRIO devices through EPICS Channel Access, besides it shall provide acquired data management for sending it to: an archiving system for off-line analysis; data sending to the plasma control system, and data sending to EPICS. The functional system defines inputs and outputs that reflects the interaction with the system. Inputs corresponds to every action that EPICS driver gets into from the user or software element, and outputs are the results available for the user (EPICS monitoring) or other software process like data archiving, or data processing by plasma control system.

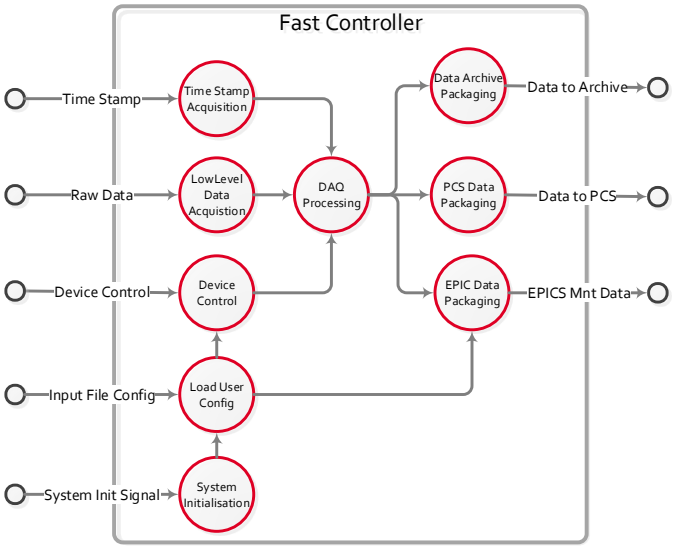

Fig. 1. Functional diagram representing the whole system under analysis.

Once dependability analysis is accomplished, there is a software integrity qualification criteria that should be reached for mitigating risks found on RAM analysis by the required software assurance development level. Following ITER guidelines recommendations [5], code implementation and deployment have been achieved. Inside the software development life cycle, software verification point process confirm if the software complies with all the initial requirement specifications determined at the initial stages. A set of software tools contributes to detect latent errors in the software, and we have chosen Polyspace Bug Finder and Polyspace Code Prover. Code Prover tool, detects and proves the absence of run-time errors in the source code using static analysis. The obtained results guarantee that 95,101% of the code is free of errors as memory corruption, etc. 4,11% of the software codification is free of errors if function input values are on expected range by the programmer, 0,78% is unreachable code, and 0,009% has errors. More detailed results and analysis will be presented in the paper.

III. SUMMARY AND CONCLUSIONS

As a summary and advancement of the paper results the following points will be studied:

- IRIO software tool RAM analysis to ensure the technical risk control and mitigation.
- Criticality analysis and assessment of mixed-critical systems for failure propagation mitigation by the usage of segregation strategies, such as virtualization techniques by using hypervisors.
- The IRIO Software Integrity Level (SWIL).
- Software verification methodology based on source code static analysis to reduce errors present in the final product.

This analysis will provide confident methodologies to take into account in future software implementations for minimizing costs and risks in such kind of nuclear environment.

REFERENCES

[1] Wallander, A. et al., “ITER instrumentation and control—Status and plans, Fusion,” Engineering and Design, Volume 85, Issues 3–4, July 2010, Pages 529-534.

[2] Sanz, D.; Ruiz, M.; Castro, R.; Vega, J.; Lopez, J.M.; Barrera, E.; Utzel, N.; Makijarvi, P., "Implementation of intelligent data acquisition systems for fusion experiment using EPICS and FlexRIO technology," in *Real Time Conference (RT), 2012 18th IEEE-NPSS* , vol., no., pp.1-8, 9-15 June 2012.

[3] Yang, C.; Zhang, M.; Zheng, W.; Yuan, T.; Zhuang, G., "Real-time data acquisition and processing system based on ITER Plant Fast Controller and FlexRIO FPGA," in *Real Time Conference (RT), 2014 19th IEEE-NPSS* , vol., no., pp.1-4, 26-30 May 2014

[4] K. Zagar, S. Hunt, P. Kolaric, R. Sabjan, A. Zagar, J. Dedic, Evaluation of high-performance network technologies for ITER, Fusion Engineering and Design, Volume 85, Issues 3–4, July 2010, Pages 557-560.

[5] Stepanov, D. “SEQA-45 - Software Engineering and Quality Assurance for CODAC,” ITER IDM ID 2NRS2K, Dec 2013, Available: http://static.iter.org/codac/pcdh7/folder%201/9-seqa-45_-_software_engineering_and_quali_2nrs2k_v3_2.pdf.