

EEE499 – Model-driven Development of Real-Time Systems

UML-RT and Papyrus-RT: Advance Structural Modeling and RTS

ROYAL MILITARY COLLEGE OF CANADA
ELECTRICAL & COMPUTER
ENGINEERING



GÉNIE ÉLECTRIQUE
ET GÉNIE INFORMATIQUE
COLLÈGE MILITAIRE ROYAL DU CANADA

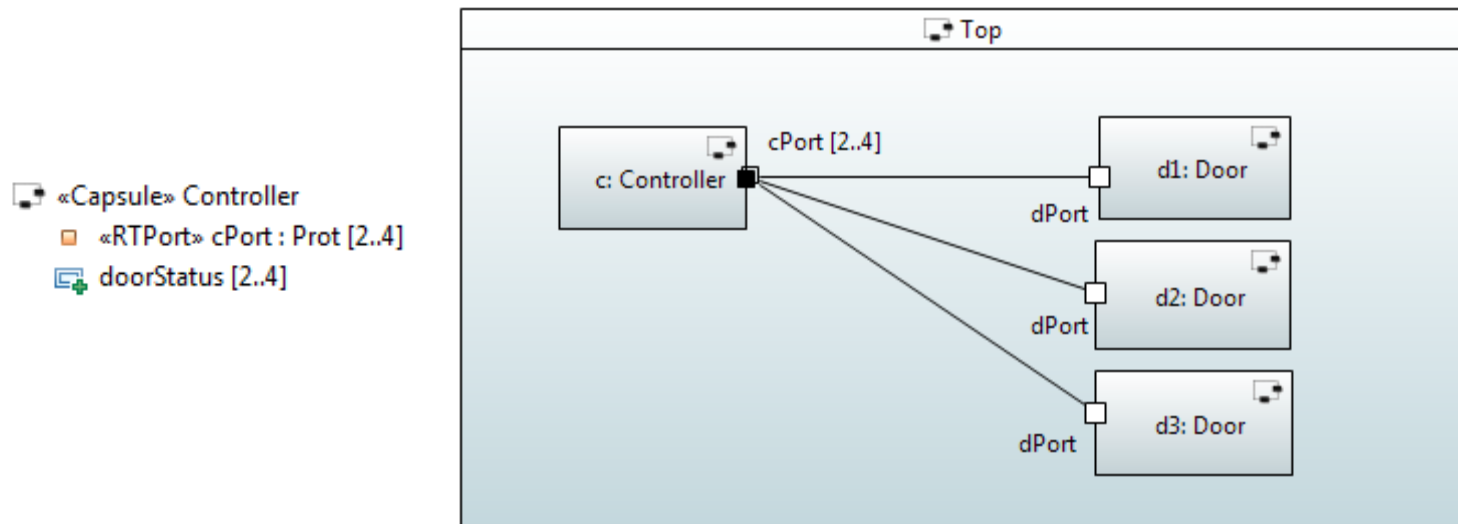


Acknowledgement

The original material for this section was developed by [Prof. Juergen Dingel](#) (Queen's University)

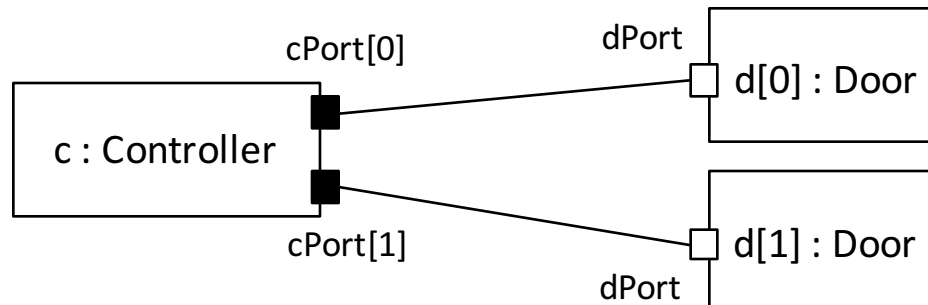
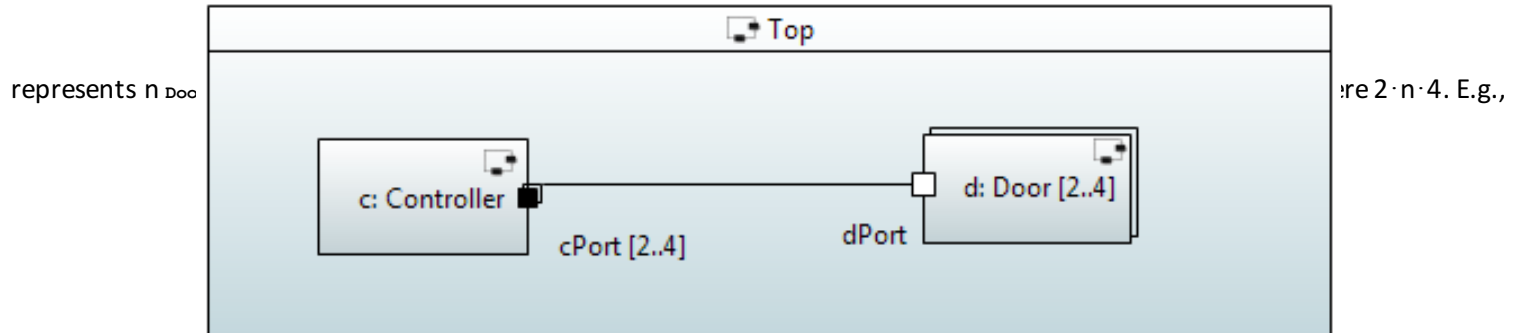
Replication

- Some elements can be replicated
 - attributes, ports, and parts (all instances of UML-meta type 'Property')
- Port replication
 - To send `m` to all doors: `cPort.m().send()`
 - To send `m` to a single, specific door (e.g., `d3`): `cPort.m().sendAt(2)`
 - To tell which port a message came in on: `msg->sapIndex0()` returns port index
 - E.g., if `d3` sends `m` to `c`, then `msg->sapIndex0()` in effect of transition triggered by `m` would return 2

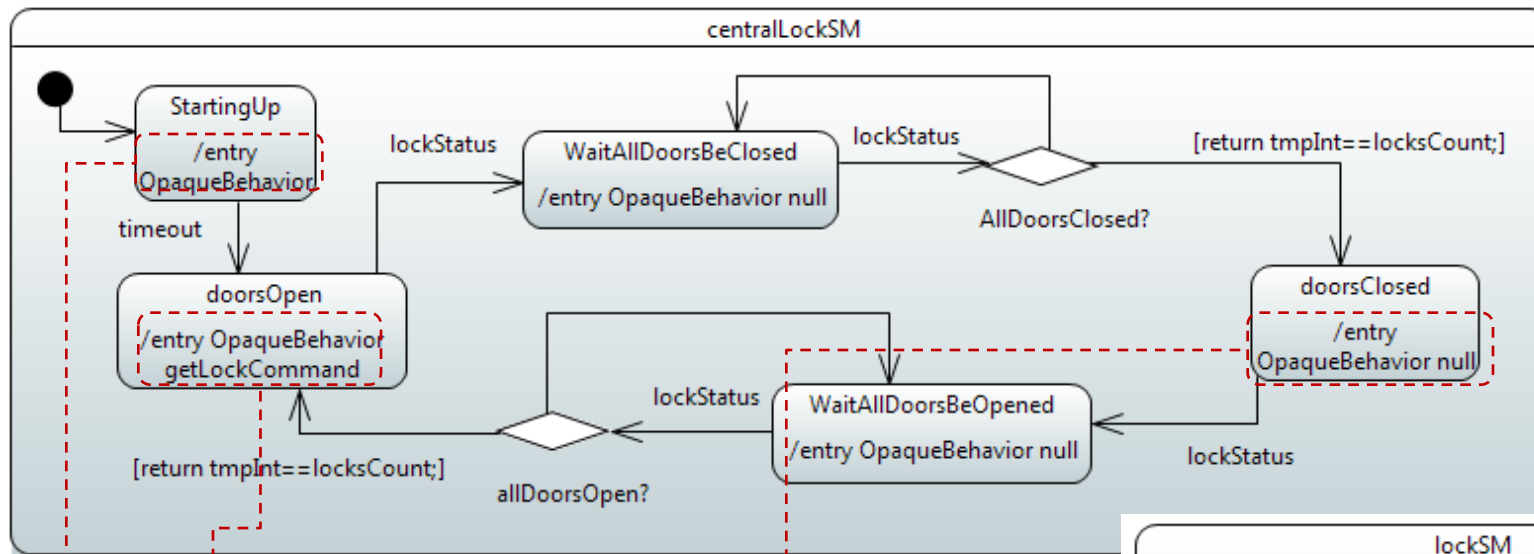
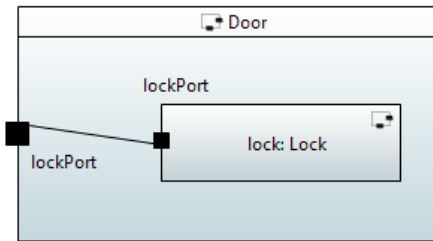
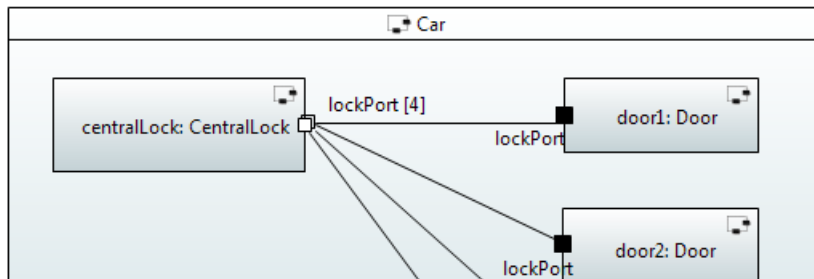


Replication (Cont'd)

- Combining port and capsule replication
 - E.g.,



Example: Door Lock



set timer

```

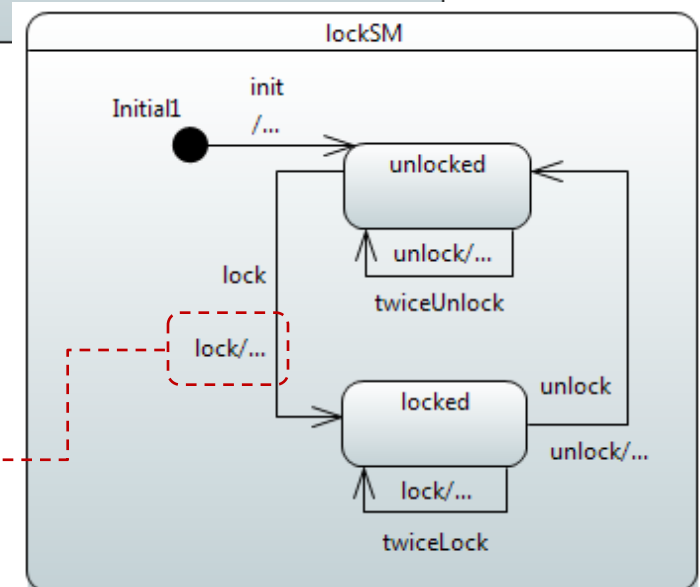
"doors open";
"hit key to lock"
getchar();
lockPort.lock().send()
    
```

```

"doors locked";
"hit key to open"
getchar();
lockPort.unlock().send()
    
```

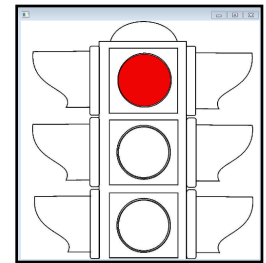
```

"lock"+i+"locked";
lockPort.lockStatus(true).send
    
```

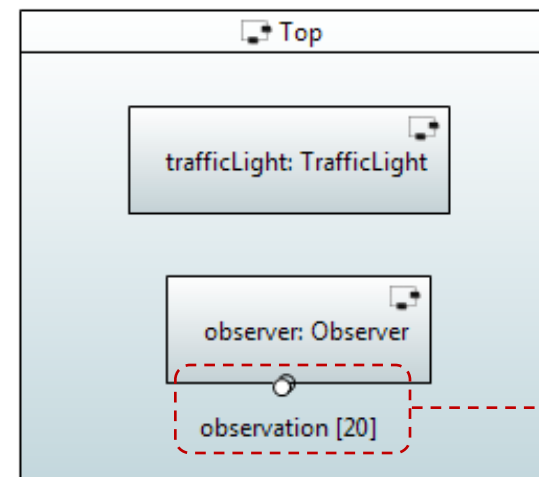
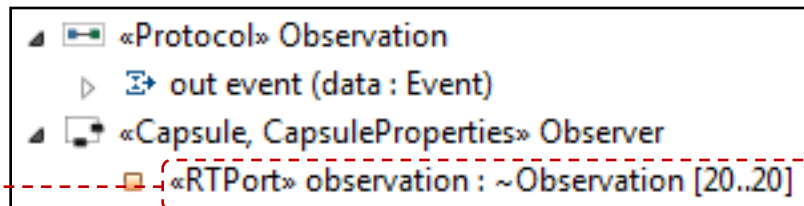
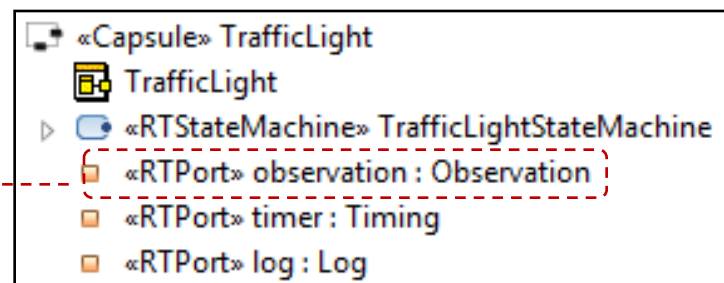


Ports: SPP and SAP

- So far, only **wired ports**
 - Connected automatically when instances are created
- **Unwired ports**
 - Connected at run-time
 - Publish/subscribe



- Port on publisher: Service Provision Point (kind = SPP)
- Port on subscriber: Service Access Point (kind = SAP)
- Register with RTS using unique service name (manually or automatic)



Run Time Services (RTS) Library

- Provides services to application that involve resources managed by the RTS
 - Capsules, communication, timing, logging, frame
- Can be found in

[Papyrus Installation Directory]/

Papyrus-RT/plugins/org.eclipse.papyrusrt.rts_0.8.0.201612120508/umlrts/include

```
$ ls
umlrtapi.hh                umlrtmutex.hh
umlrtbasicthread.hh       umlrtobjectclass.hh
umlrtcapsule.hh           umlrtobjectclassgeneric.hh
umlrtcapsuleclass.hh      umlrtoutsignal.hh
umlrtcapsuleid.hh         umlrtpool.hh
umlrtcapsulepart.hh       umlrtpriority.hh
umlrtcapsulerole.hh       umlrtprioritymessagequeue.hh
umlrtcapsuletocontrollermap.hh umlrtprotocol.hh
umlrtcommsport.hh         umlrtqueue.hh
umlrtcommsportfarend.hh   umlrtqueueelement.hh
umlrtcommsportrole.hh     umlrtrtsinterface.hh
umlrtcontroller.hh        umlrtinterfaceumlrt.hh
umlrtcontrollercommand.hh umlrtsemaphore.hh
umlrtframeprotocol.hh     umlrtsignal.hh
umlrtframeservice.hh      umlrtsignalelement.hh
umlrtguard.hh             umlrtsignalelementpool.hh
umlrthashmap.hh           umlrtslot.hh
umlrtinoutsignal.hh       umlrttimer.hh
umlrtinsignal.hh          umlrttimerid.hh
umlrtlogprotocol.hh       umlrttimerpool.hh
umlrtmain.hh              umlrttimerprotocol.hh
umlrtmessage.hh           umlrttimerqueue.hh
umlrtmessagepool.hh       umlrttimespec.hh
umlrtmessagequeue.hh      umlrtuserconfig.hh
```

Application code
(generated or hand-written)

RTS Library

Target OS

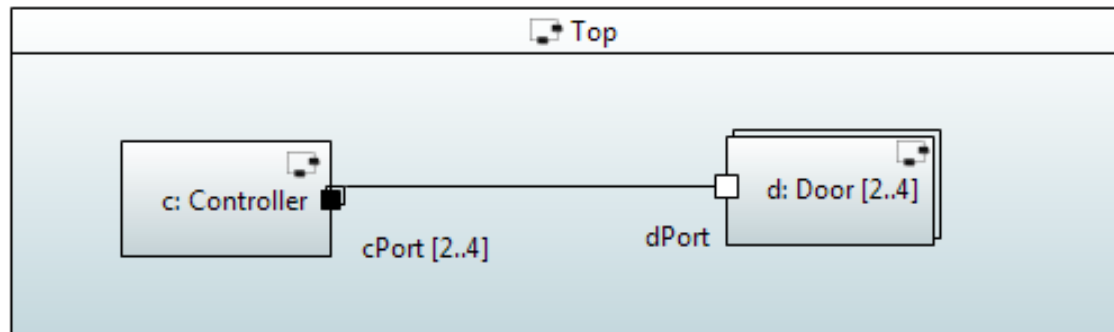
Target HW

Run Time Services (RTS) Library: Capsules

- **UMLRTCapsule** (in `umlrtcapsule.hh`)

- **Methods**

- **string getName()**
 - name of capsule part
 - **string getTypeName()**
 - name of capsule
 - **int getIndex()**
 - index of capsule part



Run Time Services (RTS) Library: Communication (1)

- **UMLRTOutSignal**

- **Methods**

- **bool send(priority)**
 - **asynchronous**
 - priority argument optional
 - if port replicated, send over all instances
 - **bool sendAt(index, priority)**
 - to specific instance of replicated port (indices are 0-based)
 - **int invoke(replyMsg)**
 - **synchronous**, i.e., sender blocks until reply is received (via `reply()`)
 - mimicks 'operation call'

- **Properties**

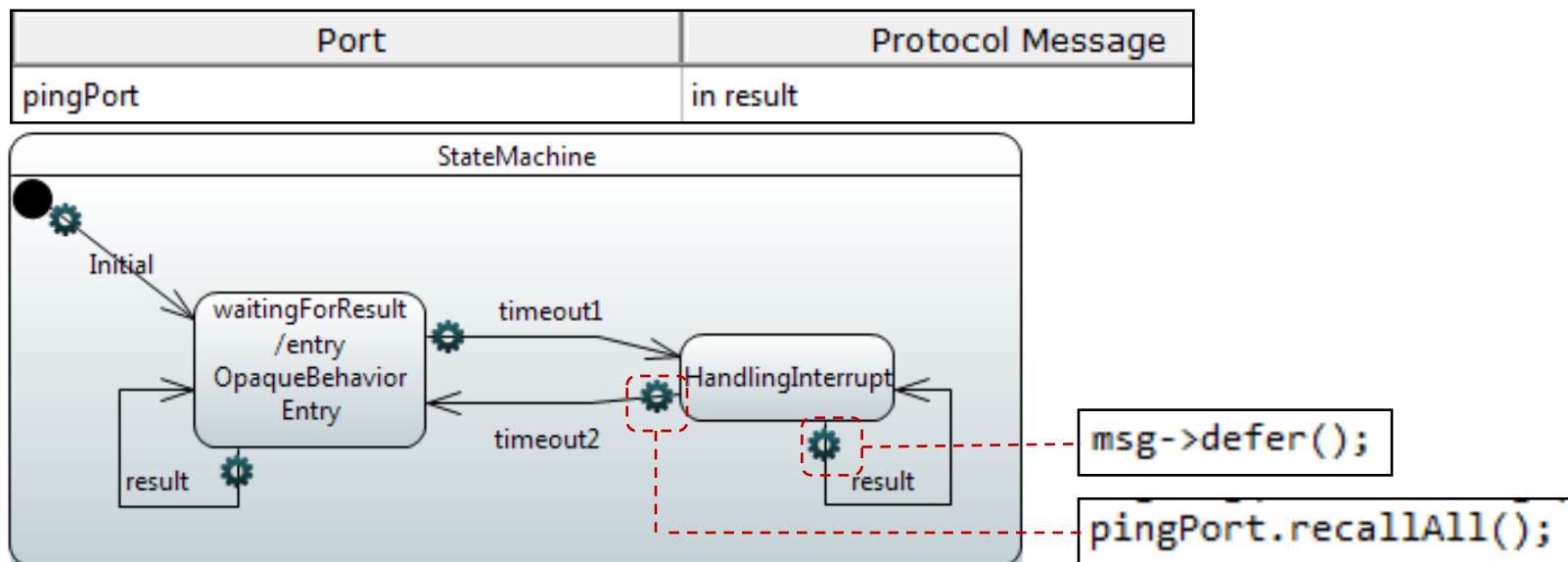
- Messages sent over same connector received in same order they've been sent (unless application is distributed)
 - Delivery of messages to unbound ports will fail
 - Delivery of messages that don't trigger transition, will be dropped with error message
 - If message data has type descriptor, it will be copied and passed **by value**

Run Time Services (RTS) Library: Communication (2)

- **UMLRTMessage**
 - Base type for messages
 - Created upon send signal event; refers to signal being sent and its 'payload'
 - Signals separated from messages, so that different messages can refer to same signal (for broadcast signals)
 - **Methods**
 - `bool defer()`
 - Put message into 'defer queue'
- **Aside: 'signals' vs 'messages'**
 - Signals: elements defined in the protocol
 - Message:
 - represents the sending of a signal
 - contains a signal and any 'payload'
 -) different messages can refer to same signal

Defer/recall

- Allows handling of messages that arrive while in 'wrong' state
- Defer message m on port p:
 - 'Wrong' state has self transition triggered by m with effect `\msg->defer()`
- Recall message m on port p:
 - When entering state in which m should be handled, execute `\p.recallAll()`



Run Time Services (RTS) Library: Communication (3)

- **UMLRTProtocol**

- Base type for protocols

- **Methods**

- **bool recall()**

- Move matching messages from one instance (port) from defer queue to message queue

- **bool recallAll()**

- Move matching messages from all instance (ports) from defer queue to message queue

- **bool registerSAP(string)**

- Non-wired ports with 'RegistrationKind=Application' have to be wired programmatically

- Registers this port as SAP port with RTS to allow for dynamic binding from SPP

- Example: 'p1.registerSAP("myService");'

- **bool registerSPP(string)**

- Registers port as SPP providing service with name 'string' and automatically connects with matching SAP ports

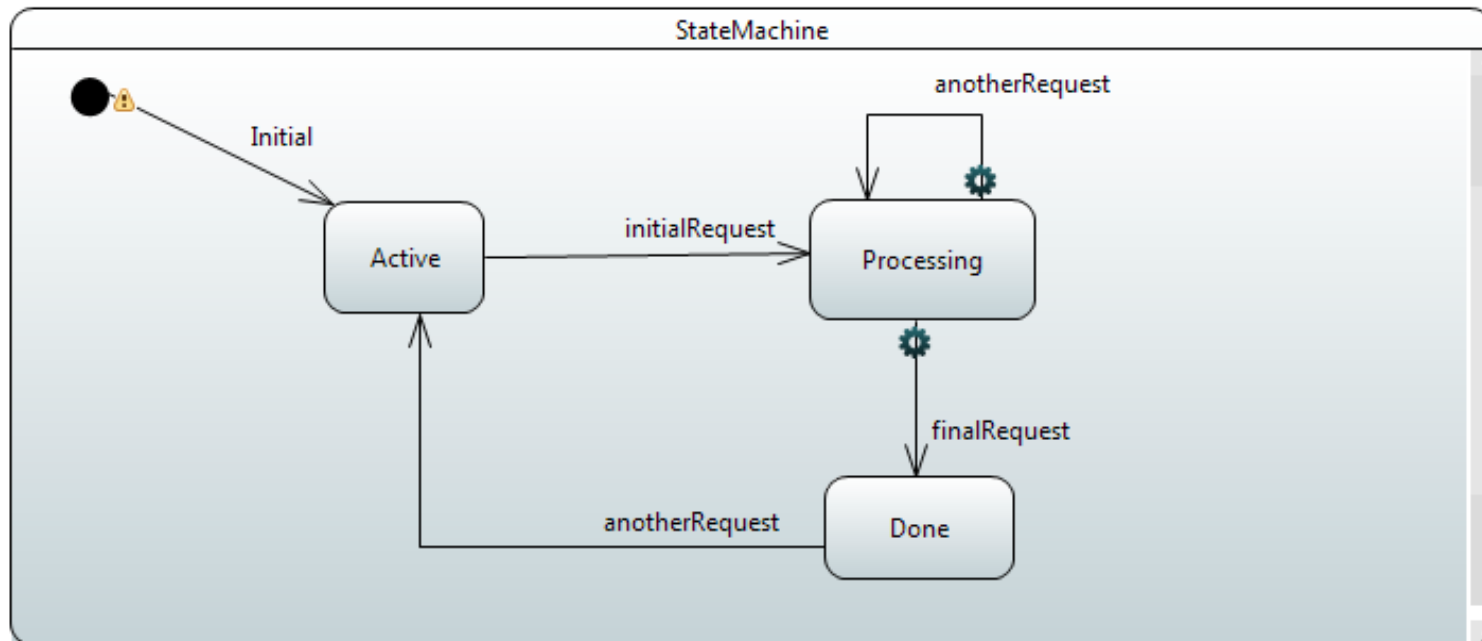
- Typically, one SPP port and multiple SAP ports

- **bool deregisterSAP()**

- **bool deregisterSPP()**

Ways to Avoid ‘Dropped Messages’

- Internal transition with trigger set to ‘any event’ (i.e., ‘*’), or
- Use ‘defer/recall’
 - In effect code of self transition of state ‘Processing’: `msg->defer()` ;
 - Then, after ‘finalRequest’: `port.recall()` ;



Run Time Services (RTS) Library: Timer Services

- **UMLRTTimerProtocol**

- Protocol for timer ports

- **Methods**

- **UMLRTTimerId informAt(UMLRTTimespec)**

- ‘one-shot’ timer, absolute

- Example: `'UMLRTTimespec now;`

- `UMLRTTimespec::getclock(now);`

- `timer.informAt(now + UMLRTTimespec(5, 0));'`

- **UMLRTTimerId informIn(UMLRTTimespec)**

- ‘one-shot’ timer, relative

- Example: `'timer.informIn(UMLRTTimespec(5, 0));'`

- **UMLRTTimerId informEvery(UMLRTTimespec)**

- Periodictimer

- Example: `'timer.informEvery(UMLRTTimespec(5, 0));'`

- **cancelTimer(UMLRTTimerId)**

- **UMLRTTimespec**

- Supports comparison (e.g., '<'>=', '==') and simple manipulation (e.g., '+', '-')

Run Time Services (RTS) Library: Logging Services

- **UMLRTLogProtocol**
 - Type of log ports
 - **Methods**
 - **log(primitiveType)**
 - With newline appended
 - **show(primitiveType)**
 - No newline appended
 - **cr(int)**
 - Output newlines

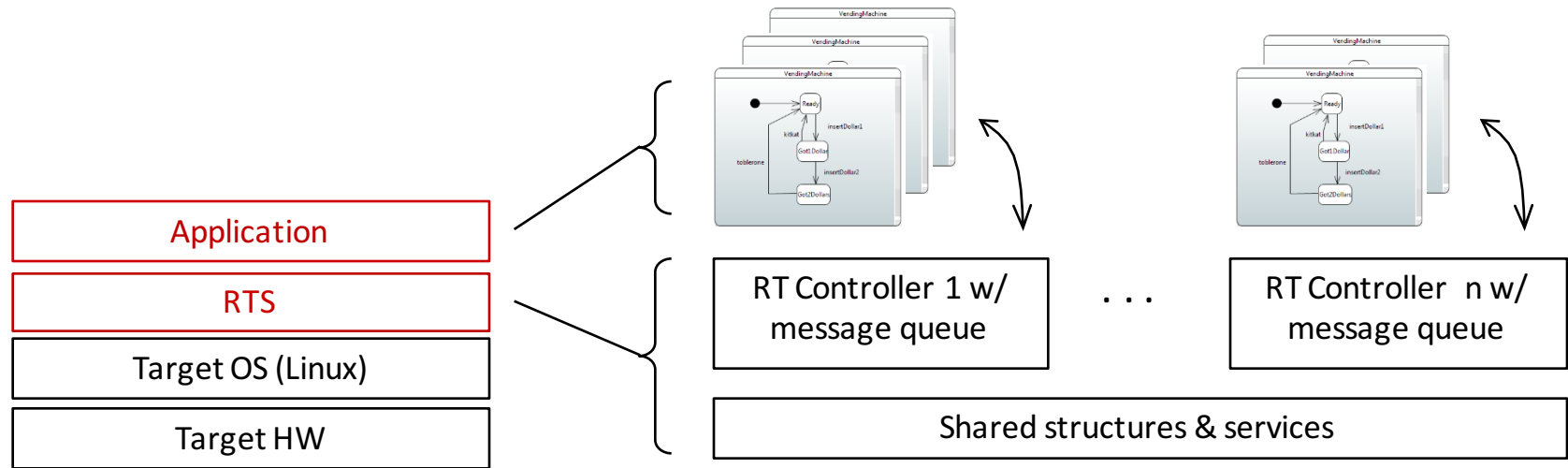
Run Time Services (RTS) Library: Frame Services

- **UMLRTFrameProtocol**
 - Type of frame ports
 - **Methods**
 - `bool destroy(UMLRTCapsuleId)`
 - `UMLRTCapsuleId incarnate(UMLRTCapsuleClass)`

The RTS is Actually Not That Large

```
Juergen Dingel@dingel420s /cygdrive/c/Users/Juergen Dingel/Programs/papyrus-rt-win_
Dec_2016/Papyrus-RT/plugins/org.eclipse.papyrusrt.rts_0.8.0.201612120508/umlrts/uml
rt
$ more *.cc | wc -l
9608
```

Creating Multi-Threaded Applications



- Specifying thread assignment
 - File 'Top.controllers' in same directory as generated executable
 - E.g.,
- Possible consequences of using multiple threads/controllers?

```
Top = MainThread
Top.pinger = pingerThread
Top.ponger = pongerThread
```