

Science & Technology Facilities Council ISIS Neutron and Muon Source

Kathryn Baker (STFC/RAL/ISIS, Chilton, Didcot, Oxfordshire)

Interrupting a State Machine

At the ISIS Neutron and Muon Source we interact with a variety of types of beamline systems for controlling the environment of samples under investigation. A state machine is an excellent way of controlling a system which has a finite number of states, a predetermined set of transitions, and known events for initiating a transition. But what happens when you want to interrupt that flow?

Abstracted Desc	ription			
User Actions	Interrupt State Set	Main State Set		Queue State Set
	Monitor	Idle	State 1	Dequeue Element

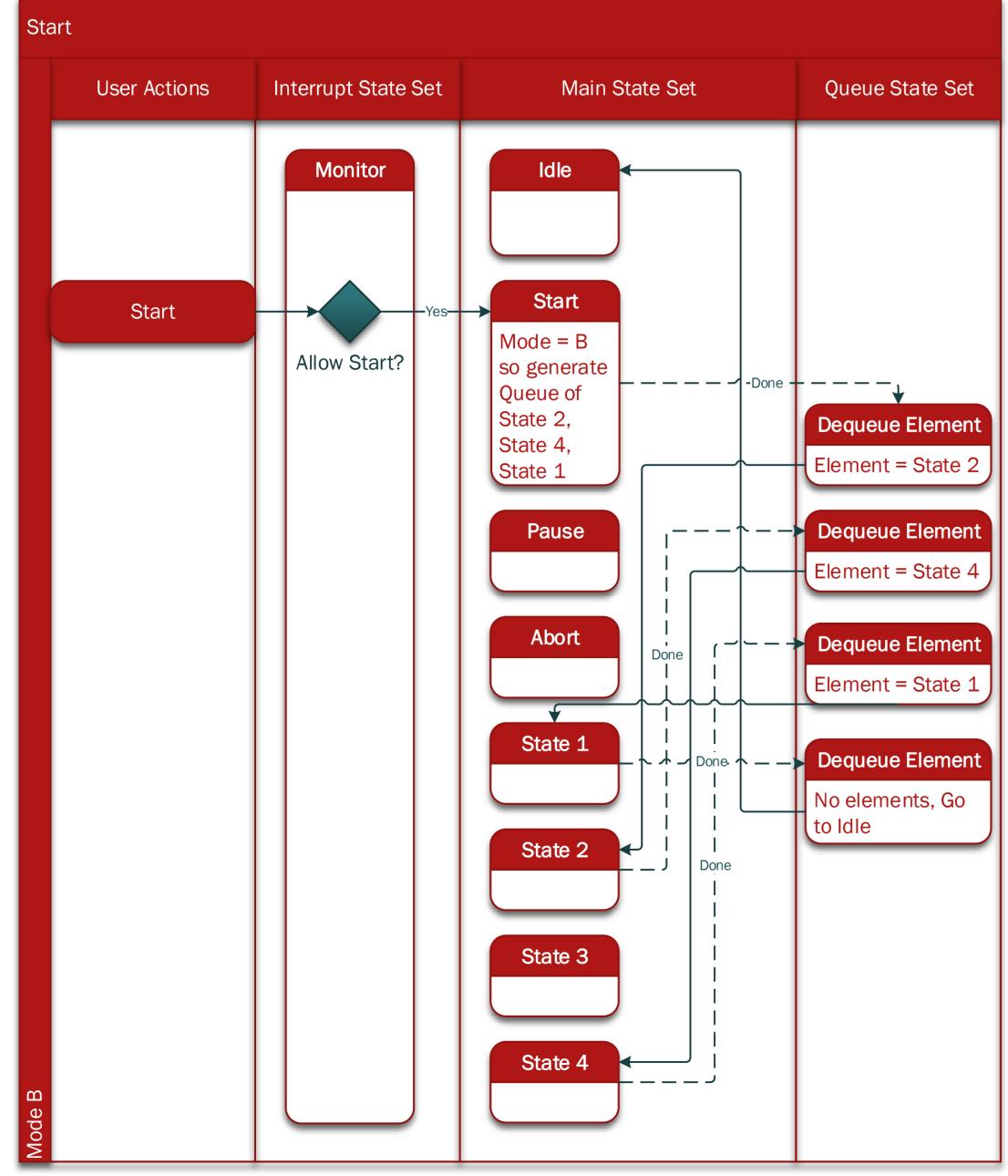
	Should any of the user	No actions are undertaken here	Do something for State 1	Dequeue first
Start	actions occur, then the associated main state is	Start	State 2	element
Pause	triggered, simply by setting	Load queue with states for the Mode	Do something for State 2	Dequeue Element
Tause	the driving variable, andthis state set quickly goes	Pause		Dequeue next element
Resume	back to looking at the interrupting values. Should the interrupt be invalid,	If running: Hold in current state Second request or resume:	State 3 Do something for State 3	Dequeue Element No elements
Abort	then this state set can issue	Continue as if not interrupted	State 4	Go to Idle
	the warning and ignore it,without delaying the main	Abort	Do something for State 4	
	code to deal with the situation too much.	Clear the queue of states Return the system to an idle state		

Figure 1: The state sets

The main state set above consists of states for being idle, start, pause, abort, state 1, state 2, state 3, and state 4. We can assume that the states handle themselves, in reality there may be more interaction between them and could need to respond to certain interrupts as well.

The interrupt state set monitors the start, pause/resume and abort statuses.

The queue state set will receive a list of states to go through from the start state, and as each state ends, will action the next state in the queue by setting the driving variable for the main state set, pausing and resuming as appropriate.



An Abstracted System

Let us consider a fairly simple abstracted system, which can be started, paused, resumed, and aborted.

The system can be run in Mode A, where a start requires going through states 1, 2 and 3; Mode B, where a start necessitates states 2, 4, and 1 to be run in that order; or Mode C which runs states 1, 3, 4 and 2. Any pause or abort has to be acted on quickly. If we consider three state sets, we can see a way of interleaving them to provide such functionality.

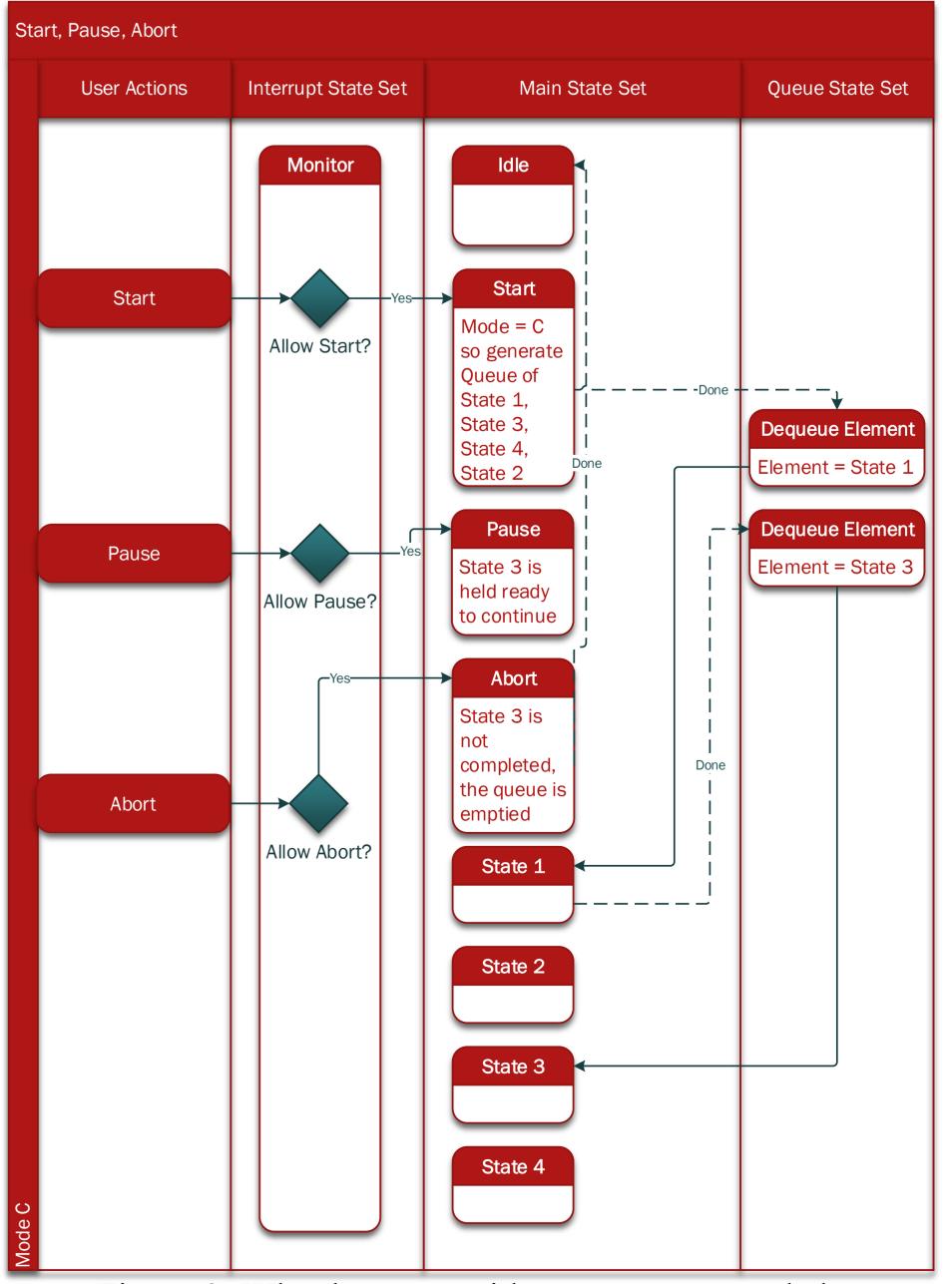


Figure 2: What happens on a start

Figure 3: What happens with a start, pause and abort

Glossary

State Machine/State Set

"A device which can be in one of a set number of stable conditions depending on its previous condition and on the present values of its inputs" [https://www.lexico.com/en/definition/state_machine]

Event

An event is something that happens, it may be a timeout or value is reached, or it may be that a button is pressed.

Transition

A move from one state to another triggered by an event

Interrupt

A break or stop of something continuous [https://www.lexico.com/en/definition/interrupt]