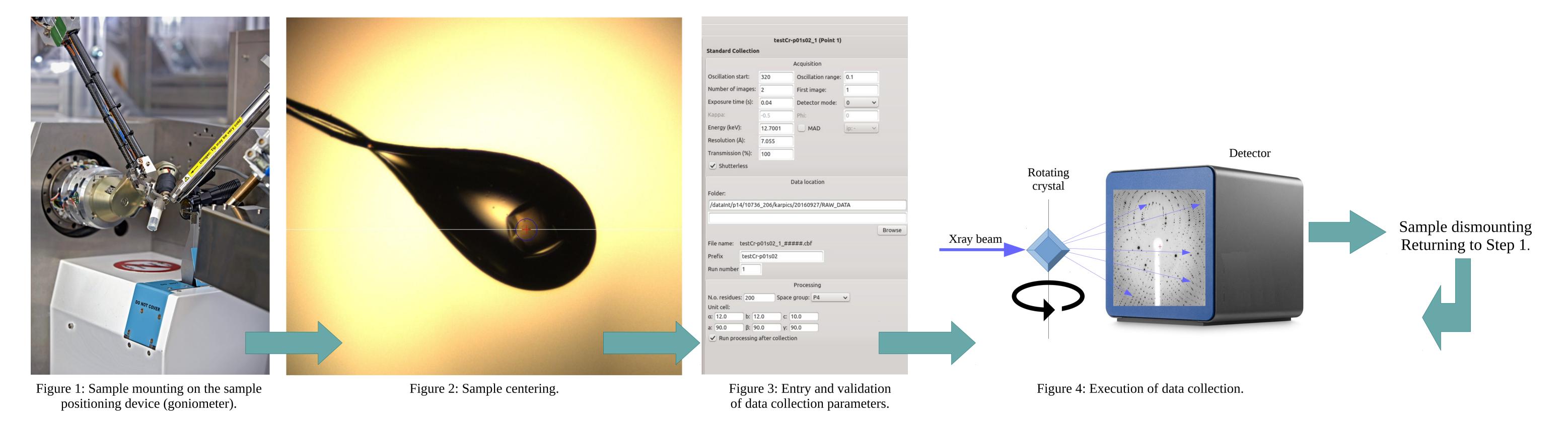
# Enhancing the MxCuBE user interface by a finite state machine (FSM) model Ivars Karpics, Gleb Bourenkov, Thomas R. Schneider European Molecular Biology Laboratory (EMBL) Hamburg Unit c/o DESY, Notkestrasse 85, 22607, Hamburg, Germany

#### 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



### **Use Case**

## **Finite State Machine**

- 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.

ect Log Test										Machine current
nple centring								ISPyB proposa	d I	92.7 mA
				Sample: m	nanually-mounted					Machine state
ega: 254.00 🛇 🔘 🛃 90.0 🗸	Kappa: 220.00 🗘 Phi: 90.00 🗘 🛃	10.0 V Close	Standard Collection			_	mxihr00	021 - Test cry ∨ Group:	Set	Studien
			Scandard Collection							Hutch temperature and humidity 23.1 C, 40.4 %
ront light: 100.00 🗘 🕎 🔁 🌌 10.0 🗸 Back light: 1.00 🗘 💱 🔁 🖉 🖉 🖉 — — —	Zoom 1 🗸 🚍 🔂 Focus: 0.00 🗘 🖲	🗐 🎜 0.01 🗸	Acquisition			Sample tree			23.1 C, 40.4 %	
			Contraction of the				Mode:	Sample changer 🛛 🗸	Show SC-details	Flux: 1.00E+00 ph/s
			Oscillation start:	254 (	Osc. range per frame:	0.1	Complex		<b>1</b> 100-0	@ 50 transmission, 100 x 100 be
		Centre	Number of images:	1 1	Total osc. range:	0	Sample:	Ľ	😂 ІЅРуВ	Cryoject in place
		<b>A</b>					Centring:	: Manual 3-click 🗸		In place Sample changer
		Save	First image:	1		Full range			AL	Dewar level in range
			Exposure time (s):	0.01	Detector mode:	16M V	Filter:		No filter 🗸	Ramdisk size
			Exposore entre (s).						Total: 128.0GB	
		Line	Kappa:	220 F	Phi:	90	>- 🔲 F			Free: 128.0GB (100%)
			Energy (keV):	12.7	MAD	(in: v)	>- 🔲 F			Storage disc space
		Grid	Energy (kev):	12.7	U MAD	ip:- ~	🖌 >- 🔲 F	Puck 3		Total: 13.0TB Free: 4.9TB (38%)
		•	Resolution (Å):	7.684			3- 🔲 1	Puck 4		Free: 4.91B (38%)

- 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].



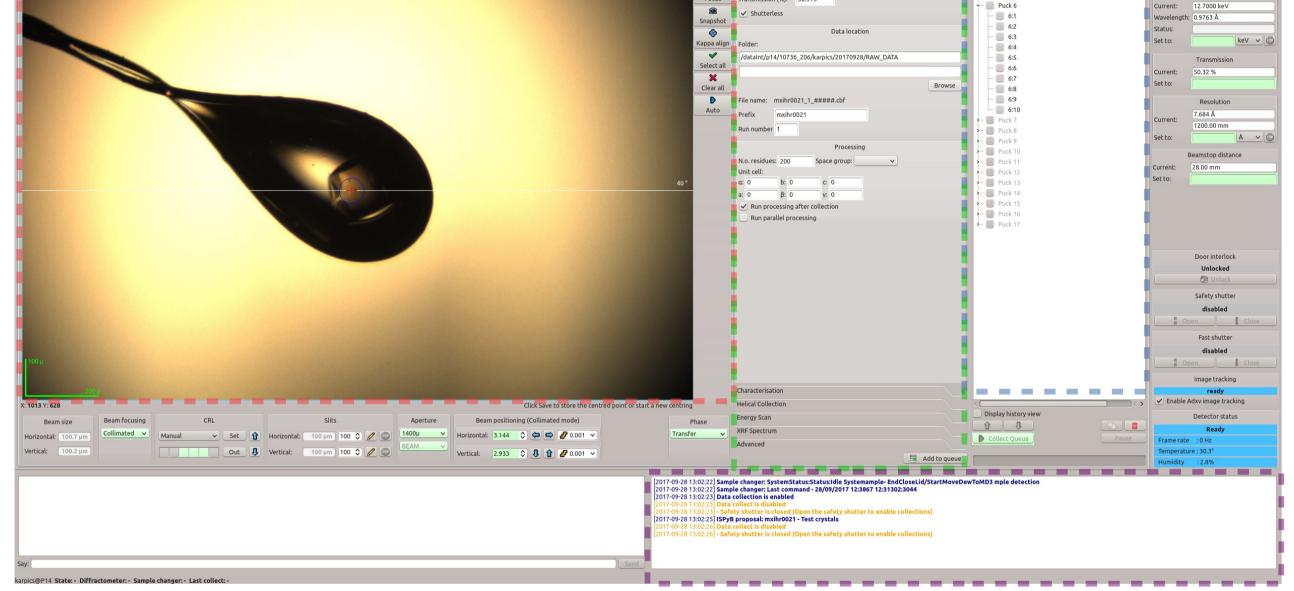


Figure 5: Graphical user interface MxCuBE as seen at EMBL beamline P14.

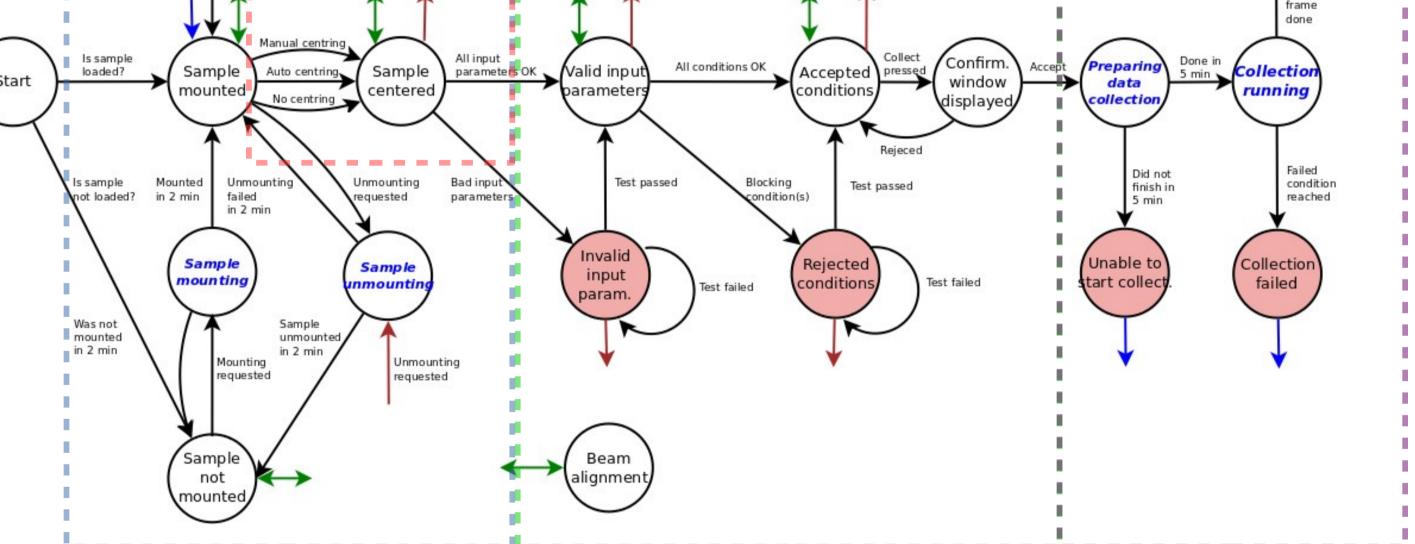
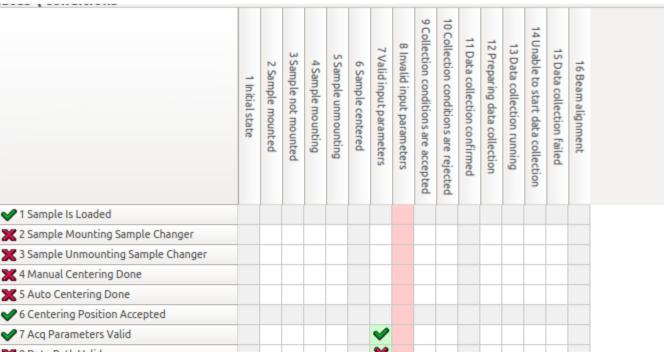


Figure 6: State graph of a user interaction during a macromolecular crystallography data collection. Circles represent FSM discrete states. Normal states are shown in white, while states painted in red are error states and require actions to return system to a normal state. Blue arrows point to *Sample 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 selfcontained hardware objects that represents beamline components.
- FSM is implemented as an object connecting all hardware objects.



# **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.

- 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].

🔀 8 Data Path Valid			×							
💢 9 Energy Ready										
💢 10 Energy In Range										
💢 11 Transmission Ready										
💢 12 Transmission In Range										
💢 13 Resolution Ready										
💢 14 Resolution In Range										
💢 15 Diffractometer Ready										
💢 16 Diffractometer Phase Transfer										
17 Diffractometer Phase Centering	1									
State	Start time		End time	e		Total time Prev		Ргеу	Previous state	
Invalid input parameters Sample centered Sample mounted Sample mounting	2017-09-28 2017-09-28 2017-09-28	12:02:51 12:02:47 12:02:44	2017-09 2017-09 2017-09	-28 12 -28 12	2:02:51 2:02:47	0:00:04 0:00:03	1 3	Samp Samp Samp	ble centered ble mounted ble mounting ble not mounted	
Sample not mounted	2017-09-28	3 12:02:40	2017-09	-28 12	2:02:44	0:00:04	1	Initia	l state	

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

#### References

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.
 D. Harel, "Statecharts: a visual formalism for complex systems", *Science of Computer Programming*, vol 8, iss 3, pp. 231–274, 1987.
 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.
 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.
 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.