STATE MACHINES … VDM

Marcel Verhoef
CHESS

STATE MACHINES … VDM
(heaven) … (hell)

Marcel Verhoef

CHESS

STATE MACHINES … VDM
(hell) … (heaven)

Marcel Verhoef

CHESS

STATE MACHINES *OR* VDM

Marcel Verhoef

CHESS

STATE MACHINES *AND* VDM

Marcel Verhoef

CHESS

STATE MACHINES *IN* VDM

Marcel Verhoef

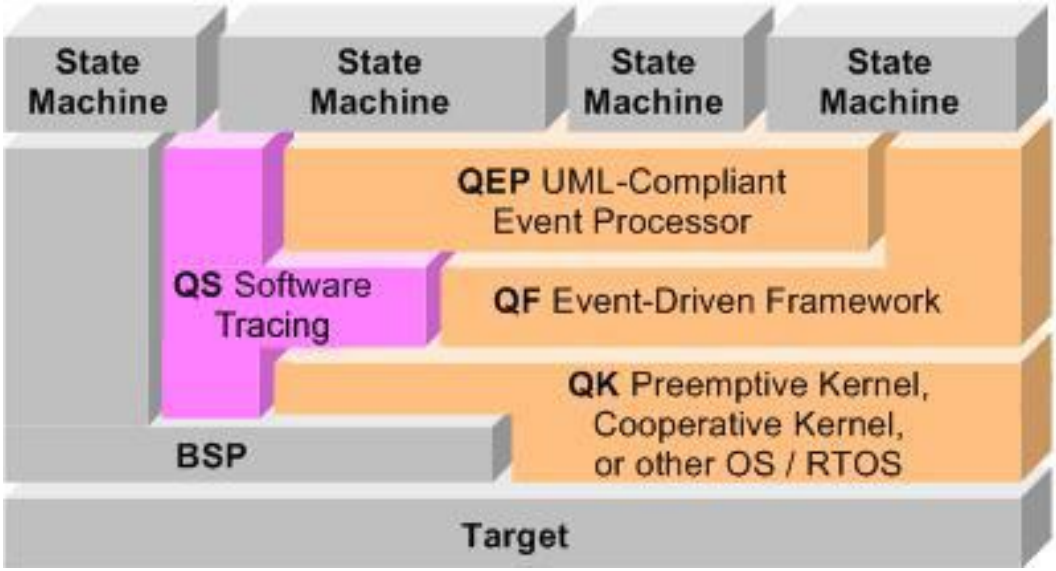
CHESS

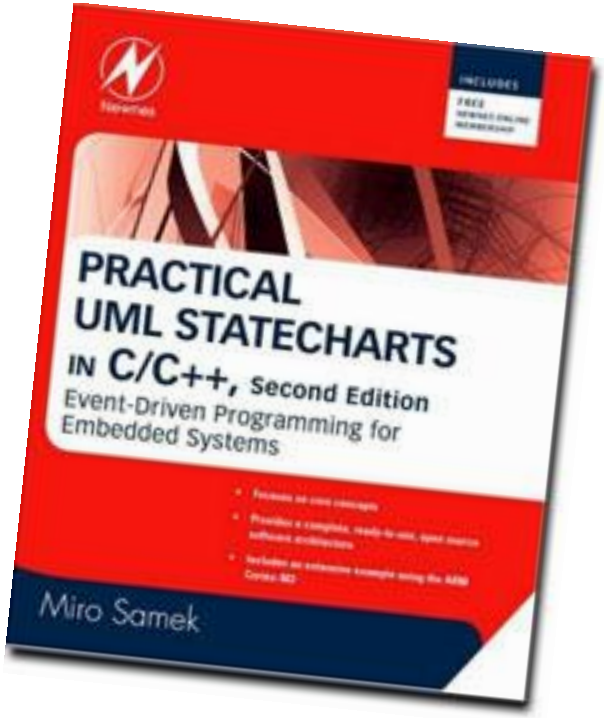
State machines in VDM

- Provide as standard library (much akin to VDMUNIT)
- Avoid “build-your-own-framework” pitfall
- Implemented in VDM++ (use OO structuring)
- Extendable for timing analysis (using VDMRT)
- Support multiple CPU deployment (using VDMRT)
- Focus on model writing productivity
- Initial analysis by interactive simulation

Existing state machine frameworks

- Quantum Framework
- http://www.state-machine.com
- Dual license strategy (open- and closed source)
- Tool support available and well-documented (book)
- Compliant to UML state machine semantics and patterns
- Extremely efficient implementations (C-, C, C++)
- Available on many (real-time) operating systems
- Supports both hierarchical and finite state machines
- Message based communication (signals and events, including inheritance and priorities)


State Machine
State Machine
State Machine
State Machine
QEP UML-Compliant Event Processor
QS Software Tracing
QF Event-Driven Framework
QK Preemptive Kernel, Cooperative Kernel, or other OS / RTOS
BSP
Target


PRACTICAL UML STATECHARTS IN C/C++, Second Edition
Event-Driven Programming for Embedded Systems
Miro Samek

Quantum Framework in VDM (1)

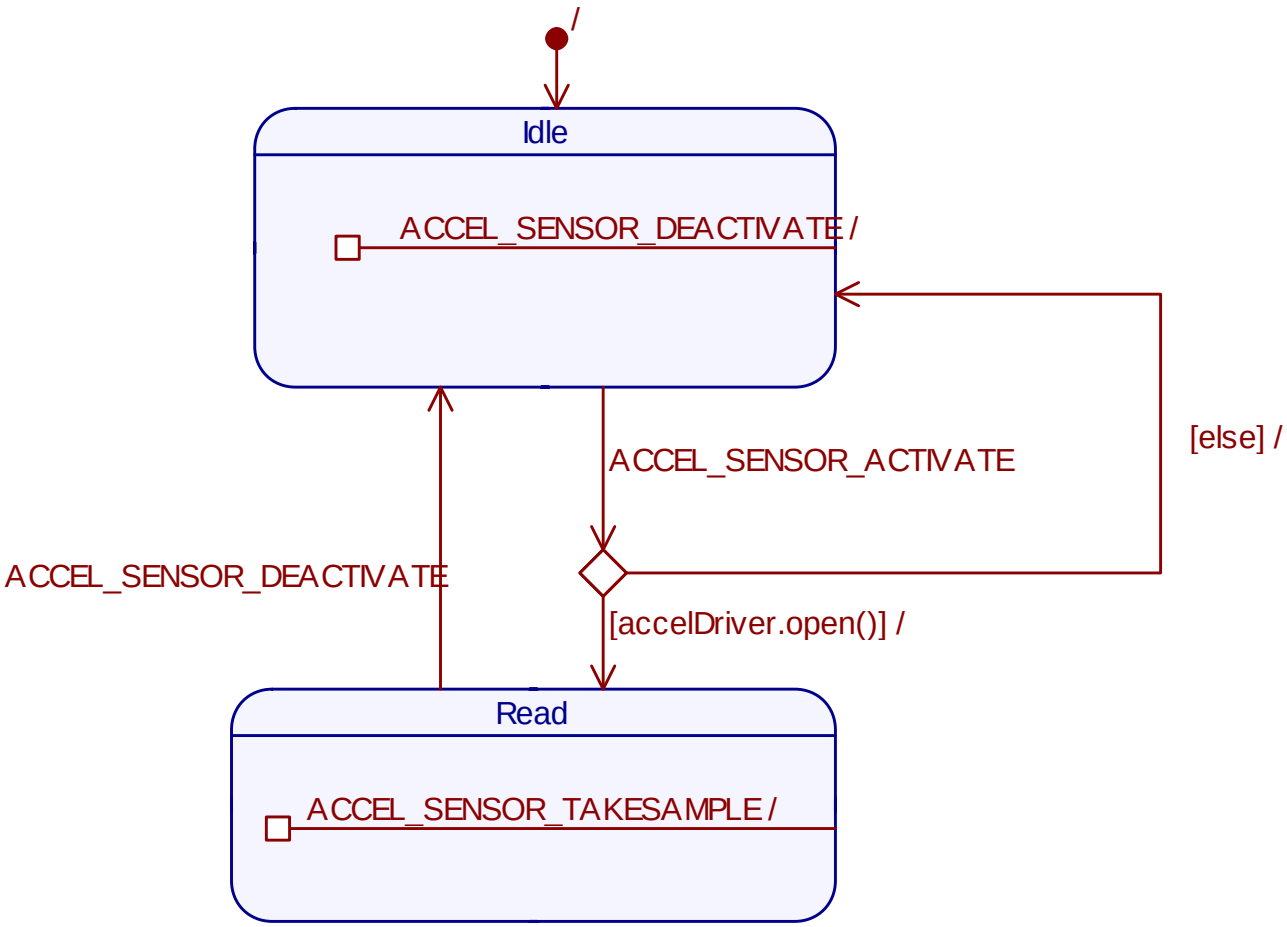
- Modeling QF in VDM was *"Not Quite Trivial"*
- QF uses pointers to (C-)functions extensively
  - Efficiency
  - Inheritance
- Introduced *StateHandler* class and use VDM++ inheritance to solve this
  - Efficiency is *not* our primary concern
- Introduced *EventHandler* class that allows run-time reconfigurable state machine behavior

Quantum Framework in VDM (2)

- Parts of QF currently available in VDM are
  - *FiniteStateMachine* (with message queue and (static and dynamic) event dispatching)
  - *StateHandler* (containing the FSM local state)
  - *EventHandler* (dynamically extend FSM behaviour)
  - *ActiveObject* (the thread executing the FSM with RTC semantics)
  - *Signal*, *Event* and *Timer*
  - *Kernel* (manage AOs, timers, publish/subscribe mechanism)
- Still missing is
  - Hierarchical state machine
  - Event inheritance and priorities
  - Distributed kernel (deployed on multiple CPUs)

Quantum Framework in VDM (3)

- Completed application
  - Commercial product (sensor data fusion application)
- On the drawing board
  - Dining philosophers (illustrative example and testcase)
  - ChessWay DESTECS case study
- Possible extensions / future work
  - animation during simulation (state and sequence diagrams)
  - Automatic UML mapping
  - Test automation and automated learning
  - Verification of the state machines (timed and untimed)


/
Idle
ACCEL_SENSOR_DEACTIVATE /
ACCEL_SENSOR_ACTIVATE
[else] /
ACCEL_SENSOR_DEACTIVATE
[accelDriver.open()] /
Read
ACCEL_SENSOR_TAKESAMPLE /

Example VDM execution log

QF: initialized
time = 0
QF: starting all AOs
QF: starting AccelDataAO
QF: starting GpsDataAO
QF: starting MovementAO
QF: starting TachoDataAO
QF: initializing AccelDataAO
QF: initializing GpsDataAO
QF: initializing TachoDataAO
QF: initializing MovementAO
QF: entering AccelDataAO.Idle
QF: entering GpsData.Idle
QF: entering TachoData.Idle
QF: entering Movement.Top
MovememementAOStrategy.init()
QF: onActivate in TachoData.Idle
QF: onActivate in AccelDataAO.Idle
QF: leaving TachoData.Idle
QF: leaving AccelDataAO.Idle

QF: entering TachoData.Read
TachoDataAO state is reset
QF: onActivate in GpsData.Idle
QF: entering AccelDataAO.Read
AccelDataAO state is reset
QF: leaving GpsData.Idle
QF: entering GpsData.Read
GpsDataAO state is reset
MovementAOStrategy.handleTachoSensorActivated()
MovementAOStrategy.handleAccelSensorActivated()
MovementAOStrategy.handleGPSSensorActivated()
time = 1
MovementAOStrategy.handleTachoSensorData received tcr = 0 at 1.00000648
distance covered = [0, 0, 0] NOT_DRIVING
MovementAOStrategy.handleAccelSensorData received [0, false] at 1.00001081
MovementAOStrategy.handleGpsSensorData received at 1.00001526 :
mk_rmc_data(false, [], "162645", 0.0, 0.0, 0.0, -1.0)
mk_gga_data("162645", 0, 0, 99.99, 0.0, 0.0)