APPLICATIONS OF EXPERT SYSTEM TECHNOLOGY IN THE ATLAS TDAQ CONTROLS FRAMEWORK

Alina Corso-Radu, Raul Murillo Garcia
University of California, Department of Physics and Astronomy
4129 Frederick Reines Hall, Irvine, CA 92697-4575, U.S.A.

Andrei Kazarov
CERN, on leave from PNPI, St.Petersburg, Russian Federation

Giovanna Lehmann Miotto, Luca Magnoni, John Erik Sloper
CERN, Genève 23, Switzerland

Keywords: Controls, Expert-system, Knowledge, Verification, Testing, Recovery, ATLAS.

Abstract: The ATLAS Trigger-DAQ system is composed of O(10000) of applications running ~1500 computers distributed over a network. To maximise the experiment run efficiency, the Trigger-DAQ control system includes advanced verification, diagnostics and complex dynamic error recovery tools, based on an expert system. The error recovery (ER) system is responsible for analysing and recovering from a variety of errors, both software and hardware, without stopping the data-gathering operations. The verification framework allows users to develop and configure tests for any component in the system with different levels of complexity. It can be used as a standalone test facility during the general TDAQ initialization procedure, and for diagnosing the problems which may occur at run time. A key role in both recovery and verification frameworks is played by the rule-based expert system, which is also known as a knowledge-based system, to analyse errors and decide on appropriate recovery actions. The system is composed of a dynamic set of rules that describe the TDAQ system behaviour and by an inference engine that takes decisions on which actions to perform. The system is currently used on a daily basis for the operation of the ATLAS experiment. The paper describes the architecture and implementation of the TDAQ error-recovery system and verification framework with emphasis on the latest developments and experience gained over the first LHC beam runs.

1 INTRODUCTION

The paper describes the verification, diagnostics and error-recovery components of the Controls system of the ATLAS Trigger-DAQ (TDAQ) system. The ATLAS is one of the experiments built to study the proton-proton collisions delivered by the Large Hadron Collider (LHC) at CERN.

The main aim of verification, diagnostics and error-recovery components, that are all based on the same rule based expert system technology, is to maximize data taking efficiency by helping the operator to control and diagnose the TDAQ smoothly, through the use of the expertise provided by the system developers.

First, the overview of the use of the expert system technology in the scope of TDAQ Controls system is given, including motivations, architecture overview and some details of implementation. Then some components are described in more detail, focusing on their recent developments and use in real experiment environment during first data-taking periods of LHC machine.

Corso-Radu A., Garcia R., Kazarov A., Lehmann Miotto G., Magnoni L. and Sloper J. (2010).
APPLICATIONS OF EXPERT SYSTEM TECHNOLOGY IN THE ATLAS TDAQ CONTROLS FRAMEWORK.
In *Proceedings of the 5th International Conference on Software and Data Technologies*, pages 512-516
DOI: 10.5220/0003008605120516

2 EXPERT SYSTEM IN TDAQ CONTROLS FRAMEWORK

2.1 Motivations for the Application of Expert System in TDAQ Controls System

Requirements

At present the TDAQ system of the ATLAS experiment (Atlas, 2008) is a very complex distributed computing system, composed of O(10000) of applications running on more than 1500 computers distributed over a network (Atlas, 2003). The system is scheduled to increase further to ~ 50 thousand applications on 3000 computers to sustain the foreseen event rates at the nominal luminosity of the LHC.

The TDAQ Controls system (Lehmann, 2009) has to guarantee the smooth and synchronous operations of all TDAQ components and has to provide the means to minimize the downtime of the system caused by runtime failures, which are inevitable for a system of such scale and complexity.

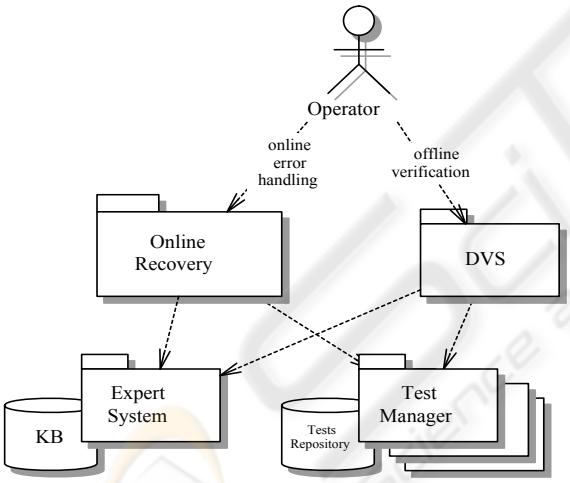

Figure 1: Use of Expert System by TDAQ Controls components.

Architecture

To accomplish its objective, the Controls system includes some high-level components which are responsible for the automation of system verification, diagnostics of failures and recovery procedures (Figure 1): DVS (Diagnostics and Verification System) (Kazarov, 2007) and Online Recovery (Sloper, 2008). These components are built on top of a common technology of a forward-chaining Expert System (ES) framework, that allows to program the behaviour of a system in terms of "*if-then*" rules and to easily extend or modify the knowledge base (KB).

The Online Recovery is composed of a global server, that handles errors that have a system-wide impact, and by local units to handle errors that can be dealt with at a subsystem level, that is, errors that do not have an immediate effect on the rest of the system.

For execution of atomic tests from a Test Repository, the Test Manager service is used, which in turn utilizes other Controls services that are not covered in this paper.

TDAQ Controls tools (Liko, 2004) are designed using a *framework* approach, thus giving the possibility to extend its basic functionality by filling the provided framework with subsystem-specific tests and rules.

Given the lifetime of the experiment which is more then 10 years, a mean of storing and reusing the expertise of the system developers, and also the operational experience with the system is important. This is achieved by storing the verification, error diagnostics and recovery procedures (in the form of rules in the knowledge base (KB)) and by using this knowledge on the running system through years. Also, the system components tests developed by subsystem experts are stored in a Test Repository and thus made available for verification and diagnostics procedures performed by TDAQ operators as schematically show on Figure 2 (more details given in chapter 3**Error! Reference source not found.**.

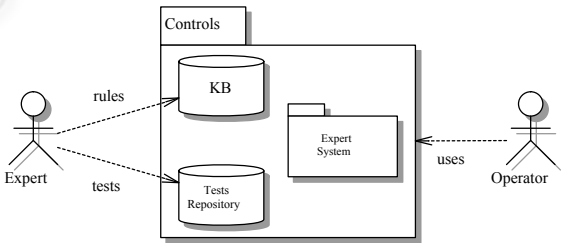

Figure 2: Storing the developers expertise in TDAQ Controls system.

2.2 CLIPS Expert System

The CLIPS expert system toolkit (CLIPS, 2010) was chosen for the implementation of the knowledge-based components of the Controls system. Its principal features and capabilities are presented below:

–originally developed in NASA

–available as open source

–can be used as a standalone application or as embeddable (C-source) library, which allows integration of ES functionality in the existing applications

–implements a number of different programming paradigms:

–"if-then" rules and a forward-chaining inference engine

–object-oriented constructs ("COOL" language)

–traditional algorithmic constructs (functions, cycles etc.)

One of the key (if not unique) features of CLIPS is the possibility to describe the architecture of your application using an object-oriented approach (including classes, multiple inheritance, methods) and to develop "generic" rules for classes of objects which will be then applied to a particular set of objects according to a particular configuration without any modification of the pre-loaded rules. This model fits very well with the TDAQ configuration object-oriented approach and gives a powerful tool for developing rule-based control applications in TDAQ framework.

2.3 Integration of CLIPS in TDAQ Controls Framework

Figure 3 presents shows how the expert s is integrated and interacts with the main TDAQ Control services.

The whole TDAQ system is fully described and configured via the Configuration Database Service according to a pre-defined configuration object-oriented schema. The corresponding schema developed in the CLIPS object language COOL is loaded to all instances of the ES. The actual set of objects for a particular configuration is loaded into the ES as a class hierarchy representing proxies of the applications and the hardware in the system. These objects, together with information gathered from the Monitoring service and test results can then be used in the ES engine to match the loaded rules. The CLIPS inference engine adopts a forward chaining approach: the agenda is how CLIPS keeps track of which rules are to be executed next run, and a rule is added to the agenda when all its conditions, given the status of objects in the systems, are satisfied.

CLIPS parses the knowledge base at run time. This allows to easily customize the behaviour of recovery procedures supplying different set of rules as arguments to the recovery applications. In a complex and dynamic framework such as the TDAQ system it is very difficult to detect a priori all the different errors that might occur and what appropriate actions should be taken. It is therefore very important that the expert system can be easily changed and customized as more data is gathered and a better understanding of the system is gained.

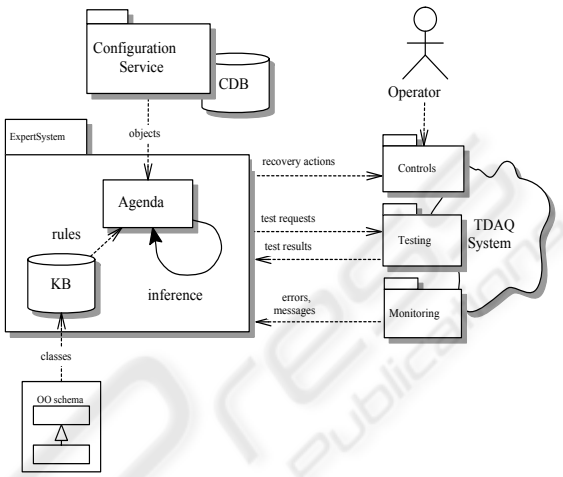

Figure 3: Integration of Expert System in Controls framework.

3 DIAGNOSTICS AND VERIFICATION SYSTEM (DVS)

DVS is a framework used to assess the correct functionality of the system, to detect and diagnose eventual problems. DVS allows the configuration of one or several tests for any component in the system by means of a configuration database. The system and the testing results can be viewed in a tree like structure using a user friendly graphical user interface.

3.1 Tests Configuration and Access Library

Having a possibility to examine and verify the functionality of any component in the system is essential for the diagnostic, i.e. for understanding the causes of a problem and for identifying the faulty components – either during the error recovery procedures on the running system or in pre-initialization phase.

A test is a binary running on a particular host in a system. It verifies a particular functionality of a TDAQ component and returns a single result value: PASSED, FAILED, UNRESOLVED, TIMEOUT and some text output. For a single component a

number of tests can be associated which can be organized in sequences, executed synchronously or asynchronously. Tests and their relationships are fully described in the configuration database as a part of a particular TDAQ configuration. The following attributes are available for test description: *parameters, host, dependencies, time-outs, scope, complexity* and *mask*. Any test can be associated either to a particular object in the configuration database or to all objects of a particular type. In the latter case the test's parameters and host can be parametrized by attribute values of the concrete objects when the test is launched.

In the spirit of the TDAQ Control services, the Tests Configuration schema and data access library allows ATLAS subsystem experts to define more advanced tests for detector-specific h/w and s/w components and thus to make the verification functionality available to other Control services, like Run Control and DVS.

3.2 DVS Core

DVS extends the functionality of the Test Manager by presenting a particular TDAQ configuration in a form of testable tree-like structure, allowing to automate testing and diagnostics of problems at different levels of the configuration. Its core components are the Test Repository database describing all tests and the expert system filled with some knowledge by an expert. The functionality is available to end-users via C++ API and via a user-friendly GUI.

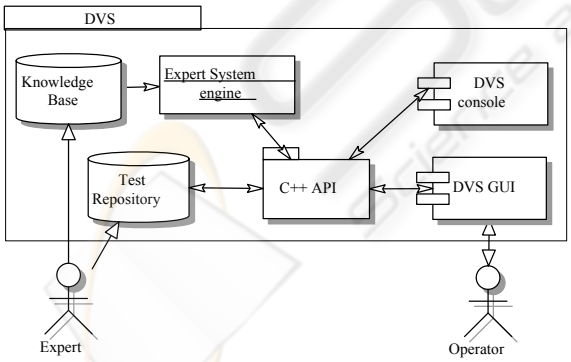

Figure 4: Architecture of DVS.

3.3 DVS Graphical User Interface

A new version of the DVS graphical user interface has been developed recently. The previous version based on Java was using JNI (Java Native Interface) for the communication with the DVS core library. This choice had proved to be not scalable enough and not easy to debug and maintain. A Qt4 based implementation replaced the previous GUI.

Figure 5 shows the main panel of the DVS GUI. In its left side the tree of testable components is displayed. The user can select a single component or a group in the tree and run all defined tests by clicking a single button. As tests finish, the components icons change colour reflecting the result. On the right hand side the test results, diagnosis and recovery advice are presented.

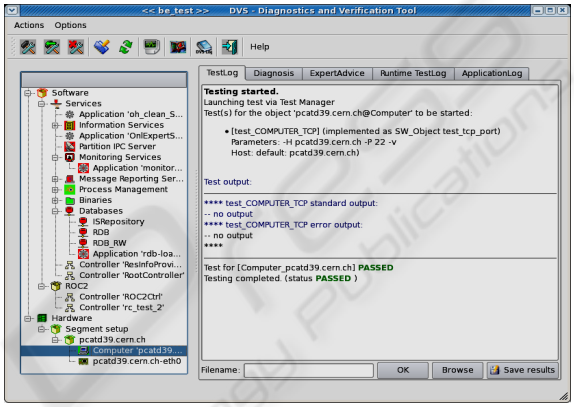

Figure 5: DVS Graphical User Interface – main panel.

Other features were implemented following the constant user feedback received: log file browser for accessing log files produced by the TDAQ applications running in a distributed environment, test scope to prevent destructive tests to be executed during data-taking sessions, test runtime output for long-running tests.

4 RECOVERY SCENARIOS IMPLEMENTED WITH ONLINE RECOVERY AND RUN CONTROL FRAMEWORK

This section introduces a simple recovery scenarios. The rule in Fig. 6 describes a general recovery mechanism in case an application has died. The left hand side (LHS) of the rule contains the conditions to identify the error situation, the right hand side (RHS) lists the actions to be taken.

In the TDAQ framework applications are arranged in a tree structure, with a controller responsible for a segment, a subset of the whole system. Following this recovery rule, when an application that satisfies the conditions described in the LHS the expert system takes care of notifying

the associated controller to ignore the application from now on.

```
if
system state is running, and
application App1 status is absent, and
application App1 has supervisor S1, and
application App1 membership in
then
notify S1 ignore App1
notify controller ignore App1
set membership App1 out
```

Figure 6: Sample recovery rule.

This approach can be applied to more advanced scenarios. In the ATLAS TDAQ framework, the expert system server holds the main responsibility of dealing with detector frontend failures. There is an automatic recovery procedure which allows for Read-Out Drivers (RODs) that are permanently busy or otherwise faulty to be recovered without stopping the run. The expert system drives the recovery operations, detecting the error conditions and holding the data acquisition triggers for the time needed to disable the failing component and to restore safe data taking operations.

5 CONCLUSIONS

We described the architecture and the implementation of the diagnostics and verification tool and the error-recovery mechanism, two components of the ATLAS Trigger-DAQ suite. The aim of these components is to increase as maximum as possible the experiment data-taking efficiency, by helping the operator to spot problems easily without the need to stop the data-taking process. The common feature of the two components is the usage of the rule-based expert system technology that allows to program the behaviour of a system in terms of "*if-then*" rules and to easily extend or modify the knowledge base. We describe the system improvements after the experience gained during the first LHC data-taking periods.

REFERENCES

ATLAS Collaboration, 2008. The ATLAS Experiment at the CERN Large Hadron Collider. *Journal of Instrumentation,* 3 (2008) S08003.

ATLAS Collaboration, 2003. ATLAS High-Level Trigger, Data Acquisition and Controls Technical Design Report. CERN Technical Design Report. Available at: http://cdsweb.cern.ch/record/616089. [Accessed 30 March 2010].

CLIPS, 2010. *CLIPS: A Tool for Building Expert Systems*. Available at : http://www.ghg.net/clips/CLIPS.html. [Accessed 30 March 2010].

Kazarov, A., Corso-Radu, A., Lehmann Miotto, G., Sloper, J. E. & Ryabov, Y., 2007. A rule-based verification and control framework in ATLAS Trigger-DAQ. *IEEE Transaction on Nuclear Science.* 54 (2007) 604-608.

Lehmann, G. et al., 2009. Configuration and Control of the ATLAS Trigger and Data Acquisition. *Proceedings of The 1st International Conference On Technology And Instrumentation In Particle Physics.* Tsukuba, Ibaraki, Japan, 12-17 Mar 2009.

Liko, D. et al., 2004. Control in the ATLAS TDAQ System. *Computing in High Energy Physics and Nuclear Physics.* Interlaken, Switzerland, 27 Sep - 1 Oct 2004.

Sloper, J. E., Lehmann Miotto, G. & Hines, E., 2008. Dynamic Error Recovery in the ATLAS TDAQ System. *IEEE Transaction on Nuclear Science.* 55 (2008) 405-410.