The Control and Data Acquisition System of the DTT experiment

G. Manduchi[1], F. Zanon[2], L. Boncagni[3],

P. Mosetti[3], G. Martini[1], G. Paccagnella[2], C. Centioli[3], R. Ambrosino[4], F. Sartori[5]

1: Consorzio RFX; 2: ENI; 3: ENEA; 4:CREATE; 5:F4E

The DTT project

- The Divertor Tokamak Test facility will be one of the largest scientific infrastructures in Italy and the most advanced tokamak in Europe after ITER.

- It is a fully superconducting Tokamak, able to confine deuterium plasmas with high flexibility. Its parameters place DTT in a position relevant for DEMO design.

	DTT	ITER	DEMO
R(m)	2.19	6.2	9.1
a(m)	0.7	2	2.93
B_t(T)	5.85	5.3	5.7
I_p(MA)	5.5	15	19.6
Ptot(MW) (ECRH/NBI/ICRH	45	120	460
Tpulse(s)	100	400	7600

DTT CODAS perspective

1. **Long Lasting experiment**
   - *Need to adopt data streaming technologies*
2. **At least 8 years before the first plasma**
   - *Decisions on priorities in CODAS development*
   - *Keep into account evolving computer technology*
3. **The exact number of signals is still unknown**
   - *Look at experiments of similar size (e.g. EAST) for an initial guess*
   - *Adopt a modular approach*
4. **Considering adopted solutions in similar experiments**
   - *The ITER experience*
5. **Take into account the need for scientific data exchange**
   - *Look around for common data format (e.g. ITER IMAS)*

Before starting: look at the lessons learnt

- **Problems related to the long development time in large experimental machines have already been tackled in other projects**
  - *CODAS development can start neither too early (wasted efforts) nor too late (not ready at first plasma)*
- **A single large project is much more likely to fail with respect to rapid prototyping**
  - *Think of CODAS as a set of coordinated tools rather than a monolithic system*
- **There is no real reason for re-inventing the wheel**
  - *It takes much longer to test and adapt a given system than to develop it*
- **Understand the requirements**
  - *Even the best Software Engineers are likely to develop a useless system if missing previous experience in Fusion Experiments*

The current DTT CODAS activities

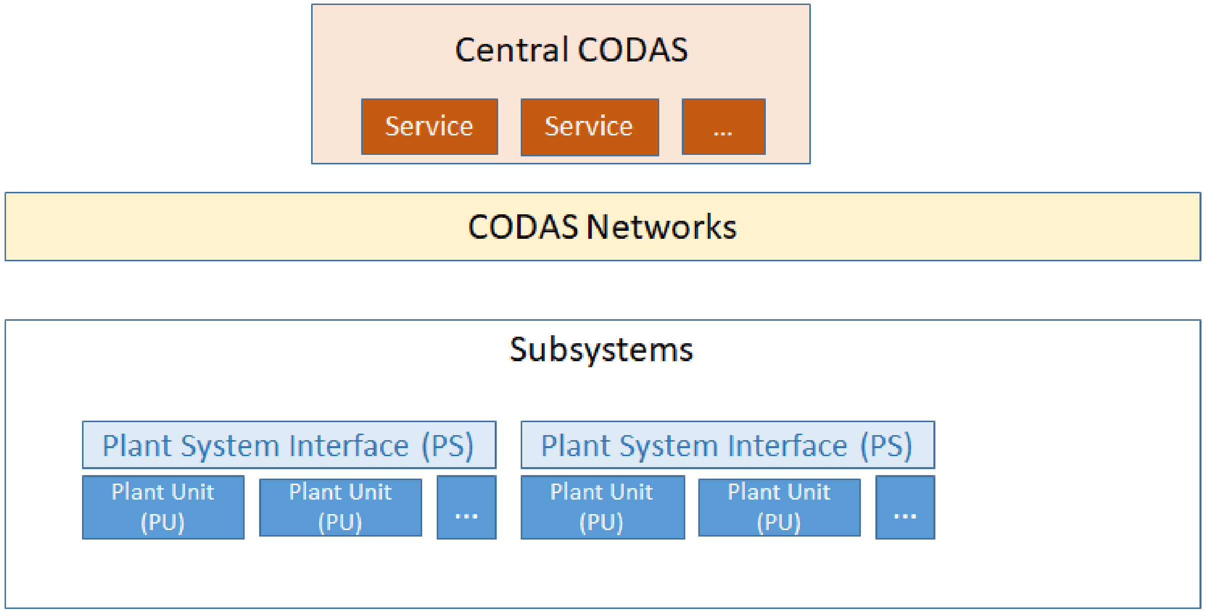
1. **Architecture definition**
   - *What are the major CODAS components and how they interact*
   - *How components synchronize with each other*
   - *The definition of the involved networks and their functions*
   - *How data are transferred, stored, and retrieved*
   - *How to achieve consistent time reconstruction*
   - *The adopted frameworks (if any) for Plasma Control and Data Acquisition*

2. **Plant interface definition**
   - *Decouple users’ views from CODAS details*
   - *Provide a set of rules before plant developments start*
   - *Suggest common approaches in hardware component selection*

3. **miniCODAS development**
   - *A useful tool for Factory and Site acceptance tests*
   - *Test architecture and components on a small scale – rapid prototyping*

CODAS Architecture: Central CODAS and Plant Systems

- **Plant System interfaces (PS) act as the interface between the Plant Units and Central CODAS**
  - *PS decouple plant-specific functionality from central CODAS services*
  - *Most development effort is foreseen in PS implementation*
  - *Good documentation about PS interface is required as soon as possible*
- **Central CODAS services prototyped in miniCODAS**



EPICS vs OPC UA

- EPICS and OPC UA Process variables (PVs) are used in the scientific and industrial environments, respectively, for network-based communication in industrial (slow) control

- EPICS is open-source and widely used in the accelerator community. EPICS and the associated Control System Studio (CSS) have been adopted in ITER

- OPC UA is an emerging industrial standard natively integrated into many Programmable Logic Controllers

- **Is a large fusion experiment more similar to a scientific or industrial application??**
  - ***Industrial***: *Vacuum, Cooling, Cryogenic*
  - ***Scientific***: *plasma control, plasma diagnostics*
  - ***Both***: *Power supplies*

- **In DTT, OPC UA has been selected for communication between industrial components**
  - *Unlike EPICS, OPC UA is well-known in the industry*
  - *Siemens WinCC OA is the selected SCADA system, natively integrating OPC UA communication*

Frameworks
Do not reinvent the wheel

- **Software Frameworks available in Fusion community**
  - *Rely on the experience gained in different machines*
  - *Provide well tested components*

- **MDSplus used for data management**
  - *Used in several experiments*
  - *Actively supported*
  - *Rugged concepts, up-to-date implementation*

- **MARTe2 used for real-time control**
  - *Relies on JET and other experiments experience*
  - *Actively supported by F4E*

- ***NOT a ready-to-use solution***
  - *A philosophy plus a set of ready components*

Framework integration

- **MDSplus, MARTe2 and WinCC OA have been integrated**
  - *Integration is required to ensure smooth operation*
  - *Specific framework functions exported to the other ones*

- **MDSplus provides data storage support for WinCC OA and MARTe2**
  - *MDSplus archiver plug-in for WinCC OA in addition to the native InfluxDB for slow signals*
  - *MARTe2 Data Source for MDSplus used for real-time data archiving and reference signals retrieval*

- **OPC UA communication implemented also by MARTe2 using open62541 library**
  - *MARTe2 Data Source components for reading and writing OPC UA variables*
  - *MARTe2 service implementing OPC UA server*

- ***High-level view of real-time components implemented in MARTe2 using MDSplus device abstraction***
  - *Automatic generation of MARTe2 configuration from the MDSplus experiment model*

Simulink integration in real-time control

- **Definition of real-time algorithms via Simulink**
  - *Simulink is the lingua franca for control engineers*
  - *Common approach in fusion experiments*

- **Seamless integration of the code generated by Simulink coder**
  - *A solution that does not depend on the Simulink version*
  - *Based on the introspection capability provided by the generated code to inquire parameters, inputs and outputs*

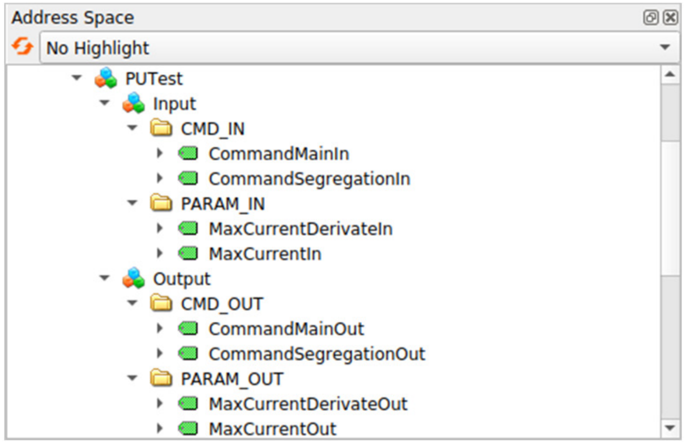
- **Integration performed at two levels**
  - *A single MARTe2 Generic Application Module (GAM) for the integration of any Simulink algorithm*
  - *Automatic generation of MDSplus devices wrapping the MARTe2 Simulink component*
  - *In all cases, generation is driven by the Simulink generated code itself –* ***No manual coding***

Plant Interface definition: industrial control

- **It is important to hide as far as possible CODAS implementation details from CODAS users**
  - *Clear separation of responsibility*
  - *Unambiguous interface*

- **Work in progress in the definition of a reference document to be included in tenders**
  - *Uniform interface via mandatory protocols (e.g. OPC UA)*
  - *Suggest common implementation solutions*

- **For Industrial components provide the structure of the exported OPC UA variable tree**
  - *A flexible naming convention is adopted*
  - *For commands prescribe double handshaking protocols*



Plant Interface definition: fast signals

- **For signals requiring a sampling rate greater than 100 Hz, an different solution from OPC UA must be defined**
  - *For all PLC-related signals OPC UA is enough*

- **UDP (unicast or multicast) will be used to communicate signals requiring a sampling rate up to 10kHz**
  - *This includes all fast signals involved in Plasma Control*
  - *A common header structure for all plant systems*
  - *Messages used also for synchronization*

- **Different solutions for faster signals**
  - *ADC and DAC converters*
  - *Synchronization carried out by PTP aware timing*
  - *Proprietary digital protocols discouraged*
  - *A standardized solution preferred (F4E ongoing activity)*

- **Camera data acquisition likely to represent a big issue in data management**

Plant Interface definition: machine protection

- **Plant interface must also include signals exchanged with the Central Machine Protection (MP) System**
  - *Central Machine Protection system is not presented here*

- **General rule: every plant system is responsible for its own protection**
  - *The exchanged machine protection signals are used only when plant failure affects other plants*

- **Only digital signals over fiber optics shall be used**
  - *Using fail-safe communication*

- **At least the following MP signals are expected**
  - *Enable (from Central MP system)*
  - *Fast shutdown (from Central MP system)*
  - *Alarm (to Central MP system)*

mini and micro CODAS

- **miniCODAS represents a small scale, but complete, CODAS installation to be used in Factory and Site acceptance tests**
  - *Successful experience with ITER miniCODAS at ITER NBTF*
  - *Good candidate for rapid prototyping of CODAS components*

- **A first implementation using an Industrial PC and D-tAcq front end (ADC, DAC, DIO)**

- **microCODAS is an even smaller scale implementation**
  - *Fast delivery of Factory Acceptance tests*
  - *Implemented on a laptop*
  - *Includes STM32 and RedPitaya boards as front end*

Thank you for your attention

questions?