Automated Validation of T&E Instrumentation Systems

SOUTHWEST RESEARCH INSTITUTE®

Austin Whittington

Benefiting government, industry and the public through innovative science and technology

Outline

- Motivation
- Automated Discovery
- Automated Validation
- Validation Workflow
- Advantages
- Conclusion

Motivation

- **Telemetry devices have constraints on what they can do**
  - Signal ranges, sample rates, etc.
- **Devices are configured using generic XML-based languages (e.g. MDL) that are capable of defining invalid configurations for a particular DAU**
- **Test cycles are short, and test objectives must be met as requirements and firmware change**
- **How to test what you think you know about the device's constraints?**

Big Picture

Current Approaches to Validation

- **APIs**
  - Special interactive communications with device
- **Negotiation**
  - Iteratively send a file back and forth
- **Constraints**
  - Know all rules beforehand to make configuration easier

Automated Discovery

- **Use existing vendor validation tools**
  - Compiler, negotiation interface
- **Supply a specially crafted series of test cases aiming for constraint coverage (fuzzing)**
- **Interpret results against predicted to define observed device constraints**
- **How much can we discover?**
  - On a random device: 50-60%
  - With a glossy sheet: 85-90%
- **When do we use this?**
  - Continuous process

How discovery works

- Start with generic ruleset defining a typical device in your target language
- Tests generated aiming for coverage of rules and boundary cases
- Either use results directly or iterate and start specializing ruleset for target device

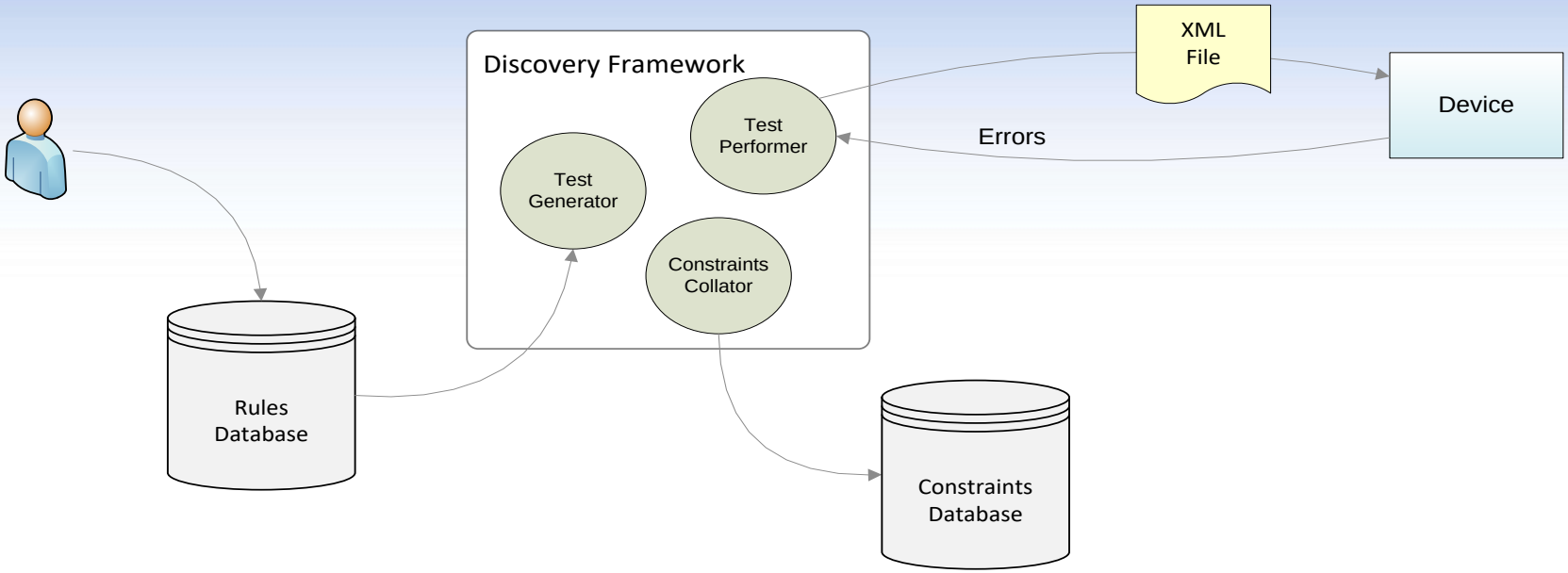
Automated Validation

- **Works broadly the same as discovery**
  - Make a series of test cases aiming for constraint coverage
- **Instead of trying to find all of the rules, try to test all of the rules**
- **Tests can be much more targeted because the constraint space is known beforehand**
  - For example, can set bounds at real values instead of bracketing

How validation works

- **Define rules matching device's true capabilities**
  - Can use discovery to aid in this process
- **Tests generated aiming for coverage of rules and boundary cases**
- **Compare actual results to expected and address issues**

Architecture



Sample Types of Rules

Rule Name	Definition	Testing Pattern
Finite	Field of numeric type which accepts a finite amount of values	Iterate through possibilities
Continuous	Field of numeric type with an upper and lower bound	Equally spaced points across the field, and outliers directly outside either bound
Forbidden	A field which cannot exist	Add and remove the disallowed field
Dependent	Values of one field change based on the values of one or more others	Cartesian combination of field values

Validation Results

- Summary of test cases and corresponding compiler/negotiation outputs

- Expected and actual test results
  - Expected to succeed and succeeds
  - Expected to fail and fails

  - Expected to succeed and fails
  - Expected to fail and succeeds

Example Validation Workflow

1. Rules checking device constraints are developed
2. Device behaves as expected in all tests
3. Firmware changes
4. Tests are re-run, noting any unexpected behavior
5. Any errors can be fixed or noted, and any changes to underlying constraints can be fixed

Advantages

- **Saves human time in running regression tests**
- **Tests using the input grammar (black box approach)**
  - Doesn't rely on knowledge of internal workings
- **Can be used with normal test suites on release of new firmware or hardware changes**

Conclusion

- **We tested in the lab on devices we already knew constraints for (thermocouple modules)**
  - Device 1: Verified 85% in a minute
  - Device 2: Verified 90% in two hours

- **Promising start to achieving automated verification of test instrumentation across the spectrum of devices**

Austin Whittington

awhittington@swri.org

+1.210.522.2847