Finite State Machine Implementation to Automate RF Operation at the TESLA Test Facility

V. Ayvazyan, K. Rehlich, S. N. Simrock, DESY, Hamburg, Germany
N. Sturm, RWTH Aachen, Germany

Abstract

At DESY the Distributed Object Oriented Control System (DOOCS)[1] was developed to solve a variety of control tasks at the TESLA Test Facility (TTF) Linac. DOOCS is designed from the device server level, to control different parts of the hardware, up to the console, where it provides a graphical user interface (GUI) to the operators.

The Finite State Machine (FSM) concept has been integrated into DOOCS to simplify the automation of the accelerator operation[2, 3].

This paper describes the FSM server implementation for automated RF operation at the TTF. The server consists of two main elements – the control manager and a tools section. The control manager is responsible for RF field control. The start up, restart and routine operation of a 10 MW klystron, that currently powers 16 superconducting cavities, is successfully controlled by a DOOCS FSM server. The FSM process includes loop phase measurement and correction, feedforward and feedback parameter adjustments, beam loading compensation, calibrations, and automatic fault recovery.

A variety of automated procedures are available in the tools section, which assist the operators in maximizing the performance of the RF system.

1 INTRODUCTION

TTF makes use of a highly collaborative approach which involves people from all over the world operating the Linac. As not all of them can be experts in TTF operation, a high degree of automation is necessary to relieve the operators from complex but well understood tasks.

Besides, TTF will become a user facility for Free Electron Laser (FEL) experiments shortly and therefore will have to provide stable modes of operation. This requires a highly automated machine because of the complexity involved.

A common approach to automation is the use of Finite State Machines. They constitute a well understood approach that is used in various industrial control systems. Since TTF has its own control system DOOCS, it was necessary to integrate the concept of FSMs into it. This resulted in the idea of implementing a high level FSM with the goal of the "One-Button Operation". The availability of all necessary data in digital form simplified the task significantly.

Outstanding properties of this approach are

- a highly modular design, so that new ideas can be incorporated easily,
- a graphical design tool with code generator as an integral part of the DOOCS framework,
- run-time display of the FSM's status and
- the implementation of operator proven and optimized procedures.

2 FINITE STATE MACHINE CONCEPTS

> A Finite State Machine is a representation of an event-driven (reactive) system. In an event-driven system, the system transitions from one state (mode) to another prescribed state, provided that the condition defining the change is true.[4]

Figure 1 shows a sample Transition (see below) Diagram of an FSM to illustrate the nomenclature used below. The

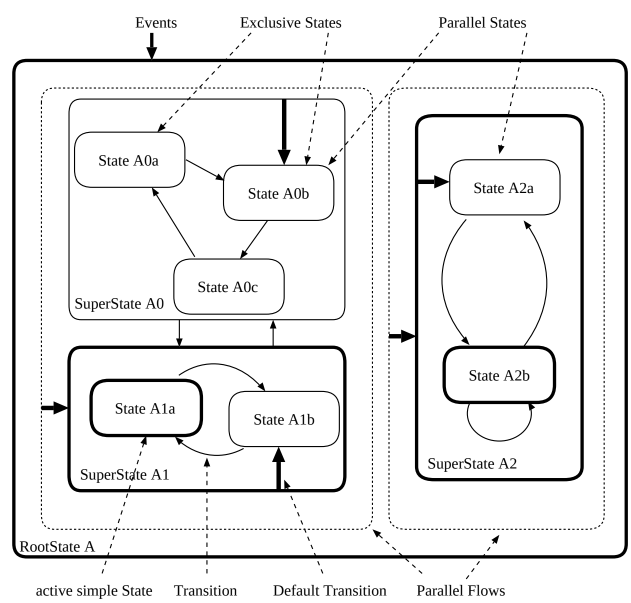

Figure 1: A sample Finite State Machine

base components of an FSM are States. There are different sorts of States called simple States, SuperStates and RootStates.

Simple States are used to describe the different modes of operation a system can be in. To give a full description of a machine's current status, one only needs to give all currently active parallel simple States, e.g., in figure 1 this is State A1a and State A2b.

A SuperState is a subsumption of all those simple States directly connected via Transitions, e.g., SuperState A2 consists of only State A2a and State A2b, since no other simple State is directly connected to these two. If there are Transitions on the level of SuperStates, too, one combines these SuperStates to Flows. The highest level States we call RootStates.

One has to keep in mind, that in reality an FSM will probably grow larger than the few levels one can describe with this nomenclature. Therefore it is absolutely possible, that there are SuperStates inside other SuperStates to create a hierarchy, e.g., State A0c could actually be a SuperState consisting of simple State A0c0 and simple State A0c1.

As already mentioned States can be connected by Transitions. These describe when to change to another State (the given condition has to be fulfilled) and what States one can change to. Since there is no Transition from State A0a to State A1a, the system cannot directly move from one of these States to the other.

Two States are exclusive, if there exists a set of Transitions to get from one State to the other. This is true for all the States in the left Flow of figure 1. The States in the right Flow are parallel to the left ones.

When a RootState or SuperState is entered there has to be some sort of initialization. This is done by default Transitions. If in our example RootState A is activated, the States A1b and A2a are activated in parallel, since the default Transitions (bold arrows) are defined that way and both Flows are entered in parallel.

3 STATE MACHINE IMPLEMENTATION IN DOOCS

The FSM automating the RF system is implemented as a DOOCS server consisting of a control manager and a collection of tools. States are implemented as objects having access to the following kinds of functions:
ENTER() for taking actions when entering the State,
DURING() for periodic actions being taken while in the current State,
<event>() for actions taken, when the specified event occurs,
EXIT() what to do when leaving the State,
<cond:act>() for actions taken, once a condition proves true.

A GUI was implemented, allowing the designer to create an FSM from scratch or to simply add States and Transitions to an existing FSM. Based on this graphical representation, a basic skeleton of server code for these new States and Transitions is created automatically (see figure 2). The programmer has just to fill in the body of the different functions.

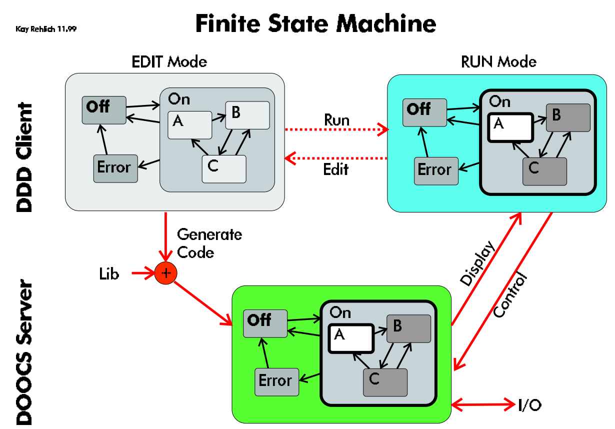

Figure 2: Interaction between client and server side of an FSM in the DOOCS

4 THE DIGITAL RF CONTROL SYSTEM

The RF control system at TTF employs a completely digital feedback system by means of seven DSPs, used to keep perturbations of the accelerating field during pulsed beam operation at a minimum. Major sources of field perturbations are fluctuations of the beam current and fluctuations of the cavities' resonance frequencies because of deformations of the cavity walls induced by microphonics (mechanical vibrations) or gradient dependent Lorentz forces.

For TESLA it was chosen to control the vector sum of 32 superconducting cavities with one klystron. At TTF right now a 10 MW klystron is used to power 16 cavities.

The control system controls the in-phase and quadrature component of the cavity field by calculating and correcting the vector-sum of 16 cavities and consists of a feedback and adaptive feedforward system. The feedforward is used because of strong repetitive and predictable beam induced transients. The feedforward tables are updated adaptively to reflect slowly changing parameters such as microphonic noise level and phase shift in the feedforward path.

5 THE RF FSM SERVER

Figure 3 gives an overview of the RootState of the RF FSM used at the TTF. One sees two parallel Flows for the control manager and the tools. The control manager, responsible for the RF field control, consists of the States

MANUAL for manual RF operation (initial State),

NULL as the main error State,

SAFETY_IL for ensuring the RF system cannot be started while the interlock is broken or temporary access to the Linac is granted and

CONTROL for automatic RF operation.

CONTROL itself is implemented as a SuperState (see figure 4), as can be recognized by the additional line above

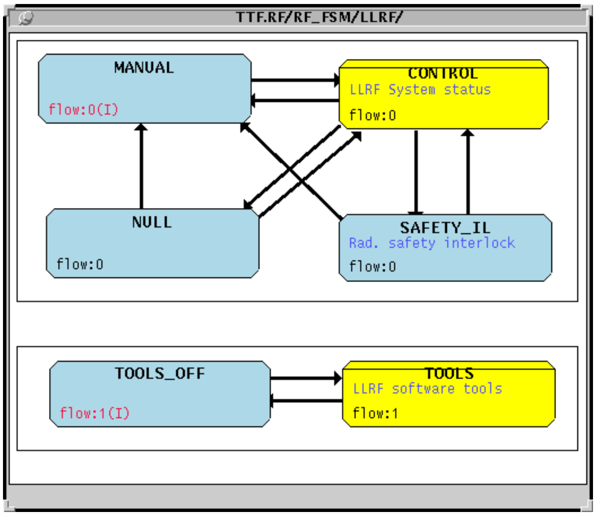

Figure 3: The RootState of the RF FSM with its two Flows as shown by the DOOCS Data Display

the name in figure 3 and consists of a Flow containing the following States:

IDLE: In this SuperState the status of different RF-related hardware is checked and the digital RF control system is initialized, i.e., the DSP server loads the appropriate programs and parameters into the DSPs. Additionally the control and calibration parameters of the RF control system are determined. (initial State)

RF_STDBY: This is the main diagnostics SuperState for RF operation. It checks and resets klystron and module interlocks and collects statistics on the reliability of the components.

HV_ON: In this State high voltage is applied to the klystron, while RF operation is not permitted.

RF_ON: When entering this SuperState the server starts adjusting the feedforward and feedback parameters, checks various other parameters like cavity detuning and loaded Q and takes care of the beam loading compensation.

The tools Flow provides the operators with some automated procedures for maximizing the performance of the RF system. These include for example, tools for cavity detuning measurement and correction.

6 OPERATIONAL EXPERIENCE

For almost one year the accelerator modules at TTF are controlled by the FSM server. The start up, restart and routine operation of our two cryomodules and the RF gun are successfully automated. We increased the up-time significantly, while decreasing the workload of the operators. Additionally the reproducibility of a high quality beam after

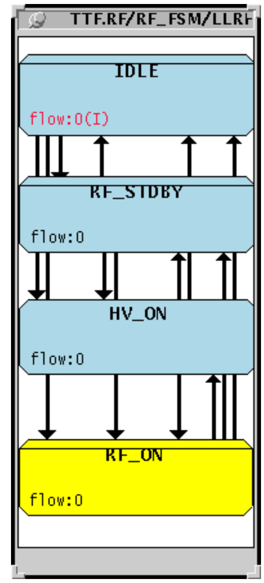

Figure 4: The CONTROL SuperState in the DOOCS Data Display representation

interlock faults was improved. This FSM is an important step on the way to a fully automated, “One-Button” driven accelerator.

7 FUTURE PLANS

Next year TTF will enter phase two by adding another four cryogenic modules each equipped with eight cavities to the Linac. After that TTF will become a user facility for FEL experiments, too. This will require improved stability of Linac operation, which will be provided by the RF FSM server.

8 ACKNOWLEDGMENT

We want to express our thanks to the TTF Operations and Control Groups for their dedicated support and helpful discussions.

9 REFERENCES

[1] G. Grygiel, O. Hensler, K. Rehlich, “DOOCS: a Distributed Object Oriented Control System on PC’s and Workstations”, ICALEPCS 97, Beijing; see also http://tesla.desy.de/doocs/

[2] D. J. Hatley and I. A. Pirbhai, “Strategies for Real-Time System Specification”, Dorset House Publishing Co., Inc., NY, 1988

[3] A. Aghababyan, V. Kocharyan, K. Rehlich, “An Integrated Finite State Machine to Automate TTF operation”, PCaPAC 2000, DESY, Hamburg
see also http://tesla.desy.de/doocs/fsm/fsm.html

[4] “Stateflow User’s Guide”, Version 4, The MathWorks Inc., September 2000, p. 2-2