An Overview of MarQ

Giles Reger
University of Manchester, UK

RV 2016

In this Talk. . .

This talk will cover

- ▶ The background of the MARQ project
- ▶ The architecture of MARQ
- ▶ And then demonstrations of
    - ▶ Building QEA specifications
    - ▶ Online monitoring of a Java API property
    - ▶ Offline monitoring of a log file

These slides and the examples used here are available at

https://github.com/selig/qea

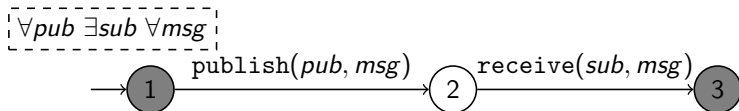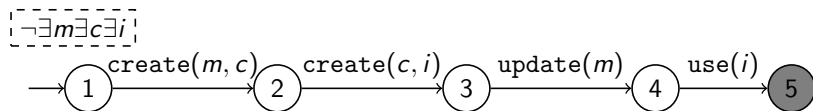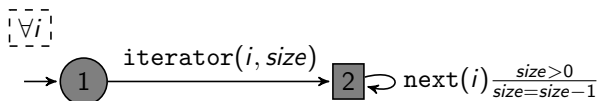# What is MarQ and where did it come from?

- MarQ stands for <u>M</u>onitoring <u>at</u> <u>r</u>untime with <u>Q</u>EA
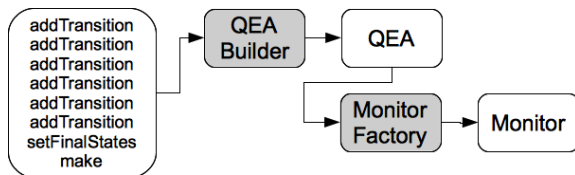- Quantified Event Automata is a formalism for RV introduced in 2012 by these people
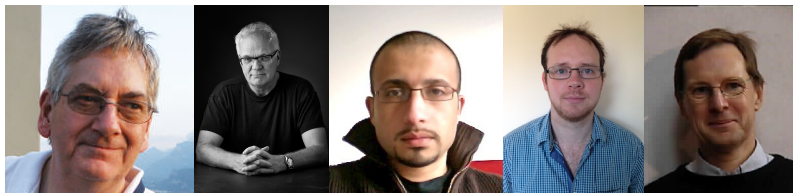
- QEA grew out of discussions trying to find a middle ground between the *expressiveness* of RuleR and the *efficiency* of JavaMOP
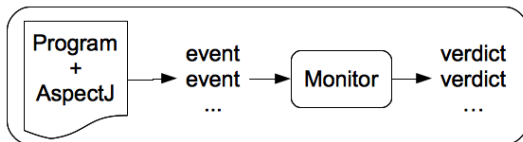
Quantified event automata

- Not a talk on QEA; other material on my website covers QEA
- QEA will be introduced via the following three properties

$\forall i$

$\rightarrow$ (1) $\xrightarrow{\text{iterator}(i, size)}$ [2] $\circlearrowleft$ $\text{next}(i)\frac{size>0}{size=size-1}$

$\neg \exists m \exists c \exists i$

$\rightarrow$ (1) $\xrightarrow{\text{create}(m, c)}$ (2) $\xrightarrow{\text{create}(c, i)}$ (3) $\xrightarrow{\text{update}(m)}$ (4) $\xrightarrow{\text{use}(i)}$ (5)

$\forall pub \; \exists sub \; \forall msg$

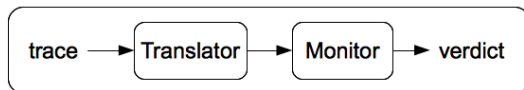$\rightarrow$ (1) $\xrightarrow{\text{publish}(pub, msg)}$ (2) $\xrightarrow{\text{receive}(sub, msg)}$ (3)

Architecture

Demonstration

- Go to the demonstration...

Summary of MarQ demonstration

We have briefly seen

- ▶ How to define QEAs
- ▶ How to use monitors with AspectJ
- ▶ How to monitor log files

We have also heard that

- ▶ MarQ is reasonably efficient (see competitions) but could be better (extensions are planned)
- ▶ There are improvements to usability needed and planned:
  - ▶ Better DSL for QEA
  - ▶ Language for defining guards and assignments
  - ▶ Automatic generation of AspectJ

Other plans include

- ▶ Translations between QEA and other RV specification languages