



Practical Statecharts in C/C++

By Miro Samek
CMP Books, 2002

Errata for first printing (June 2002)

Errors in the Accompanying Code:

File	Is / Should Be
<CD>:\Cpp\QF\Source\qhsm.cpp line 44 (the same error occurred also in Exercise 4.9)	is: for (s = myState; s != 0; s = TRIGGER(mySource, Q_EMPTY_SIG)) { should be: for (s = myState; s != 0; s = TRIGGER(s, Q_EMPTY_SIG)) {
<CD>:\C\QF\Source\qhsm.c lines 58-59	is: for (s = me->state__; s != 0; /* traverse the state hierarchy */ (QState)(*me->source__)((QHsm *)me, &pkgStdEvt[Q_EMPTY_SIG])) should be: for (s = me->state__; s != 0; s = TRIGGER(s, Q_EMPTY_SIG)) {
<CD>:\C\QF\DOS\port.h line 25 the same error occurs in <CD>:\C\QF\Borland\port.h line 25	is: #define QF_QUEUE_SIGNAL(q_) (pkgRdyMask = (q_)->osEvent__) should be: #define QF_QUEUE_SIGNAL(q_) \ pkgReadyList = (q_)-> osEvent__; \ QF_UNPROTECT()

Errors in the Text:

Location	Is	Should be
Page ix, middle of the page	such C or C++	such as C or C++
Page 11, Listing 1.1, line 11	<code>*instance()</code>	<code>*instance()</code>
Page 28, line 2 from the bottom	... a new event. it can store it a new event, it can store it ...
Page 34, line 1 from the top	... supertates in inheritance has supertates as inheritance has ...
Page 38, Figure 2.4 caption	... heating superstat	... heating superstate
Page 38, Figure 2.1	:Bird and :Fish object boxes	: <u>Bird</u> and : <u>Fish</u> object boxes
Page 45, Figure 2.6	0..9, POIN transition	0..9, POINT transition
Page 63, Footnote 5	line 6 of Listing 3.2	line 5 of Listing 3.2
Pages 65, 67, 68	multiple occurances "state class" and "context class"	"State class" and "Context class"
Page 70, Listing 3.4 line 18	<code>CParser4 : Fsm((FSM_STATE)initial) {}</code>	<code>CParser4 : Fsm((State)&CParser::initial) {}</code>
Page 72, 75, 76, 78, and 79	multiple occurreces of "state table", "nested switch statement", "optimal FSM", and "simplified state pattern"	"State Table", "Nested switch Statement", "Optimal FSM", and "Simplified State pattern"
Page 74, Exercise 3.9	the reference [Gomez 00] is missing in the Bibliography	[Gomez 00] Gomez, Martin, "Embedded State Machine Implementation", Embedded Systems Programming, December 2000, pp. 40–50.
Page 85, Listing 4.1 line 5	<code>#define QSTATE QHsm::QState</code>	the macro QSTATE is defined in line 38
Page 93, Middle of the page	(Listing 4.1, line 7)	(Listing 4.1, line 6)
Page 89, line 9	Please note the casting of QState, ...	Please note the casting on QSTATE, ...

Location	Is	Should be
from the bottom		
Page 105, Figure 4.4		add internal transition SLASH to states slash in parts (a) and (b)
Pages 106 and 107	Sidebar "Design by Contract in C/C++"	the sidebar should extend over the first paragraph on page 107
Pages 109, Exercise 4.10	Q_TRAN_DYN() is an optimization only with respect to Q_TRAN(), which...	Q_TRAN() is an optimization only with respect to Q_TRAN_DYN(), which...
Page 110, Footnote 17	...to find his was out.	...to find his way out.
Page 142, line 8 from the bottom	(See Section 4.4.4 in Chapter 4)	(See Section 4.3.3 in Chapter 4)
Page 143, Footnote 7	... EXCEPTION event.	... EXCEPTION event.
Page 156, middle of the page	... because the container cannot only choose because the container can not only choose ...
Page 159, 3 rd bullet point from the top	(e.g., in certain states of the component state machine)	(e.g., in certain states of the container state machine)
Page 168, line 11 from the top	According to the traditional LSP, any subclass should be freely substitutable for its subclass.	According to the traditional LSP, any superclass should be freely substitutable for its subclass.
Page 170, Figure 6.2	Transition PERCENT from state "operand2" targets state "ready".	Transition PERCENT from state "operand2" should target state "result".
Page 171, Middle of the page	PERCENT (line 18)	PERCENT (line 8)
Page 173, line 8 from the bottom	... with the inheritance of statecharts . (extra space before ".")	... with the inheritance of statecharts.
Page 191, line 6 from the bottom	[Dijkstra 65, 71]	[Dijkstra 71]
Page 199, middle of the page	... direct-method invocation: direct method invocation: ...

Location	Is	Should be
Page 215, quotation from Fred Brooks	The worst buildings are those whose budget was too great for the purposes to be served.	The worst buildings are those whose budget was too great for the purposes to be served.
Page 213, line 5 from the bottom	event	even
Page 216, line 14 from the bottom	I need to explanation ...	I need to explain ...
Page 217, Sidebar line 7 from the bottom	theses monsters	these monsters
Page 225, Exercise 8.1	... faulty() function to the entry action faulty() function in the entry action ...
Page 228, Footnote 17	... but is not requirement but there is not requirement ...
Page 238, line 5 from the bottom	Listing 8.2 shows three methods ...	Listing 8.2 shows the methods ...
Page 259, line 4 from the bottom	Enter an endless background loop background processing...	Perform background processing...
Page 271, Listing 9.2 Line 13	#include "qevt.h"	#include "qevent.h"
Page 293, line 1 from the top	(Listing 9.8, line 17)	(Listing 9.8, line 16)
Page 308, Listing 10.2 line 3	QActive((QPseudostate)initial)	QActive((QPseudostate)&Table::initial)
Page 309, Figure 10.1	RIGHT(n) in the upper left corner	RIGHT(RIGHT(n))
Page 309, Listing 10.3 line 12	Q_INIT(serving);	Q_INIT(&Table::serving);
Page 310.	return (QSTATE)top;	return (QSTATE)&Table::top;

Location	Is	Should be
Listing 10.3 line 61		
Page 311, Listing 10.4 line 3	<code>QActive((QPseudostate)initial)</code>	<code>QActive((QPseudostate)&Philosopher::initial)</code>
Page 312, Listing 10.5 line 6	<code>Q_INIT(thinking);</code>	<code>Q_INIT(&Philosopher::thinking);</code>
Page 312, Listing 10.5 line 15	<code>Q_TRAN(hungry);</code>	<code>Q_TRAN(&Philosopher::hungry);</code>
Page 312, Listing 10.5 line 18	<code>return (QSTATE)top;</code>	<code>return (QSTATE)&Philosopher::top;</code>
Page 312, Listing 10.5 line 35	<code>Q_TRAN(eating);</code>	<code>Q_TRAN(&Philosopher::eating);</code>
Page 312, Listing 10.5 line 31	<code>return (QSTATE)top;</code>	<code>return (QSTATE)&Philosopher::top;</code>
Page 313, Listing 10.5 line 45	<code>Q_TRAN(thinking);</code>	<code>Q_TRAN(&Philosopher::thinking);</code>
Page 313, Listing 10.5 line 53	<code>return (QSTATE)top;</code>	<code>return (QSTATE)&Philosopher::top;</code>
Page 325, Footnote 1	... hard-to-maintain code practically defeats the purpose hard-to-maintain code and practically defeats the purpose ...
Page 337, Listing A.1	line numbering repeats line 11	
Page 348, Exercise A.6	The shape class provides ...	The Shape class provides ...
Page 349, line 4, from the top	... some per-formance boost.	... some performance boost.