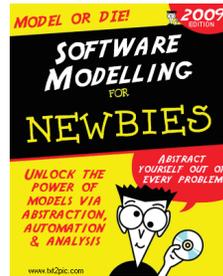


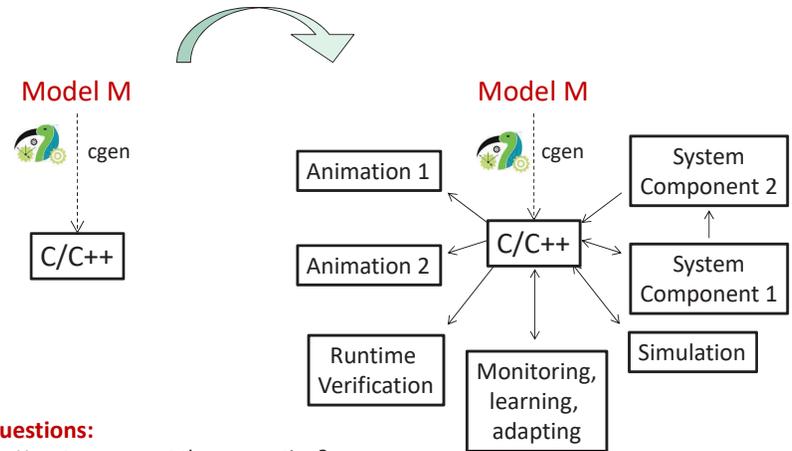
CISC836: Models in Software Development: Methods, Techniques and Tools



UML-RT and RSARTE Part VI: Integrating Models

Juergen Dingel
Feb 2020

From Isolated to Connected



Questions:

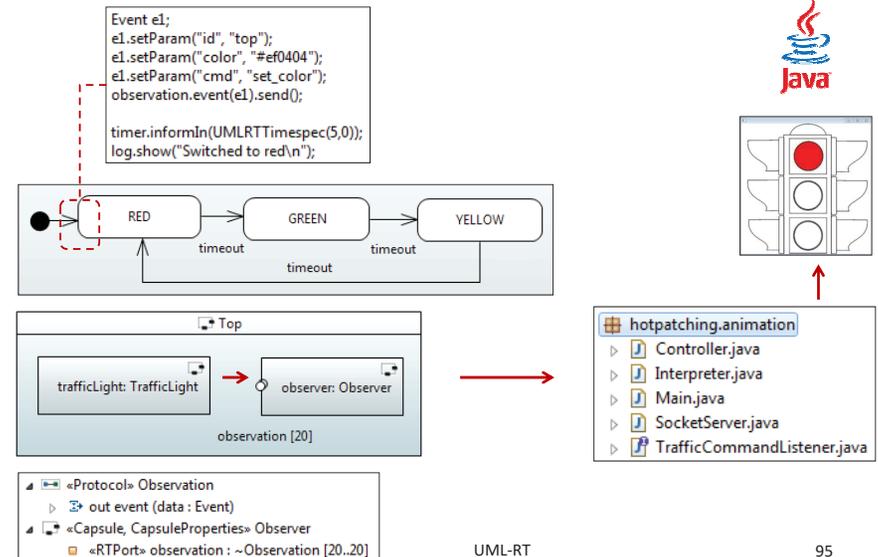
1. How to represent the connection?
2. Which communication mechanisms to support?

Representing Connections: Option 1 Observer Capsule

- **Dedicated, yet general capsule**
 - Serving as generic 'gateway'
- **Connector**
 - Dynamic, automatic, name-based registration (SAP/SPP)
- **Messages**
 - Single outgoing message 'event(data)' with general data format
 - Incoming messages can be added

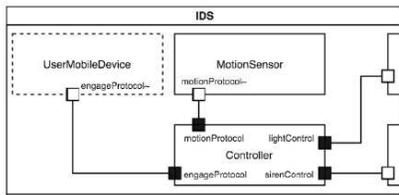
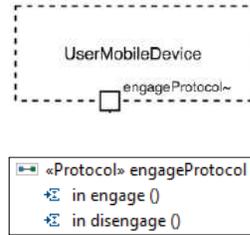


Observer Capsule: Examples

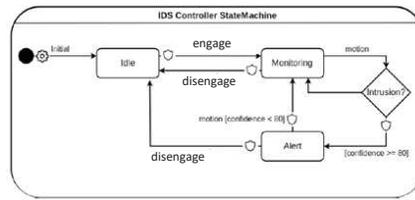


Representing Connections: Option 2 Proxy Capsule

- **Dedicated, application-specific capsule**
 - Representing specific, application-relevant, external component
- **Messages**
 - Component-specific protocol



CISC 836, Winter 2020

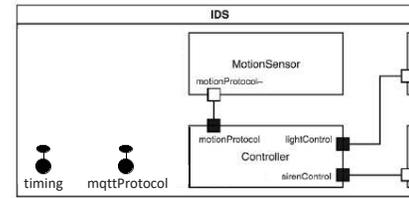


UML-RT

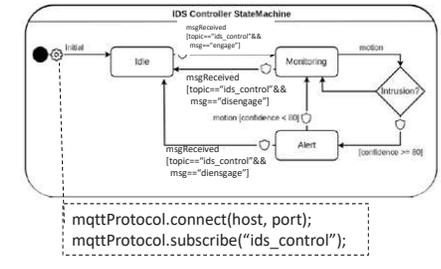
96

Representing Connections: Option 3 System Ports

- **Protocol-specific extension to RTS**
 - Similar to, e.g., timing services
- **Messages**
 - RTS developer defined



CISC 836, Winter 2020



UML-RT

97

Representing Connections: Summary

	Shown in capsule diagram?	Messages	Suitability
Observer capsule	Yes	Single, pre-defined out message	Animation, RT monitoring, (steering)
Proxy capsule	Yes	Capsule-specific, defined by user	Integration
System port	No	Protocol-specific, defined by RTS developer	Integration

CISC 836, Winter 2020

UML-RT

98

Connecting: Communication Mechanisms

- **Shared memory**
- **Communication protocols**
 - TCP/IP
 - Publish/subscribe
 - MQTT
 - DDS
 - Request/response
 - CoAP
- **Serialization**
 - JSON
 - Widely used, ASCII representation
 - Flatbuffers (Google)
 - Interface Definition Language, binary encoding

CISC 836, Winter 2020

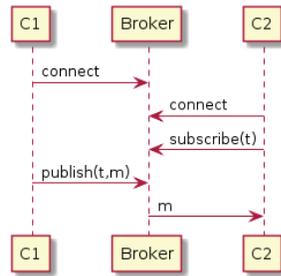
UML-RT

99

Communication Protocols: MQTT

Message Queue Telemetry Transport (MQTT)

- Publish/subscribe protocol
- Light-weight, low resource requirements
- Easy to use:
 - connect(brokerAddr, brokerPort), disconnect(),
 - subscribe(topic), unsubscribe(),
 - publish(topic, message)
- Standardized (ISO)



Broker

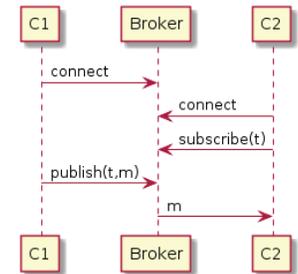
Ensures that all subscribers to a specific topic receive every message published to that topic

Topic	Subscribers
"Temperature/bedroom"	Component 2
...	...

Communication Protocols: MQTT (Cont'd)

Topics as filters

- Can be **hierarchical**, e.g.,
 - office/floor1/room1/temperature/
 - will match
 - office/floor1
- Can contain **wildcards**, e.g.,
 - single level: office/floor1+/temperature
 - multi level: office/*/temperature



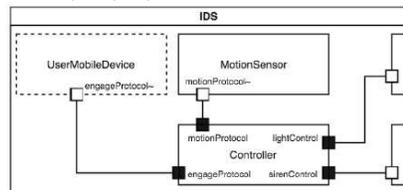
Implementations

- E.g.,
 - Eclipse Paho: MQTT clients for 10 different languages
 - Eclipse Mosquitto: MQTT broker

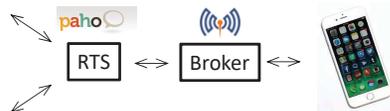
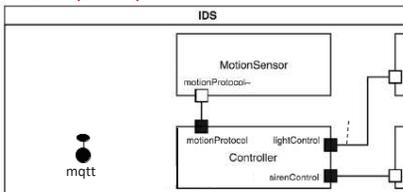


TCP/IP and MQTT in Papyrus-RT

MQTT proxy capsule w/ JSON



MQTT system ports



RTS Library Extension: MQTTProxyCapsule

MQTTProxyCapsule (in mqttproxycapsule.hh)

Properties

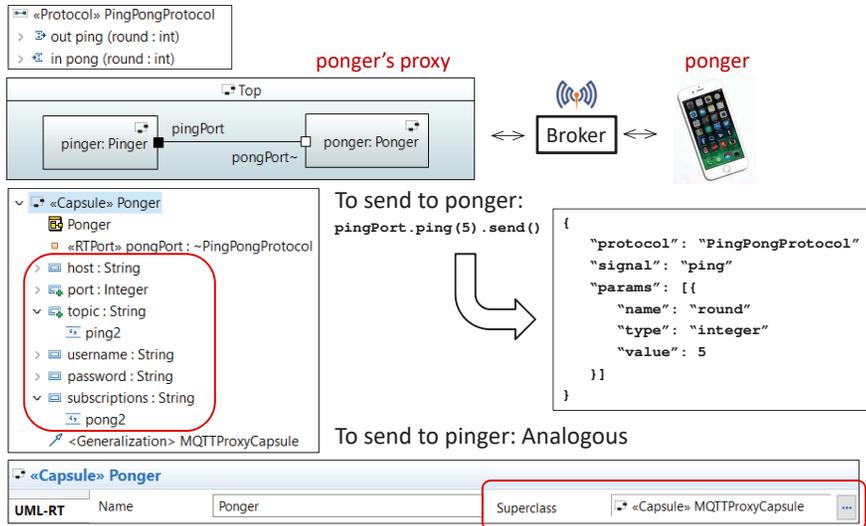
- `host: String, port: Integer` // to connect to broker
- `topic: String` // topic of the messages sent
- `subscriptions: String` // topics subscribed to
- `username, password: String`

Assumptions

- Proxy capsule
 - extends `MQTTProxyCapsule`
 - must have exactly one port (conjugate)
- Implementing external component
 - subscribed to topics listed in 'subscriptions'
 - publishes input messages listed in port's protocol to topic 'topic'
- 'subscriptions' has format "topic1; topic2; ...; topicn"

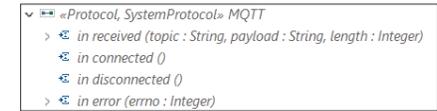


RTS Library Extension: MQTTProxyCapsule (Cont'd)

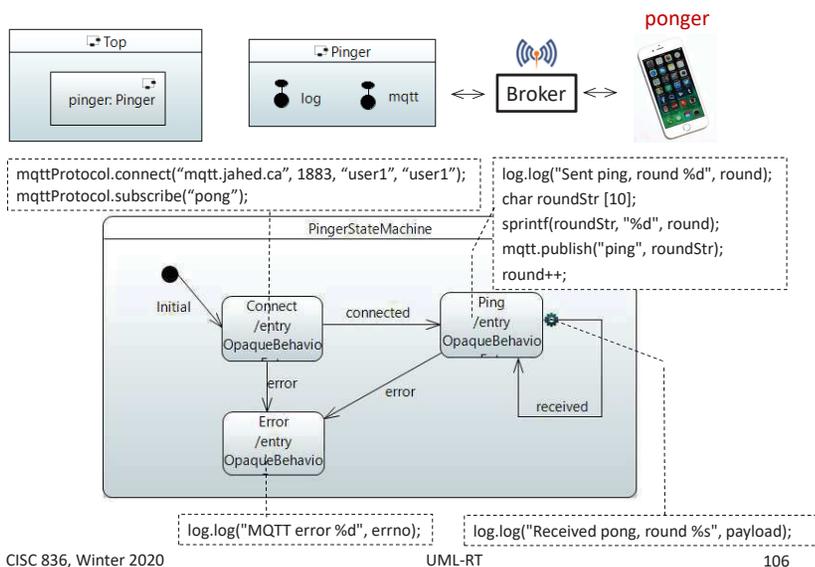


RTS Library Extension: MQTT System Port

- **UMLRTMQTTProtocol** (in `umlrtmqttprotocol.hh`)
 - Type of MQTT system port
 - **Methods implemented**
 - `connect(host:String, port:Integer, uname:String, pwd:String)`
- E.g.: `mqtt.connect("mqtt.jahed.ca", 1883, "user1", "user1");`
 - `disconnect()`
 - `subscribe(topic:String)`
- E.g.: `mqtt.subscribe("pong");`
 - `publish(topic:string, msg:String)`
- E.g.: `mqtt.publish("ping", roundStr);`
 - **Messages generated**
 - `connected()`
 - `disconnected()`
 - `error(errno:Integer)`
 - `received(topic:String, payload:String, length:Integer)`

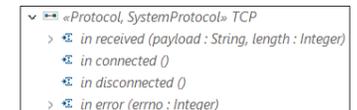


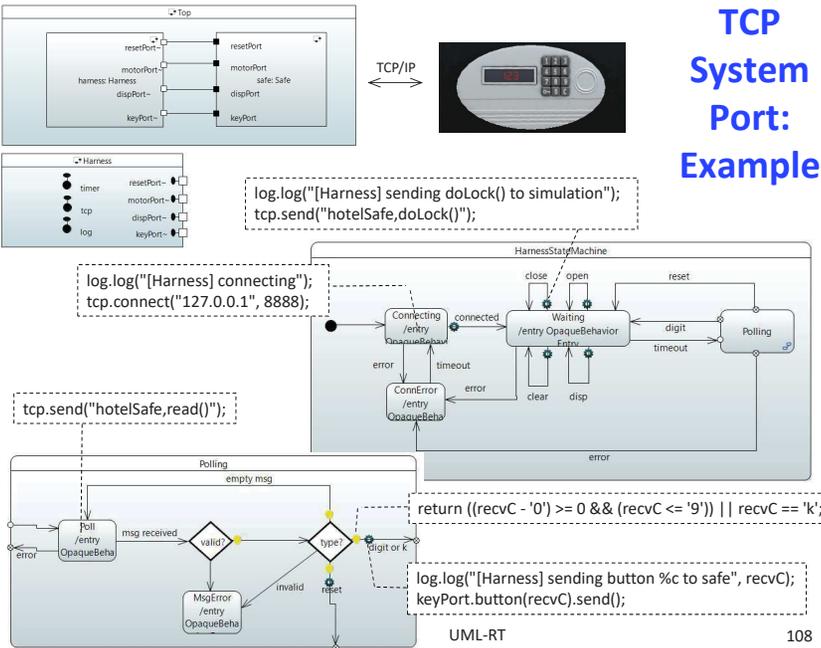
MQTT System Port: Example



RTS Library Extension: TCP System Port

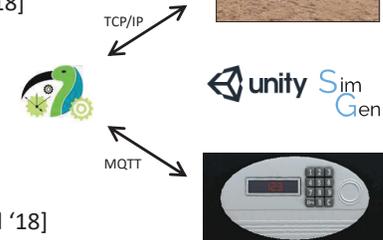
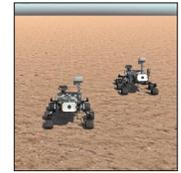
- **UMLRTTCPProtocol** (in `umlrttcpprotocol.hh`)
 - Type of TCP system port
 - **Methods implemented**
 - `connect(host:String, port:Integer)`
- E.g.: `tcp.connect("127.0.0.1", 8888);`
 - `disconnect()`
 - `send(msg:String)`
- E.g.: `tcp.send("hotelSafe,doLock()");`
 - **Messages generated**
 - `connected()`
 - `disconnected()`
 - `error(errno:Integer)`
 - `received(topic:String, payload:String)`





TCP/IP and MQTT in Papyrus-RT: Examples

- System ports
 - TCP/IP
 - Rover challenge problem [MDETools'18], [Pasternak et al '18]



- Proxy capsules & system ports
 - Hotel safe simulation [Dingel et al '18]

[MDETools'18] Bagherzadeh, Bordeleau, Dingel, Famelis, Garcia-Dominguez, Oliveira, Posse, Seidewitz, Selic. 2nd Workshop on MDE Tools. MODELS'18. 2018

[Pasternak et al '18] Pasternak, Kahani, Bagherzadeh, Dingel, Cody. "SimGen: A tool for generating simulations and visualizations of embedded systems on the Unity Game Engine". MODELS'18 Demo Track. 2018

[Dingel et al '18] Dingel, Jahed, Posse. "Making modeling cool again". Tutorial at MODELS'18. Materials available at <https://github.com/kjahed/Models18-MMCA>

Other Communication Protocols

- Data Distribution Service (DDS)
 - Also pub/sub
 - Compared to MQTT
 - Decentralized
 - For message-intensive M2M communication
 - Focus on high performance
- Constrained Application Protocol (CoAP)
 - Request/reply ala HTTP
 - For resource-constrained environments

Communication Mechanisms: Summary

	Shared Memory	TCP/IP	MQTT	DDS	CoAP	Serialization	
						JSON	FlatBuffers
Observer Capsule	✓	✓					
Proxy Capsule			✓	✓	✓	✓	✓
System Ports		✓	✓				

✓ Available at <https://github.com/kjahed/>

✓ Experimental prototype. Please contact us