ICALEPCS 2011

Operational status display and automation tools for FERMI@Elettra*

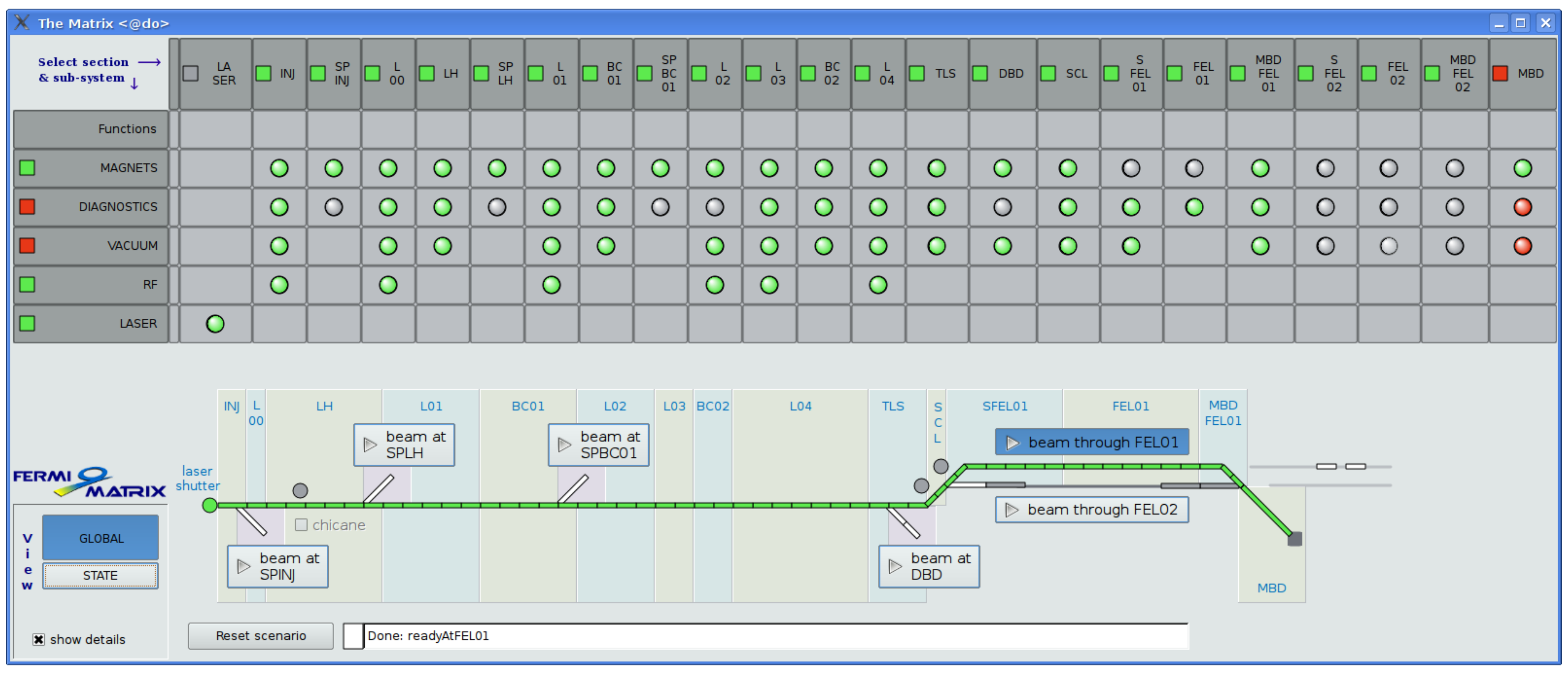
Claudio Scafuri
Sincrotrone Trieste, Trieste, Italy

Detecting and locating faults and malfunctions of an accelerator is a difficult and time consuming task. The situation is even more difficult during the commissioning phase of a new accelerator, when the plants are not yet well known. Faults involving single devices are easy to detect, however a fault free machine does not imply that it is ready to run: the definition of malfunction depends on what is the expected behavior of the plant. In the case of FERMI@Elettra, in which the electron beam goes to different branches of the machine depending on the programmed activity, the configuration of the plant determines the rules for detecting malfunctions.

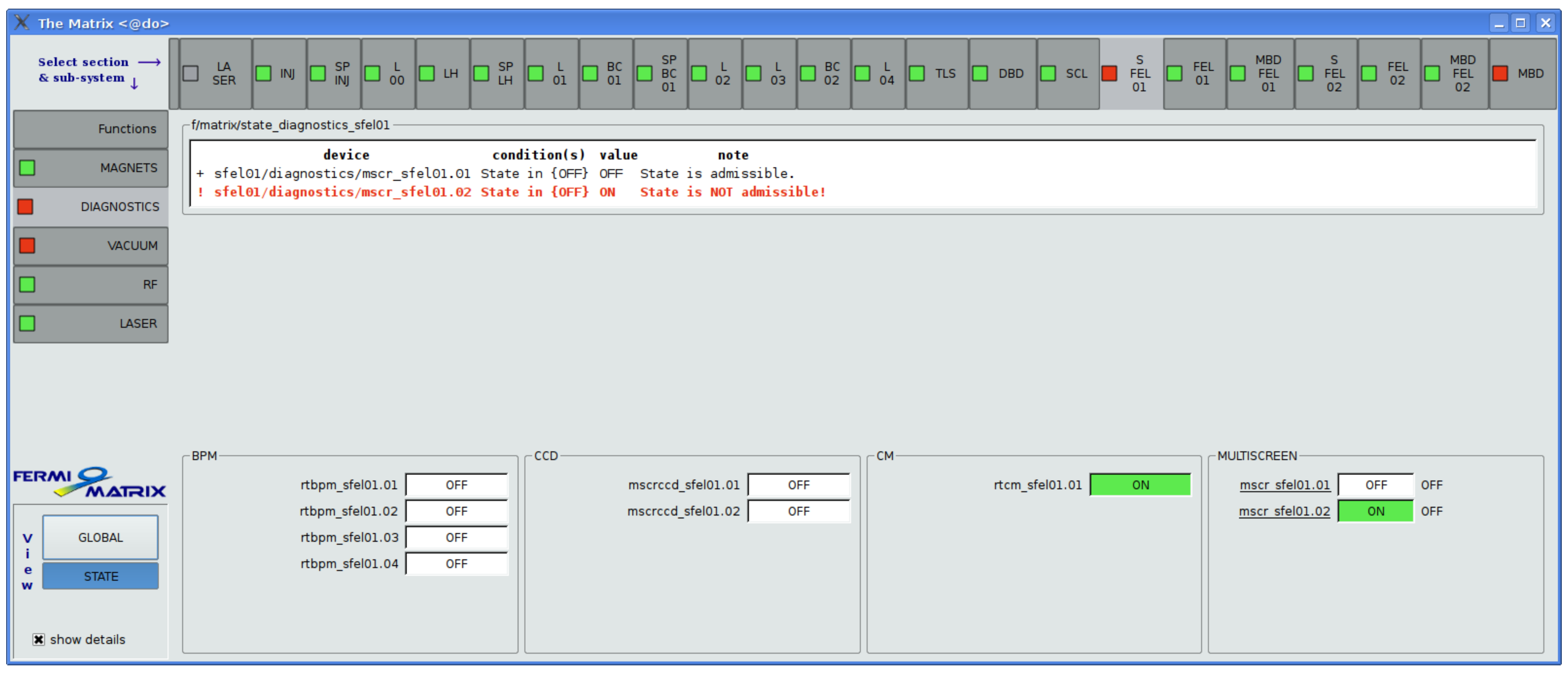
In order to help the detection of faults and malfunctions and to display the status of the plant a tool , known as the "Matrix", has been developed. It is composed by a graphical front-end which displays a synthetic view of the plant status grouped by subsystem and location along the accelerator, and by a back-end calculation engine.

The graphical front-end gives also the possibility, once a problem is detected, to focus on its details.

The calculation engine is composed by a set of objects known as Sequencers. The calculation rules have been determined by analyzing the various subsystems and global working of the accelerator with plant and operations experts. The Sequencer is designed so that it can also issue commands to the plant. This will be used in the next releases of the Matrix for actively switching from one accelerator configuration to another.



The Matrix : view of the accelerator status – active *Scenario* is “beam through FEL01”



f/matrix/state_diagnostics_sfel01

```
                device                 condition(s)  value          note
+ sfel01/diagnostics/mscr_sfel01.01  State in {OFF}  OFF   State is admissible.
! sfel01/diagnostics/mscr_sfel01.02  State in {OFF}  ON    State is NOT admissible!
```

BPM
rtbpm_sfel01.01 OFF
rtbpm_sfel01.02 OFF
rtbpm_sfel01.03 OFF
rtbpm_sfel01.04 OFF

CCD
mscrccd_sfel01.01 OFF
mscrccd_sfel01.02 OFF

CM
rtcm_sfel01.01 ON

MULTISCREEN
mscr_sfel01.01 OFF OFF
mscr_sfel01.02 ON OFF

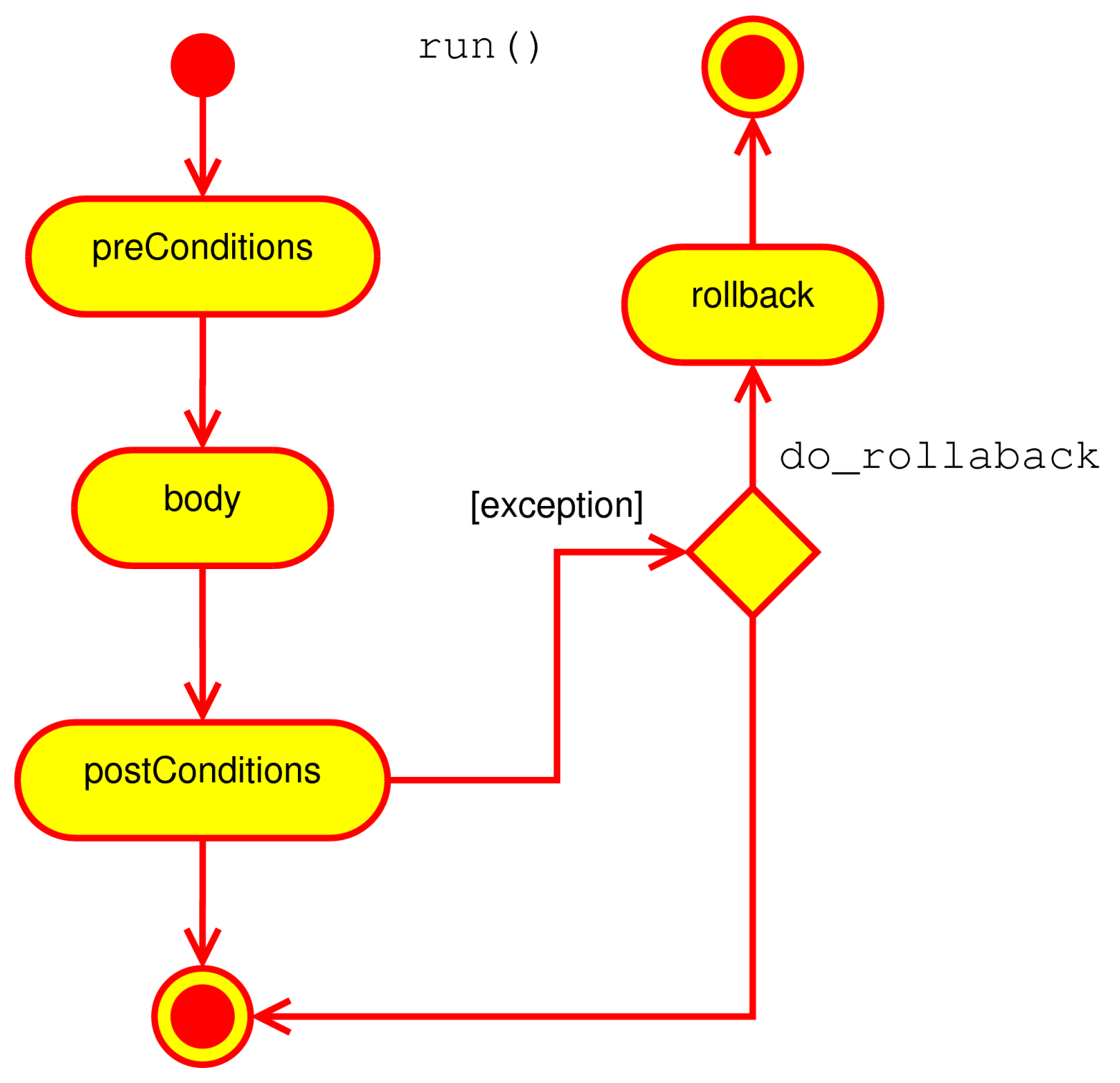
The Matrix : details of a problem

The *Scenario* is a coherent collection of rules used to check that the accelerator is ready to transport the beam to a desired destination, such as one of the various electron spectrometers, or one of the two undulator chains. Each rule is in charge of a well defined section and subsystem of the accelerator. Many of the rules are re-used in different scenarios, reflecting the fact the the electron beam must go through the same sections of the accelerator. The *Scenario* is selected and activated by the control room operator.

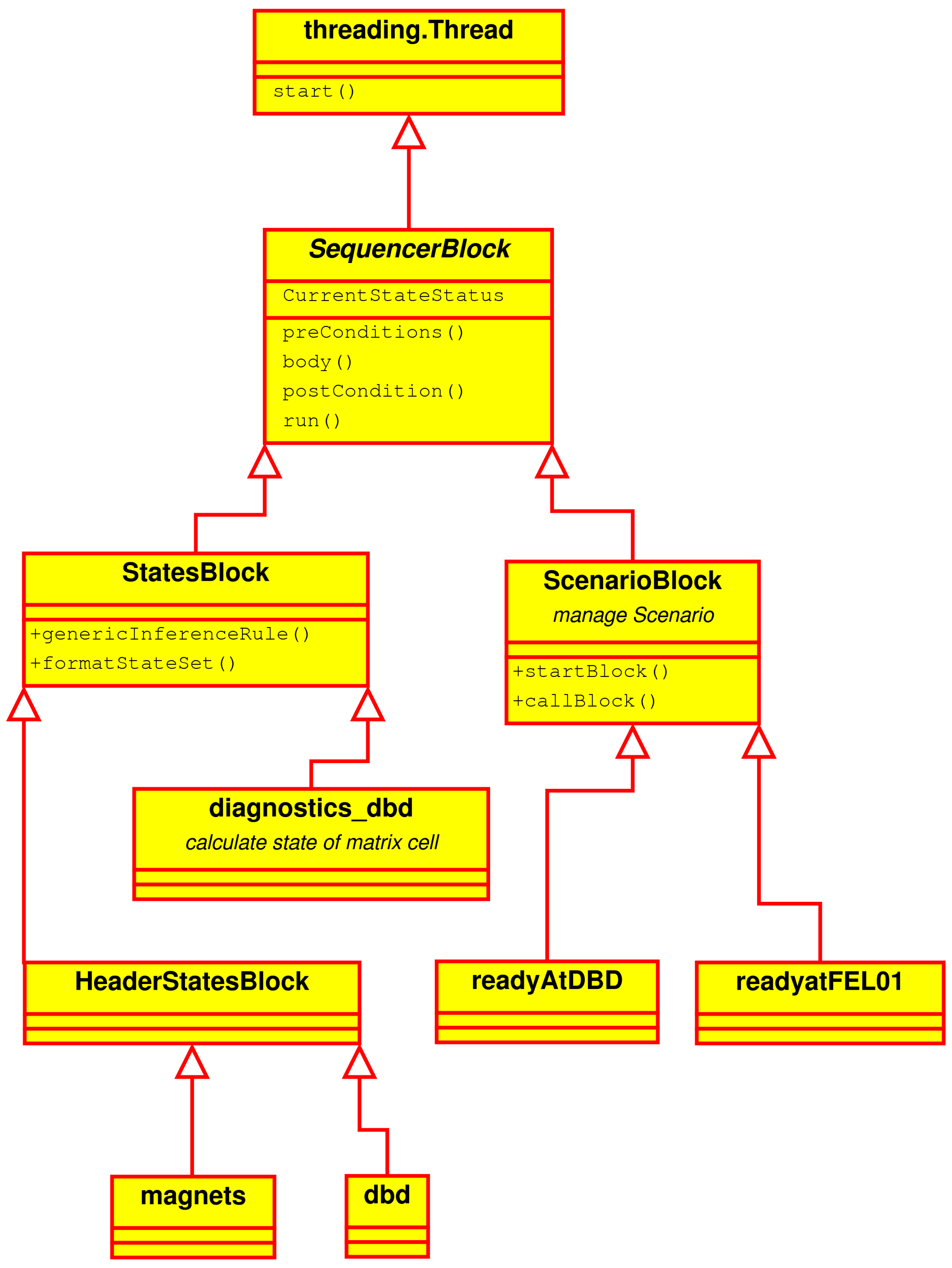
The Matrix needs one *StatesBlock* object for each of its active cells. All these objects are managed by a dedicated Tango device server written in python, which exports a simple and clean interface for interacting with a *SequencerBlock* object. This deployment scheme makes all the stuff needed by the Matrix generally available as standard control system components. The most important technical reason for this deployment scheme is that it ensures that there is exactly one instance of each of the configured *SequencerBlock* running in the control system of the plant: in this way, when the *SequencerBlock* objects will be used to modify the plant configuration they will do it in an ordered and consistent manner.

The Matrix has been used during the commissioning shifts and has shown to be an effective tool for monitoring the general status of the plant. The effectiveness of the Matrix depends strictly on the quality and completeness of the Scenarios: a good collaboration and mutual understanding between machine experts and programmers is of paramount importance in order to correctly and completely analyze and define the various rules. We expect to improve and extend the Scenarios and the diagnostics capabilities of the Matrix with the knowledge gained during the commissioning of FERMI@Elettra.

The next important development of the Matrix will be the automation of operations. As the new accelerator will start operations for FEL light users, standard procedures will be defined for setting up the plant and guaranteeing optimal and repeatable operating conditions. These procedure will be analyzed and implemented by means of *SequencerBlock* objects and supervised by the Matrix. The ultimate goal is to make FERMI@Elettra operable by a team of two control room operators.



Calculation engine: main sequence diagram



Calculation engine: class diagram

The Sequencer is the building block of the calculation engine used by the Matrix. It is developed in python starting from the SequencerBlock class which derives from standard python threading.Thread class. The *SequencerBlock* class overrides the run() method (this is required by the Threading class), implementing the activity diagram shown in UML format. The run() sequence codifies and formalizes the standard steps that must be done in order to verify equipment conditions or actively modify them; it is foressen in fact that Sequencers will be used also to automate the operations of FERMI@elettra such as start-up, shutdown, changing of beam path, etc.

* This work was supported in part by the Italian Ministry of University and Research under grants FIRB-RBAP045JF2 and FIRB-RBAP06AWK3

for additional information, please contact:
Claudio Scafuri : claudio.scafuri@elettra.trieste.it