ROD, ROS and Dataflow Display tool – Specifications

v0.2

May 18th, 2009

Abstract

This document presents specifications for a tool that would allow to graphically represent the status of the ATLAS data flow and error conditions, starting from the RODs and ROS, all the way to the SFOs. To provide a quick overview, but at the same time allowing to cope with the complexity of the system, the tool should provide several levels of display. One should be able to navigate from the schematic top level display, down to more specific areas, showing the system with higher granularity. The display should also show the propagation of the backpressure and be able to derive its conditions from multiple sources of information, such as the Information Server, the Message Reporting System and other tools such as the DAQ Expert System.

1. Introduction

The purpose of the proposed tool ("ROD, ROS and Dataflow Display", or "R2D2" for short) is to give an overview of the ATLAS data flow and error status in a single panel that would display the main components of the ATLAS DAQ. It should show the different components in a way that reflects their logical relationships. It should show the RODs and ROSs, the Data Flow Manager, the L2 and L2 supervisors, the pROS, the Event Builder, the Event Filter, the SFOs, the ROI builder and the sources of ROIs. Since many of these components are rather complicated, (eg. the L2 and the Event Filter farms), the top level of the display must show a simplified view of the system with the above elements only. It must also allow to show their constituents, such as farm nodes or individual ROSes after some navigation, via graphical interaction with the display.

The error conditions should be propagated up to the top level display. For instance an error condition in one of the ROBins should be propagated to the top level panel, by showing the ROS top level representation in RED.

The R2D2 tool must have built-in the logical relationship of the various components of the system. Most of the information necessary to flag error conditions is already available in the Information Server (IS) or via the Message Reporting System (MRS).

In many cases, the errors or warnings are not raised by the faulty element/application, but by the first element/application that tries to interact with the faulty element. This is difficult to avoid without a second level of logic, and this is one of the main purpose of the proposed tool: to help identify quickly the source of the error condition. The underlying logic of the display should contain the expert knowledge that tells which component of the DAQ system should be flagged in error. Finally the R2D2 tool should be designed in such a way that it can either rely on its own simple algorithms to decide error conditions and should also be able to rely on an external expert system.

2. Graphical Display

The quality of the graphics should be compatible with the displays in the ATLAS control room, both the monitors and the wall display. The choice of color should work well with the projectors in the ATLAS control room, as this is foreseen to become one of the main displays in the control room. In Fig. 1 we present a prototype for the top level display. It shows the main components of the ATLAS data and trigger flow and their relationship. The display is composed of 2 types of elements: boxes and lines. Each box and each line must be able to change color.

For the boxes we foresee 5 different colors:

1. RED when there is a critical error condition that will block the data or trigger flow,

2. ORANGE when a condition is not critical but requires attention,
3. GREEN when there is no error condition,
4. GREY when the element is not active in the current ATLAS partition
5. WHITE when there is no associated condition.

We foresee that the lines can turn RED to display backpressure, or XOFF condition. Lines should be BLACK otherwise.

In order to be able to show the entire ATLAS data flow on a single panel, this representation is necessarily simplified. Most of the boxes represent in a simplified way their constituents, like for instance the Event Filter box, which represents the entire Event Filter farm.

The logic of the propagation of the state of the children to the parent box has to be defined carefully and needs to be dependent on the box. For instance in the ROS sector, a RED condition should propagate to the parent. On the other hand for the L2 Farm sector, a single L2PU can be in RED, but should not necessarily cause the top level L2 Farm box to turn RED, as the functionality of the L2 Farm would not be impaired by a single L2PU in error.

As a general rule, the color of the top level box should reflect the highest priority condition of any of its components. The priority of the conditions is given by the itemized list above, with RED condition being the highest priority condition and WHITE being the lowest one. A possible exception to this rule is the propagation of the RED condition, which in some specific case should lead to ORANGE or even GREEN of the parent box.

Finally the boxes should be able to display a few key numbers.

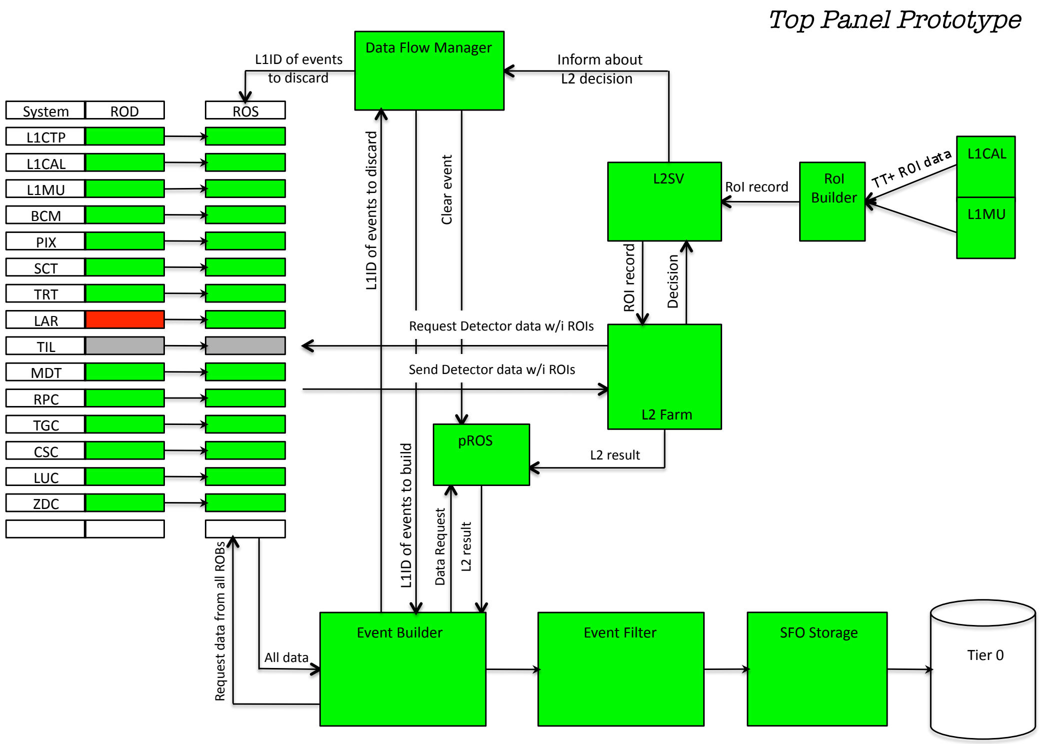

Fig. 1: *Prototype of top level display.*

3. Navigability

Many of the boxes on the top level panel are necessarily simplified representations. Therefore some of the boxes should be clickable and provide additional graphical display to allow to navigate down to the problematic component. The elements that are foreseen to be navigable are: the ROSes, the Event Builder, the Event Filter, the SFO, the L2Farm, the L2SV, and possibly the pROS. In some cases there might be several level of navigation like for instance foreseen for the ROS sector as illustrated in Fig.2.

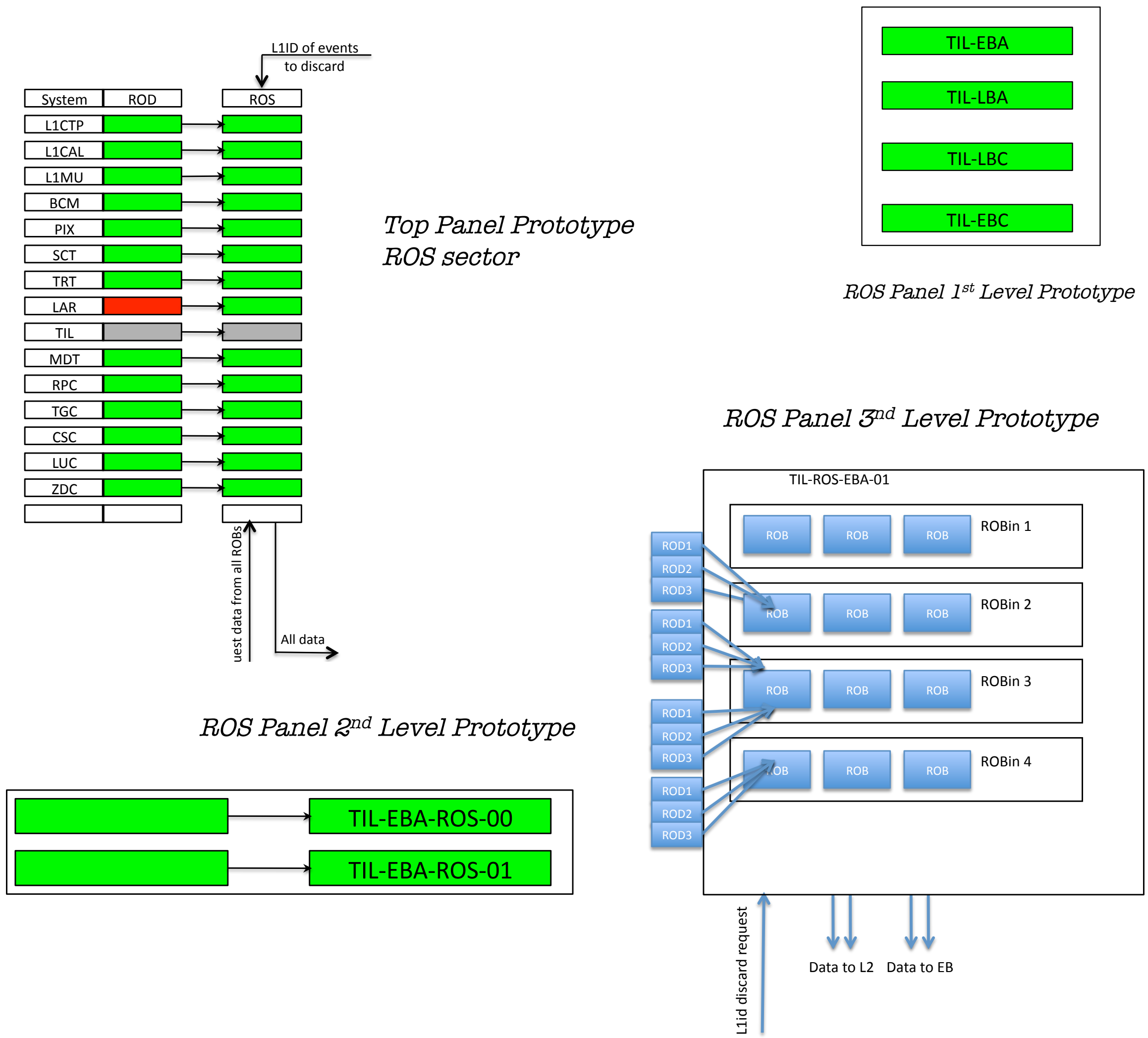

Fig.2: *Illustration of the different levels of navigation in the ROS sector and the corresponding panels. The top level panel (top left) shows the condition of each detector. By clicking on a sub-detector, one gets the top right panel, with the status per TTC-partition. By clicking on the a particular TTC partition, one gets a display of all the ROSes inside a TTC partition (bottom left). Finally by clicking on a particular ROS, one gets a display of the specific ROS, with the status of each ROB and ROBin card. The tool needs access to the Online Configuration Database (OKS) in order to access the mapping between the ROBs and the detector RODs.*

4. Interface with Other Panels

The R2D2 tool should be able to call an already existing panel. If there is an already existing panel allowing to see the details concerning for instance the SFOs, then upon clicking the SFO top level box one should get the already existing SFO panel.

5. Sources of Error Conditions

The R2D2 tool should be able to display error conditions extracted from several sources. The following sources of information should be used:

1. The information server of the ATLAS partition. The variables from the information server can be either from general TDAQ components or quantities published for example by the sub-detectors RODs.
2. The Message Reporting System (MRS)
3. The ATLAS DAQ expert system.

The tool should also know about the mapping between the RODs and the ROSes and will therefore need to access the online configuration database (OKS).

6. Logic Behind the Display

In order to determine whether one or several components are in error condition, one would ideally like to combined all possible sources of information. This includes variables published in the Information Server, but also error/warning messages generated by C++ code running anywhere in the system and reported to MRS. Therefore the R2D2 tool should be able to subscribe to specific types of error/warning messages seen by MRS. The decision to flag a certain element in red can be based on very simple criteria, for instance one number in IS smaller or higher than a certain threshold, but could also be a complicated combination of different variables or the result of another externally provided expert tool, for instance the ATLAS DAQ expert system.

While the graphical display should be rather stable once designed, the logic behind the display is likely to evolve and improve over time. Therefore it is recommended to make the graphical part of the tool independent from the logical part.

Any logic of internal to the tool should be configurable, with a configuration file. The criteria for the display conditions should not be hardcoded.

7. Performance and Environment

The R2D2 tool should run on linux, in the environment provided in Point 1 and be based on programming languages with long term support by the TDAQ group. The update rate of the display should be in the range 0.3 – 1 Hz.

Appendix A: Error conditions for the ROS, RODs, ROBs

We define a first list of simple criteria to define the conditions of the ROBs. The information in the table below can be found in IS, in the server 'DF'. For the ROBins the data is stored in structures of the type: DF.DF-ROS-LAR-EMBC-00.DataChannel9 (object of type RobinDataChannelInfo). The information from the DFM, needed for some of the criteria below are stored in the DFM object: DF.DFM-1. The specific thresholds, in the column value, should be configurable, eg. 0.2 for the pagesInUse or 33553432 for the difference in L1ID.

Name	**Value**	**STATUS**
rolXoffStat	1	RED
bufferFull	1	RED
rolDownStat	1	RED
pagesInUse	0.2* pagesInUse > pagesFree	ORANGE
fragstruncated	fragstruncated> 0.1 * fragsreceived	ORANGE
fragscorrupted	fragscorrupted> 0.1 * fragsreceived	ORANGE
lastl1id	|lastl1id-mostRecentId|>33 554 432	RED
numberOfNotDeleted	numberOfNotDeleted> 0.5 * DF.DFM-1.cleared events	ORANGE
numberOfNotDeleted	numberOfNotDeleted> 0.9 * DF.DFM-1.cleared events	RED

Table 1: *A list of simple criteria to give error and warning conditions in the R2D2 tool.*