

Figure 2 illustrates the JetFsm state machine logic. A JetFsm pulse can be ended in one of four ways:

1. Acquisition completed normally.
2. The pulse was aborted by the controlling subsystem.
3. An error occurred in the data acquisition.
4. The software timed out on further data triggers.

In all cases, the software will return to the ‘waiting for pulse’ state.

OPERATING MODES

The data acquisition software normally works in ‘remote’ mode, ie unattended, and driven by JET pulses. Pulse initialisation and data acquisition parameters are delivered to the PC through the HTTP ‘Black Box’ protocol at the start of pulse, and data is collected afterwards.

The software can also be operated by the local user in ‘local’ mode. This is required at JET for calibration of the equipment, and for system commissioning. This mode can be combined with visualization of the data that has just been acquired.

Thirdly, some of the software can operate in ‘calibration’ mode, with scripted synchronisation of data acquisition and motorisation control. This is very useful for repetitive calibrations, which are frequently required.

FRAMEWORK BENEFITS

Although Windows is not a real-time operating system [9], the JetFsm framework achieves soft real-time [10] performance by virtue of the efficiency of native C++ code.

The framework includes generic, C++ templated, data storage memory management. This supports multi-dimensional datasets of arbitrary numeric types, with little, or no, run-time processing overhead required.

The data management is integrated with extensive polymorphic, multi-dimensional, data visualisation.

Data, pulse initialisation and log files are stored locally, with a ‘recycle bin’ deletion of oldest files. This prevents the local data from filling up the hard disk.

The framework automatically disables non-essential Windows services (such as virus scanning) during the acquisition, because these are likely to interfere with real-time acquisition capability.

The framework is mature, having been used at JET for 7 years, used in ~20 software packages and deployed onto ~60 PC systems. More systems are being developed and installed all the time.

An alternative, file-based protocol is also supported. This is convenient for testing, but also allows software to be deployed and integrated at other sites that do not use the JET HTTP protocol. An HTTP client component is also available for more rigorous testing of acquisition software.

The framework architecture is such that it would be straightforward to add support for other communication protocols.

DATA ACQUISITION EQUIPMENT

These are data acquisition systems which could be of interest to users outside of JET, or the Fusion community:

Acquisition hardware	Manufacturer Reference
Wright Instruments MK2 and MK2 enhanced CCD controller.	Formerly Jonathan Wright Instruments (Wembley, London, England).
XCAM REM2 CCD controller	http://www.xcam.co.uk/
Roper (Photometrics) Cascade CCD camera	http://www.photomet.com/pm_products/index.php
Princeton Instruments CCD controller types.	http://www.piaction.com/
Photron APX-RS CMOS camera.	http://www.photron.com/
Cedip Infra-red camera.	http://www.cedip-infrared.com/
Hunt Engineering Infra-red camera.	http://www.hunt-thermal.co.uk/
Ortec Gamma Ray Spectrometer.	http://www.ortec-online.com/pdf/dsp_pro.pdf

Control of the motorisation of these spectrometers may be integrated with the data acquisition:

Motorised spectrometer	Manufacturer Reference
Acton Research Spectrometer	http://www.piaction.com/products/sp ec/spectrapro/default.aspx
McPherson Spectrometer	http://www.mcphersoninc.com/
Spex Spectrometer	http://www.jobinyvon.com/

Several additional JET-specific data acquisition systems exist.

There are also a number of real-time data analysis systems, e.g. 3 different CCD spectroscopy data analysis systems [11]. These systems will take JET data acquisition system data, during the JET pulse, in order to process the raw data into physics results. These physics results can then be used for plasma control feedback experiments in order to generate stable fusion conditions.

INTEGRATION SOFTWARE

PCDataNet provide two options for integrating non-C++ software with the JET plant control:

1. JetFsmExt – This is a C++ application, built around the framework. It will invoke system commands in response to JET pulse signals.
2. JETCOM – This is a Windows COM component wrapped around a JetFsm object. This can be used for integrating software developed in languages other than C++.

SUGGESTED ENHANCEMENTS

ITER (www.iter.org), the much larger Tokamak which will be the successor to JET, is currently under construction. It is likely to have much more demanding data acquisition requirements, because fusion pulses will have a duration of ~1 hour.

This means that the real-time data acquisition model that is currently used by the framework, with data being stored in RAM for archiving post-pulse, is not viable.

1. Data will need to archive continuously in real-time, or in near real-time.
2. The framework could use RAM as a real-time buffer, with data being continuously streamed to archive.
3. The framework, and applications that are based on it, are currently supported for 32-bit (x86) architecture but could be ported to 64-bit (x64) architecture.
4. The visualisation and GUI interface code is based on Microsoft's MFC class library. It would be preferable to replace this with a cross-platform GUI library.
5. It would be desirable to make the framework cross-platform capable, with (at least) Linux supported.

EXTERNALLY-DEVELOPED PROJECTS

When software is developed by organisations outside JET, there are two ways of providing software support:

1. The organisation which developed the software remains responsible for support of the system, during the lifetime of JET.
2. Responsibility for maintaining the software is transferred to JET. In this case, the organisation would be responsible for providing software engineering documentation, to a standard specified by JET. This type of project is referred to as a 'handover' project.

Both of these approaches have been used by PCDataNet co-ordinated projects. PCDataNet has been involved in a total of 11 externally developed, and completed, enhancement projects that have used the JetFsm framework. For 8 of these 11, project handover was required by UKAEA management. A further 5 handover projects are either in progress, or in prospect, at this time.

Lessons learnt from co-ordinating external projects were:

1. It is **vital** that the handover, or non-handover, status of each project be agreed by all parties **at the outset** of the project
2. It is very important to provide detailed and clear and complete documentation to remote developers, describing all aspects of software development in the JET environment. PCDataNet provide extensive documentation aimed at clarifying the process.
3. Where project handover is planned, it is essential that developers understand what is expected of them to achieve this. This would include specifications of code quality, commenting and software documentation.

4. Good channels of communication, between the remote project and JET staff are very important throughout the duration of the project.
5. Language barriers can be a significant obstacle on multinational projects such as JET. A technically knowledgeable translator may be required at meetings. Project documents may need translation.
6. If software developers lack experience with software engineering quality assurance methodologies, PCDataNet can recommend appropriate training.
7. Automatic testing should be used wherever possible. Such testing might use CPPUNIT [12], for example.

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.

ACKNOWLEDGEMENTS

This work has been conducted under the European Fusion Development Agreement

REFERENCES:

- [1] <http://st-www.cs.uiuc.edu/users/johnson/frameworks.html#Definition>
- [2] <http://www.iop.org/Jet/fulltext/EFDR02004.pdf>
- [3] The HTTP 'Black Box' Protocol for Control and Data Acquisition at JET, C Hogben et. al., PCaPAC 20-23 Oct 2008.
- [4] <http://www.fusion.org.uk/>
- [5] <http://www.efda.org>
- [6] http://www.efda.org/about_efda/associated_laboratories.php
- [7] <http://jacow.org/ica07/PAPERS/TOPA01.PDF>
- [8] <http://www.nist.gov/dads/HTML/finiteStateMachine.html>
- [9] <http://www.123exp-computing.com/t/03974064946/>
- [10] <http://www.real-time.org/hardandsoftrealtime.htm>
- [11] Real-time analysis of charge-exchange spectroscopy data at JET_Rev. Sci. Instrum. 74, 1783 (2003); DOI:10.1063/1.1534406
- [12] <http://sourceforge.net/projects/cppunit>