MARTe in JET
June 14, 2010

June 14, 2010 - EFDA Feedback Control Working Group Meeting

D. Alves\textsuperscript{1}  G. De Tommasi\textsuperscript{2}  A. Neto\textsuperscript{1}  F. Sartori\textsuperscript{3}  
R. Vitelli\textsuperscript{4}  L. Z Abeo\textsuperscript{5} and EFDA-JET PPCC contributors

\textsuperscript{1}Associação EURATOM/IST, Instituto de Plasmas e Fusão Nuclear
\textsuperscript{2}EURATOM-ENEA-CREATE, Università di Napoli Federico II
\textsuperscript{3}Fusion For Energy
\textsuperscript{4}Università di Roma Tor Vergata
\textsuperscript{5}ITER Organization
Where we started from - mid 90s

From JETRT to MARTe
The eXtreme Shape Controller

MARTe in JET
The new Vertical Stabilization System - VS5
EFCC Voltage Amplifier Controller
PPCC Control systems in the mid 90s

PPCC systems for plasma magnetic control

The two main systems run at JET by the Plasma Position and Current Control Group were (and still are!):

**Shape Controller (SC)**  C code deployed on a VxWorks/VME/Motorola68k platform

**the Vertical Stabilization System (VS)**  C code deployed on 4 Texas Instruments DSPs

- The code was *tailored* for the specific platform
- Lack of modularity
- Different software solutions to interface with the JET software infrastructure (pre-pulse system configuration, post-pulse data collection,...)

F. Sartori et al., The Joint European Torus - Plasma position and shape control in the world’s largest tokamak, *IEEE Control Systems Magazine*, vol. 26(2), Apr. 206
A new framework for RT applications

Motivation

- In 2001/2002 the revamping of the SC was planned in order to add the eXtreme Shape Controller algorithm (XSC)
- Within the PPCC group, it was decided to move to a common framework for the development of real-time applications

Aims

- Standardize the development of real-time applications
- Increase the code reusability
- Give the possibility to separate the user application from the software required to interface with the plant infrastructure
- Reduce the time needed for commissioning.

Requirements

The new framework would have been:

- portable (multi-OS and multi-platform)
- modular – the user application would have been easily plugged into an executor of real-time application
- written in C++ (at that time C++ was not a CODAS standard)
The JETRT framework was developed in 2002/2003 to deploy the XSC. JETRT is based on the cross-platform BaseLib library (developed within the PPCC group).
Separation between application and infrastructure software

Why we want to separate application from infrastructure software?

> Scientist (process experts) can abstract from the plant interfaces
> Increase code reusability
> Achieve standardization
As a result we have a **Real-time Application Plug-in** that can be used to:

- ▶
- ▶
- ▶
As a result we have a **Real-time Application Plug-in** that can be used to:

- perform offline validation against a plat model
  - ...
  - ...
  - ...

Real-time application plug-in
As a result we have a **Real-time Application Plug-in** that can be used to:

- perform offline validation against a plat model
- perform real-time validation with hardware-in-the-loop
As a result we have a **Real-time Application Plug-in** that can be used to:

- perform offline validation against a plat model
- perform real-time validation with hardware-in-the-loop
- run the real-time system on the plant
The new SC (including the XSC) has been deployed on a 400 MHz G4 PowerPC running VxWorks.

- 2 ms control loop (but it can easily run at 1 ms)

Commissioning of the JETRT framework and of the XSC

- Thanks to portability, an exhaustive debug of both the JETRT framework and the XSC was performed offline on a Windows-based platform.
- Only 3 days of testing on the plant were needed to commission the new system.
Although it was a first attempt, JETRT didn't provide a real separation between the user application from the plant-interface software!

**From JETRT to MARTe**

- *More* modularity $\rightarrow$ Generic Application Modules (GAMs)
- *Real* separation $\rightarrow$ Dynamic Data Buffer
Since 2006 several MARTe–based systems have been deployed at JET:

- Walls (2006 - old version)
- XSC strike–points sweeping algorithm (2007 - old version)
- Betap–Li (2008)
- VS system (2008/2009)
- EFCC controller (2009)
Elongated tokamak plasmas are susceptible to a vertical axisymmetric instability

- Dedicated VS system required
- Essential system for operation
- Growth rate of 1000 s$^{-1}$
- Loss of control can produce forces in the order of the 100’s of tonnes
Elongated tokamak plasmas are susceptible to a vertical axisymmetric instability

- **192 signals** acquired by ADCs and transferred at each cycle
- **50 µs** control loop cycle time with jitter < 1 µs
- Always in real-time (24 hours per day)
  - $1.728 \times 10^9$ 50 µs cycles/day
  - Crucial for ITER very long pulses
VS Requirements

- 18 GAM instances
  - Altogether execute in less than 40 $\mu s$
  - Synchronization always achieved within 0.8 $\mu s$

- 192 signals acquired by ADCs and transferred at each loop

- Enable advanced experimental features
  - ELM pacing
  - Complex time windows with different controller features and settings
MARTe for the new VS

Outline
Where we started from
From JETRT to MARTe
The XSC
MARTe in JET
VS5
EFCC Controller
EFCC Voltage Controller - Introduction

▶ EFCCs, what do they do?
  ▶ They change magnetic field topology at the plasma boundary

▶ Why is it important?
  ▶ Instability mitigation and ELM control

▶ How?
  ▶ By controlling the current in the EFCCs we can control the magnetic field

▶ Who?
  ▶ The session leader sets the required current waveforms
EFCC controller – Schematic
EFCC controller – Hardware

- VME-based technology
- VPLS (timing)
- VX5100 (cpu)
- MPV956 (analog I/O)
- MPV922 (digital I/O)
EFCC controller – Software
EFCC controller – Configuration

MARTe in JET
G. De Tommasi

Outline
Where we started from
From JETRT to MARTe
The XSC
MARTe in JET
VS5
EFCC Controller
EFCC controller – Experimental results
Conclusions

- Brief MARTe history (from 2001-today)
- The development has been mainly driven by JET needs
- First “porting” to different machines has been done
Conclusions

- Brief MARTe history (from 2001-today)
- The development has been mainly driven by JET needs
- First “porting” to different machines has been done
- Additional effort is needed to make it a commercial framework

THE END

Thank you!