Control of Nuclear Fusion Experiments
Gonçalo Nuno Cerqueira Olim Marote Quintal
Instituto Superior Técnico, Lisbon, Portugal
November 2014

Abstract
This thesis is part of the development of a new slow control system and its nodes. This system is composed of a control unit and three nodes. The system also archives the data acquired by the nodes.

The software’s architecture is based on ITER’s concept, the EPICS framework is used to publish data on ISTTOK’s internal network. The CSS software was used to develop and run the new GUIs.

One of the nodes controls the gallium system, composed of six temperature sensors and four valves. The second node receives data from two pressure gauges, the first monitors the pressure inside ISTTOK’s chamber and the second monitors the pressure between the primary and secondary pumps. The third node, developed in this thesis, acquires data from four thermocouples installed along ISTTOK’s shell, giving the temperature difference between the shell and environment. This last node was projected to also acquire the charge data of ISTTOK’s capacitor bank but due to some problems it is not ready to be use.

At present, the system has the required software running but more work is necessary in order to integrate all elements of the full slow control system. The state machine needs to be improved, in particular with more feedback and the output implementation. It is also necessary to implement the interaction with the fast control system.

Overall, this thesis’s goals were achieve, nominally the creation of a control unit with all the basic software installed and the integration of a few nodes within the system.

Keywords: Fusion, Control, Tokamak, Plasmas, ISTTOK

1. Introduction
1.1. Thesis Aim
This thesis is part of a larger project to create a new slow control system to replace the system currently in use at the ISTTOK. This project aims to implement a new prototype system, taking special care to overcome some of the limitations of the current system. In specific, this thesis deals with the core framework and some complementary software for the new system. The goal for this prototype is to integrate some peripheral nodes and develop a GUI to display the data gathered by the nodes.

1.2. Foundations of nuclear fusion
The basic principle in nuclear fusion is the fusion of two nuclei into one and during this process there is a release or abortion of energy. The first clue to this phenomena was provided by Einstein in 1905 with his famous relation [1] between energy and mass, equation 1, derived from his special theory of relativity.

\[ \Delta E = \Delta m \cdot c^2 \] (1)

The fusion concept uses the last equation in it concept, the fusion process using hydrogen and its isotopes are the the fusion process with most yielding. Presently our knowledge shows that equation 2 is the easiest and with the most yielding between several hypotheses using hydrogen.

\[ _1^1D^2 + _1^1T^3 \rightarrow _2^4He^4(3.5MeV) + _0^1n(14.1MeV) \] (2)

To be able to create this fusion reaction, several conditions must be met. In stars, the gravity force is able to compress the hydrogen to densities and temperatures so high that allows for the creation of a hot plasma and the ignition of a nuclear fusion reaction in a steady state. On earth, the gravity force is not enough to achieve such extreme conditions, otherwise we would not be here.

Several ways are being studied, the most promising is the one is the technique using magnetic fields to confine the fusion plasma, allowing it to achieve the fusion conditions. In the engineering approach to fusion several machines architecture were developed. Today the most promising architecture is tokamak, figure 1.

In 1998, the JET[3] a machine that uses the tokamak design achieve the today’s world record in power production of 16MW[4]. Though not enough to have an ignition, the ignition achieve when the
the machine is at least producing enough energy as the energy spent to maintaining it turn ON. The next step in this scientific area is ITER project, this projects aims to construct a tokamak capable of achieve the ignition.

1.3. ISTTOK

ISTTOK is a small tokamak that started operating in 1993 at IST, its parameters can be seen in Table 1. This small tokamak [5] gave to Instituto de Plasmas e Fusão Nuclear (IPFN), at the time called Centro de Fusão Nuclear (CFN), the opportunity to test fusion related diagnostic systems. Thus allowing a broader Portuguese participation in International fusion related projects. It is also an attraction pole for physics students in a university environment, providing basic training in fusion and plasmas.

<table>
<thead>
<tr>
<th>Table 1: Main parameters of ISTTOK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Major radius</td>
</tr>
<tr>
<td>Copper shell radius</td>
</tr>
<tr>
<td>Minor radius</td>
</tr>
<tr>
<td>Toroidal magnetic field</td>
</tr>
<tr>
<td>Plasma current</td>
</tr>
<tr>
<td>Central plasma density</td>
</tr>
<tr>
<td>Electron temperature</td>
</tr>
<tr>
<td>Transformer flux</td>
</tr>
<tr>
<td>Standard discharge duration</td>
</tr>
</tbody>
</table>

ISTTOK started installation in 1990 with the acquisition of the several parts from TORTUR tokamak and other parts were developed in IPFN. In the table 1 are stated the ISTTOK main parameters.

2. ISTTOK Control

The control system is a solution that enables us to control and operate all the machinery in very small scales of time and/or operate it for long periods of time, which would otherwise be imposable.

In a tokamak there are several parameters that would be imposable for a human to have direct control over due to their time scales, an example is the plasma’s position. On the other hand, a tokamak typically needs to always be at a ultra high vacuum, this means that the control system should always be monitoring the vacuum status.

ISTTOK’s control system and data acquisition system were projected [6] to have three main operation modes, Figure 2, a mode to start, shut-down and for vacuum maintenance, another one to perform cleaning discharges and a last one to perform the power discharges.

Figure 2: State machine that is implemented and running in ISTTOK for its operation modes, from [7]

The ISTTOK control system can be separated into several systems working together, the plasma control or fast control, the technical control or slow control, the FireSignal, the Triggers and the data storage handled by the FireSignal.

The slow control system or technical control presently installed on ISTTOK [7] uses a vacuum control system, the EDWARDS 2032, that proved its reliability since its working from 1991 without majors problems. The vacuum unit is equipped with an Intel® 8085 8-bit HMOS microprocessor with 6.36KBytes of RAM memory, a EEPROM20 with 8.19KBytes and serial port. The communication with the fast control system requires the use of the trigger system and the FireSignal.

The stated machine programmed in the slow control unit is the depicted in Figure 3 is represent the state machine that is program in the slow control unit.

In the PROCESS stage programmed is executed by 24/7 and is responsible for maintaining the ISTTOK’s pressure at $10^{-7}$mbar this stage is represents
the high vacuum sub-stage of vacuum stage.

In cleaning discharge state is to clean the primary chamber from impurities, that typically get embedded in shell during higher pressures. Before changing the slow control state with the command "E", the operator has to perform some tasks. In the cleaning process, the shell is heated by a low temperature plasma that is not well confined and by a heating blanket. During this process the operator sends several commands to the control unit control to initialize the necessary systems.

In power discharge stage is necessary to use fast control system. It is also in this state that is created the plasma with scientific relevance. To begin this a power discharge, operator gives a command the "B" to the slow control unit and it transits to the state WAIT-VME. After the fast control system acknowledge and configures all its systems according to the operator instructions it sends a signal to the slow control through the FireSignal and the VME triggers system, from this point the system is fully automated.

The fast control system utilizes the ATCA® technology [9, 10] that was devolved recently in the laboratory and is composed by a board [11] capable of real-time control and two acquisitions board. This system is resposable for several diagnostics, main rogowski coil, Mirnov probes, electric probes, tomography, sine probe, cosine probe, h-alpha bolometer, interferometer and loop voltage.

This control system gives the operator several options to be configure through two objects named Discharge Configuration and Advanced Configuration. This objects are access remotely and are available in the ISTTOK’s internal network with a browser.

The middleware used is the FireSignal [12] and system of triggers [13].

The FireSignal was design to control and operate physics experiments in a full modular concept and avoids dependency on a particular technology. Each hardware client uses the plug-and-play philosophy to connect to the FireSignal central server, programmed in Java. It is this central unit of middleware responsible for managing all the commands, for the system configuration, access to the database and data broadcast. For the users it exists a graphical user interface (GUI) access to the data in a cooperative environment.

The interaction with the slow control is very limited and presently only two signal are exchange and there is no data record from this control unit. The slow control unit receives a trigger signal from the FireSignal, this signal is send to it through a trigger signal that was developed to synchronize several systems. Presently this is the only input send to the control unit.

However for the fast control system was developed and installed a driver in the three boards, for the FireSignal access to all data stored in the fast control system at the end of each discharge.

2.1. ISTTOK’s Slow Control Future

2.1.1 System Limitations

The current slow control system is outdated. As mentioned previously, the programming language, EVPL, lacks modern capabilities of high level programming languages. The interaction with the operator is done through a terminal, also lacking a GUI.

The interaction with the FireSignal is minimal and is based on a triggering system during the power discharge. There is no storage of the data acquired in this unit, an example is the background pressure data during the vacuum stage or the cleaning discharge state.

The control unit’s expandability is limited to its control bus that is fully occupied and its processing capability limits the instructions’ complexity. As mentioned it has an Intel® 8085 8-bit HMOS microprocessor with 6.36KBytes of RAM memory, which compared to the today’s systems is extremely outdated.

The slow control unit is also based on a centralised philosophy. This means that if it is required an intervention to this unit the all its sub-systems have to stop consequent ISTTOK has to stop.

Finally, this unit was discontinued by its manufacturer, implying that in the event of a malfunction, it would not be possible to replace, and even simple repairs can be challenging.

Due to these several limitations, ISTTOK requires a new control system that can overcome all these limitations and that ensures support and simplifies compatibility problems in the development of
new systems for other tokamaks.

2.1.2 ITER Approach
The ITER project is being developed by several countries that together represent more than half the world. Consequently, integration represents a major challenge to the project. The core system is being developed by the host, the ITER Organization, and must be prepared to support interfaces to control the local systems created by the member states.

A global architecture [14] was developed, with standards, protocols and methodology to have better process integration and automated operation from a central location. The architecture developed for the control system is called Control, Data Access and Communication or CODAC [15, 16] and will provide continuous supervision, data monitoring, visualization, storage and handling, alarm handling, error logging, plant’s system automation, operation state management and schedule management, automated pulse execution and real-time plasma feedback control functions for the overall ITER operation.

The chosen solution to this problem was the use of a common software framework, interface and GUI which allows all device systems to be independent of the hardware where they are running. The used framework is called EPICS [17] and uses for communication middleware the channel access (CA). Control System Studio (CSS) was opted as the central GUI development framework.

This framework also has integration with the MARTe framework [18], which means a deeper integration would be possible between the fast control system and the slow control.

Opting for using these technologies also brings ISTTOK nearer to the ITER fusion community, with its support and increasing the compatibility between ISTTOK’s system and ITER’s system.

3. New slow control software
The most important requirements for the new slow control system are reliability, resilience, external connectivity, modularity, compatibility with the installed systems and easy programmability. On the downside the installed system is old and its core is a bought system whose production and assistance have discontinued.

A new system is required to overcome the downsides of the current system, it should reunite all the necessary traits already described. Figure 4 represents a schematic of a system running experimental physics and industrial control system (EPICS) and other tools, inside the blue boxes are future possibilities for expansion.

The installed prototype control unit, and currently running, has three peripheral nodes, depicted in the figure, these are the gallium node, the temperature sensors node and the pressure sensors node. These nodes are using the RS-232 protocol and are further explained in Section 4.

![Figure 4: New control system scheme. The computer with a monitor represents a user terminal, the control unit is where EPICS is installed and the integrated circuit with the name dsPIC30F is the IPFN dsPIC board.](image)

3.1. System Structure
Returning to Figure 4, channel access (CA) is the protocol that all communication is based on. This protocol works in a higher layer, with the TCP/IP and UDP protocols behind. It uses the client/server and publish/subscribe techniques, which allow sharing of information between several hundred computers.

EPIC framework implements and uses this CA protocol to communicate with others units. The PV are variables in this framework and they are composed of several fields, each field with a particular information about the PV.

This framework is also prepared to have software modules and extensions. Typically the modules give EPICS new capabilities, like asynchronous communication, and the extensions are extensions to other software, such as the extension to MATLAB or LabVIEW.

The new slow control unit, that works as a server in this framework, has four input output controllers (IOCs), a data archiver, the Best Ever Archive Toolset, yet (BEAUTY) and an alarm handler, the Best Ever Alarm System Toolkit (BEAST).

The operator terminal also has an instance of EPICS running to access to the PV, though this instance is integrated in the graphical suite, the Control System Studio (CSS). CSS is a collection of tools: Alarm handler, archive engine and oper-
ator interface (OPI). Though some of its tools are in reality individual applications, the Alarm Server and the Archive Engine are examples of stand-alone applications.

3.2. Serial Communication

The peripheral nodes present in this prototype are using the RS-232 protocol. Due to this requirement it was necessary install the Asynchronous Driver Support (asynDriver) module in EPICS. In this project, the asynDriver was used in development of the device driver to communicate with IPFN dsPIC board version 2.

This driver is divided into files one to configure driver and the other with the source code of the driver itself. With this structure the file with the driver source code can be used to create several processes, one for each IOC linked to a different port. The message protocol implemented has the following structure, "Code_Number Code_Number (...) UP_Number CRC", in table 2 has all codes currently implemented.

Table 2: The codes of the developed message protocol exchange between the control unit and the peripheral nodes.

<table>
<thead>
<tr>
<th>Message code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td>The value followed is a pressure</td>
</tr>
<tr>
<td>PRD</td>
<td>The value followed is a wave period</td>
</tr>
<tr>
<td>TE</td>
<td>The value followed is a temperature</td>
</tr>
<tr>
<td>UP</td>
<td>The value followed is the running time of the device</td>
</tr>
<tr>
<td>VL</td>
<td>The value followed indicates a valve state</td>
</tr>
</tbody>
</table>

3.3. State Machine

To implement a state machine in EPICS, it is necessary to install another module, the State Notation Language (SNL), also known as the Sequencer, which consists of the SNL compiler and runtime system. This is a set of tools that provides a human readable programming language and is smoothly integrated with EPICS base, depending and building upon it.

With SNL the created program is structured as a set of finite state machines, called state_sets. These states are defined in the IsttokSeqExec.stt, IsttokSequenceExecution.stt and PulseSequenceExecution.stt files located in the src folder, Figure ?? for reference. In turn, the finite states are defined under conditions (when). Once one of the conditions is fulfilled, the state changes to another state, respectively executing the programmed actions once the transition is triggered. Several states were created, Figure 5, this is a conceptual state machine working as a prototype. All the transitions in this state machine have as origin an order given by a user.

3.4. Archiver

The chosen archive system is one of the most popular for EPICS, BEAUTY. This system is part of the built-in tools in CSS, and was developed as a replacement for the Channel Archiver.

Figure 5: State machine of the new control system prototype.

Figure 6: Information flow in the Archive System.

The BEAUTY system records data from the front-end computer through the CA and stores the data in a MySQL database. Figure 6 represents a flow schematic of the archive system. A user was created in the database with read/write permissions to be used by the ArchiveEngine otherwise the ArchiveEngine could not access the database. This engine also supports other types of Relational Databases (RDB), like Oracle or PostgreSQL.

The archive system has an xml configuration file where the PV that should be stored is defined. The PVs are being saved every 30s or when the last saved value has a high difference from the one published in CA from the server. This difference is defined in a PV field called ADEL.

The databrowser is the CSS viewer for database and the configuration only needs the user credentials to access the database, this user typically only has permission to read.
3.5. Alarm Handler

The tool used for this system is the Best Ever Alarm System Toolkit or BEAST. It is based on the original EPICS alarm handler, ALH, combined with new ideas from the CSS project.

Process Variables are used as Alarm triggers. The way this works is that when the PVs are outside a certain predetermined range of values, a minor severity occurrence is triggered. If these values go beyond a broader range of predetermined values a major severity occurrence is triggered.

These limits are defined in the IOC’s by the user through the HIHI, LOLO fields and the HIGH, LOW fields. The first two fields define the major severity range, and the last two the minor severity range.

The Alarm Handler, besides detecting PVs and triggering the alarm, also manages this alarm, i.e. it keeps the alarm on, until a user acknowledgement, or the return of the values to the appropriate ranges, and can generate a system occurrence Log message.

3.6. The Graphical User Interface

Control System Studio is a collection of tools: Alarm handler, archive engine, as well as operator interface (OPI) used to in the development of the Graphical User Interface (GUI). To install it is a simple operation of zip extraction and copy to the desire folder, when running it the user will asked for the working directory, this directory holds the developed OPIs, scrips and all the files necessary to run the GUI.

This program is a suite of several tools, as already stated, it also has several layers running. One of the layers is the Data Access Layer (DAL), it is this layer that accesses the CA and manages all the publishing and subscribing, simplifying the user’s and developer’s tasks. The DAL can also be used to access other protocols, different from CA, however this capacity was not explored.

There were created several OPIs: one for gallium experiment enabling to a user to control the four valves in the experiment and monitoring six temperature sensors; another for gallium experiment with its schematic holding the information of the monitoring values; one OPI to substitute the pressure monitor shown in one of ISTTOK screens; an OPI to control the state machine and with an information resume of all important information currently acquire; and finally one last OPI, to work as a menu to navigate through the OPIs. All the OPIs were developed using the BOY toolkit.

The archiving process already discussed is installed and implemented in the server. But the BEAUTY toolkit has another feature, the data browser a built-in GUI that permits access to the ISTTOK archive located in the EPICS server.

4. Hardware Platform Description

The control unit running EPICS and working as a supervisor/central node, is an Intel® Atom™ CPU 330 @ 1.60 GHz, Dual Core, 1 Gbyte of random access memory (RAM), 4 Serial Ports and two Ethernet ports. It is connected to internal ISTTOK private network and to a peripheral system composed by 3 IPFN dsPIC board [19] by RS-323 protocol.

![Figure 7: IPFN dsPIC board version 1](image)

These 3 peripheral nodes have an eurocard format with several interfaces, in particularly the RS-485, RS-323 ones and as well a DIN96 eurocard connector. The input power supply is an unregulated DC 5 V to 35 V with a power supply regulator allowing other output of 5 V at a maximum of 1 A. The micro controller is a Microship dsPIC30F4013 with several 10bit ADCs that can access 4 multiplexed sample and holders. The second version of this board has a dsPIC node with power h-bridge and Microship dsPIC30F4011 support. The third version is similar to the second version although its dimensions have been reduced to 100x100mm. These two micro controllers can be programmed in C using MPLAB with its compiler, and the Pickit 3 to program the device.

This board was choosen for its versatility, availability and low cost and fulfils all the integrated nodes requirements.

5. Integrated Peripheral Nodes

5.1. Pressure sensors

ISTTOK has mounted two Pfeiffer MPT100 pressure sensors, which have a range of $5 \times 10^{-9}$ mBar to 1000 mBar. These sensors allow the software and the operator to monitorize the pressure in ISTTOK’s chamber, and between the two vacuum pumps.

The micro controller acts as a master when communicating with the two sensors, the slaves, by the RS-485 protocol, Figure 8. The ISTTOK chamber sensor has the address 001 and the one between the vacuum pumps has the address 002. Every 500 ms
the micro controller asks for values first to 001 sensor, then waits for the new value. After receiving
the value of the 001 sensor, it asks to 002 sensor for his value and again waits for a new value, this way
the risk of collisions is avoid.

After all data have been processed it will be automatically sent to EPICS with the protocol describe
in Section 3.2.

5.2. Temperature sensors
The ISTTOK has 4 sensor working of type K thermocouples. Note that there are more thermo-
couples installed, though they are not working at present time, thought system is prepare to with-
stand their possible use. Currently, only one is being monitored with a multimeter and its data are
not being stored.

It was developed a board that aims to improve this monitorization, it connects the thermocou-
lies to the IPFN dsPIC board, Figure 9, which is responsible for the signal digitalization and data
transmission to EPICS. Every 500 ms the micro controller toggles between reading the ADC read-
ing or sending the store data to EPICS.

The board has 100x100mm and it is by the IPFN dsPIC board power supply with 5 V and 4 mA. It
has 8 differential input channels, it is entirely analogue and prepares the thermocouple’s signal to be
digitalized by IPFN dsPIC board, which would otherwise be impossible due to the thermocouple’s low
voltage output. The board output has a range of 0.4 V to 3.5 V, though during the ISTTOK operation
it is expected a maximum of 2.7 V, this last value represent the maximum temperature that ISTTOK
achieves.

Figure 10: Acquire signal in EPICS from the thermocouple system

The signal in EPICS is depicted in Figure 10 and from the analyses of the acquire signal it was deter-
mine that this system has a 1°C, 8-bit, precision.

5.3. Capacitor Bank (ELCO)
The ELCO is the power source for primary field and was updated recently [20]. Currently the slow
control system and ISTTOK’s operator do not have feedback from this system therefore an independent
system was developed to measure the bank’s voltage. The connection scheme is similar to the ther-
ocouple system, Figure 9, only instead of having thermocouples in the input, it has the capacitor
bank terminals and the output of the this developed board is an optic signal. It was opted for using
an optic output for insulation purposes between the developed board the IPFN dsPIC board.

Due to the fiber optic requirement for this project were developed two boards, one board is called the
transmitter board and the other the receiver board. The transmitter board is responsible for reading the
bank’s voltage, for modulated the input reading into a pulse-position wave and for sending the signal to
the receiver board. The receiver was developed has the dimensions of 100 mm × 50 mm, a female eu-
rocard connector for docking into the IPFN dsPIC board without significantly increasing the size. This
last board is also used to connect the thermocouple board to the node as it is the same node receiving
from both systems.

The data send by the node to EPICS is a value representing a wave’s period of a pulse-position
modulation of the signal received through the optic fiber. However, during the final test to the capaci-
tor board, it was detected an anomalous signal.

In Figure 11 is plotted a discharge of the capacitor bank and as can be observed the signal has an upper
bond and a lower bond. This signal appears to be
Figure 11: This plot as the raw data acquisition from the capacitor bank for one discharge.

as a result of a mix of three distinct signals.

After a meticulous examination of all possible origins of this error, it due of the board’s ground being decoupled from the signal ground and also due to the high amplitude of the input signal. Consequently, these two contributions make the amplifier saturate in both limits simulating a maximum curve and a minimum curve, as the acquire data has.

The best approach to solve is to redesign the input stage of the circuit to a differential amplifier scheme. Other solutions were tried but none succeeded.

5.4. Gallium experiment

Experimental set-up of the gallium experiment [21], it was conceptualized in 2006 and has since been developed and suffered some improvements and its goal was the study of gallium-plasma interactions.

In this project this experiment was integrated in the EPICS system. This experiment already had a dsPIC running and necessary alterations were done only in firmware.

This experiment has 8 temperature sensors and 4 valve, the information from the sensors flows just in one direction between the dsPIC and the EPICS. This isn’t the case for the valves since they need to be control by user/operator through EPICS. In this case the node sends every second the information about valve state to EPICS though EPICS send a message only when is necessary to change the valve state.

6. Conclusions

During this thesis it was possible to developed a system prototype for the ISTTOK slow control system. This system is still in development stage but shows promising signs and is important to ISTTOK to have a substitute.

This system fulfils all the ISTTOK necessities in reliability, resilience and can surpass all the limitations of the installed one. The external connectivity is not only possible but is nativity implemented through CA protocol, connectivity through other protocol is also possible but is require little more work. Presently it has its own database though in future it can be integrated with the ISTTOK database that uses the PostgreSQL technology. This new control system is based on modular concept and distribute control, it supports several control unit sharing information between them. It has compatibility with the installed systems, in the case of slow control communication specially with MARTe there is already integration, it is possible to configure MARTe and read its variables through the CA. It has modern programmable language, all documentation and source code are available as all system is based on freeware though documentation sometimes can is not the best and it terms of supports the ITER community can always give a hand. Another problem of the slow control unit in use is its hardware, however in concept of the new system this problem do not exist, due its software approach. It can run in any computer provide that has an operating system, Perl, GNU make and C++.

In the prototype control unit was integrated three nodes. One was develop on house to measure the temperature of ISTTOK’s shell and voltage of the ELCO. The temperature measuring met the expected precision and with this system is now possible to easily track the shell temperature. The measuring of ELCO’s voltage has a problem with the acquire signal and its electric schematic needs to revise, its problem derives from both grounds are totally decouple some workarounds were tried however without any result, probably the resolution for this problem will be a redesign of electric schematic and change the input stage. The vacuum node and the gallium node are acquiring without any complication for some time now, and their values are also being archiving.

6.1. Future Work

The next step for this project should be the improvement of the state machine with it start deciding and implementing input variables that should work as conditions to change between states. Then the implementation of the output PV should take place, without any connection to world, dummy PV and verify that all work correctly. It also should be test system response in case of abortion or emergency.

The archive system should be upgraded to be integrated with the ISTTOK database. This mean that an interface with the FireSignal mildware has to be develop.

The communication with the fast control has to be also implemented though there are some work
than in this area. This way smoother integration with the fast control system, would be possible.

It possible to developed OPI to be access by a browser to accomplish that first is needed to solve the compatibility problems between the Java, the OPIs and tomcat.

In parallel with each integration is necessary to develop a GUI for it or integrate the new information in a existing OPI.

References