Redesigning a Radio Frequency Control System with TACO
Why and How?

J. Meyer, J.L. Pons, J.L. Revol
European Synchrotron Radiation Facility (ESRF), BP 220
F-38043 Grenoble Cedex
FRANCE

Abstract

During the construction phase the ESRF bought 3 radio frequency transmitters as turnkey systems. Today they still operate according to the original specifications, nevertheless, a decision was taken to redesign the control system of these transmitters.

The main reasons for this decision were maintenance and upgrade problems for electronics and software, the difficulties encountered with the technical evolution of the radio frequency transmitters and an insufficient integration into the accelerator control system.

The goal of the new system was to build an easy to maintain and flexible control system, which can evolve technically, offer full user access to configuration and sequencing and is integrated into TACO (the ESRF control system) and its tools.

This paper analyses the reasons which led to the decision to redesign the radio frequency control system and the special needs we had to take into account in a research environment. Based on the new object oriented radio frequency control system, we will present our analysis, solutions and conclusions, from the hardware interface level up to the graphical user interface.

1 Introduction

Four radio frequency transmitters, with a maximum power of 1.3 MW each, are presently installed at the ESRF. Three on the storage ring and one on the booster.
In 1991 three radio frequency transmitters were bought as turnkey systems [10]. The hardware and software which was bought still operates according to their specifications. The control system still operates reliably and was designed as a very complex and powerful system for the 1991s.
Since August 1997 a fourth transmitter operates on the storage ring [2] with a new control system. The other three transmitters will be redesigned one by one without stopping the ESRF operation.

2 Reasons for a new design

Several reasons led to the decision to redesign the old transmitter control system [1]:

- The company, where the transmitters have been bought, does not exist any more.

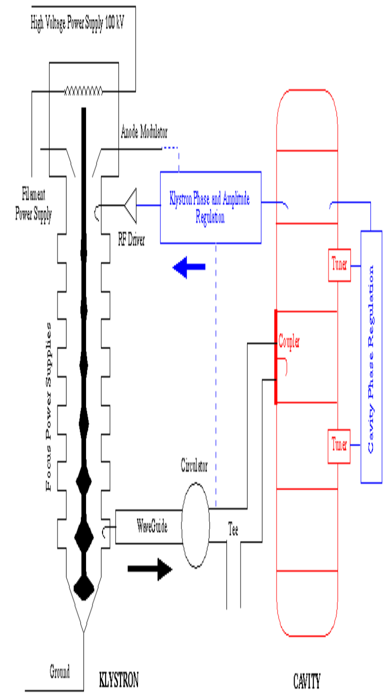

Figure. 1. Scheme of a radio frequency transmitter

- To run the control system with a single 68020 CPU it was optimised for speed. The result is a very complex system with a high level of interconnectivity between all tasks. Code changes are nearly impossible without high risk for operation.
- To optimise the operation speed, no device drivers were used. Direct memory access was implemented. Neither the low level electronics can be upgraded without major code modifications, nor can the OS-9 operating system be upgraded, because memory access is protected in the actual releases.
- Technical improvements or more detailed diagnostics cannot be integrated into such a closed and non modular system. Access on individual hardware parts or sequencing is not possible. At present new diagnostics are build in parallel.
- The hardware protection is a part of the control system.

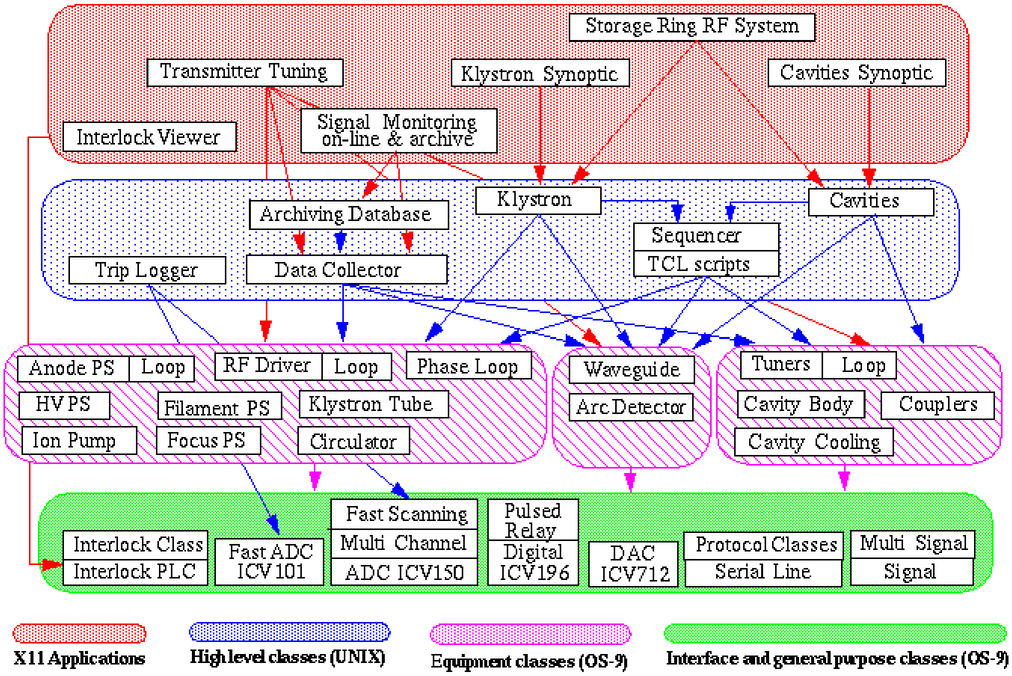

Figure. 2. Control system architecture and class structure

Hardware interlocks and software protection are tightly coupled. Modifications on hardware protection are not possible without detailed analysis of all software interactions in the control system.

- Configuration and diagnostics are only possible with local applications running under OS-9. The information is not available in the control room.
- A very poor integration to the surrounding ESRF control system and its tools. The different operation modes of the ESRF storage ring require more and more detailed equipment access for complex operations.

3 Design analysis

3.1 Why was the life cycle of the turnkey system so short?

After analysing the old transmitter control system design, two major reasons were found for the short life cycle of 6 years.

1. The radio frequency systems have been bought for a production process. The control system was not designed to evolve. Research institutes however need a much higher degree of flexibility. The system must evolve and improve all the time.
2. During the ESRF construction phase, the controls people were not closely enough involved in the radio frequency control system design.
   - The software structure was not defined together with the supplier. A working system was delivered, but the software structure is much too complex to be flexible.
   - When delivered, the implemented software structure was not verified by the ESRF. Only the specified functionality of the system was verified. Today we face a high level of interconnectivity where every modification is a risk for operation.
   - I/O electronics were not specified according to a set of maintainable I/O cards, used at ESRF. Only the real-time operating system and the VME standard were imposed.

3.2 How to guarantee a longer life cycle?

In order not to repeat the same mistakes we drew the following conclusions for the new control system design:

- **Design the new control system object oriented.** This gives the best results for modularity and code reuse.
- **Use in-house standards of hardware and software as much as possible.** TACO (the ESRF control system) [3][4] was used for all control aspects. This ensures a complete integration into the existing ESRF accelerator control system.
  Only standard ESRF I/O cards were used with their TACO interface classes, a set of debugged and maintained classes for the ESRF accelerator and beamline control systems.
- **Do not outsource, but hire extra man power for the in-house development.** This allows a much closer control over design and development. Easy maintenance is possible if in-house people are involved in the project. Technology transfer to an external company is

not necessary. Later changes and improvements can be very efficient and cost effective.

- **Separate hardware protection from the control system software.** It must be possible to shutdown the control system without stopping the transmitter operation. The hardware must be still protected by the independent interlock system.
- **Implement regulation loops as software loops, if the transmitter can operate for a short period of time without software regulation.** To improve the transmitter tuning a high degree of flexibility was requested.
- **Give the system responsibles maximum access to the control system.** 90% of changes are requested for configuration changes, interlock changes, modified hardware interactions (sequences) and logging or trip diagnostic modifications. The turn around time for these kind of interventions is shortest if the users are able to modify the system themselves according to their needs.

4 Solutions

A general overview of the new control system structure is shown in Figure 2. The new radio frequency transmitter control system differs considerably in its implementation from the old system. Some of the more important differences are:

1. The hardware interlock system is based on a PLC which was programmed by the radio frequency group according to control's specifications. A device server [1] reads from the PLC while graphical applications are used to edit and visualise interlock logic diagrams. The hardware responsibles are able to modify and diagnose all hardware interlocks without intervention from controls people.
2. The object oriented design and the use of TACO allowed the reuse of all I/O interface classes, some general purpose classes (Figure 3) and a set of general purpose applications (example: archiver and fast data logger).

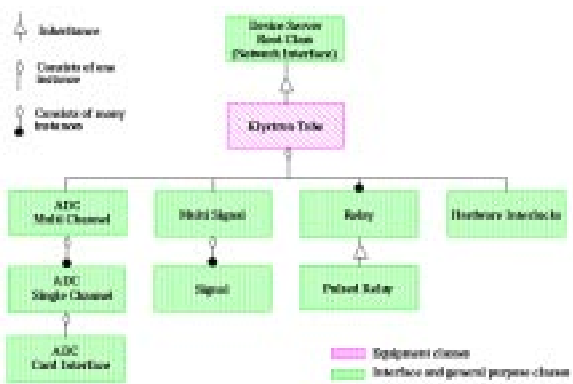

Figure. 3. Klystron tube device server class structure

3. To start and stop a radio frequency transmitter a lot of sequencing is necessary. A sequencer device server was designed to run interpreted Tcl [5] scripts using the existing Tcl interface to TACO [6] (Figure 4). All necessary TCL sequences to run the transmitter were written by the radio frequency group. Such Tcl scripts are simple, interactive and very flexible for modifications.

```
#
# Example Tcl script to switch on a Focus Power Supply
# and to apply a voltage setpoint.
#

# Set the name of the device to access
#
set focus_1 "sr/rf-foc/tra3-1"

# Set the voltage setpoint to apply
#
set voltage_setpoint 10.0

# Read state of the focus power supply
#
set state_value [DevState $focus_1]
puts "$focus_1 : The focus power supply state is $state_value"

# If the focus power supply is in OFF switch it ON.
#
if { ($state_value == $DEVOFF) } {
        puts "$focus_1 : Switching power supply ON"

        DevOn $focus_1
        }

# Apply the voltage setpoint
#
DevSetVoltage $focus_1 $voltage_setpoint

# wait for 500 miliseconds before reading the voltage value.
#
after 500

# Read the voltage value of the focus power supply
#
set voltage_value [DevReadSignal $focus_1 "Voltage"]
puts "$focus_1 : The voltage value is $voltage_value Volt"
```

Figure. 4. Tcl script example

4. Synoptic applications only display information. All intelligence was moved from the operation applications to high level object classes (Figure 5) and flexible TCL sequences. A graphical synoptic editor LOOX Maker [7] was used, which allows to create pretty synoptics and modify them without touching the application's code (Figure 6).

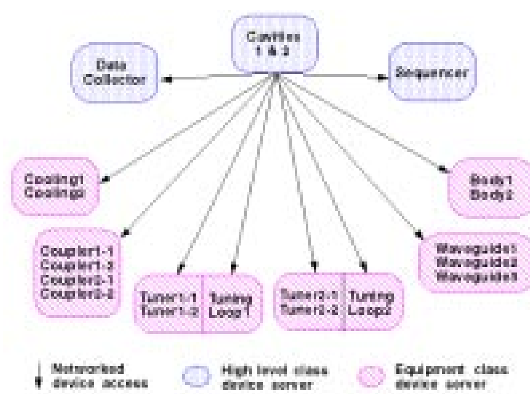

Figure. 5. Cavities devices access structure

5. Off-line diagnostics are implemented in three layers to fit all needs. Data logging after transmitter trips was installed with sampling rates of 200kHz and 10Hz. Archiving can be configured with a maximum sam-

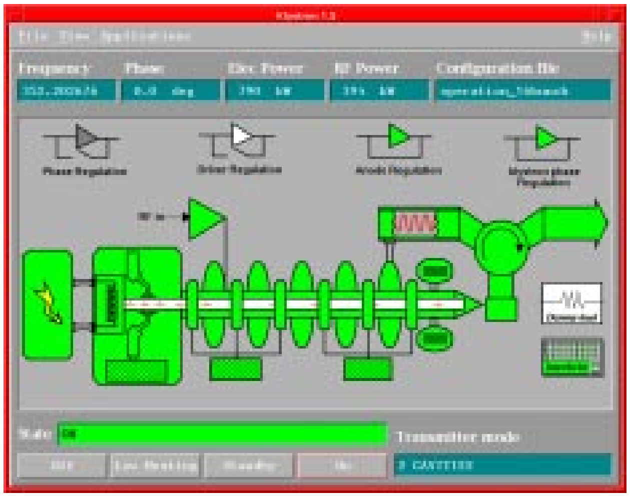

Figure. 6. Klystron synoptic

pling rate of 0.1Hz for trend analysis.
All tuning, monitoring, archiving and trip diagnostics are handled by a set of general purpose applications, which can be configured and adapted by the users to fit their needs.

5 Conclusion

For a project of this size (Table I) in a research environment in-house development is a good approach. The gain is high flexibility, easy in-house maintenance and low costs for further improvements.
With a simplified software development methodology (specification document → design → design review → implementation → tests → documentation) and software standards (coding style, standard object interfaces and version control) good results are possible without high overhead.
The close contact with the hardware responsibles and the possibility to react on specification changes allowed modifications even in the development deadlines.
A similar conclusion was already drawn four years ago with the redesign of the ESRF LINAC control system [8].

Table I
Implementation statistics

Development time	3.5 man years
Costs for VME crate and I/O cards	˜37000 $US
Classes written	36
Classes used	48
Applications written	6
Configuration resources installed	˜3000

Giving the transmitter responsibles maximum access to the control system has shown several advantages:

1. They do not see the system as a black box. By actively changing the system behaviour, they understand the internal structure. Better diagnostics are done in case of system failures.
2. The turn around time for modification requests was reduced. Most of the requests can be treated by the radio frequency group.
3. Easy commissioning with a tight schedule for ESRF operation. Responsibilities were shared with the radio frequency group for interlock installation, software configuration, sequencing and configuration of diagnostics and archiving.

Big spin-off for other TACO based projects. General purpose classes and applications for tuning, monitoring and diagnostics, which were developed for this project, are reused in different other projects for accelerator or beamline control.
The decision to use an open control system structure as a network of TACO objects, on all levels, offers high modularity. Parts of the system can be improved or even replaced without interfering with other objects. In the near future external objects could be integrated with the CORBA [9] standard.

Acknowledgements

The authors would like to express their thanks to A.Beteva, I.Fladmark, P.Pinel and JM.Chaize from Computing Services and to the whole radio frequency group for their help in design and implementation of the radio frequency transmitter control system.

References

[1] “ESRF RF-Transmitter Automation System”, Cap Gemini SCS, CGSCS-Project No. C10024, March 1993
[2] “Construction of a Third RF Acceleration Unit for the ESRF Storage Ring”, EPAC, Barcelona (Spain), 1996
[3] A.Götz, “TACO”, http://www.esrf.fr/computing/cs/taco/taco.html, ESRF, 1995
[4] A.Götz, W.D.Klotz, P.Mäkijärvi, J.Meyer, E.Taurel “TACO: An object oriented system for PC’s running Linux, Windows/NT, OS-9, LynxOS or VxWorks”,PCaPAC Conference, DESY, Hamburg (Germany), November 1996
[5] John K. Ousterhout, “Tcl and the Tk toolkit”, Addison-Wesley Publishing Company, 1994
[6] G.Pepellin, “Tcl/Tk using TACO API” http://www.esrf.fr/computing/cs/taco/tcltk/TclTk_api.html, ESRF, 1996
[7] “LOOX Programmer‘s and User‘s Guide Version 3.2”, LOOX Software, January 1996
[8] J.M.Chaize, F.Barale, L.Claustre, P.Pinel “An Object-Oriented LINAC Control System”, ICALEPCS, Berlin (Germany), October 1993.
[9] “The Common Object Request Broker: Architecture and Specification, Revision 2.0”, *OMG Document 97-02-25*
[10] J.Jacob et al. “Commissioning and Operation of one Booster abd two Storage Ring RF Acceleration Units at the ESRF”, EPAC, Berlin (Germany), March 1992