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

Queen's University

I am therefore I compute

A Model-Based Architecture for Interactive Run-Time Monitoring

Nicolas Hili, Mojtaba Bagherzadeh, Juergen Dingel

Outline

- Motivation
- Implementation
- Progress
- Demo
- Conclusion and Future Work

Motivation

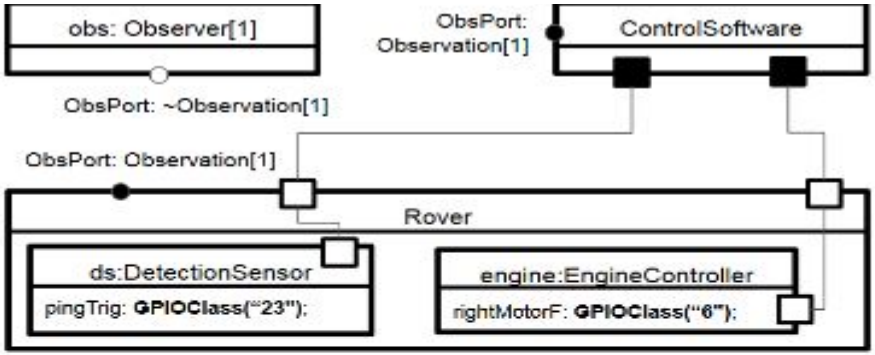
- Observing the system execution plays a critical role in the maintenance and debugging of real-time embedded systems.
- In addition to addressing the main requirements, observing of RTE systems should cause predictable overhead.
- UMLRT as a well-known modelling language for RTE system does not provide any built-in feature to address the observation.

Goal

Design and implement an observability architecture to meet the following requirements:

- Efficient observation with predictable overhead.
- Configurable for different workload and requirements.
- Supports active observation (Steering).
- Easy to use
- Supports different communication channel
- Being extensible to UMLRT language

Implementation- Observer Capsule



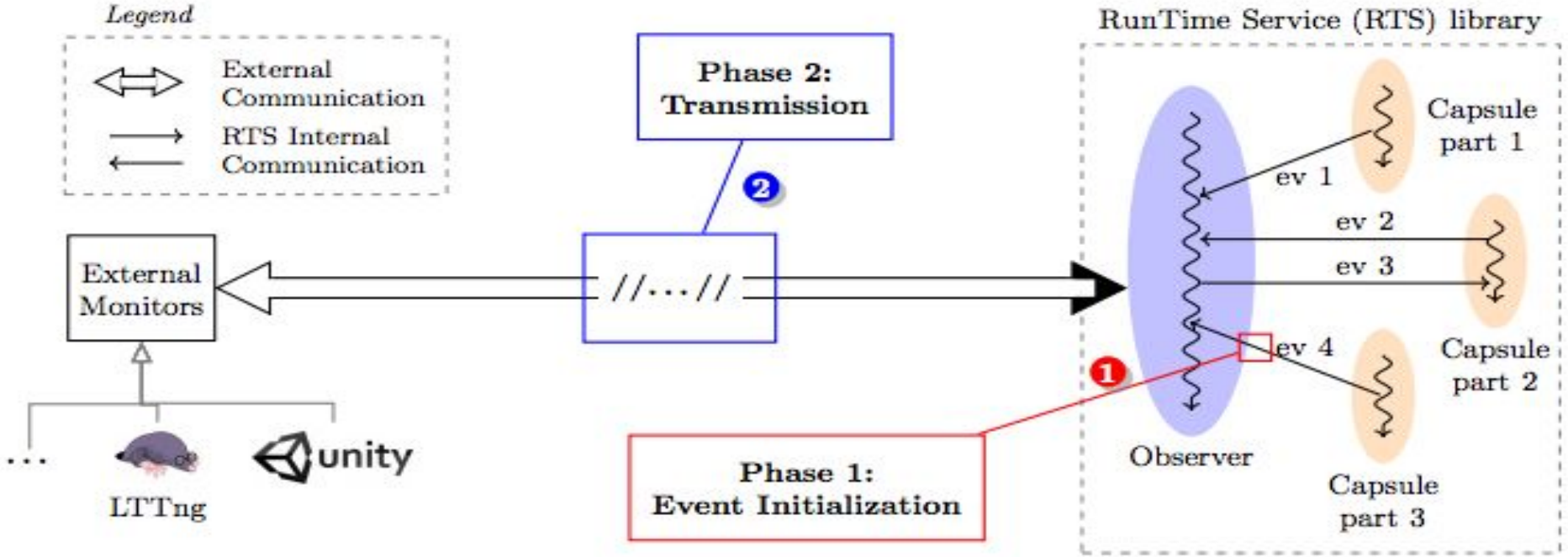
What is it ?

✓ Capsule implemented in UML-RT;
✓ Rely on the SAP / SPP communication;
✓ Intended to be integrated into the Papyrus-RT RunTime Service.

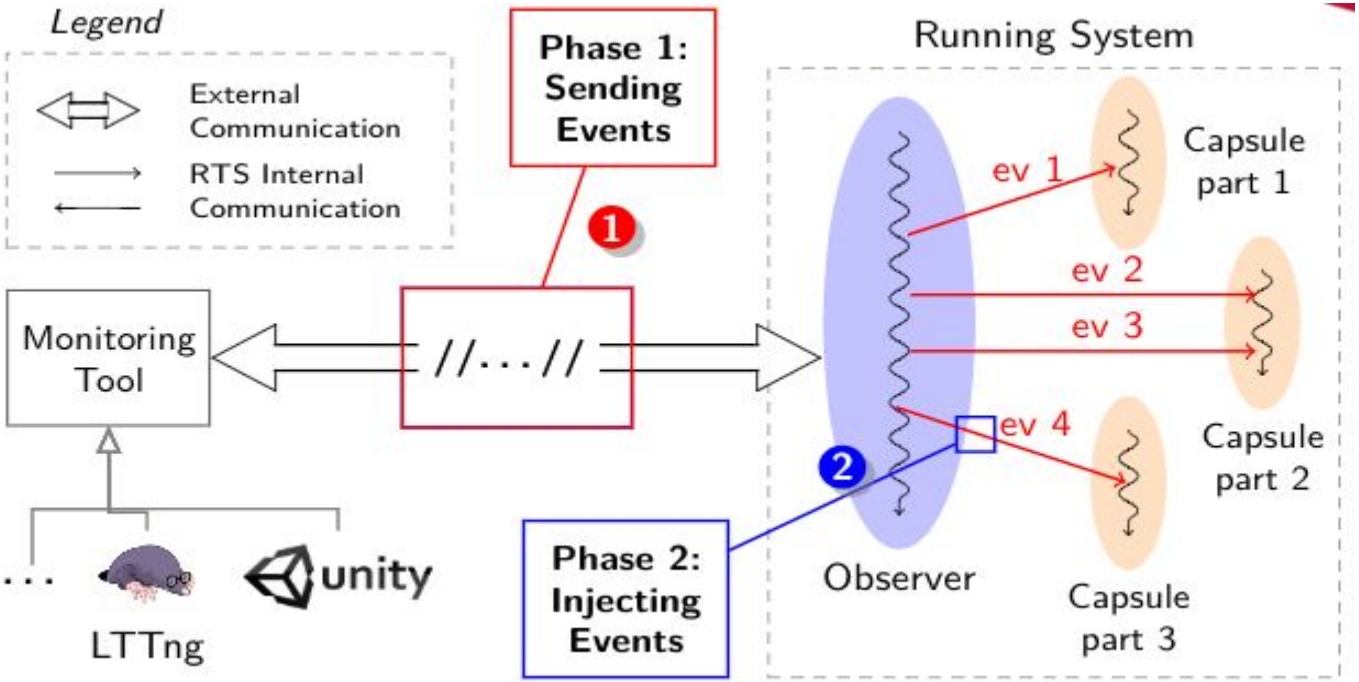
What for ?

✓ Gathering events from registered capsules;
✓ Transmitting events to external monitoring tools in a unified way;
✓ Reversible: events can be injected into the running system.

How it works (Observing)



How it works (Steering)



Conclusion and Future work

- Observer notation and concept are defined and integrated with PapyrusRT.
- Different experiments are done using Observer.(e.g., Rover, Unity, Model-level Debugger Communication)
- We used the observer for teaching and performed users study which shows 93% of users see the observer useful to understand the system behaviour.
- The performance overhead for different use cases is measured that is between 15% and 20%.
- We are working with ZeligSoft to extend the Observer concept to the core of the UMLRT language and PapyrusRT.

Thank You!