

Integrating external finite state machines into a StateWORKS System

The problem

In the StateWORKS philosophy the control of the entire application is concentrated in a system of cooperating state machines. The basic features of the system are:

- The target structure should be a hierarchical one, where possible.
- The interface between Master and Slave state machines has to be of a Command – State nature.

Applications based on the StateWORKS RTDB Library fulfill these requirements as the RTDB objects are designed accordingly.

Sometimes a part of the application is realized in external devices. For instance, many modern actuators contain some control functionality, a LON-node performs simple control tasks, etc. Integration of such external control-system elements into the StateWORKS specification and RTDB Library based implementation increases the reliability, maintainability and understandability of such applications.

The StateWORKS development environment possesses mechanisms for smooth integration of external control devices into the entire state machine specification.

Specifying a system of state machines

The system running in the RTDB should “see” an external state machine as if it were in the RTDB. Thus, an image of the external state machine has to exist in the RTDB.

When specifying the system of state machines we should consider all cooperating state machines, including the external ones. An external state machine will be represented in the RTDB by a VFSM object and a corresponding CMD object. The IO-handler which is the interface to the external state machine, let's call it ExtVfsm-handler, will play a vital role in this process..

We may specify the entire application in StateWORKS Studio in this way, but we can also test such a system as an entity using SWLab and Monitors provided by the Studio. There are no special precautions required for external VFSMs. During testing they are just parts of the system running in SWLab.

External VFSMs in the RTDB base application

While implementing the application we provide a ExtVFSM-handler responsible for:

- Setting the image of the external state machine in the RTDB into service mode.
- Passing the commands sent by the Master to the external state machine.
- Suppling the state values delivered by the external state machine to its image in the RTDB (which is in fact done by means of “service values”).

Demo application

The concept is demonstrated by a small application shown in Figure 1. It consists of a Master state machine which is an actual state machine running in the RTDB. It controls two instances of a state machine of the same type: VFSM:01 and VFSM:02 which both run outside the RTDB. The external

state machines: VFSM:01 and VFSM:02 are represented in the RTDB by two dummies: ExternalVFSM:01 and ExternalVFSM:02 which stay for ever in the state Init. On start-up they are set into service mode by the ExtVFSM-handler which passes states coming from VFSM:01 and VFSM:02 as service values. As those dummy state machines in the RTDB are in the service mode the Master of them “sees” as their states the service values, i.e. the Master “sees” the state of the external state machines.

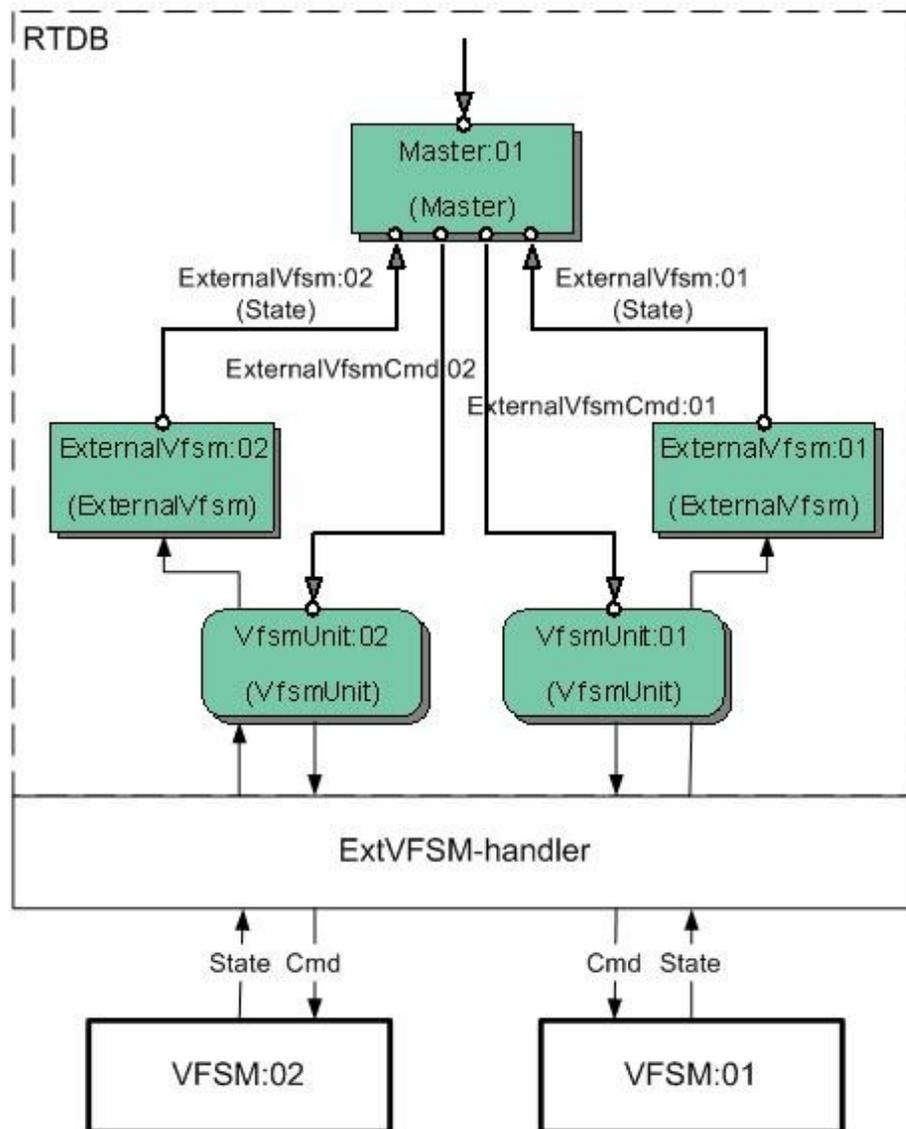


Figure 1: Demo Application with two external finite state machines

States of the external states machines are passed to the the RTDB (ExternalVFSM:01 and ExternalVFSM:02) via a ExtVFSM-handler which interface to the RTDB is defined in VfsmUnit. On the same way, commands are passed to the external state machines.

The Configuration of the tested application is shown below.

```
//Configuration for Cmd object(s)
CMD Name = "ExternalVfsm:Cmd:01"
  Type = "externalvfsm"
CMD Name = "ExternalVfsm:Cmd:02"
  Type = "externalvfsm"
CMD Name = "Master:Cmd"
  Type = "master"

//Configuration for ExternalVfsm object(s)
VFSM Name = "ExternalVfsm:01"
  Type = "ExternalVfsm"
  MyCmd = "ExternalVfsm:Cmd:01"
VFSM Name = "ExternalVfsm:02"
  Type = "ExternalVfsm"
  MyCmd = "ExternalVfsm:Cmd:02"

//Configuration for Master object(s)
VFSM Name = "Master:01"
  Type = "Master"
  MyCmd = "Master:Cmd"
  Externalvfsm1 = "ExternalVfsm:01"
  Externalvfsm2 = "ExternalVfsm:02"
  ExternalVfsmCmd1 = "ExternalVfsm:Cmd:01"
  ExternalVfsmCmd2 = "ExternalVfsm:Cmd:02"

//Configuration for VfsmUnit object(s)
UNIT Name = "VfsmUnit:01"
  Type = "VfsmUnit"
  CommPort = "0"
  PhysAddr = 0
  ExternalCmd = "ExternalVfsm:Cmd:01"
  ExternalVfsm = "ExternalVfsm:01"
UNIT Name = "VfsmUnit:02"
  Type = "VfsmUnit"
  CommPort = "0"
  PhysAddr = 0
  ExternalCmd = "ExternalVfsm:Cmd:02"
  ExternalVfsm = "ExternalVfsm:02"
```

Of course, in the first step we may test this application in SWLab. We will use a system like that shown in Figure 2. It is just a typical Master-Slave system. The changeover to the application where the Slaves are external state machines is straightforward as described above: setting the Slave state machines into service mode does it. And of course we need the ExtVFSM-handler as an interface to the external state machines.

In practice, the ExtVFSM-handler software will need to be tailored to the characteristics of the external hardware and of its interface to the main system. For example, external sub-systems might be implemented in commercial Programmable Logic Controller (P.L.C.) units, where some simple control tasks are added to the basic functions of input-output handling and safety interlocks which

are required to stay in the P.L.C. Domain. The interface might use a serial port, CAN, LON or some other LAN. We can supply detailed advice to our customers about the writing of the necessary software.

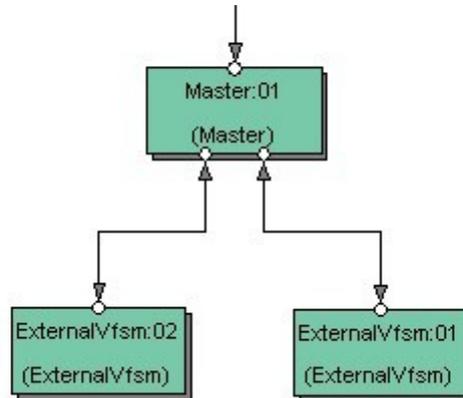


Figure 2: Demo application: variant running in the RTDB

Conclusions

The described interface to external state machines allows us to build StateWORKS applications which integrate smoothly any parts of a control system (seen as state machines) which, for some reason must run outside the RTDB, into the RTDB. This powerful feature of the StateWORKS run-time system results in homogeneous software applications, based on common rules for behavior specification and for interfaces between involved control elements (always regarded as state machines).