Queen's School of Computing
Queen's University
Kingston, Ontario, Canada

Queen's University

Sum Ergo Computo
I am therefore I compute

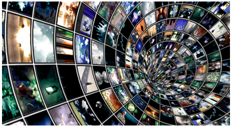
Evaluation of UML-RT and Papyrus-RT for Modelling Self-Adaptive Systems

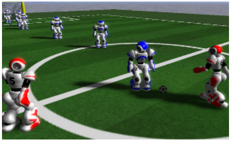
Nafiseh Kahani, Nicolas Hili, James R. Cordy, Juergen Dingel
{kahani,hili,cordy,dingel}@cs.queensu.ca
School of Computing, Queen's University,
Kingston, Ontario, Canada

MiSE Workshop
May, 2017

Self-Adaptive Software (SAS) Systems

SAS systems can adapt their behaviour, because of evolving their current configurations or in response to the changing environment's context to provide the required services and maintain the intended properties.





This capability can be achieved by reconsidering environmental or system requirements, and then applying the proper adaptation process to cope with the changing conditions.

Problem Statement

Design Complexity

☞ The **uncertain** and **dynamic** nature of SAS systems makes their design a **complex task**.

▼

Model-Driven Engineering (MDE)

✓ MDE techniques can be applied to **manage** the **increased design complexity** posed by the adaptation process and increase **reliability**. **However...**

▼

Suitability ?

➟ Despite a strong interest in modelling SAS systems for real-time, **few evaluations** of modelling languages **exist** to assess their **suitability** for providing an appropriate set of concepts for modelling adaptations.

Objectives

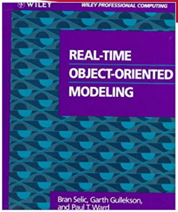
- ✏ Evaluation of the suitability of UML for Real-Time (UML-RT) features for modelling SAS systems;
  - ➟ Each feature is related to the structural and behavioural adaptations it covers.

- ✏ Assessment of the current state of support of these features by Eclipse Papyrus for Real-Time (Papyrus-RT).

Outline

UML-RT: History

- Real-time Object-Oriented Modeling (ROOM)
  - Early 1990 ties
  - ObjecTime Ltd

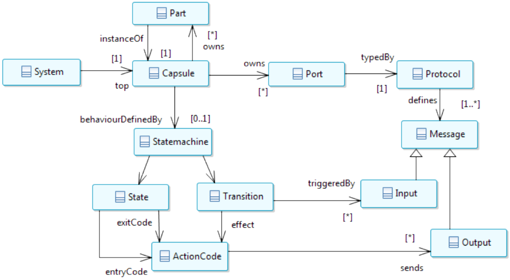


[SGW94] B. Selic, G. Gullekson, and P.T. Ward.
Real-Time Object-Oriented Modelling. Wiley. 1994

- Major influence on UML 2

- Supported by many tools:
  - Rational Rose for Real-Time (RoseRT)
  - RSARTE
  - Eclipse Papyrus for Real-Time (Papyrus-RT)

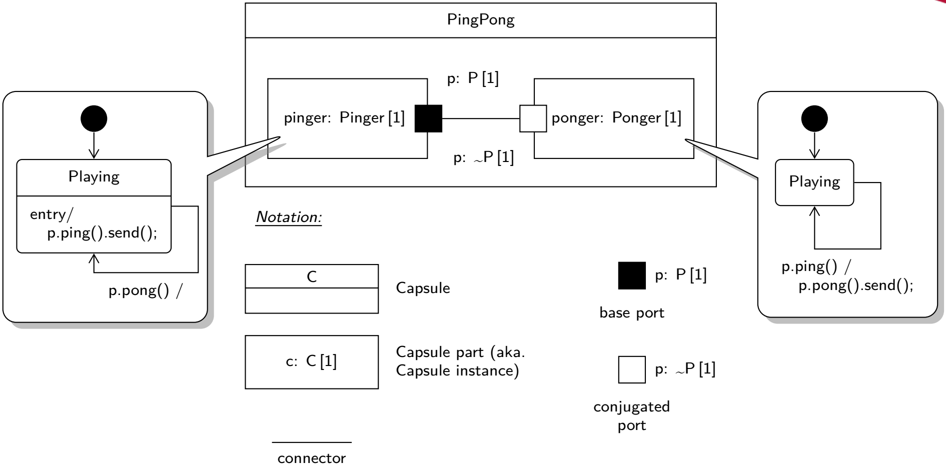
UML-RT: Core Concepts



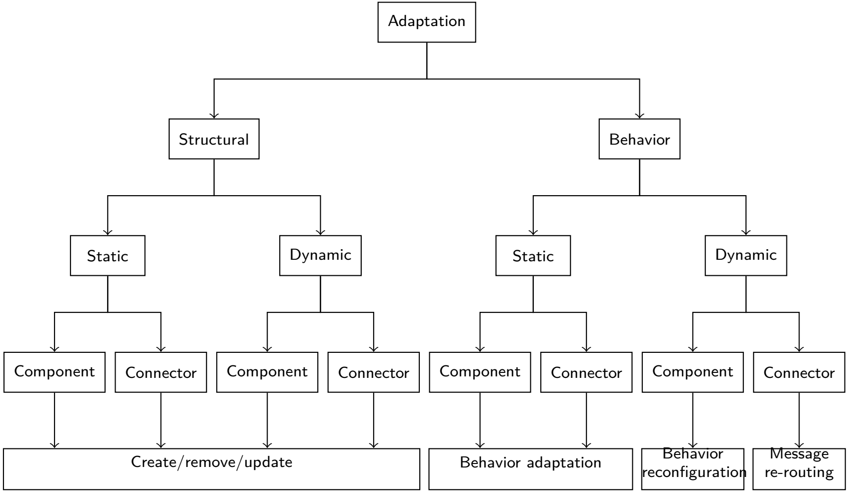
[Dingel17] Dingel, J. (2017). CISC836: Models in Software Development: Methods, Techniques and Tools

- System: collection of capsules (instances) communicating via messages
- Capsule (instances):
  - Active class (object) whose behaviour is defined by a state machine
  - Communicates by sending and receiving messages through its ports
- Port: typed by protocol defining input and output messages
- State machine:
  - Transition triggered by incoming messages
  - Action code contains statements that send messages over certain ports

UML-RT Example

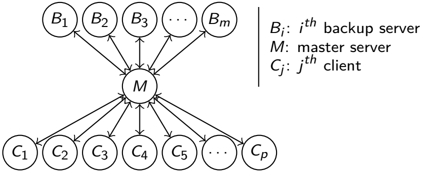


Taxonomy of Adaptations for Modelling SAS Systems

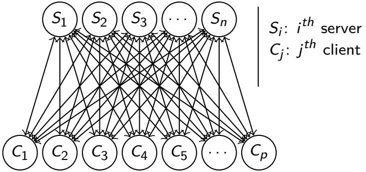


Case Study: Adaptive Failover System

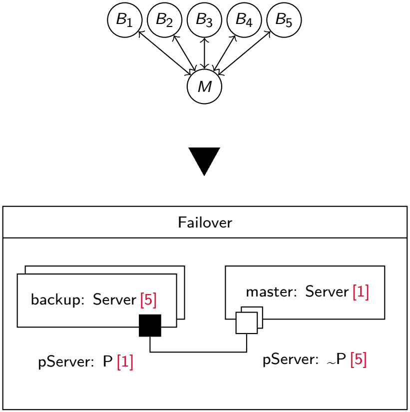
- Failover: a system involving a set of component servers to handle client requests;
- Goal: maximizing availability when designing fault-tolerant systems;
- Two different replication modes:
  - **Passive replication:** one **master** server and $m = (n-1)$ **backup** servers;



  - **Active replication:** all the servers can process client requests.



Static Structural Adaptations for Comps.& Connectors



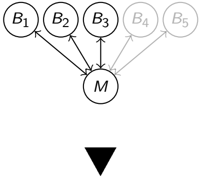
Static replication:

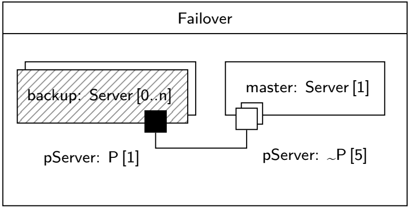
- ≈ *multiplicity* of a capsule part/port;
- Statically defined within the model;
- Action language:

```
// Without replication
pServer.msg().send() // unicast

// With replication
pServer.msg().send(); // broadcast
pServer.msg().sendAt(index); // unicast
```

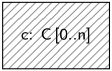
- No support for unrelated connectors

Dynamic Structural Adaptations for Components





Notation:



Optional capsule part

Dynamic replication:

- Optional/plugin capsules;
- Dynamically created instances at run-time;
- Action language

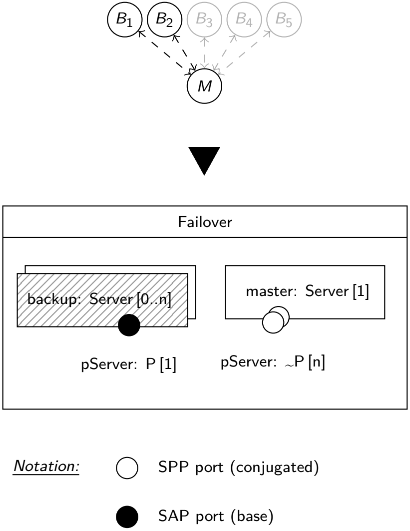
Incarnation:

```
// Incarnating a capsule
UMLRTCapsuleId backupId =
  frame.incarnate(backup, Server);

if(!backupId.isValid())
  context()->perror("incarnate failed.");

printf("capsule: @%p\n",
  backupId.getCapsule());
```

Destruction:

```
// Destroying a capsule
if(!frame.destroy(backupId))
  context()->perror("destroy failed.");
```

Dynamic Structural Adaptations for Connectors



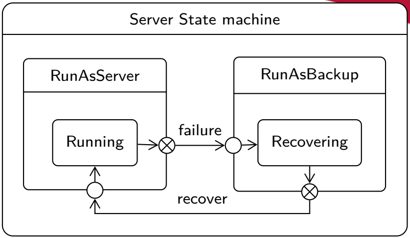
Dynamic binding:

- SAP/SPP ports;
- Capsule parts are dynamically bound at run-time (no explicit connectors);
- SAP/SPP ports act like normal ports (no specific action language).

Specific constraints:

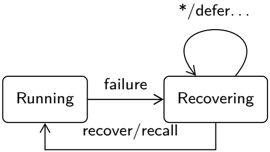
- One SPP port can be bound to several SAP ports;
- Ports must have the same type and name;

Static & Dynamic Behavioural Adaptations



Static adaptations:

- *Component:*
  - Hierarchical state machines;
  - Component / State machine inheritance.
- *Connector:*
  - protocol inheritance.

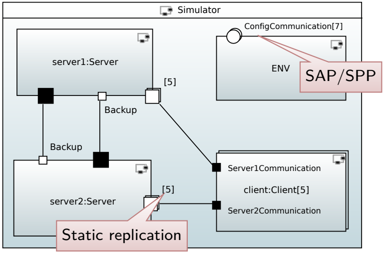



```
// Deferring messages
msg->defer();

// Recalling messages from a port
port->recall(...)
```

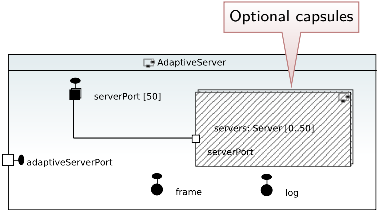
Dynamic adaptations:

- *Component:*
  - No support for behavioural reconfiguration ($\approx$ traits).
- *Connector:*
  - No explicit support in UML-RT;
  - Refer/recall mechanism.

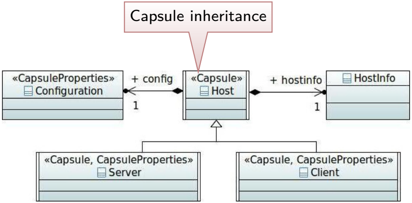
Implementations



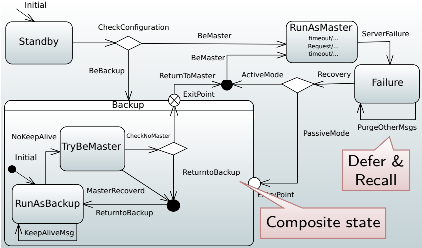
Failover w/o load balancer



Failover with load balancer

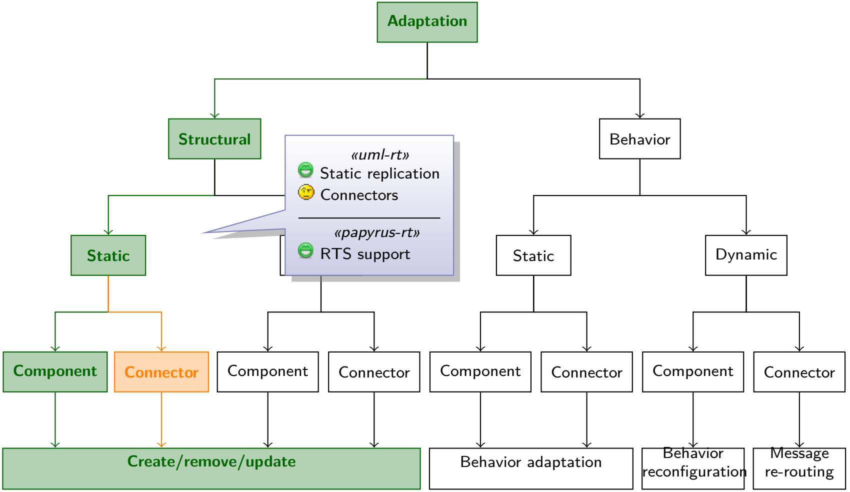


Class diagram inheritance



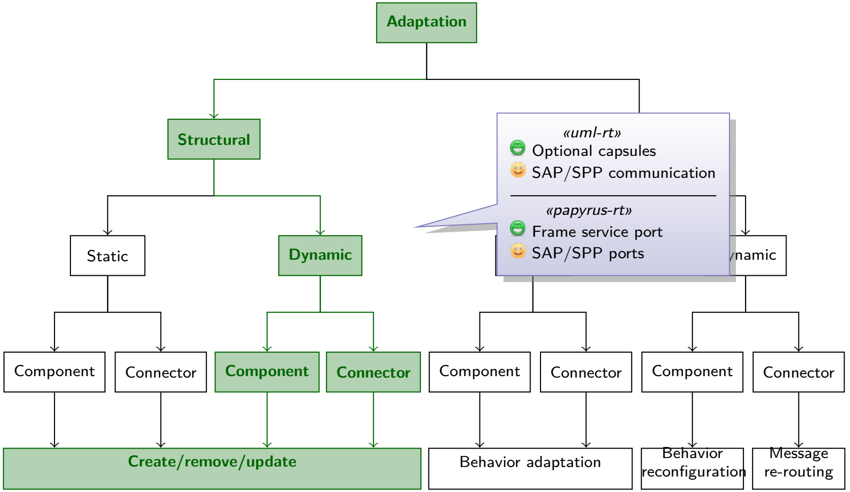
Server state machine

Evaluation



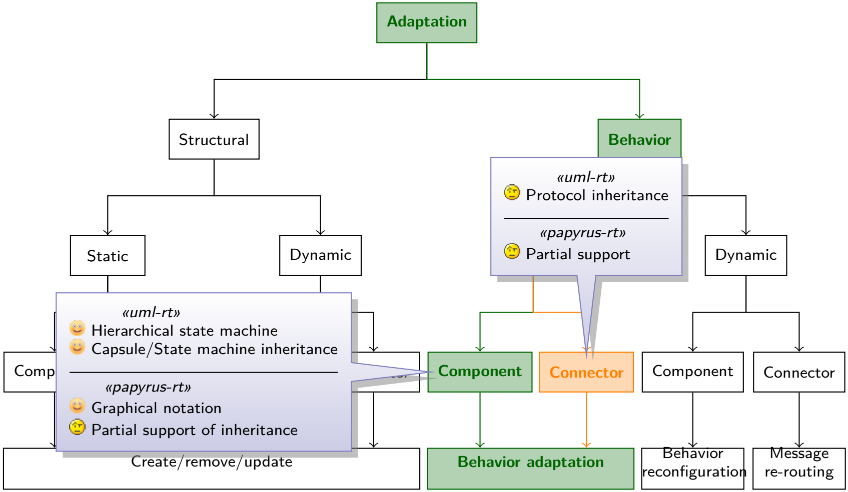
Adaptation
Structural
Behavior
«uml-rt»
Static replication
Connectors
«papyrus-rt»
RTS support
Static
Static
Dynamic
Component
Connector
Component
Connector
Component
Connector
Component
Connector
Create/remove/update
Behavior adaptation
Behavior reconfiguration
Message re-routing

Evaluation

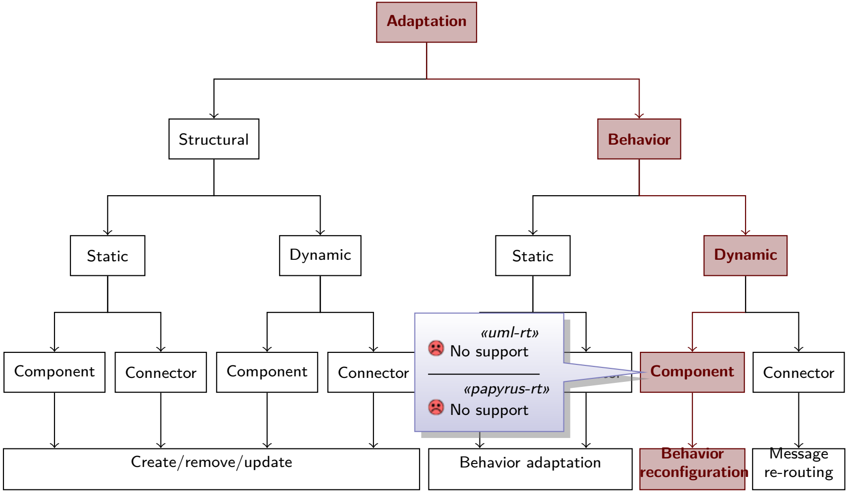


Adaptation
Structural
Static
Dynamic
Component
Connector
Component
Connector
Component
Connector
Component
Connector
Create/remove/update
Behavior adaptation
Behavior reconfiguration
Message re-routing
«uml-rt»
Optional capsules
SAP/SPP communication
«papyrus-rt»
Frame service port
SAP/SPP ports

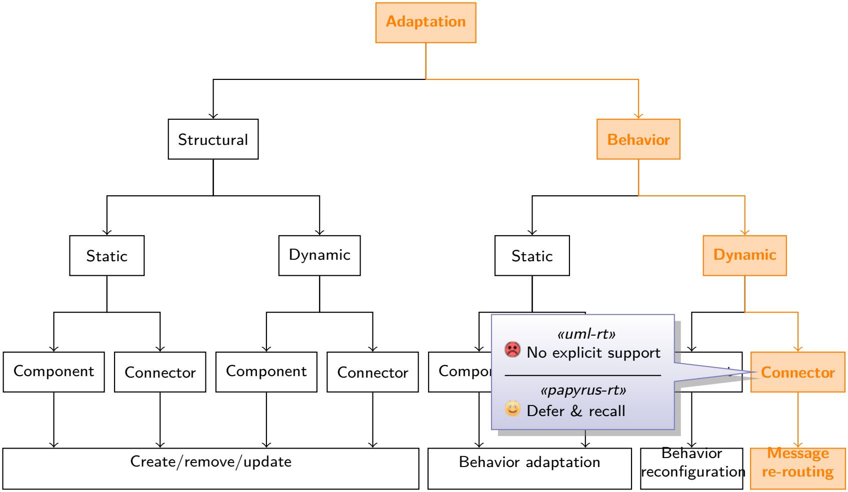
Evaluation


Adaptation
Structural
Behavior
Static
Dynamic
Dynamic
«uml-rt»
Protocol inheritance
«papyrus-rt»
Partial support
«uml-rt»
Hierarchical state machine
Capsule/State machine inheritance
«papyrus-rt»
Graphical notation
Partial support of inheritance
Comp
Component
Connector
Component
Connector
Create/remove/update
Behavior adaptation
Behavior reconfiguration
Message re-routing

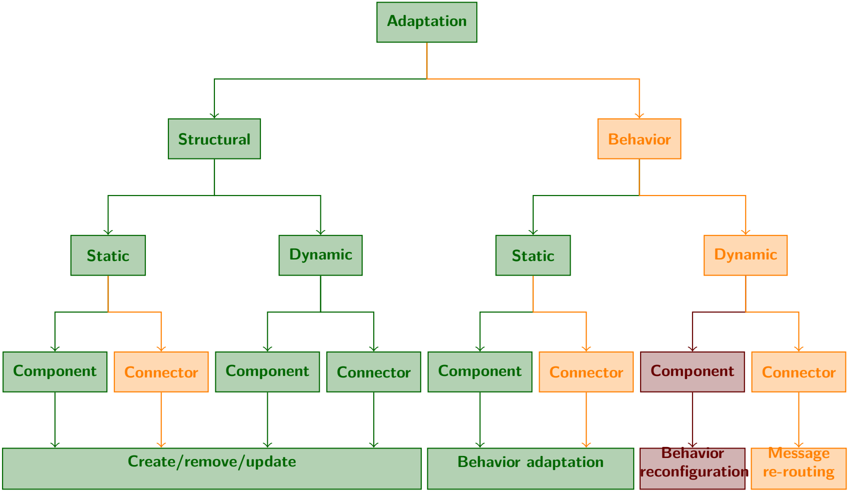
Evaluation


Adaptation
Structural
Behavior
Static
Dynamic
Static
Dynamic
Component
Connector
Component
Connector
Component
Connector
«uml-rt»
No support
«papyrus-rt»
No support
Create/remove/update
Behavior adaptation
Behavior reconfiguration
Message re-routing

Evaluation


Adaptation
Structural
Behavior
Static
Dynamic
Static
Dynamic
Component
Connector
Component
Connector
Connector
Create/remove/update
Behavior adaptation
Behavior reconfiguration
Message re-routing
«uml-rt»
No explicit support
«papyrus-rt»
Defer & recall

Evaluation


Adaptation
Structural
Behavior
Static
Dynamic
Static
Dynamic
Component
Connector
Component
Connector
Component
Connector
Component
Connector
Create/remove/update
Behavior adaptation
Behavior reconfiguration
Message re-routing

Outline

Summary

- Evaluation of the suitability of UML for Real-Time (UML-RT) features for modelling SAS systems;
  - ➡ Most of the adaptations are covered by UML-RT
  - ➡ UML-RT still lacks of support for behavioural adaptation at run-time
- Assessment of the current state of support of these features by Eclipse Papyrus for Real-Time (Papyrus-RT);
  - ➡ Mostly supported by service ports and the RTS
  - ➡ Some RTS mechanisms can be used to support behavioural adaptations

Queen's Computing

Queen's University
Kingston, Ontario, Canada

Queen's University

Sum Ergo Computo
I am therefore I compute

Thank you !