

# THE SOFTWARE AND HARDWARE ARCHITECTURAL DESIGN OF THE VESSEL THERMAL MAP REAL-TIME SYSTEM IN JET

D. Alves, A. Neto, D.F. Valcárcel

Associação EURATOM/IST, Instituto de Plasmas e Fusão Nuclear Laboratório Associado, Instituto Superior Técnico, Universidade Técnica de Lisboa, 1049-001, Lisboa, Portugal

S. Jachmich

Laboratory for Plasma Physics, Ecole Royale Militaire/Koninklijke Militaire School, EURATOM-Association “Belgian State”, Brussels, Belgium,  
Partner in the Trilateral Euregio Cluster (TEC)

G. Arnoux, P. Card, S. Devaux, R. Felton, A. Goodyear, D. Kinna, P. Lomas, P. McCullen, A. Stephen, K.-D. Zastrow and JET EFDA contributors\*

EURATOM-CCFE Fusion Association, Culham Science Centre, Abingdon OX14 3DB, United Kingdom

## Abstract

The installation of ITER-relevant materials for the Plasma Facing Components (PFCs) in the Joint European Torus (JET) is expected to have a strong impact on the operation and protection of the experiment. In particular, the use of all-beryllium tiles, which deteriorate at a substantially lower temperature than the formerly installed Carbon Fibre Composite (CFC) tiles, imposes strict thermal restrictions on the PFCs during operation. Prompt and precise responses are therefore required whenever anomalous temperatures are detected.

The new Vessel Thermal Map (VTM) real-time application collects the temperature measurements provided by dedicated pyrometers and Infra-Red (IR) cameras, groups them according to spatial location and probable offending heat source and raises alarms that will trigger appropriate protective responses. In the context of JET's global scheme for the protection of the new wall, the system is required to run on a 10 millisecond cycle communicating with other systems through the Real-Time Data Network (RTDN).

In order to meet these requirements a Commercial Off-The-Shelf (COTS) solution has been adopted based on standard x86 multi-core technology, Linux and the Multi-threaded Application Real-Time executor (MARTe) software framework. This paper presents an overview of the system with particular technical focus on the configuration of its real-time capability and the benefits of the modular development approach and advanced tools provided by the MARTe framework.

## INTRODUCTION

As the fusion scientific community steers their efforts towards the operation of the ITER tokamak, JET aims to provide an important contribution as it will operate with

similar PFC materials. In particular, demonstrating the predicted reduction of the tritium retention levels when compared with the previous CFC-based wall[1]. JET's new all metal wall surface made of solid beryllium, tungsten and tungsten coated CFC is much less robust than the previous one therefore raising serious challenges in high power operational conditions.

The Protection of the ITER-like Wall (PIW) project was launched with the aim of providing the necessary tools to ensure the integrity of the vessel during JET's scientific campaigns. These include a set of 13 IR cameras and 9 pyrometer diagnostics that, together with their real-time image processing systems[2], provide the temperatures of PFCs.

The VTM collects these temperature measurements, groups them according to spatial location and probable offending heat source, and raises alarms that trigger the appropriate protective responses coordinated by the Real-Time Protection Sequencer[3] (RTPS) system, see Figure 1. RTPS drives the Local Managers (LM) for all of JET's non-inductive heating systems: Lower Hybrid (LH), Radio Frequency (RF) and Neutral Beam (NB), the Plasma Density Local Manager (PDLM) and the Plasma Position and Current Control (PPCC) system.

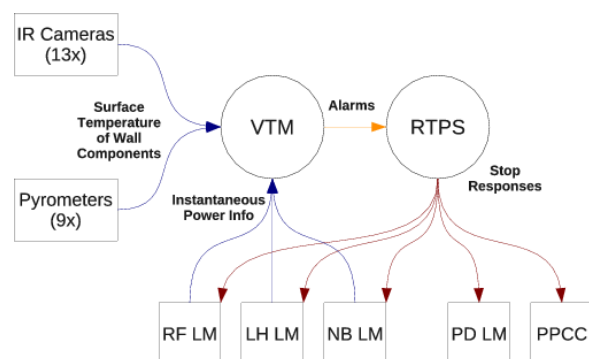


Figure 1: VTM in the context of the PIW project.

\* See the Appendix of F. Romanelli et al., Proceedings of the 23rd IAEA Fusion Energy Conference 2010, Daejeon, Korea

## VTM FUNCTIONAL OVERVIEW

Temperature measurements of physical tile groups are processed by the VTM in macro sets called Logical Tiles (LTs). These macro sets take advantage of toroidal symmetry to establish the equivalence of temperature measurements that share the same poloidal position although taken at different toroidal locations. This feature allows for different camera views that, although not “looking at” particular physical tiles, monitor “equivalent” ones. Also, in case of measurement loss (e.g. when a camera fails in the middle of an experiment), VTM checks to see if the minimum amount of measurements for each LT is still fulfilled and, if so, the experiment proceeds without significant risk to the machine. After checking the validity of temperature measurements, VTM estimates LT temperatures as the maximum of all temperature measurements within the set.

At a higher level, sets of LTs are grouped into *Wall Segments* (WSs). Similar to the LT estimated temperature, the WS temperature is defined as the maximum of all LT temperatures within the set. Alarm triggering temperature thresholds are placed on WS estimated temperatures. VTM has the capability of processing a maximum number of 60 WSs (30 fixed and 30 user-definable) and more than 400 tiles.

Figure 3 shows part VTM’s expert graphical user interface where WS related quantities and the mapping between physical and logical tiles are controlled.

The system was specified to run at 100 Hz, twice the frequency of the real-time image processing units’ temperature outputs and its real-time I/O was specified to be performed solely via the ATM-based RTDN[4].

## HARDWARE

Because of its technical specifications, and after considering other options such as VxWorks® and VME (VV) or RTAI and ATCA (RA), VTM was chosen to be implemented on a standard 4GB x86 multicore (6 cores) Linux-based PC with a PCI Gb ethernet Network Interface Card (NIC) and a PCI ATM NIC. Despite JET’s long history of implementing the VV combination for real-time systems and the recent success of using the RA combination for the new Vertical Stabilisation system, the chosen path proved advantageous in terms of cost, development effort, debugging tools availability and with support within the Linux and MARTE’s community.

## SOFTWARE

The VTM system was built using the MARTE real-time framework[5] on a multicore vanilla (2.6.35.9) Linux platform. Real-Time performance with a traditionally non real-time operating system is achieved by configuring cpu isolation in terms of processes, threads and Interrupt ReQuest (IRQ) affinities, see Table 1.

The synchronization of the VTM system with JET’s central timing is done via the RTDN. Central timing is read

Table 1: CPU Affinities

CPU #	Task
0	Linux & MARTe services
1	ATM IRQs
2	ATM synch Rx thread
3	ATM Rx threads
4	ATM Rx threads
5	Real-Time thread

by the Real-Time General Services (RTGS) system directly from a VME electronics module and made available to the network. VTM not only time stamps its internal data using this clock but also triggers its own “control” cycle on the arrival of this ATM packet. Preliminary results show that worst case jitters of less than 50  $\mu$ s on RTGS are propagated to worse case jitters of less than 100  $\mu$ s on the VTM, see Figure 2.

Lab tests were performed using the available Linux kernel’s real-time patch but those showed no performance improvements when compared to the above mentioned configuration.

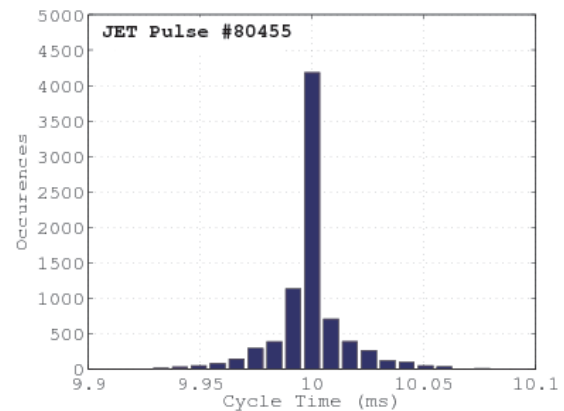


Figure 2: VTM Cycle Time.

Furthermore, the VTM system receives a total of 19 ATM packets in real-time (13 packets from cameras, 2 from pyrometers, 3 from additional heating systems plus the synchronization packet). A standard Linux (kernel module) ATM low-level driver is used together with a (user space) MARTE thread-based high-level driver socket implementation for receiving.

The system has been configured to boot remotely using the *tftp* protocol and mount its file system using NFS. These approaches attempt to make the VTM more resilient to hardware faults.

MARTE is a multi-platform framework for the development and deployment of data-driven, flexible and modular real-time applications. It is based on a real-time oriented C++ library called BaseLib2. It currently supports the VxWorks®, Linux, Linux/RTAI®, Solaris® and Windows® operating systems. Ideally, and assuming the

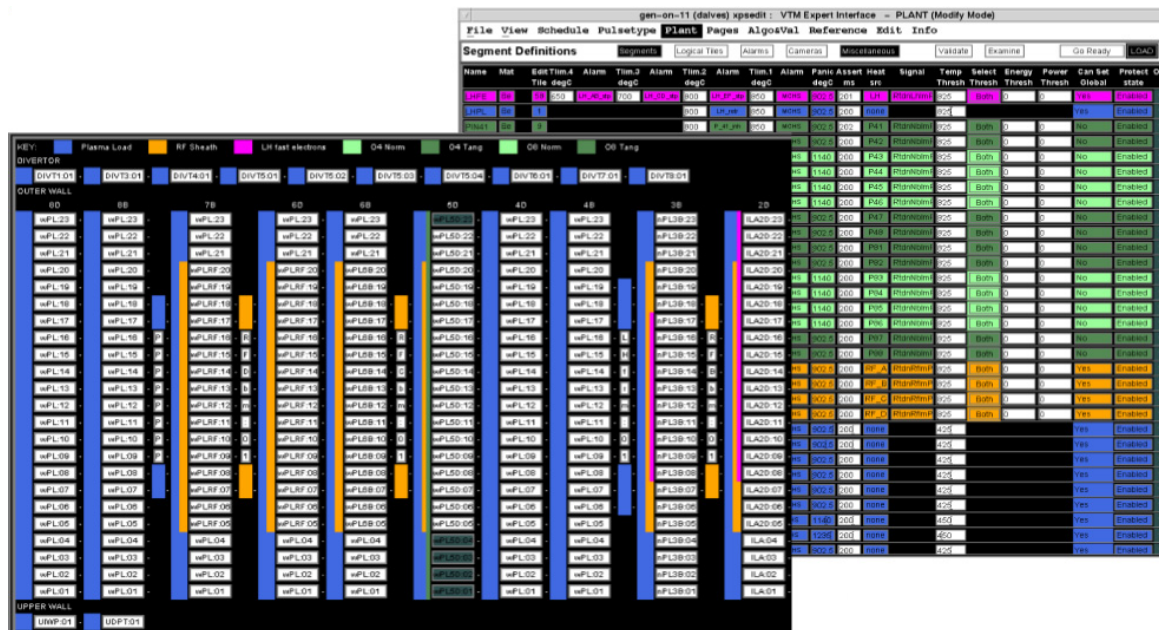


Figure 3: VTM Expert Graphical User Interface.

required hardware components are supported, the developer is only required to write the pieces of software dealing with the logic/algorithmic aspects specific to the control application itself.

MARTE, from the user point of view, is primarily a sequential executor of *Generic Application Modules* (GAMs) in a real-time priority context. MARTE also defines standard high-level interfaces for various activities including driver I/O, http display and messaging. It provides a configurable internal state machine which, in the case of the VTM, is driven remotely by JET’s global state machine. According to its current state, the appropriate list of GAMs is executed. Deeply embedded in MARTE’s philosophy is the idea that GAMs should not be aware of one another and would perform specific and self-contained tasks communicating with each other exclusively by writing signals to and reading signals from the *Dynamic Data Buffer* (DDB). This buffer is basically a signal pool with a standard access interface. MARTE also contains a driver pool of high-level interfaces for common (and/or maybe less common user implementable) activities, e.g., I/O. Finally, the External Time Triggering Service (ETTS) is the entity that unleashes the execution of the GAMs. It can either be interrupt driven or polling a specific event. Figure 4 shows not only some of MARTE’s architecture but also the specific VTM GAMs:

**ATM Synch input GAM** - is unblocked by the ETTS on the arrival of the ATM synch packet every 10 ms;

**ATM input GAMs** - make the ATM packets’ data available as DDB signals;

**Digital Filter GAM** - filters DDB signals;

**Logical Tile GAM** - estimates LT temperatures based on the measurements;

**Wall Segment GAM** - estimates WS temperatures based on LT temperatures;

**Alarm GAM** - issues alarms based on WS temperatures and thresholds;

**Event GAM** - records and displays alarm events;

**ATM output GAMs** - output the alarms and WS temperatures to the RTDN;

**Data Collection GAM** - record DDB signals to be collected for posterity;

**Web Statistics GAM** - http display of DDB signal statistics.

To provide a convenient and intuitive real-time display of the vessel temperature map, an Ajax based http service (see Figure 5) has been developed. Planar images of both JET’s inner and outer wall are used as canvas for painting each LTs’ temperature using the color code indicated at the bottom of the Figure. This service has been tested with a single browser client polling at 4 Hz and no impact was observed in the system’s real-time performance.

### PRELIMINARY RESULTS

As a proof of concept, dedicated plasma pulses were performed to test the end-to-end response of PIW’s protection apparatus. Simulated camera measurements based on visible light were provided to the VTM and reference pulses were used to place VTM’s alarm triggering temperature thresholds at convenient levels. Figure 6 illustrates the achievement of pulse 80455 where a Divertor Hot-Spot (DHS) was detected by the VTM at the bottom of the machine and the correspondent alarm triggered the appropriate predefined PPCC response moving the plasma current’s

centroid upwards and reducing the gap between the plasma boundary and the top of the vessel.

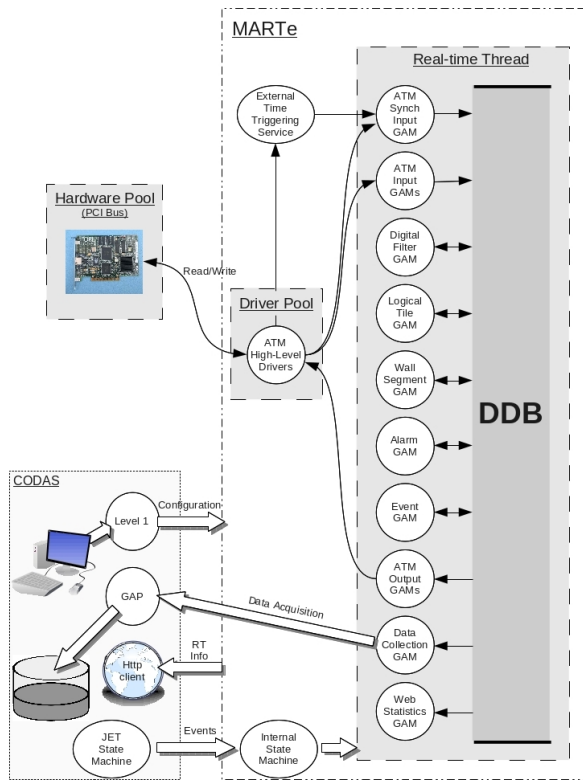


Figure 4: VTM MArTE.

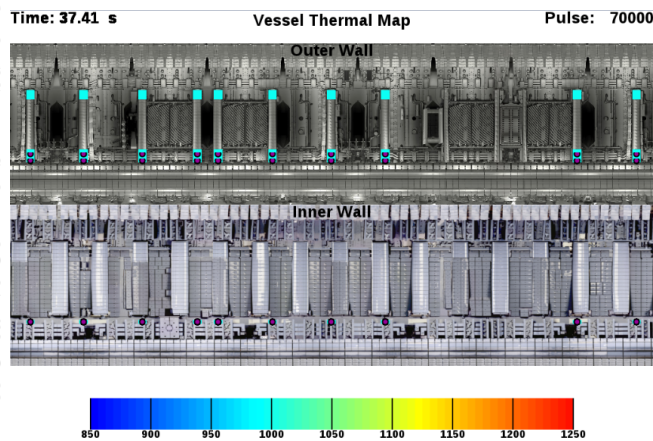


Figure 5: Real-Time temperature map display - http/ajax based polling @ 4 Hz without compromising real-time performance.

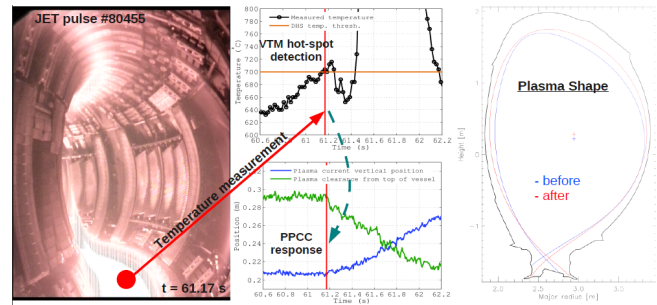


Figure 6: Preliminary results - upon the detection of a thermal event, VTM signaled RTPS, which in turn drove PPCC into a new plasma shape and position.

### CONCLUSIONS

The VTM has been demonstrated to perform the role it was designed for in the context of the PIW project. The non traditional configuration of its real-time capability, using a plain Linux vanilla kernel together with the user-space MArTe-based implementation makes it an interesting proof that with little effort it is possible to have an application meeting real-time requirements. Its Ajax-based visual temperature monitoring service with no impact in real-time performance takes this matter to an even higher level demonstrating the potential of the multicore approach.

In the near future this system will be part of the standard machine protection ensemble fundamental to JET operations.

### REFERENCES

- [1] G. Matthews et al, "JET ITER-like Wall - Overview and experimental programme", Proceedings of the 13th International Workshop on Plasma-Facing Materials and Components for Fusion Applications, Rosenheim, Germany.
- [2] M. Jouve et al, "Real-time protection of the ITER-like Wall at JET", this conference.
- [3] A. V. Stephen et al, "Centralised Coordinated Control To Protect The JET ITER-like Wall", this conference.
- [4] R. Felton et al, "Real-time plasma control at JET using an ATM network", 11<sup>th</sup> IEEE NPSS Real-Time Conference, Santa Fé 1999.
- [5] A. Neto et al, Nuclear Science, IEEE Transactions on 57 (2010).