Towards a Benchmark Repository for Runtime Monitoring

Giles Reger

June 5, 2016

Outline

Two Talks

I want to have two related discussions:

1. The Competition
2. Towards a Benchmark Repository

Section 1

The Competition

CRV16 Status

Slight change to structure:

- Separate *implicit* and *general* subtracks for C

Since 2015, have a wiki where all benchmarks/specifications live.

Participants (so far):

C		Java	Offline
Implicit	General		
E-ACSL	E-ACSL	BeepBeep	BeepBeep
	LABMC	MarQ	MarQ
		Larva	CRL
		Mufin	

Issue : Low Participation

2014		
C	Java	Offline
3	4	4
2015		
C	Java	Offline
2	2	4
2016		
C	Java	Offline
2	4	3

Issue : Low Participation

2014		
C	Java	Offline
RiTHM-1 E-ACSL ~~ARTiMon -1~~ RTC	Larva JUnitRV JavaMOP ~~prm4j~~ ~~Java-MaC~~ MarQ	~~ZOT+SOLOIST~~ ~~LogFire~~ RiTHM-2 MonPoly ~~ARTiMon -2~~ STePr ~~Breach~~ MarQ
3	4	4
2015		
C	Java	Offline
2	2	4
2016		
C	Java	Offline
2	4	3

Issue : Low Participation

2014		
C	Java	Offline
3	4	4
2015		
C	Java	Offline
~~MarQ~~ E-ACSL ~~RiTHM-v2.0~~ RV-Monitor ~~TimeSquareTrace~~ ~~RTC~~	MarQ ~~TJT~~ ~~Java-MOP~~ Mufin	MarQ ~~RiTHM-v2.0~~ OCLR-Check RV-Monitor ~~OptySim~~ ~~AgMon~~ ~~Breach~~ LogFire
2	2	4
2016		
C	Java	Offline
2	4	3

Issue : Low Participation

2014		
C	Java	Offline
3	4	4
2015		
C	Java	Offline
2	2	4
2016		
C	Java	Offline
2	4	3

What's going on?

- Too much work?
- Too frequent?
- Not relevant?
- Badly advertised? Badly run?

Issue : Low Participation

What's going on?

- Too much work?
  - Structure requires participants to submit benchmarks
  - Then write lots of specifications
  - Will address this in the second half of talk
- Too frequent?
- Not relevant?
- Badly advertised? Badly run?

Issue : Low Participation

What's going on?

- Too much work?
- Too frequent?
  - I didn't win last year and haven't improved my tools
  - Didn't last year's competition only just finish?
- Not relevant?
- Badly advertised? Badly run?

Issue : Low Participation

What's going on?

- Too much work?
- Too frequent?
- Not relevant?
  - Scope is quite restricted... but RV is very broad
  - Similar benchmarks and tools each year
- Badly advertised? Badly run?

Issue : Low Participation

What's going on?

- Too much work?
- Too frequent?
- Not relevant?
- Badly advertised? Badly run?
  - Who are we reaching?
  - Are people happy...?

Proposal

1. Hold the current performance-driven competition every two years i.e. the next one will be in 2018

2. In alternate years hold one-off *special interest* competitions or show-cases

   For example:

   - Usability
   - Techniques combined with static analysis
   - Hardware monitoring
   - Monitoring uncertain systems (unreliable observation)
   - Focus on concurrency
   - Certification of monitoring
   - Application domains

   Idea: motivate research in an area (other fields have done this)

Is this still too frequent?

Proposal

1. Hold the current performance-driven competition every two years i.e. the next one will be in 2018

2. In alternate years hold one-off *special interest* competitions or show-cases

   For example: How to evaluate these things?

   - Usability
   - Techniques combined with static analysis
   - Hardware monitoring
   - Monitoring uncertain systems (unreliable observation)
   - Focus on concurrency
   - Certification of monitoring
   - Application domains

   Idea: motivate research in an area (other fields have done this)

Is this still too frequent?

Other Discussion Points

- Can we give further incentives for participation?
- How much does performance matter? (Time, Memory)
- What is an acceptable entry? ...hand-coded monitors? ...does it matter? ... issue of analysability
- How do we check that entered specifications are *correct*? We have not relationship between specification languages and (currently) a single trace/program.

Section 2

A Repository

Vision

- We have an online database containing a set of properties
- Each property is specified in various specification languages
- Each property has artefacts (traces or programs) attached to it that are known to satisfy or violate the property
- Database is populated by tool developers
- Further support for meta-data, trace formats, translation, evaluation etc

Reason 1: Support for Evaluation

- Evaluation selects a set of properties and runs monitors for selected attached specifications on selected artefacts

- The competition can use the subset of the database that corresponds to entered monitoring tools without any additional effort from entrants
- ... assuming entrants have put something in the database, but this can be done at any time
- The contents of the competition is known up-front,

- Researchers writing a paper can easily access benchmarks to compare their tool on
- Even more importantly, they have a medium to communicate benchmarks and solutions of interest

Reason 2: Insights into Languages etc

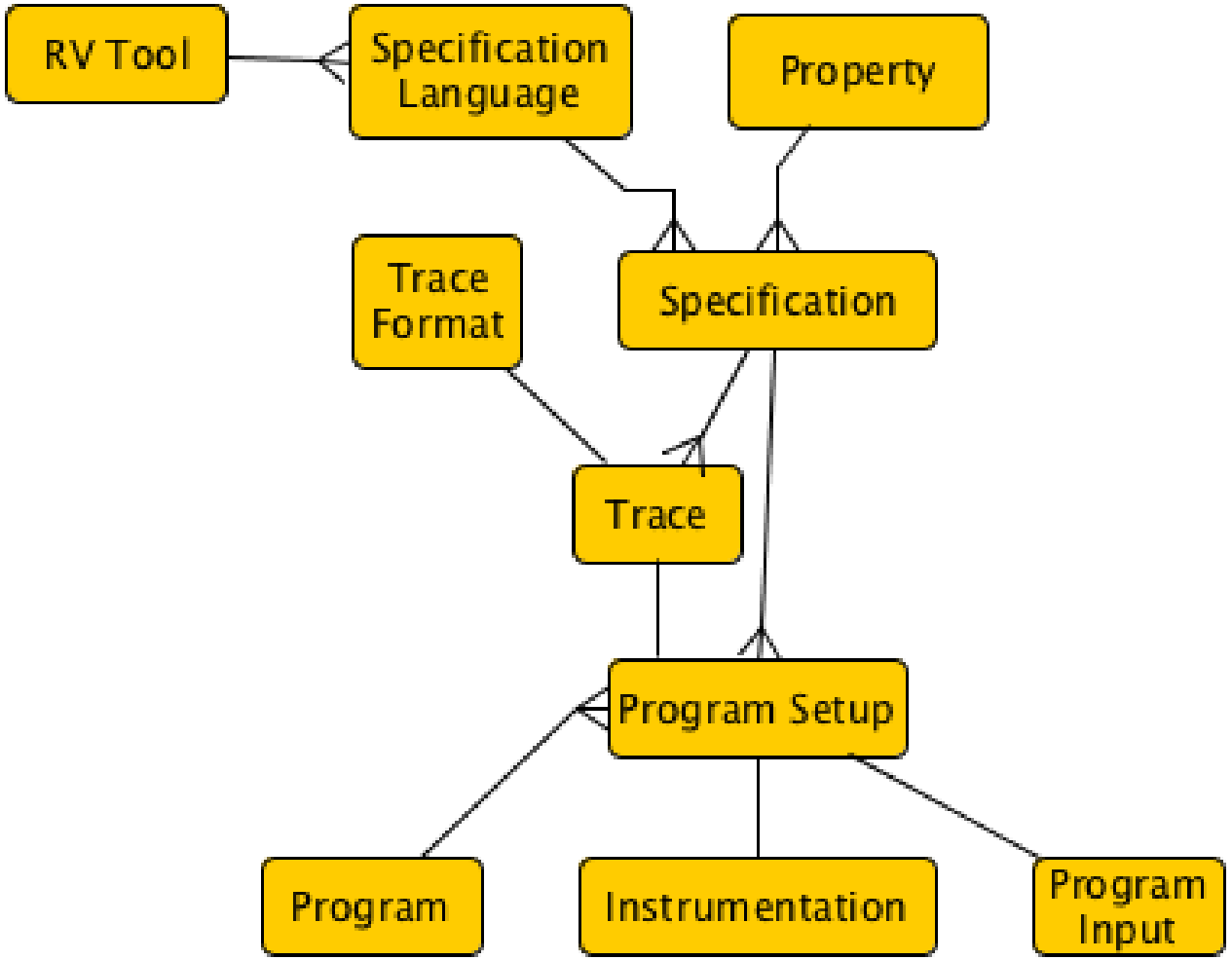
- Additional benefit.... we get a set of properties specified in various ways
- There is a general question of how specification languages are related. We now have a lot of data to help with this question
- Can use this information to develop translators that translate specifications in one language to another language, which can be used to further populate the database
- Which takes us another step towards *general* (intermediate) specification languages

Reason 3: We said we would

The development of a common infrastructure that enables the development of a collection of runtime verification problems and benchmarks for the comparison of algorithms and tools, and to increase their collaboration

The creation and maintenance of a collection of examples in the form of benchmarks, classified according to the taxonomy and expressed in the languages in the common format.

Design


RV Tool
Specification Language
Property
Trace Format
Specification
Trace
Program Setup
Program
Instrumentation
Program Input

First Step

- I propose, as a first step, to collect *properties* and *specifications*
- i.e. the first step does not collect artefacts

- Providing tagging mechanism to begin to form a classification
- Do not fix this up-front... we don't know what it should look like yet

- Hopefully this will give us some insights into what RV taxonomy should look like

Property Object

- Unique name
- Alphabet of observations
- English description of ordering constraints
- Short examples of correct and incorrect behaviour
- List of *tags*
- List of references to papers mentioning this property
- List of *Specification Object* items specifying this property

Specification Object

- *Property Object* of property being specified
- *Specification Language*
- The formal specification
- Description/Explanation
- List of references to papers mentioning this specification

- Later... links to artefacts

From Here...

- Anybody interested in *active* collaboration?
- Create input mechanism for this first step
- Send out requests for input
  - Direct input
  - Papers containing specifications/properties, which I will manually extract... bootstrapping
- Extend to full Artefacts
  - Depend on progress by WG2 Tasks (which need restarting)