International Journal of Modern Physics C

THE EMU MESSAGES DISPLAY FOR THE DELPHI EXPERIMENT AT LEP

TIZIANO ROVELLI
Physics Department, Bologna University and INFN,
Via Irnerio 46, I-40126 Bologna, Italy

Received (to be inserted
Revised by Publisher)

ABSTRACT

The DELPHI Data Acquisition and Control System uses the Error and Message handling Utility (EMU) as a standard distributed system which deals with messages generated by the controlling and monitoring processes. A tool which allows one to look at EMU messages as they are produced and to retrieve past messages is described. This utility, being based on a configuration file, can be used both by the central and individual detector operators. Versions exist with both MOTIF (X-windows) and MHI ("VT100") user interfaces.

Keywords: online, slow control, monitoring.

1. Introduction

The increasing complexity of the multipurpose detectors for high energy physics experiments demands more and more emphasis on their control and on proper diagnostic tools. The DELPHI[1] apparatus, located at the intersection 8 of the CERN LEP accelerator, is a composite 4π detector. It consists of many subdetectors able to give three-dimensional information with high granularity, for particle tracking, identification, calorimetry, and muon detection. In the DELPHI Online environment, the Error and Message handling Utility (EMU) deals with messages generated by controlling and monitoring processes. This paper describes a tool which allows one to look at EMU messages as they are produced and to retrieve past messages. It is running, at present:

- centrally on the Slow Control (SC) VAXStation;
- centrally on the Central Partition (CP) VAXStation;
- locally on the individual detectors VAXStations.

On the SC VAXStation every detector has its own window where safety or diagnostic messages of general interest concerning the detector appear.

On the CP VAXStation messages relating to the data acquisition and to the Trigger Partition (TP) are displayed. A separate window has been created for communication with LEP.
Local displays include messages that are useful for the detector operator. They are selected and sent in appropriate windows (data acquisition, gas, high voltage, etc.). In each configuration alarm conditions appear in dedicated windows.

The next section summarizes EMU and the controlling and monitoring processes, showing how they interact with it. The DELPHI slow control and gas system are recalled in Section 3. Section 4 describes the connections to the Fastbus and data flow. Section 5 and 6 describe the EMU message display utility.

2. EMU

EMU is a run-time messages reporting system. A full description on how to use EMU in a generic environment and a general exposition of the principles and conventions adopted to use EMU in the DELPHI Online Equipment Computer (EC) can be found in Ref 2,3. The message pipeline implemented in EMU for the DELPHI Online, can be summarized as follows:

(i) Injection: As a general rule all injected monitoring and controlling programs call EMU message injection routines to send EMU messages. Only the name of the message and, if necessary, some parameters values are required in these calls. These injection routines call the Decoding. The only exception is represented by the messages coming from Fastbus Intersegment Processors (FIPs), which are directly sent into the EMU Router (Section 2.).

(ii) Decoding: The EMU Decoder process looks into an existing local database (message file), and retrieves the message text which correspond to the injected message name. The Decoder builds the final message adding the appropriate parameters and properties (strings that characterize the messages[2]), and then sends it to the Router.

(iii) Routing: The EMU Router process, following previously built database (route map file) instructions, transfers the message to the proper destination, anywhere in the DELPHI Online Cluster.

(iv) Destinations: Possible EMU message destinations are physical devices or other programs. The message can be sent either in formatted or unformatted mode. Actually, in the Online Cluster, the destinations are:

   (a) Log-files: All messages are routed into log-files without any formatting. These log-files are mainly local log-files where the messages useful for detector purposes are routed (such as detector hardware and software debugging) and central operator log-files where general utility messages are routed (such as alarms, general safety, detector voltages, and gas messages).

   (b) Terminals: Logged-on terminals are alerted whenever an alarm message is injected.

   (c) Programs: EMU messages can be sent to other programs. In the Online environment this facility is used to forward alarm EMU messages to other processes which communicate with GSS (see below), beep selected detector experts, or broadcast to selected logged-on users.

3. DELPHI Slow Control and Gas System

The DELPHI SC system[4] is designed to monitor the apparatus, to allow an operator to control the experiment, to take automatic action if required in case of an alarm, and to record onto the Delphi database slowly varying values of parameters required for data analysis.

At the lowest level the system consists of G64 crates[5] which act as the interface between the hardware and the VAX-based Elementary Processes (EPs).

An EP, running as a batch job in an EC, controls and monitors one subsystem of the apparatus (high voltage, temperature, etc.). Communication between G64 and VAX is based on Remote Procedure Calls[6] (RPCs), which use CATS and OSI standards[7] for communication over Ethernet.

Being easily configurable, all software, in spite of the wide hardware variety, is highly standardized in the majority of subsystems. Interaction with the EPs is possible via SMI[8] and HIPE.[9] One of the tasks of the EP is to inform detector and central operators about changes in the monitored hardware. This is implemented by sending messages to EMU.

The general safety of the LEP point 8 apparatus is monitored by an independent system provided by the GSS (General Surveillance and Security system) group of CERN ECP. Communication between DELPHI and the GSS system is implemented via two programs that send and receive EMU messages, informing GSS of detected alarms, and perhaps forcing a GSS action (such as beeping the SLIMOS[a]).

The Delphi Gas System[10] is independent of the Delphi SC. All subsystems within the gas system are autonomous in order to guarantee safe operation in case of network problems. The gas system is linked with the SC system via SMI and an alarm server in the VAX-based Gas Interface. This server relies on EMU messages to inform both the operator and the GSS system (which will perform hardware action for certain alarms). In this area, the EMU system is an essential part of the safety structure.

4. Fastbus Readout and Data Flow

Error messages can be generated by all processes running in the fastbus (FB) processors (FIPs). Each time a process detects an error it calls a library[11] which sends the error to a local error server (`Error_Server`).

On each FIP a single error server runs collecting the errors detected by all the processes running on the same FIP. Whenever receives an error, it stores it in a log file local to the FIP, and sends a message over Ethernet to a special server (`Emu_FbServer`) on the SC VAXStation.

A Single EMU FB server receives errors from all the DELPHI FIPs' error servers.

[a] The person from the detector group who takes care of the safety.

On receipt of an error message the EMU FB server will append all the necessary EMU properties to the error, and inject it directly in the EMU router mailbox.

The data flow[13] deals with the data collection of each partition from Fastbus and controls the data until its final destination (mass storage and on-line monitoring). The data flow has been built using Model Buffer Manager[14] (MBM), that is a part of the MODEL package having as main purpose the control of data acquisition data flow. Actually the central data logger records the data on disk, and an asynchronous copy procedure running as a permanent task copies the data from disk to tape. Both the data flow processes and the asynchronous tape copy procedure send EMU messages concerning their actions and status.

5. EMU Messages Display Utility (EMDU)

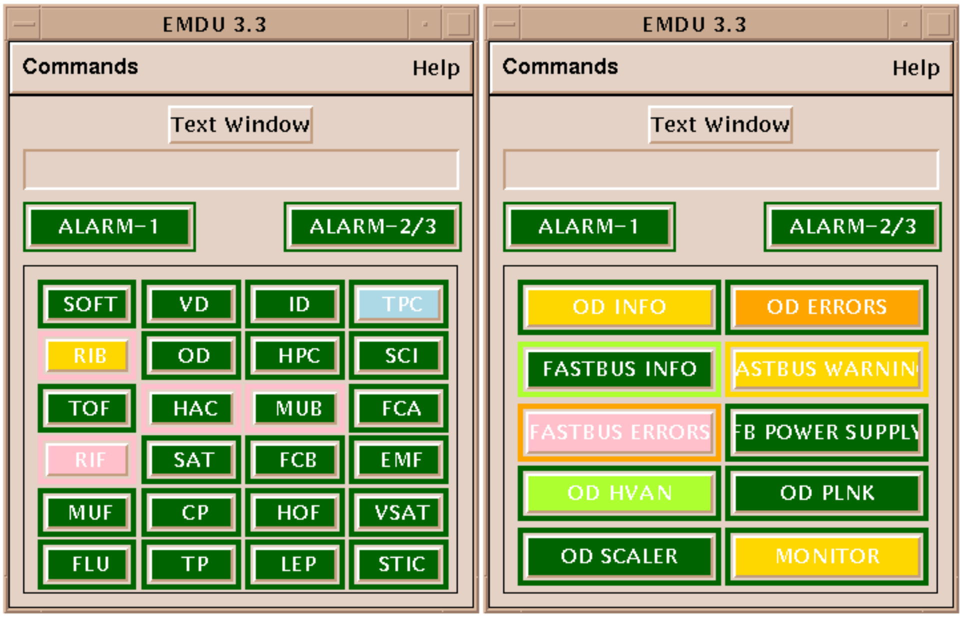

Fig. 1. EMDU configuration for Central Slow Control
and for a detector (outer detector)

EMDU has been developed in the DELPHI Online environment to provide a tool that has, as its main task, the capability to get, select, and show the incoming EMU messages (Fig. 1.).

EMDU reads new messages from a selected log-file (EMU write messages in a number of log-files) waiting, at the end-of-file, for a log-file update.[15] This solution solves limitations of the previous programs allowing different EMDU programs to run in every DELPHI Online Cluster node, and different EMDU programs to receive the same EMU message (pointing on the same log-file).

The user interface was built with MOTIF[16] and with the Xt Intrinsic - a set of routines that provide access to lower levels of the X Window System.[17] In addition a version of the EMDU program using MHI[18] as user interface exists and will be described shortly later (section 6).

5.1. The Configuration file

When the EMDU program is starting up, it will search for the configuration file. This file is composed of a set of commands, with its parameters and qualifiers, created according to the Command Definition Utility[19] (CDU) rules. EMDU parses these commands using the Command Language[19] (CLI) routines.

A command defines the way of starting, with a choice between the following possibilities:

- The program will search for the EMU messages (in the log-file specified with the appropriate command), and will take into account those which have been injected between an initial time, specified as a command qualifier, and the time considered present time by the running program (reaching the log-file EOF). These messages will be selected and shown in the appropriate window, according to the specifications of the dedicated command. Then it will wait for new messages.
- The program will reach the EOF of the log-file specified, and it will wait for incoming messages.
- The program will search for the messages between an initial time and a final time, both specified as command qualifiers. All the incoming messages will be ignored.
- The program will show a selection box where the previous options can be chosen interactively. Then, depending on the choice, different boxes will appear to select, if necessary, the log-file to scan and the time period(s) to be searched for past messages, so allowing to take over the log-file specifications.

Another command defines the windows and the characteristic of each one. This command supports many qualifiers to select, according to the EMU conventions in the DELPHI environment,[3] the selection criteria of the messages and, being the EMU messages formed by different parts,[2] which parts of the message should be shown.

A command sets the log-file name from which the messages will be extracted.

The selected messages can be arranged, in every window, in two different ways:

- By time i.e. with the most recent messages first.
- By severity[b] i.e. with the highest severity (often the most important) first.

A command selects these different ways of messages sorting.

The last command allows to make use of the possibility to associate a timeout to each incoming message. Messages that were injected before the specified timeout

[b] The message severity is a EMU property defined in the DELPHI online environment to distinguish the different messages gravity (alarm, warning, information, etc.).

are not displayed. Different timeouts will be possible, depending on the gravity and the type[18] of the messages.

5.2. Window and Alarm Buttons

When a new message has arrived, and window selection criteria are satisfied, the user is notified via a color change (according to the message severity) of the corresponding button on the display. Clicking on this button, a separate window and two clocks on the left- and right-hand side of the 'Text Window' will appear. The window will show the message(s) and the clocks will indicate the injection time of the oldest, and of the latest message. Closing the window the border of the display button will change colour (referring to the message(s) with the highest severity) to remind the operator that there are some messages in the window.

5.3. Message Window and Text Window Button

Over the 'alarms' buttons there is a 'text' window, that may always be seen. In this window the messages that have to be visible to the operator such as EMDU status, results of operator actions, etc., will appear.

Many of these messages and others will be recorded on a special window, the 'text' window. This window will appear on the screen by clicking on the 'text' window button, the upper button in the EMDU. The 'text' window can be considered as a kind of EMDU run history.

5.4. Commands menu

Clicking over the word 'Command' gives the possibility to choose between several actions concerning windows, log-files, timeouts.
Some lower level menus allow to activate and deactivate windows, and to modify all the default or log-file set quantities, according to the DELPHI EMU standards. The possibility to change the log-file to search for different kinds of messages and the update of the log-file version to which the program is pointing[c] is foreseen. Also the timeouts associated to the messages can be enabled, disabled or modified by menu.

5.5. Help menu

The active buttons (and their border) will assume different colours when messages with different severities arrive (have been seen). By clicking on the word 'Help' it is possible to get a box containing all the possible message severities with the corresponding colour.
It is also possible to get the standard DECwindow help widget showing the main improvements and differences between the different program versions.
Moreover a detailed description of the program, especially for the users, can be

[c] A program log-file up to date attempt occurs automatically at the new file creation time.

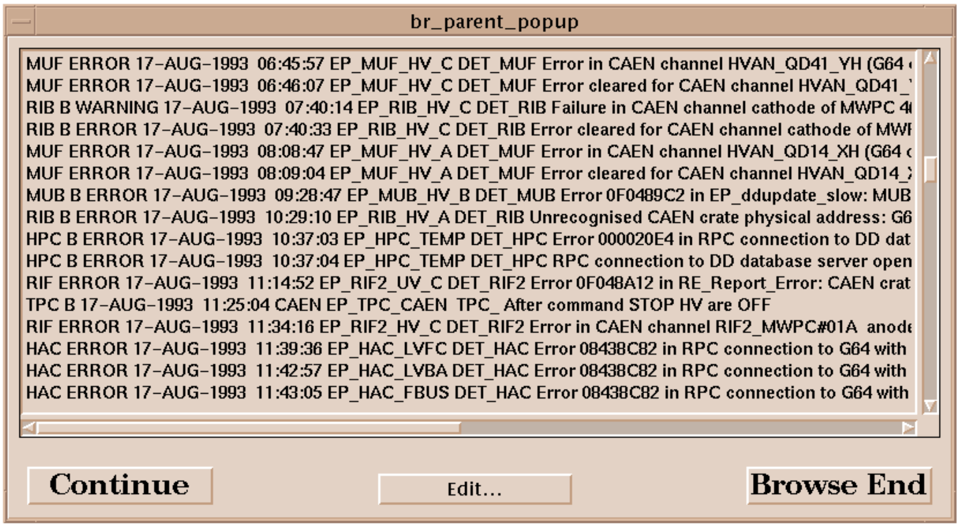

Fig. 2. Browse utility.

found in Ref. 20.

5.6 Browse Utility

This utility retrieves past EMU messages in any log-file. It will search for them in a given time interval, and according to given selection criteria. Selected messages will be shown according to given display options. While this utility is active all incoming EMU messages will be ignored.

The history browse utility can be invoked either when the program is starting up with the right command in the configuration file (Section 5.1.), or by selecting the appropriate actions on log-file in the Commands menu (Section 5.4.).
In the latter case, a file selection menu will appear and then, when the file has been selected, a widget with two adjustable times will pop up. These times represent the temporal interval between which the messages will be searched for. Any valid combination of absolute and relative[d] times are allowed. After the selection of the time in this case (or directly in the first one), the 'browse widget' (Fig. 2) will pop up. This widget contains a window where the messages appear, and three buttons which offer the following possibilities:

- To start the messages' search or to continue the messages' search when the searching is paused[e];
- To terminate the browse utility;
- To modify the criteria that define the messages selection and the pieces of the message that will be visualized.

[d] A time will be considered relative if the Month is set to 0.
[e] The messages search is paused after every hundred messages selected.

If all messages within the given time interval have been scanned, the search terminates, and only the 'Browse End' button will be active. If 'Browse End' is selected a selection box will pop up where one can choose whether to remain in the browse utility or to (re)start looking for the incoming EMU messages.

6. MHI User Interface.

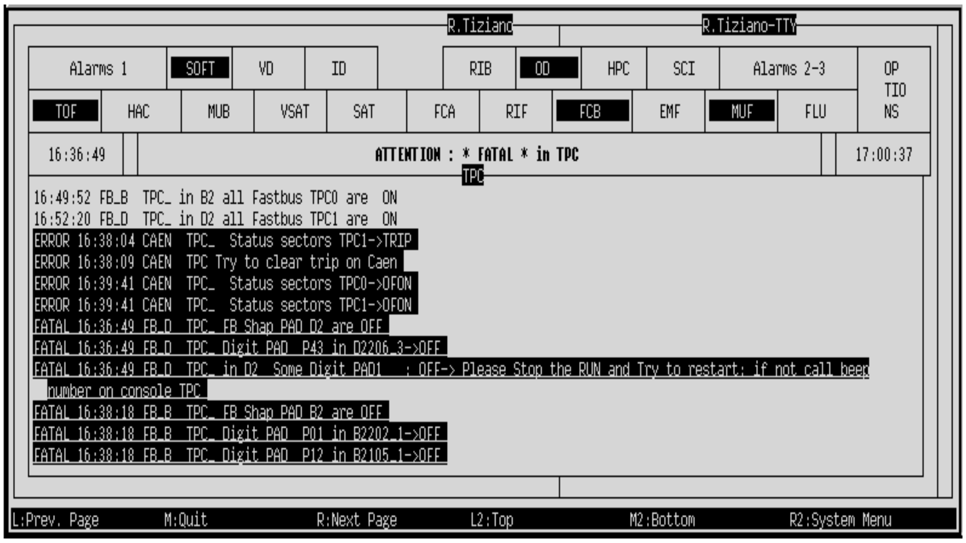

Fig. 3. Central Slow Control configuration
with MHI user interface.

Since an X terminal may not be available in all laboratories, a version of this package built with MHI is also available (Fig. 3.). With this user interface every alphanumeric terminal can be used. The program functionality is essentially the same of the X version with the exception of the browse and the help facility. When a new message arrives the window concerned starts to blink. Moving the cursor over the window and selecting it (the mouse is substituted by the arrow keys and the first row of the keypad) all messages will be shown into an extended window.

Conclusions

This product allows an effective on-line control of the different parts, both hardware and software, of a complex apparatus. By having sophisticated messages selection criteria that can be changed at any time, each part of the experiment can be monitored separately. Moreover the easy configurability and the possibility to search for past messages makes this product a powerful tool for discovering and debugging software faults and detector problems.

Acknowledgements

I would like to thank the members of the central DAS who have been very helpful with their advice in the realization of this tool. Special thanks go to T. Adye for his active collaboration in the initial phase of this project, and acknowledgements to M. Jonker for specific discussions in its development phase.
The author is grateful to R. Jones of the ECP group, and A. Aimar of the SL group for their high quality support. The author would also like to acknowledge the original contribution of K. Spang and the work of P. Burkimsher, author of EMU.
Finally, I wish to express my sincere thanks to F.L. Navarria for his support in this work.

References

1. P. Aarnio et al., Nucl. Instrum. Methods **A303** (1991) 233
2. *EMU The MODEL Error Message Utility,* CERN-DD, December 1990.
3. F. Harris and K. Spang, *The Use of EMU in the DELPHI Online System,* DELPHI 89–19 DAS–95, 16 March 1989.
4. Ph. Charpentier et al., *The Fastbus Intersegment Processor,* presented at the IEEE Conference, Washington, October 1986.
5. T. Adye, A. Augustinus, M. Dönszelmann, T. Rovelli, R.L. Sekulin and G. Smith, *The Slow Controls of the DELPHI Experiment at LEP,* Proceedings of the International Conference on Computing in High Energy Physics '92, Annecy, France, September 1992.
6. A. Ball, G. Smith and J. Zaslavsky, *Input Output Cards for use with the MAC Chassis,* EF, November 1985;
   J. Anthonioz-Blanc, S. Brobecker and J. Joosten, DD, May 1985.
7. T. Adye et al., *Online Communications in DELPHI Experiment,* Comp. Phys. Comm., **57** (1989) 466;
   T.J. Berners-Lee, *RPC User Manual,* CERN DD/OC, July 1989.
8. G. Heiman and T.J.Berners-Lee, *The CATS TS standard calling sequence v. 4.0a,* CERN/DD/CS, May 1989.
9. J. Barlow et al., *Run Control in MODEL: The State Manager,* IEEE Trans. Nucl. Sci., **36** (1989) 1549-1553.
10. M. Dönszelmann, *DELPHI HIPE System User Manual,* DELPHI 92–26 DAS 124, 2 March 1992.
11. A. Augustinus, *An overview of the remote procedure calls from the equipment computers,* January 1989.
    A. Augustinus, private communication.
12. C. Gaspar, private communication.
13. T. Adye et al., *Architecture and Performance of the DELPHI Data Acquisition and*

Control System DELPHI 91-93 DAS 113, Proceedings of the International Conference on Computing in High Energy Physics '91, Tsukuba, Japan, March 1991.

14. C. Boissat and P. Vande Vyvre, *The MODEL Buffer Manager User Guide,* CERN ECP/DS, January 1991.
15. T. Adye, *RMSTRIG routines,* written especially for the EMDU.
16. Open Software Foundation, *OSF/Motif Programmer's Guide,* Prentice Hall, Englewood Cliff, New Jersey 07632 (1990).
17. A. Nye and T. O'Reilly, *X Toolkit Intrinsics Programming Manual,* O'Reilly & Associates, Inc. (1990).
18. C. Boissat, R. Jones and G. Mornacchi, *Model Human Interface User's Guide,* OC/DD Group CERN (1989).
19. VMS Command Definition Utility Manual, June 1989.
    VMS Utility Routine Manual, June 1990. VMS Utility Routine Manual, June 1990.
20. T. Rovelli, *The EMU Messages Display Utility*, DELPHI 93-30 DAS 142, 24 March, 1993.