Model-based design and validation of control systems
December 11, 2009

December 11, 2009 - ITER Plasma Control Workshop

G. De Tommasi[1] A. Neto[2] F. Sartori[3] L. Zabeo[4]
and EFDA-JET PPCC contributors
[1]EURATOM-ENEA-CREATE, Università di Napoli Federico II
[2]Associação EURATOM/IST, Instituto de Plasmas e Fusão Nuclear
[3]Fusion For Energy [4]CCFE Fusion Association, Culham Science Centre

Outline

Motivations

Standardization of RT control systems

Model-based Design and Validation

Motivations

Main aim

Reduction of the time needed for commissioning on the machine

"Tools"

- Standard architecture for real-time control systems
  - Complete separation between the algorithmic part of a real-time application from the plant-interface software
  - Multi-platform development environment
- **Massive offline validation against plant models**

Separation between application and infrastructure software

Why we want to separate application from infrastructure software?

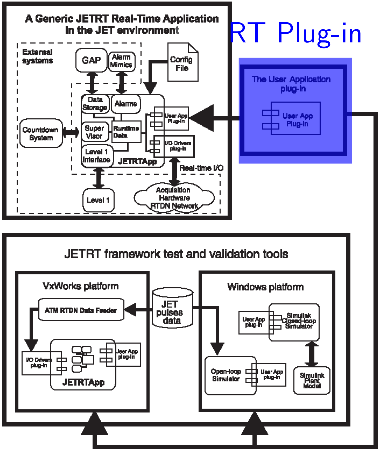
Defining a standard API for real-time control application we:

- Scientist (process experts) can abstract from the plant interfaces
- Increase code reusability
- Standardization (I)

Real-time application plug-in

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

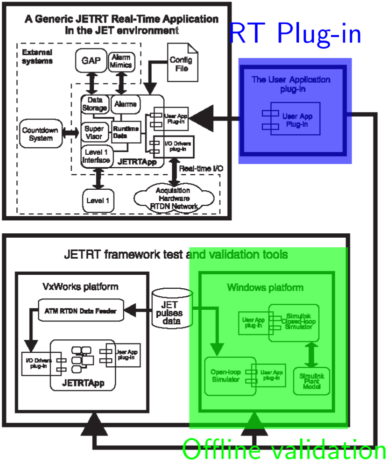
-
-
-



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



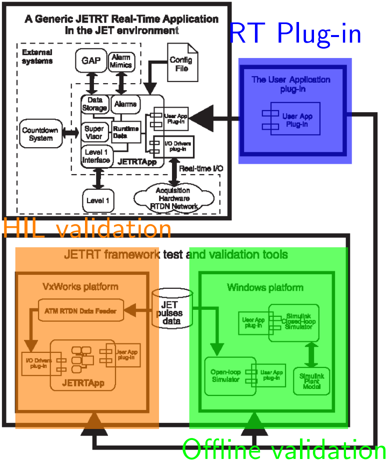
RT Plug-in

Offline validation

Real-time application plug-in

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

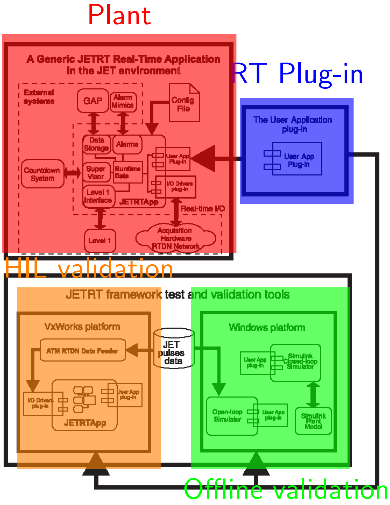
- perform offline validation against a plat model
- perform real-time validation with hardware-in-the-loop
-



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
- run the real-time system on the plant



Multi-platform development environment

Why we want a portable framework?

- Real-time applications can be tested offline on standard office devices (Intel/Windows) (⇒ sensible reduction of time needed for commissioning)
- A large range of platforms can be supported:
    - *High performance* and *Standard* platforms
    - *Prototypes* and *COTS* platforms
    - *Open source* and *Closed source*
    - . . .
- Standardization (II)

Modelling in control system design and validation

Modelling helps you . . .

- to define control system requirements
- to design the control algorithms
- to make performance analyses
- to validate real-time implementation of the control systems
- to perform offline analyses to forecast experimental behaviour

Modelling for system requirements definition

When system requirements have to be defined, the development of a simplified plant model

- forces the definition of I/O (interfaces)
- helps the definition of the *operational scenarios*
- gives the possibility to define non-envisaged situations
- can be used to formally define the system requirements

- This task is performed by using offline environments
- Rapid prototyping techniques can be used to product a prototype of the control system
- Hardware-in-the-loop (HIL) simulations can be performed to do a preliminary performance assessment

Modelling for control algorithm design

Detailed models must be developed

- to design control algorithm
- to assess control system performance

This task is performed by using offline environments (Matlab/Simulink. . .)

Modelling for control system validation

Real-time version of (simplified) models can be effectively used

- to perform offline *live* debug of the software implementation
- to peform offline validation of control system with or without hardware-in-the-loop

Given a multi-platform modular framework, two possible *approaches*

1. The real-time control application plug-in can be validated offline in a simulation environment (e.g. Matlab/Simulink)
2. A plant model plug-in can be added in the plant system

Conclusive remarks

- Models are essential for the design of control algorithm, but there is something more. . .
- . . .before the design phase, models can help you to define system requirements. . .
- . . .after the design phase, models help you to test and validate the real control system

You need a SW/HW architecture that allows you

- to test the real-time application in a simulation environment
- and/or to embed the (simplified) plant model into the real-time system