

V. Martin, V. Gervaise, V. Moncada, J.M Traverre, S. Devaux, G. Arnoux,  
D. Kinna and JET EFDA contributors

# A New Integrated Software Platform for the Analysis of JET ITER-like Wall Imaging Diagnostics Data

“This document is intended for publication in the open literature. It is made available on the understanding that it may not be further circulated and extracts or references may not be published prior to publication of the original when applicable, or without the consent of the Publications Officer, EFDA, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK.”

“Enquiries about Copyright and reproduction should be addressed to the Publications Officer, EFDA, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK.”

The contents of this preprint and all other JET EFDA Preprints and Conference Papers are available to view online free at [www.iop.org/Jet](http://www.iop.org/Jet). This site has full search facilities and e-mail alert options. The diagrams contained within the PDFs on this site are hyperlinked from the year 1996 onwards.

# A New Integrated Software Platform for the Analysis of JET ITER-like Wall Imaging Diagnostics Data

V. Martin<sup>1</sup>, V. Gervaise<sup>1</sup>, V. Moncada<sup>1</sup>, J.M Traverre<sup>1</sup>, S. Devaux<sup>2</sup>, G. Arnoux<sup>3</sup>,  
D. Kinna<sup>3</sup> and JET EFDA contributors\*

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

<sup>1</sup>*CEA-EURATOM, IRFM, Saint-Paul-Lez-Durance, F-13108, France*

<sup>2</sup>*Max-Planck-Institut für Plasmaphysik, EURATOM-Assoziation, D-85748 Garching, Germany*

<sup>3</sup>*EURATOM-CCFE Fusion Association, Culham Science Centre, OX14 3DB, Abingdon, OXON, UK*

*\* See annex of F. Romanelli et al, "Overview of JET Results",  
(23rd IAEA Fusion Energy Conference, Daejeon, Republic of Korea (2010)).*

Preprint of Paper to be submitted for publication in Proceedings of the  
27th Symposium on Fusion Technology (SOFT), Liege, Belgium  
24th September 2012 - 28th September 2012



## **ABSTRACT**

During the last shutdown, the JET imaging systems have been significantly upgraded. JET is now equipped with 30 video cameras (visible, infrared, spectro-photometry, fast imaging) covering up to 70% of the inner wall. Consequently, it has become necessary to also upgrade the data access and analysis tools. To this end, a software platform named PInUP (Plasma Imaging data Understanding Platform) has been adapted to the JET computing environment. PInUP integrates all necessary tools to efficiently read, display, and analyze data from imaging diagnostic data bases in a graphical user interface. The software has demonstrated to be a crucial tool for the good operation of JET during the last campaign and is now daily used by operators and physicists.

## **1. INTRODUCTION**

During the last shutdown when moving from a full carbon to the new metallic ITER-like wall, the JET imaging systems have been significantly upgraded. JET is now equipped with 30 video cameras covering up to 70% of the Tungsten and Beryllium plasma facing components. Most of the new installed cameras are dedicated to the PIW (Protection of ITER-like Wall) project [1] for real time surface temperature measurements. Others infrared cameras have also been installed for physics studies, as seen in table 1. Since most of the imaging data acquisition systems - old and new ones – remain on different strategies, it has become necessary to also upgrade the data access and analysis tools. The main challenge was therefore to provide a unique tool able to read and display all types of imaging data from the different data repositories with time-line synchronization and advanced image analysis features. To this end, a software platform named PInUP (Plasma Imaging data Understanding Platform) developed at CEA and already used at Tore Supra has been adapted to the JET computing environment before the beginning of the JET campaign in fall 2011. This software is adapted to process large amount of multi-sensor data (e.g. several types of cameras and other diagnostic data) and is used for image analysis during plasma experiments and for edge plasma physics studies [2].

This paper describes the software architecture and main functionalities, its integration at JET and some use cases of basic functions for routine operation of the new real time video protection system in control room.

## **2. THE PINUP SOFTWARE**

### ***2.1 CONTEXT***

One major issue for the physicists is to efficiently analyze the produced images together with signals coming from other diagnostics without requiring deep programming skills. There already exist several tools on the Jet Analysis Cluster (JAC) for images analysis. However, as shown in table 1, these software applications are each dedicated to specific imaging diagnostics:

In order to perform cross-analysis from multiple diagnostics, operators and physicists need a platform that integrates all analysis tools into a unique Graphical User Interface (GUI), and allows physicists to develop their own tools inside.

### ***2.2. FUNCTIONALITIES***

PInUP gathers all necessary tools to efficiently read display and analyze data in a graphical user interface

as seen in figure 1. Among the major functionalities, the platform proposes to:

- Read several kinds of signal and video formats: raw ASCII files, MPEG video files, image sequences.
- Read all JET pulse data (JPF, PPF).
- Display and synchronize heterogeneous data (e.g. synchronization of videos coming from several cameras with 2D plots)
- Create and modify regions of interest (ROIs) of different shapes (e.g. rectangles, circles, polygons) and extract features from an image sequence as: a region histogram, the temporal evolution of some pixel statistics (minimum, maximum, mean and variance values), or the pixel profile along a segment.
- Use basic to more complex image processing techniques as pixel-wise calculations (e.g. for temperature conversion), geometrical operations (rotations, reflections, warping, etc.), or motion correction algorithm to compensate camera shaking.
- Develop custom analysis scripts and routines using an integrated Python environment.

### **2.3 SOFTWARE ARCHITECTURE**

The PInUP software is fully written in C++ and only uses open source libraries without viral licenses (no GPL). It can be compiled under the Linux (almost all distributions) and Windows operating systems, using either the GCC compiler (since version 4.5.2) or MS Visual Studio© (since 2005 version).

The software is based on a modular architecture. The core of PInUP is divided into several generic interconnected modules (i.e. C++ libraries) which forms the Software Development Kit (SDK) of the platform. This SDK defines the interfaces needed to develop new dynamic modules called plugins. This allows every user to develop new C++ plugins adding specific features to the software. For instance, if a new camera is installed, a new plug-in can be easily added to handle the generated data. The figure 2 presents the general software architecture.

### **3. PINUP INTEGRATION IN THE JET ENVIRONMENT**

PInUP grants the access to almost every kind of data available from a JET Analysis Cluster (JAC) as to know JPF (raw data), PPF (processed data) using a specific user ID as well as video files stored on a dedicated server.

A specific plugin has been developed to browse these JET data. This plugin also organizes the imaging diagnostics into several categories: protection, operation, IR, spectro-photometric and fast visible cameras. When browsing the cameras for a given pulse number, a red/green light beside the camera name indicates whether current data are available or not. Each type of camera can be used in several configurations. As shown in table 2, the combination of capture size, pixel depth, time stamping strategy and image acquisition rate is not the same for all cameras or even fixed for a camera. Specific and adapted loaders have thus been implemented in order to handle all these possibilities. At the opening of a movie, the right loader is automatically selected by the software. Thus, coping with different video formats is completely transparent to the user.

Unlike many software applications available at JET, PInUP loaders do not download the entire video file before reading it. Here, videos files are streamed and displayed dynamically. Thus, the amount

of memory required for playing a video is significantly reduced. This inexpensive way to play large videos allows PInUP to open several files at the same time, even from a terminal with low memory resources.

The ability to play videos from different cameras at the same time is even more interesting if these films can be synchronized in time, e.g. for hot spots or flying debris tracking using several cameras. But as explained above, JET videos may have different formats. This is also valid for video time vectors. For example, the time vector of a KL9 camera can be variable. It may be divided into three windows: a first one where the frame rate is low (slow window), a second where the frame rate is much higher (fast window), and in the last window where the frame rate is low again (slow window). Regarding the KL8 camera, the frame rate is steady and the time vector is composed of disjoint time windows (up to 8).

After understanding the different formats of time vectors, PInUP is able to merge time vectors of the selected videos to be synchronized. The merged time vector starts at the smallest timestamp, stops at the largest timestamp, and increments at the highest frequency. Synchronized videos are displayed in a single player, with a unique slider (see figure 1). When reading synchronized videos, the one with the slowest frequency will remain fixed on an image until the slider (which is incremented at the highest frequency of the selected films) reaches the timestamp of its next image.

The time vector (if recorded) of any other JPF/PPF does not require neither further treatment nor interpretation to be exploited. Thus, the PInUP software is not only able to synchronize videos, but also to synchronize videos and any time stamped JPF/PPF.

#### **4. PINUP AND THE PIW PROJECT**

PInUP was initially deployed at JET as a part of the PIW project. The goal of this new real time video protection system [3] is to compute temperature from the images of the so-called 12 protection cameras (visible cameras equipped with a NIR filter). These outputs are then sent to a decision system able to trigger alarms based on temperature thresholding and to eventually stop the plasma if temperature limits are reached.

The set of developed tools ensure essential functions for the routine operation of this new system as the offline display of recorded videos for event analysis. The basic use of PInUP by operators in control room concerns the supervision of the real time process unit (RTPU). This system provides temperature measurements from images grabbed by the protection cameras. More precisely, maximums of temperature are computed for specific ROIs, these ROIs being predefined by people in charge of the diagnostics. The RTPU is able to automatically distinguish between several types of events as neutron impact and dangerous events as hot spots. However, other unexpected events can occur (e.g. dust production) leading to false temperature estimation and thus to false alarms. It is therefore essential for operators to check the behavior of the system and to take adequate decision as ROI setting changes or system parameter tuning. To this end, a specific feature has been integrated in PInUP giving the possibility to track the position of the hot spot event detected in each ROI along the pulse. As shown in figure 3, for pulse JET Pulse No: 81511, the VTM has triggered a PANIC alarm at time  $t=7.95s$ . Using PInUP, the operator can quickly investigate this alarm just after the pulse. On the upper part of the GUI, the ROI drawn by the operators are superimposed on the video, here a protection camera with a divertor view, and the time trace of the hot spot candidate for the selected region is plotted on

the bottom part. Using the time slider, it is easy to retrieve on both the image and the plot the triggered PANIC alarm. After a quick analysis, it has been concluded that the strong edge plasma radiation observed in the image has led to a false hot spot detection triggering an alarm at  $t=7.95s$ . Operations can then safely continue.

## CONCLUSION

With the increasingly growth of imaging diagnostics installed at JET, the needs in adapted analysis tool is becoming a major issue. The PInUP software has demonstrated to be a crucial tool for the good operation of JET during the last campaign and is now used by operators and physicists. Moreover, the modular design of its architecture eases the development and the integration of new algorithms according to the physicist requests. PInUP is also more and more used by physicists for edge plasma studies based on imaging diagnostic data. For instance, PInUP has been used to compare results of power deposition calculations and real images from JET IR diagnostics [4]. For the study of RF Beryllium sources presented in [5], JET visible images from colour cameras have been processed using specific noise removal and background subtraction functions implemented in PInUP.

## REFERENCES

- [1]. G. Arnoux et al., A protection system for the JET ITER-like wall based on imaging diagnostics, in proc. HTPD, 2012.
- [2]. V. Martin et al., Integrated software for imaging data analysis applied to edge plasma physic and operational safety, in Fusion Engineering and Design, **86**, 4-5, PP. 270-278, 2011.
- [3]. M. Jouve et al., Real time protection of the ITER-like Wall et JET, in proceedings of ICALEPCS, Grenoble, France, pp. 1096, 2011.
- [4]. Firdaouss et al., Modelling of power deposition on the JET ITER like wall using the code PFCFLUX, in proc. of PSI, Aachen, Germany, 2012.
- [5]. Klepper et al., RF Sheath-Enhanced Beryllium Sources at JET's ICRH Antennas, in proceedings of PSI, Aachen, Germany, 2012.

JET imaging diagnostic	Software name	Functionalities
IR cameras (KL7 and KL3)	IRdisp (IDL)	Video player, temperature conversion, projection of 3D components, few image processing functions
IR cameras (KL7, KL9, KL3)	KL9Control (Python)	Video player, temperature conversion, heat flux calculation, motion correction
VIS+NIR filter (protection), colour cameras	Jetmovie (C/C++)	Video player only
Fast visible camera (KL8)	KL8 Viewer (IDL)	Video player only
JETview (VIS+NIR, spectro, colour, fast visible cameras)	KL8 Viewer (IDL)	Video player only

*Table 1: JET image analysis related softwares.*



Camera type	No. of cameras	Image size (pixels)	Pixel depth (bits)	Image acquisition rate (Hz)	Time vector
VIS + NIR filter (PIW)	13	720x255	8 (DL)	50	Can be empty or false
Colour (operation)	3	720x576	24 (YCrCb)	50	No specificity
IR (KL7)	1	Up to 640x512	16 (DL)	Up to 500	No specificity
IR (KL9)	2	Up to 320x256	16 (DL)	Up to 28k	Needs to be corrected and shifted. Can be variable
IR (KL10)	1	320x255	16 (DL)	40	No specificity
VIS + spectral filters	7	1024x1024	8 or 16 (DL)	33	No specificity
Fast visible (KL8)	1	Up to 1024x1024	8	1k-250k	Disjoint time windows (up to 8) from triggers

Table 2: Imaging diagnostics used during the 2011/2012 JET campaign.

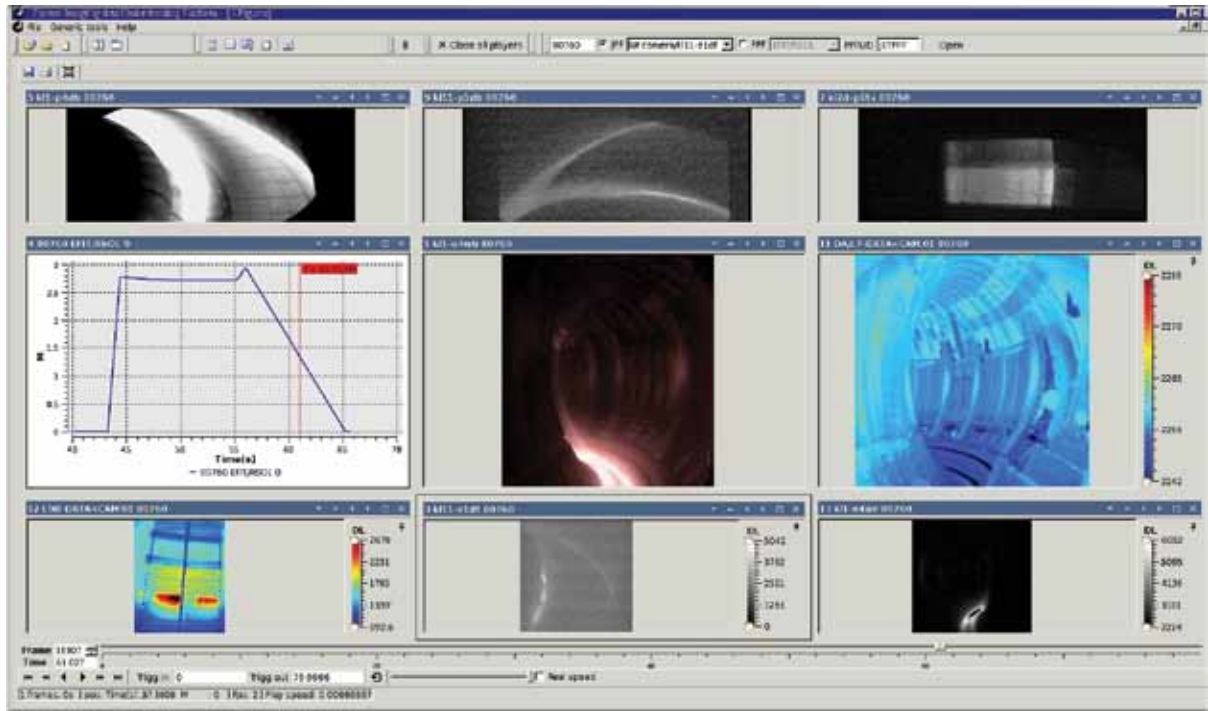


Figure 1: PInUP graphical user interface showing 8 videos of the same pulse (JET Pulse No: 80760) from different cameras, all time-synchronized with one PPF plot (EFIT/RSOL).

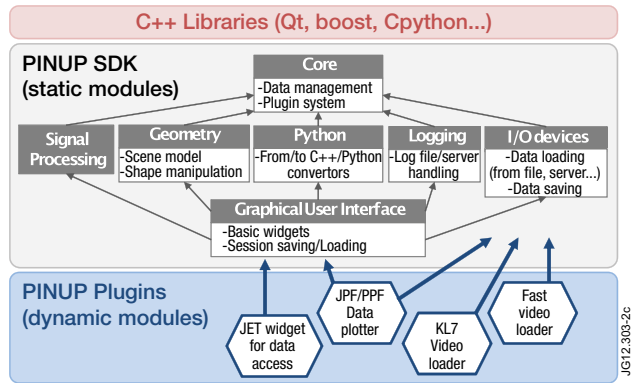


Figure 2: the PinUP architecture.

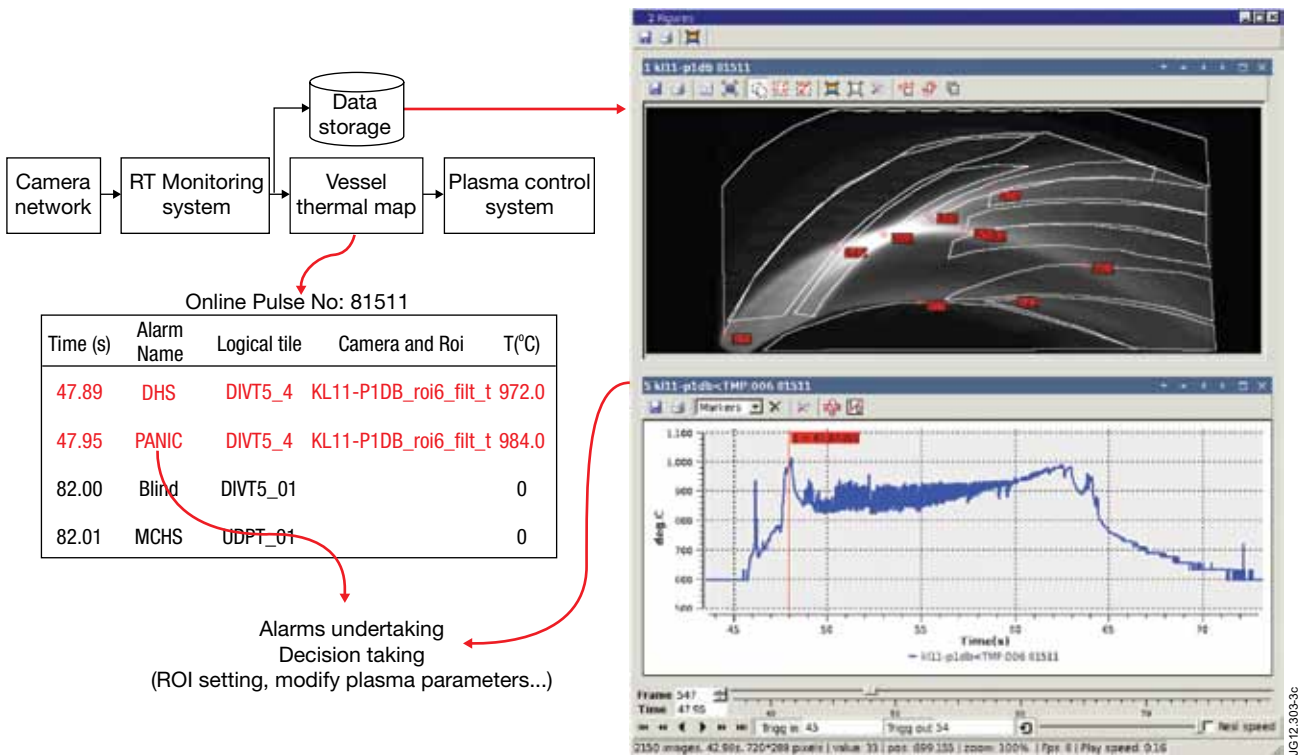


Figure 3: Typical use of PinUP for the analysis of the real time video protection system outputs.