



Practical UML Statecharts
in C/C++, Second Edition:
Event Driven Programming
for Embedded Systems

By Miro Samek

Newnes 2008

ISBN-13: 978-0-7506-8706-5

Quantum Leaps, LLC

103 Cobble Ridge Drive

Chapel Hill, NC 27516

USA

info@quantum-leaps.com

866-450-LEAP

www.quantum-leaps.com

www.state-machine.com

User Story



In my experience writing state-machines they usually involved lots of tedious and fragile event handling and state transition code that always resulted in bugs during the maintenance phase. Quantum Framework makes all of that a distant memory since the state machine infrastructure code is already implemented and well tested. The implementation is very compact and efficient yet provides you with lots of features for such a small package. The Quantum Framework code is well organized and with the port files has allowed us to easily make any customizations that we have needed. All of this has allowed us to focus on our design and functionality instead of reinventing the wheel with another custom state machine.



We used Quantum Framework software for developing a new state machine based object using C on multiple platforms. The implementation turned out to be very straight forward once our complicated FSMs were rethought into nice HSM models. In doing so, the state count and the number of different messages needing to be sent were reduced and the code readability increased. This helped to make it much easier to bring on new people to work on the product during the development cycle. The HSM model is very intuitive and easy to explain to other engineers who have picked it up very quickly. One of the best parts about this model is that the maintenance code turns out to be much more easily added without adding lots of spaghetti code to add in something new.

Without using QF, I don't believe we could have delivered on our given schedule dates with the same level of quality. ”

—Jeff Karau, Sr. Software Engineer, General Dynamics C4 Systems