The JetFsm Data Acquisition, and proposed usage for ITER

P.J.L. Heesterman[1], K. Kneupner[1], D. Kinna[1], C. Hogben[1,], S. Griph[1], and JET EFDA contibuters*

JET-EFDA, Culham Science Centre, OX14 3DB, Abingdon, UK

[1]EURATOM-UKAEA Fusion Association, Culham Science Centre, Abingdon, Oxon OX14 3DB, UK

*See the Appendix of F. Romanelli et al., Fusion Energy Conference 2008 (Proc. 22nd Int. Conf. Geneva, 2008) IAEA, (2008)

INTRODUCTION

JET is the world's largest nuclear fusion device, operated by UKAEA Culham [1] for EFDA [2], with extensive collaboration from other fusion laboratories within Europe, and elsewhere.

A comprehensive C++ framework [3], using the HTTP-based 'Black Box' protocol [4][5], has been developed to allow Windows® PC-based control, data acquisition and data visualisation of JET diagnostic enhancements.

This framework has been used in-house to develop a large number of applications, which will be briefly itemised.

The framework has also been extensively used to allow collaborative software development to take place in association with remote laboratories. The paper will give a summary of experience gained through this process.

The 'Black Box' protocol is also used at JET on platforms other than Windows – e.g. Linux. The largest use is on Windows, partly as a consequence of the availability of commercial data acquisition hardware drivers.

CONTACT

Peter.Heesterman@Jet.Uk

BACKGROUND

The JET experiment operates in a pulsed manner, pulses of ~90s duration, occurring ~2 times per hour. During each pulse, a large number of data acquisition systems are acquiring data. Many of these are controlled and data is collected by means of the HTTP protocol. A typical time sequence of JET pulses is shown in figure 1.

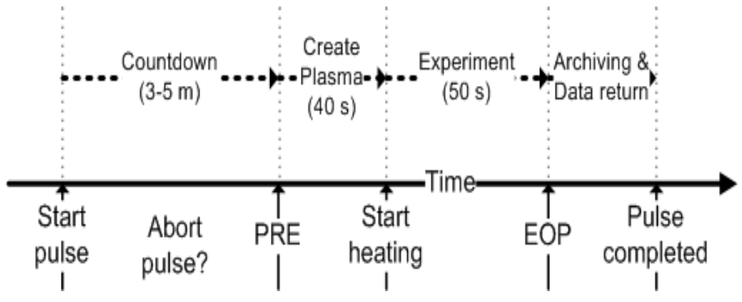

Figure 1 JET pulse time-line.

Figure 2 illustrates the architecture of a data acquisition system based on the JetFsm framework. An HTTP server communicates with HTTP client(s) on the Solaris subsystem. In turn, the server communicates with the JetFsm framework to control the acquisition system.

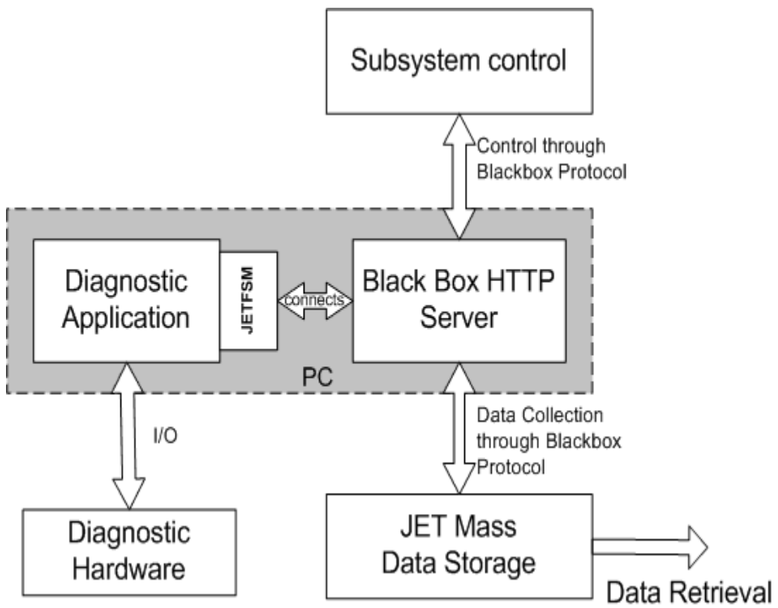

Figure 2 Diagnostic application system architecture

FRAMEWORK BENEFITS

- Efficiency of native C++ code.
- Generic, C++ templated data storage memory management.
- Polymorphic, multi-dimensional, data visualisation.
- 'Recycle bin' deletion of oldest local files.
- Disables non-essential Windows services during the acquisition
- Mature, used at JET for 7 years; about 20 acquisition codes are deployed onto ~60 PCs.
- Supports an alternative, file-based protocol. This is helpful for testing, and allows easy deployment of the software at sites where the HTTP protocol is not used.

THE JETFSM FRAMEWORK

The JetFsm software framework provides the skeleton of an application that can be customized by a data acquisition application developer, by means of C++ inheritance. Figure 3 illustrates how this works.

The framework is divided into a number of areas:

1. Data acquisition
2. Pulse control.
3. Data memory management.
4. User interface and data visualisation.
5. Operating system access.

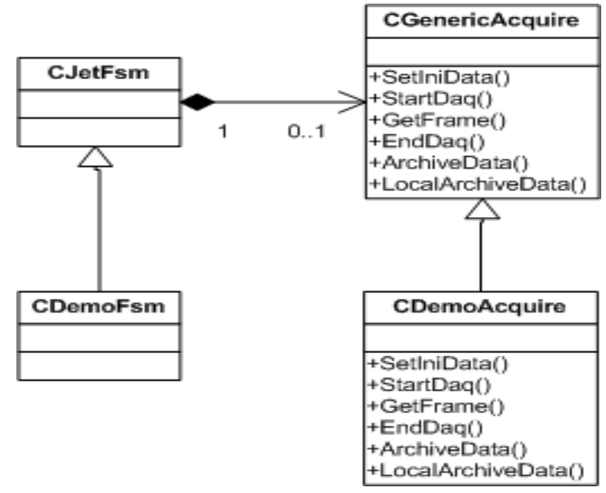

Figure 3 Example acquisition class inheritance

The state machine has 8 states:

1. Waiting for pulse (i.e. idle).
2. Waiting for (hardware) trigger.
3. Pulse (i.e. acquiring data).
4. Archiving.
5. Abort.
6. Error.
7. Fatal error.
8. Pre-initialising.

Figure 4 shows the transitions between these states.

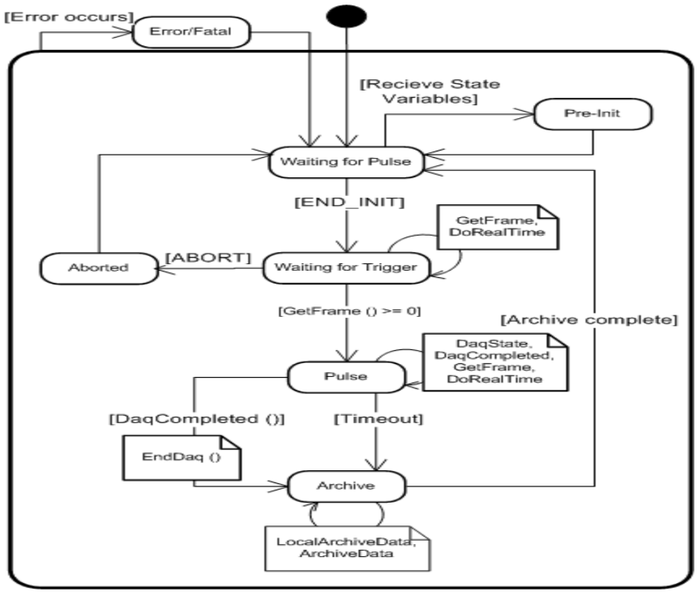

Figure 4 The JetFsm Finite State Machine

The framework includes extensive, multi-dimensional data visualisation capability; some example (spectrometer calibration) data is shown in figure 5.

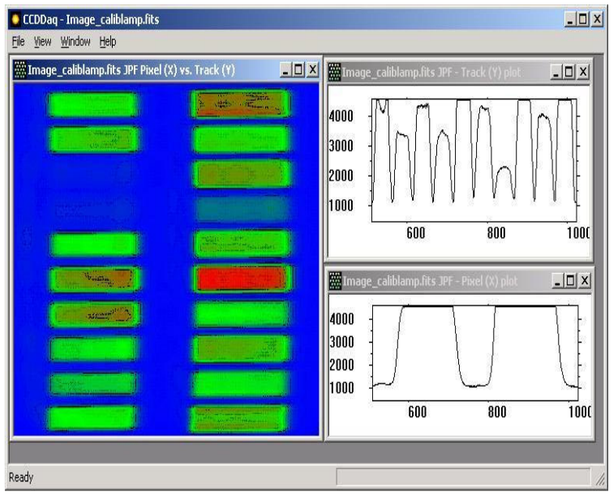

Figure 5. Example data visualisation

SUPPORTED DATA ACQUISITION HARDWARE

- Wright Instruments' MK2 and MK2 Enhanced CCD controller.
- XCam REM2 CCD Controller.
- Roper (Photronics) CCD Camera.
- Photron APX-RS CMOS Camera.
- Cedip Infra-Red camera.
- Hunt Engineering Infra-Red camera.
- Ortec Gamma Ray Spectrometer.

SPECTROMETRS

- Acton Research
- McPherson
- Spex

REAL-TIME DATA ANALYSIS

CCD spectrometer real-time data analysis [8]

LIDAR [9] real-time data analysis

ELM [10] detection

EXTERNAL PROJECT EXPERIENCE

11 external projects have been completed. These have included both 'Black Box' and 'handover' project types. A further 5 are in progress at this time. Lessons learnt are:

- The handover, or non-handover, status of each project must be agreed by all parties from the **outset.**
- Provide detailed and clear and complete documentation [10] to remote developers, describing software development in the JET environment.
- Where project handover is planned, it is essential that developers understand what is expected of them. Good channels of communication, between the remote project, and JET staff, are very important throughout the duration of the project.
- Language barriers can be a significant obstacle on multinational projects such as JET.
- Automatic testing should be used wherever possible. Such testing might use CPPUnit [11], for example.
- Projects should budget for spare acquisition hardware.

SUGGESTED ENHANCEMENTS (FOR ITER)

ITER (www.iter.org) is likely to generate much more demanding data acquisition and data return requirements due to the long duration (~1 hour) of its fusion pulses. Suggested future software enhancements to the framework that would make this possible are:

1. Data will need to archive continuously in real-time, or in near real-time. With today's multi-processor PCs, this will not prevent real-time data acquisition.
2. The framework could use RAM as a real-time buffer. Data is being continuously streamed to archive; the RAM of archived data is no longer required.
3. The framework, and applications that are based on it, are currently supported for 32-bit (x86) processor architecture. The framework, and some of the applications, to the 64-bit (x64) processor architecture supported by Windows Vista®, would overcome the 32-bit (4GB) address range limit.
4. The existing visualisation and GUI interface code is based around Microsoft's MFC class library. It would be preferable to replace this usage with a cross-platform GUI library.
5. It would be desirable to make the framework cross-platform capable, with (at least) Linux being supported.

CONCLUSIONS

- The JetFsm framework is a powerful medium for developing real-time scientific data acquisition applications, especially in an environment where unattended (remote) data acquisition is required.
- The framework is suitable for applications development at ITER (and elsewhere) although it would need to be extended in order to fulfil the demanding requirements of ITER. Note that extending the framework, in this way, would easily make the existing applications available for use in the same environment.
- The framework has been extensively used by external developers to develop applications for use at JET. However, the provision of an excellent software framework is not sufficient for this project model to work well. A lot of attention also needs to be directed at project co-ordination, quality management, and documentation.

ACKNOWLEDGEMENT

This work has been conducted under the European Fusion Development Agreement.

REFERENCES

1. http://www.fusion.org.uk/
2. http://www.efda.org
3. http://st-www.cs.uiuc.edu/users/johnson/frameworks.html#Definition
4. http://www.iop.org/Jet/fulltext/EFDR02004.pdf
5. The HTTP 'Black Box' Protocol for Control and Data Acquisition at JET, C Hogben et. al., PCaPAC 20-23 Oct 2008.
6. http://www.123exp-computing.com/t/03974064946/
7. http://www.real-time.org/hardandsoftrealtime.htm
8. Real-time analysis of charge-exchange spectroscopy data at JET_Rev. Sci. Instrum. 74, 1783 (2003); DOI:10.1063/1.1534406
9. http://www.jet.efda.org/pages/focus/lidar/index.html
10. http://www.jet.efda.org/pages/focus/elms/index.html
11. http://sourceforge.net/projects/cppunit