Designing a HEP Experiment Control System, lessons to be learned from 10 years evolution and operation of the Delphi experiment.

A. Augustinus[1], *Ph. Charpentier*[1], *M. Donszelmann*[1], *C. Gaspar*[1], *Ph. Gavillet*[1], *M. Jonker*[1], *T. Adye*[2], *B. Franek*[2], *R. Sekulin*[2], *G. Smith*[2]

[1] CERN, Geneva, Switzerland
[2] RAL, Didcot, England

Abstract

The DELPHI experiment, one of the four experiments on the LEP electron positron collider at CERN, started its operation in august 1989. Over the more than 10 years of operation the Experiment Control System has been evolving constantly.

We will describe the Delphi Experiment Control System and the evolution from a system with almost completely separated components (data acquisition, slow controls, trigger, ...) to a fully integrated Experiment Control System. We will explain how this was possible by taking advantage of new technologies in various domains, gain of expertise and a better understanding of the behaviour of the components. To conclude we will extract some lessons that we have learned in the evolution and the operation of the system and should be retained for the design of Experiment Control Systems for future big HEP experiments.

Keywords: Data Acquisition, Control System, Experiment Control

1 The DELPHI Experiment Control System

The DELPHI[1] Experiment Control System (ECS) consists of several subsystems, the two main components are the data acquisition system and the slow controls system. Other systems we consider part of the ECS are the control of the trigger system and the communication with the LEP accelerator. The readout electronics consists of over 250000 channels and the slow controls system handles several thousand channels of power supplies, temperature probes, etc.

- **The Data Acquisition System**

The DELPHI Data Acquisition System (DAS)[2] is organised as a set of 20 partitions (equivalent to a sub-detector, or part of it) and a so-called Central Partition dedicated to event building and event selection (3rd level trigger). Each partition can, at the front-end level, be split into several entities. Each partition can run independently, e.g. for debugging or calibration purposes.

The Fastbus electronics architecture is standard across all sub-detectors. There are about 180 Fastbus crates in the experiment. The control software of all levels runs in the same type of embedded processor: Fastbus Inter-segment Processor (FIP). DELPHI uses 72 of these processors running the OS9 operating system, and using TCP/IP to connect to the LAN.

- **The Slow Controls System**

The DELPHI Slow Controls System (SC)[3] is a highly modular and distributed system, providing an automated system for monitoring and controlling the technical aspects of the experiment. The standard front-end systems are CERN developed G64/MAC crates, using the G64 bus with M6809 processor and a limited set of I/O cards. The processor runs a remotely configurable program, without an operating system. Connection to the LAN is via a CERN developed Ethernet card and a CERN implementation of RPC. DELPHI uses over 50 of these crates, mainly for the sub-detector SC. Similar systems are used for control and monitoring of the DELPHI solenoid system and some general purpose tasks. A unified, independent gas system uses a further 30 G64 crates, but with a more powerful M68340 processor and the OS9 operating system. The gas and liquid systems of the RICH detectors are controlled and monitored via PLC (Siemens S5 series) systems.

For each sub-detector a set of supervisory processes monitors and controls their subsystems such as high voltage, temperature, etc. The sub-detectors SC can run independently, giving full control to an expert; or under central control giving, usually limited, control to central operators.

The DELPHI Trigger System[4], with its central and local level, and timing and decision part, is considered part of the ECS. It includes readout of numerous counters (monitoring background, trigger rates, etc.). Also the communication with the LEP accelerator, to retrieve machine parameters (status, etc.) and to send experiment parameters (background, etc.) is considered integral part of the ECS. Also the data quality monitoring could be considered as part of the ECS.

• **Software aspects**

Already in the design phase of the DELPHI ECS it was decided to use a single high level system: SMI (State Management Interface)[5]. SMI is based on the concept of *objects* that can be in one definite *state* and to which *action requests* to change this state can be sent at any moment. The behaviour of objects is described as a finite state machine through a dedicated language (SML). Closely related objects are grouped in so-called *SMI domains*.

Graphical user interfaces represent the state of objects to the operator. For a central operator all detailed information (e.g. from a sub-detector) is summarised in a single object, while an expert operator will have access to all objects. The operator uses these interfaces to give commands by sending *action requests*.

The communication with and between the SMI processes is done through a TCP/IP based communication package called Distributed Information Management (DIM), based on the publish/subscribe paradigm. This package is described elsewhere in these conference proceedings[6].

2 The Evolution of the DELPHI Experiment Control System

As mentioned before, the DELPHI experiment started taking data in 1989. At that moment an ECS did not yet exist, and all control was done via simple, mainly line mode, interfaces by sub-detector experts. SMI was only available in a very rudimentary form and synchronisation at that time was mainly done by sub-detector experts via telephone or intercom. Things have well evolved since[7].

• **Evolution**

The next step, as SMI became fully available, was to properly define SMI domains for DAS and SC. These domains fully describe the behaviour of a sub-detector DAS partition or of a sub-detector SC system. Also so-called central domains for DAS and SC were defined grouping all sub-detectors. This allowed central supervision and control of all sub-detectors from a single place and by a crew of only a few central operators.

At this moment communication with and inside SMI was revealed to be inflexible and non-scalable. To overcome these drawbacks the DIM communication package was developed. Because of well defined APIs, the replacement of the various coexisting means of communication with this DIM package was largely transparent for the users and led to a gain in performance and maintainability. DIM has now evolved into a universal communication package within the DELPHI experiment, fulfilling almost all our communication needs with great ease and satisfaction.

The trigger system, the LEP communication system and the data quality monitoring became more and more integrated in the ECS through the use of SMI and DIM.

During the lifetime of the experiment both hardware and software modifications/upgrades were made. For example the upgrade of the front-end controllers of the gas control system from the standard SC M6809 processor to a more powerful M68340 based CPU card was triggered by the obsolescence in the hardware and the development environment. Another modification was induced, half way through the lifetime of the experiment, by the migration from VAX to Alpha architecture of the experiments online computer cluster as certain software packages or interfaces were no longer supported.

• **Automation**

With the definition of all central SMI domains (DAS, SC, Trigger, LEP), we created a level of abstraction for the operators in hiding detailed information by summarising these details in states of abstract objects. Information from a diversity of sources is represented in a standard way and, standard commands allow to act on a variety of hardware. Having done that, it was fairly easy to carry the level of abstraction even further, and implement automation on top of this highest level. This saw the development of a so-called autopilot for the DAS, taking care of all standard operations and error recoveries that would otherwise be done by the operator.

The highest level of automation was done in interconnecting the various domains, to automate standard actions during the normal LEP cycle. This automation is defined in SML, in a new SMI domain called *Big Brother* that issues commands to and synchronises the various central domains. This automation has as its main advantage that a minimum of time is lost while waiting for subsystems to become ready, maximising efficiency. It has the additional advantage that well defined commands are issued at well defined moments, ensuring consistency.

The DELPHI experiment has operated in this manner, with only minor modifications to its ECS, for the last few years, with constantly increasing efficiency.

3 Future Experiment Control Systems

As a conlusion we list a collection of points that we think one should bear in mind when designing an Experiment Control System, based on our experience of running the DELPHI detector.

• **Partitioning and Central Control**

A well thought out partitioning of the system should evidently be a requirement for any ECS. There will always be a need to run part of the detector (e.g. one or more sub-detectors) in stand-alone mode without disturbing the central operation. While this is already very useful during the commissioning phase it will certainly be necessary during the operational phase for debugging and calibration purposes. Also when part of the experiment is out of operation the rest of the detector should operate unhindered. The need for centralised control is evident and a small crew of operators should be able to run the experiment. Operation should be highly automated with for example one single command being necessary to start a run.

• **Uniformity across subsystems**

As already is clear from the above, the various subsystems such as data acquisition, slow controls, trigger control, interface to external services such as the accelerator, infrastructure, should all be considered part of the ECS and wherever possible use the same software. This will ease integration and automation, an additional advantage is a reduction in maintenance efforts. An effort should be made to provide uniform user interfaces both across subsystems and across interface hierarchies i.e. central and local displays.

• **Uniformity across sub-detectors**

A difficult point, but one on which it is worth spending effort, is to aim as much as possible for standardised hardware and software across all sub-detectors. The obvious advantage is, for hardware, likely cost reduction at purchase and a certain cost reduction due to a smaller stock of spares. Also having to interface software only to a limited variety of hardware implies a reduction in effort. For software the obvious advantage is the easier maintenance, but probably also a gain of efficiency in the development phase (sub-detectors can share expertise). The current trend towards the use of commercial (or 'shareware' like Epics) packages naturally encourages this uniformity.

• **Strong and stable central support**

During the lifetime of the experiment we experienced a great benefit from having a strong and stable central support team. This team should provide the users (sub-detector experts) not only with strong guidelines but also with frameworks and tools in order to enforce uniformity across all systems. In the ideal case an expert should only have to configure (or exceptionally slightly modify) packages delivered by a central team. This implies that at a relatively early stage requirements should be collected from the users so that, instead of concentrating on solving their particular problem, they join effort at an early stage with the central team to devise common solutions for common problems thus avoiding duplication. A strong and stable central team together with the uniformity mentioned will also be able to solve more easily problems arising after the departure of the original experts, an unavoidable fact on the timescales of current HEP experiments.

• **Flexibility**

An ECS should never be considered as 'finished', it should be flexible enough to cope with the many changes that will happen over the lifetime of an experiment. These changes can be of a wide variety; they can be changes in the experiment itself such as addition or modification of a sub-detector, modification or upgrade of hardware, or changes induced by a better knowledge of the experiment such as different working points, change of operational procedures. They can be changes induced by modifications in the environment such as a new version of an operating system, or voluntary changes to profit from new technologies. This implies that already while designing an ECS this need for flexibility should be taken into account, it should be easy to modify/reconfigure (also by someone who is not the original author). In this case good documentation is essential.

• **Day-to-day operation**

During normal operation the experiment is supposed to be run by a small crew of not necessarily expert operators. The ECS should provide enough abstraction to allow non-expert operators to run the experiment. This implies that a lot of details should be hidden from the operator while still easily available for experts, and that information should be represented in a uniform way, with all sub-detectors appearing similar. Operators should not be inundated with alarm or error messages; effort should be put in filtering these messages to reduce the risk of missing important messages. Interaction with the experiment should be highly automated, and commands should, again, be uniform with the same commands to do similar things on different sub-detectors. Operators should have adequate documentation describing how to handle anomalies and should be properly trained for example through use of ECS simulation sessions.

References

1 DELPHI collaboration, "The DELPHI detector at LEP", **NIM A303** (1991) pp.233-276.

2 T.Adye et al., "Architecture and Performance of the DELPHI Data Acquisition and Control System", **Proceedings CHEP'91**, Tsukuba, Japan, March 1991.

3 T.Adye et al., "The Design and Operation of the Slow Controls for the DELPHI Experiment at LEP", **NIM A349** (1994) pp.160-182.

4 J.A.Fuster et al., "Architecture and Performance of the DELPHI Trigger System", **Proceedings IEEE'92**, Orlando, Florida, October 1992.

5 B.Franek and C.Gaspar, "SMI++ - Object oriented framework for designing Control Systems for HEP experiments", **Proceedings CHEP'97**, Berlin, Germany, April 1997.

6 C.Gaspar et al., "DIM, a Portable and Efficient Package for Information Publishing, Data Transfer and Interprocess Communications", **These proceedings**.

7 A.Augustinus et al., "The evolution of the DELPHI Experiment Control System, or how to survive 10 years of running", **proceedings ICALEPCS'99**, Trieste, Italy, October 1999.