

## What is StateWORKS

### **StateWORKS is not**

***another tool to model a state machine and to produce code.***

Several tools on the market allow development of state machine software. They use incomplete state transition tables and are not able to offer a standard solution for a system of state machines. The run time system must be coded.

The tools on the market describe the behavior of a state machine in terms of events that trigger an action and / or state transition. Representative of this way of reasoning is the state transition table in the form:

state event	here	there
event1	do this <sup>1)</sup> go there <sup>2)</sup>	

- 1) A fuller description of the action conditions  
2) A fuller description of the state transition conditions

The table reads: if the state machine is in state **here** and gets the **event1** it does **this** and goes to the state **there**. Unfortunately, the true control task is more complex and the actions to be performed as well as the transition conditions are, in general, more complex. For example the event might be "key pressed" but its effects will depend on "which key?" and whether the "system is enabled". Therefore, the action and the next state require footnotes (1 and 2 in the example) where the full story is described (the complete conditions are defined). The input events will often only be relevant to one state transition vector, so the table ends up having very many rows. Of course, this kind of table must be coded; there is no way to translate it into a pure software table.

These tools model state machines but not the entire software. Therefore the code they produce is just a piece of software in form of a function that must be integrated into another software.

A non-trivial control system application requires more than one state machine. These tools do not offer a complete solution for a system of state machines but leave it to the programmer who should connect several state machines using events and messages, i.e. a programmer should get it ready (code) in some way.

### **StateWORKS is**

***a development environment to specify a system of state machines and an execution environment to run the specified system. StateWORKS does not produce code.***

StateWORKS has both a development environment and a run time system. The designer specifies the behavior and a system using the development environment. The run time system

is a ready-to-use program to perform control as specified by the StateWORKS editor. The remaining coding (if any) is limited to I/O and User interface functions.

By using the concept of a virtual environment (VFSM) and the rules of the logic algebra<sup>1</sup> completed by a complement value<sup>2</sup> StateWORKS editor allows you to fully specify the behavior of a state machine as a formal model (without footnotes).

The VFSM idea is based on a total separation between the data and the control flow. This means that a system of state machines as specified in StateWORKS uses only control flow (no data flow). This, in turn, permits the definition of a standard Master/Slave<sup>3</sup> interface between state machines. Using this interface the StateWORKS editor allows you to specify a system of state machines using terms like commands and states; programming language terms such as events, message, semaphores, etc. are not known in the StateWORKS state machine specification, but only in the external software handling I/O, communications, etc.

The specification of the state machines behavior and of the state machines system is executed by a standard program (run time system) which loads the specification results on startup and controls the application.

The run time system is a fixed, ready-to-use program which frequently requires the addition of I/O and User interface software. The rules for writing I/O and User interface software are imposed by a TCP/IP interface and/or the StateWORKS I/O library functions, depending on the project's requirements.

Some products use the term "executable model" to signify a model which can be tested by means of an interpreter, in simulation runs: StateWORKS also does this, but with a vital difference - execution takes place with high efficiency in the run-time system and not merely for testing.

## **Summary**

There are vital differences between StateWORKS and other "similar" approaches. The similarity is limited to the names: State and State Machine. StateWORKS is not merely a simple software development tool.

***StateWORKS is a specification and run time system for an application control.***

***StateWORKS is a true implementation of an executable specification.***

---

<sup>1</sup> See the technical note "The Virtual environment".

<sup>2</sup> See the technical note „Complement control values in the VFSM concept“

<sup>3</sup> See the next technical note "Hierarchical system of state machines".