Enhancing the MxCuBE user interface by a finite state machine (FSM) model

Ivars Karpics, Gleb Bourenkov, Thomas R. Schneider
European Molecular Biology Laboratory (EMBL) Hamburg Unit c/o DESY,
Notkestrasse 85, 22607, Hamburg, Germany

Abstract

The acquisition of X-ray diffraction data from macromolecular crystals is a major activity at many synchrotrons and requires user interfaces that provide robust and easy-to-use control of the experimental setup. Building on the modular design of the MxCuBE [1] beamline user interface, we have implemented a finite state machine model that allows to describe and monitor the interaction of the user with the beamline in a typical experiment. Using a finite state machine, the path of user interaction can be rationalized and error conditions and recovery procedures can be systematically dealt with.

Typical steps of a macromolecular crystallography (MX) data collection

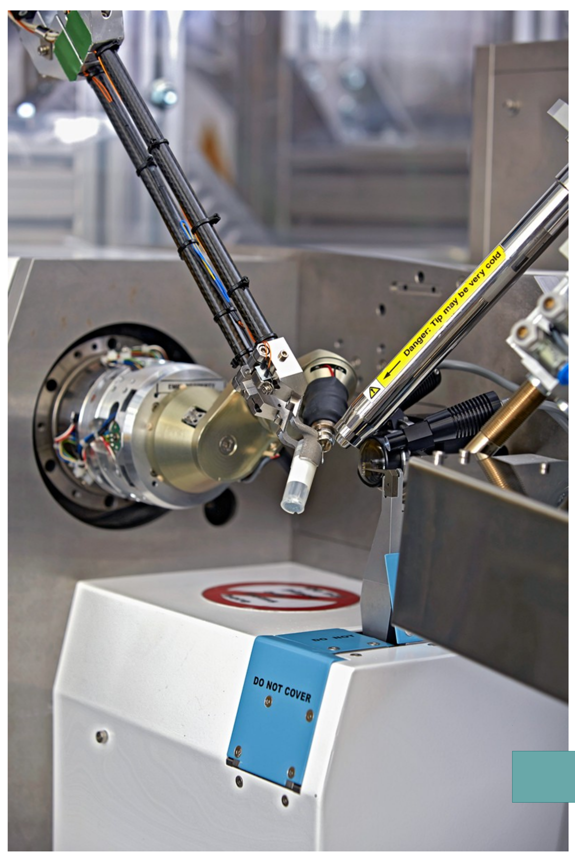

Figure 1: Sample mounting on the sample positioning device (goniometer).

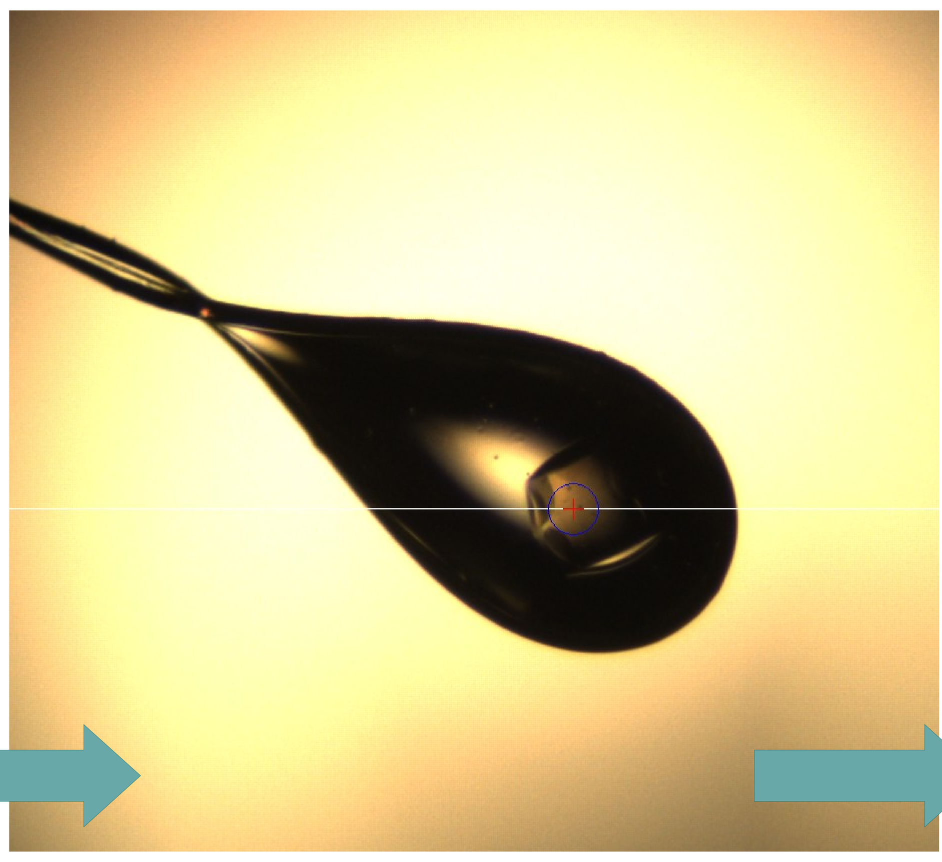

Figure 2: Sample centering.

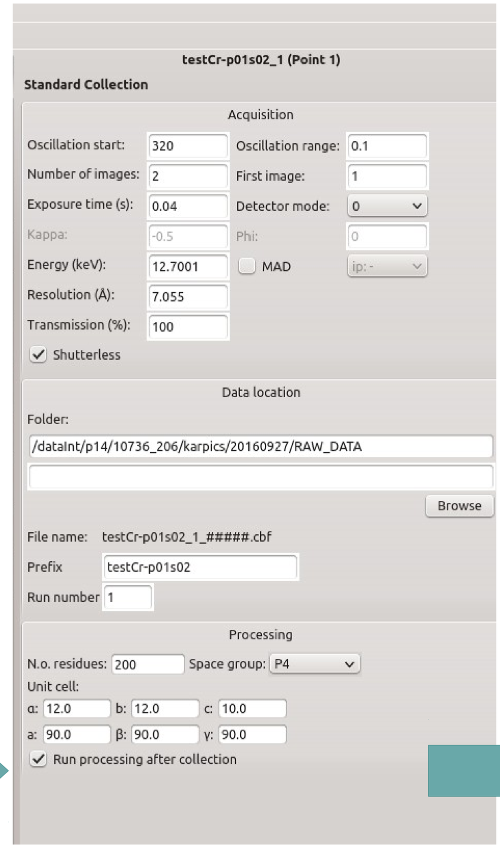

Figure 3: Entry and validation of data collection parameters.

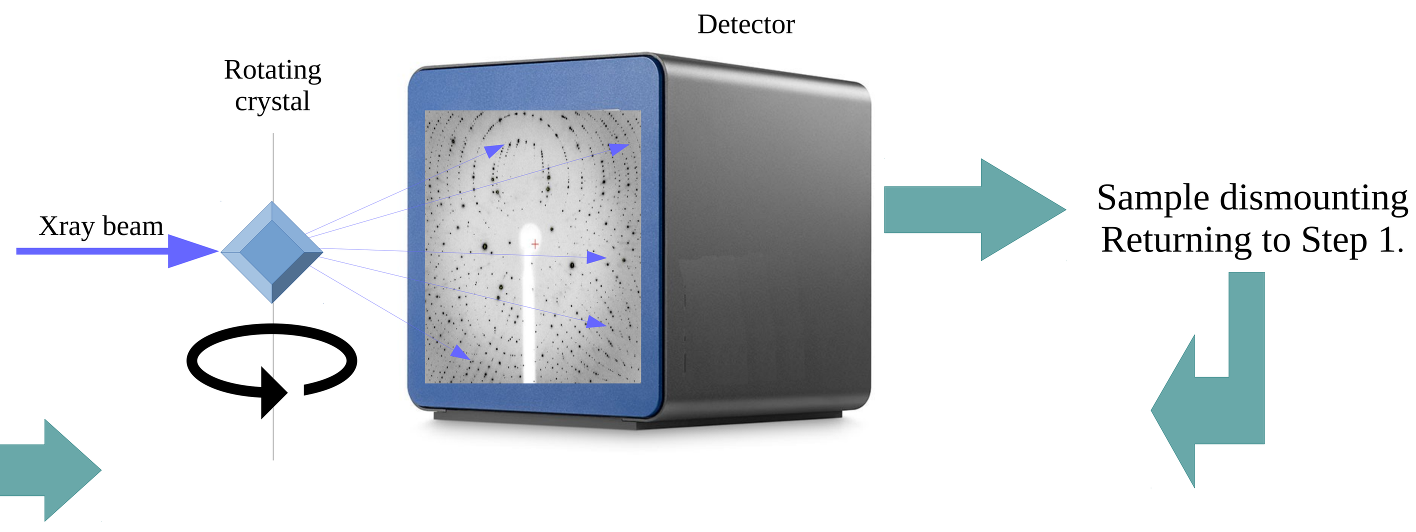

Figure 4: Execution of data collection.

Use Case

- The MxCuBE graphical user interface (GUI) for MX beamlines contains numerous widgets to control the settings of the beamline hardware and set the collection parameters [Fig. 5].
- The many different ways of collecting data on a given crystals and the interdependencies between different components of the beamline result in a highly complex system for which stable operation is not trivial to achieve.
- Many users of MX beamlines are inexperienced posing high requirements in terms of making the operation robust and supporting recovery from errors.

Finite State Machine

- An FSM is a mathematical model of a closed or open loop discrete-event system with defined states [2]. FSM graph contains states and transitions between them.
- It is widely used to define, analyse and control the functioning of a system.
- Applications include software engineering and experimental control systems. For example, the usage of FSMs are described in [3, 4, 5].
- A FSM describing user interaction with MX graphical user interface MxCuBE has been created [Fig. 6].

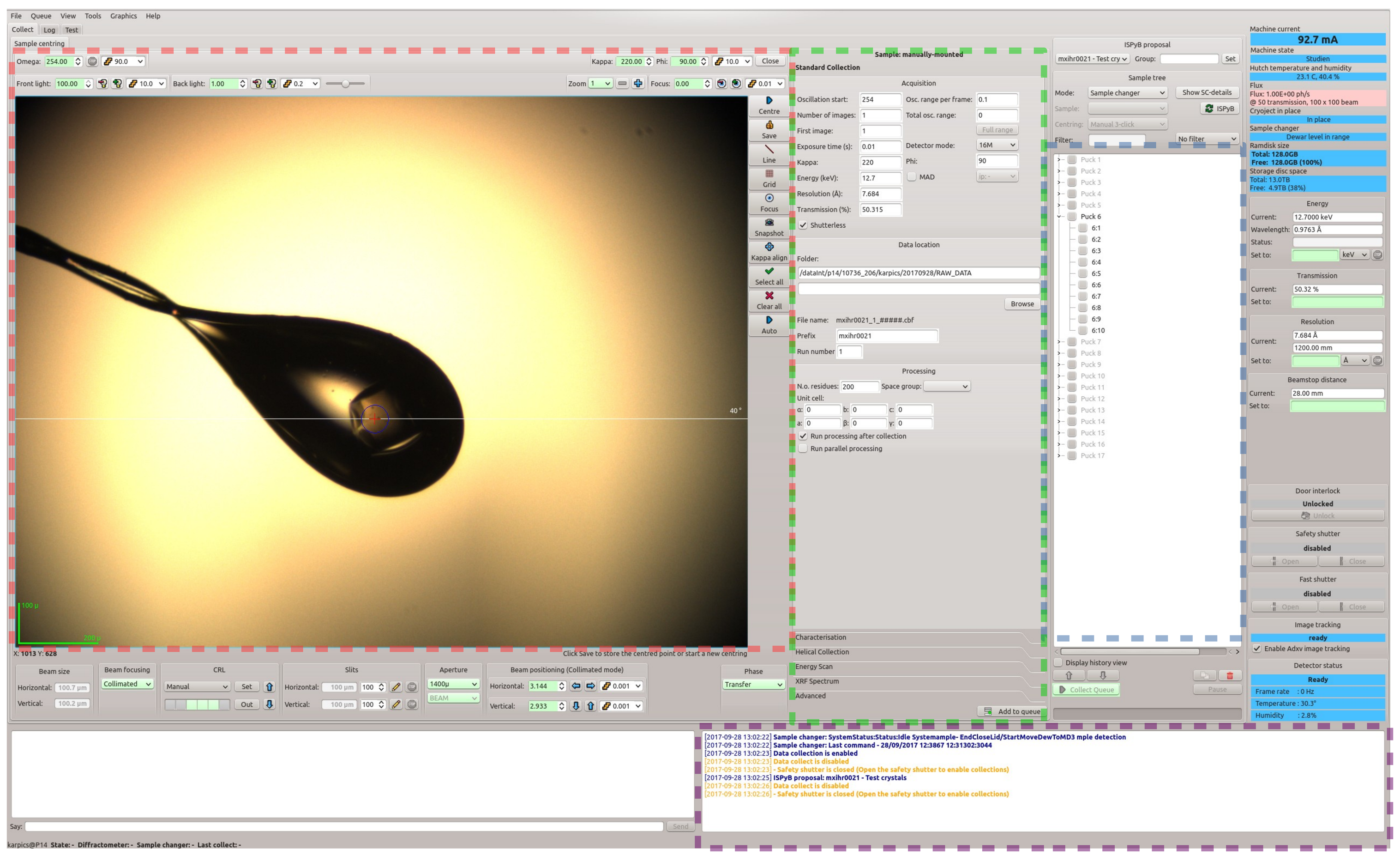

Figure 5: Graphical user interface MxCuBE as seen at EMBL beamline P14.

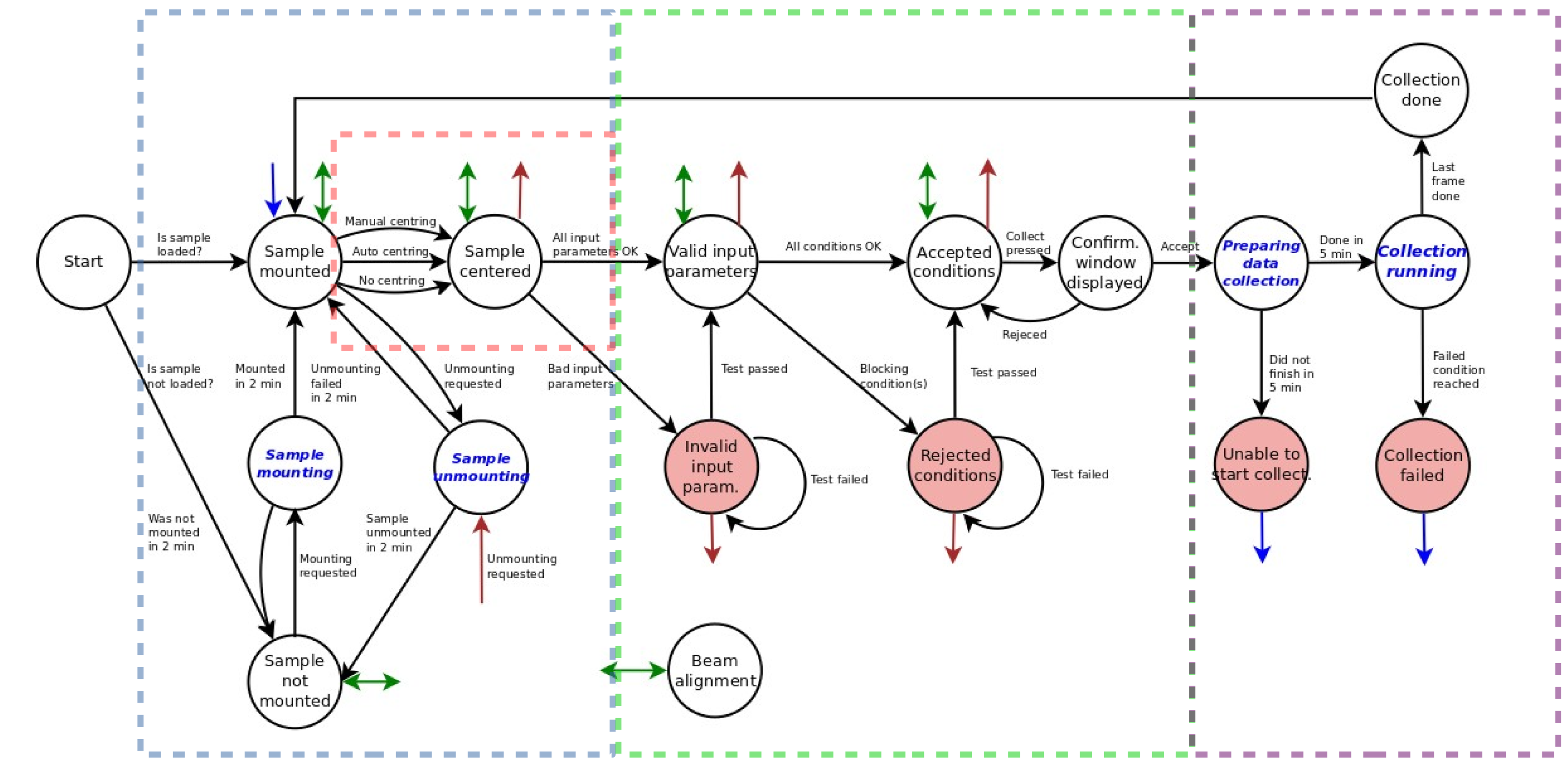

Figure 6: State graph of a user interaction during a macromolecular crystallography data collection. Circles represent FSM discrete states. Normal states are shown in white, while states painted in red are error states and require actions to return system to a normal state. Blue arrows point to *Sample mounted* state and are executed automatically, red arrows indicate the request from a user to unmount a sample.

Implementation in MxCuBE

- MxCuBE is logically divided into a hardware access and a graphical representation layer.
- Hardware access level contains self-contained hardware objects that represents beamline components.
- FSM is implemented as an object connecting all hardware objects.
- Transitions are triggered upon request when the conditions as evaluated by individual hardware objects are fulfilled.
- For debugging purposes a window with information about the state of the FSM is available [Fig 8].

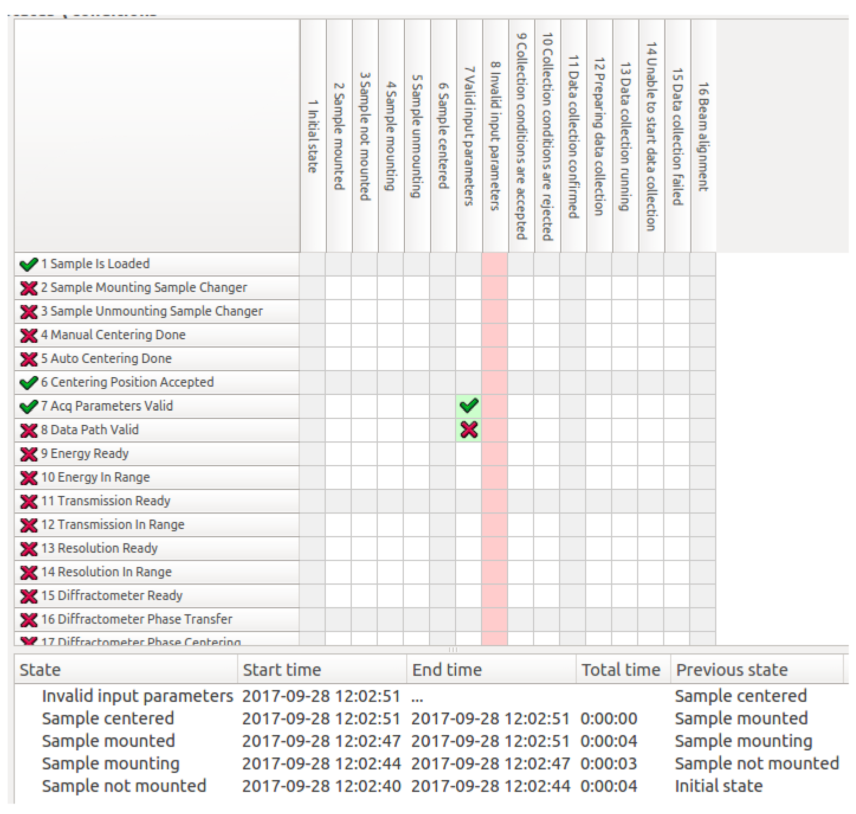

Figure 8: Window for monitoring the FSM described in [Fig. 6].

Conclusion and Perspectives

- An idealized Finite State Machine model for the interaction of a beamline user with a beamline to collect diffraction data from a crystal is presented.
- Resulting description is helpful for the user of the beamline, the beamline scientist supporting the beamline user, and for the developers of the beamline control interface.
- For the (inexperienced) beamline user, being informed about the current state/current transition is useful especially in situations in which the beamline is seemingly idle or blocked
- The clean information about error state and - when possible - suggested recovery procedures make the beamline user more autonomous.
- For the developer, the state history provides an important tool for debugging.
- Gathering statistics about the behavior of beamline components as seen via the states assumed and transitions take can be used to build a knowledge base for pin-pointing fault-causing beamline components.
- Describing subsystems as FSMs can be useful both for achieving a better understanding of the needs and for optimizing procedures in terms of efficiency and robustness.

References

[1] J. Gabadinho *et al.*, "MxCuBE: a synchrotron beamline control environment customized for macromolecular crystallography experiments", *Journal of synchrotron radiation*. vol. 17, pt. 5, pp. 700–707, 2010.
[2] D. Harel, "Statecharts: a visual formalism for complex systems", *Science of Computer Programming*, vol 8, iss 3, pp. 231–274, 1987.
[3] F. Calheiros, P. Golonka, and F. Varela, "Automating The Configuration Of The Control Systems Of The Lhc Experiments", in Proc. ICALEPCS2007, paper RPPA04, pp. 529–531.
[4] G. De Cataldo, A. Augustinus, M. Boccioli, P. Chochula, and L. Stig Jirdén, "Finite State Machines for Integration and Control in ALICE", in Proc. ICALEPCS2007, paper RPPB21, pp. 650–652.
[5] B. C. Heisen *et al.*, "Karabo: An Integrated Software Framework Combining Control, Data Management, And Scientific Computing Tasks", in Proc. ICALEPCS2013, paper FRCOAAB02, pp. 1465–1468.