State Machine Framework and its Use for Driving LHC Operational States

M. Misiowiec, V. Baggiolini, M. Solfaroli Camillocci, CERN, Geneva, Switzerland

LHC follows a complex operational cycle with 12 major phases that include equipment tests, preparation, beam injection, ramping and squeezing. Finally followed by the physics phase. This cycle is modelled and enforced with a state machine, whereby each operational phase is represented by a state. On each transition, before entering the next state, a series of conditions is verified to make sure the LHC is ready to move on. Java State Machine Framework was developed to cater for building independent or embedded state machines. They safely drive between the states executing tasks bound to transitions and broadcast related information to clients. State Machine Framework encourages users to create their own actions. Simple configuration management allows the operators to define and maintain complex models themselves.

State Machine Framework

State Machine is composed of states connected by transitions and associated with actions. Exactly one state is initial. State machine holds a status with an active state as its most essential attribute.

Configuration

XML document, readable and simple to maintain, was the choice for LHC Operational States. It lets operators manage the configuration fully on their own which has proven stimulating for maintenance of the instance. XML configuration is stored in SVN under version control.

Actions Development

Actions are:
- tasks – operations executed in the environment
- conditions – checks performed on the environment

Actions are developed to the dedicated API by domain experts and associated by configuration. Variety of information is provided to the action execution, e.g. client credentials. An action can be placed in one of three locations with regard to a transition or state: OnExit, OnTransition or OnEntry.

Conclusions

State Machine Framework and its first instance have been operationally used for almost a year now. Over the period it has become an integral constituent of the LHC controls software architecture, running impeccably throughout its open architecture promoted active participation of the users, LHC operators, in the development and maintenance of both, the instance and framework. State Machine Framework is a general purpose library aimed at both standalone or embedded use wherever state machine concepts need putting in place. It has let operators manage the configuration fully on their own which has proven stimulating for maintenance of the instance. XML configuration is stored in SVN under version control.

Deployment in LHC

LHC Sequence executes preconfigured series of tasks – sequences. They move the machine through its operational lifecycle. Each phase of the lifecycle, an LHC operational state, is guarded by a set of conditions – checks – supervised by LHC Operational States machine service.

Designed for Quality

Safety

State machines open to a parallel use of many clients are prone to errors. Diligent effort was taken to design flawless concurrency engine serving multiple clients. A year of intensive use has confirmed a bug-free environment.

Embedding

Framework supports an embedded use within another application. The instance is hooked into the controlling application using Spring context. All the client calls are executed locally, between Java threads. Embedded mode is achieved by configuration.

Interoperability

Various middleware protocols are supported to communicate status messages from the SM instance.

Perfect for:

- building a state machine service in a multi-tier system
- embedding state machine within any application
- outsourcing actions development
- easy maintenance of XML configuration
- heavily concurrent environment.