Automatic Data Generation for Web Application Validation.

Shoichiro Fujiwara,
Software Innovation Laboratory,
Fujitsu Laboratories.

Outline

1. Background and motivation for automatic web application validation
2. SWEEP: a validation tool for server-side Java applications
3. Driver/stub data generation for SWEEP
4. Concluding remarks

Target Domain & Phase

- Domain: Web applications (server-side Java)
- Phase: Integration or system test

Web App Validation

- Checking whether application behaviors around each event fulfill the specifications.

Motivation and Approach toward Automatic Validation

Outline

1. Background and motivation for automatic web application validation
2. SWEEP: a validation tool for server-side Java applications
3. Driver/stub data generation for SWEEP
4. Concluding remarks
Validation Steps

1. Rewrite specifications into properties.
2. Generate data for the target application to drive.
3. Construct a validation environment.
4. Validate the target.

Rewriting Specifications into Properties

<table>
<thead>
<tr>
<th>Specifications</th>
<th>1. Rewrite manually</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Validator</th>
<th>Target application</th>
<th>Test result</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Data generator</th>
<th>Data to drive target applications</th>
</tr>
</thead>
</table>

Generate Data for the Target Application

- Extract constraints from design information in addition to properties.
  - Schemas of DB & frontends, ER designs like cardinality
  - Solve the constraints to generate frontend/DB data.

Validation Environment Construction

- Utilize framework, design and properties.
  - Web application in actual use

Java PathFinder

- A model checker for Java byte code
  - Developed by NASA Ames Research Center
  - Target programs run on the VM (JPF VM)

Properties for "Search"

- Menu
  - New product
    - Search
      - (Category=Food)
  - Product regmt
  - New product
    - Registration (code = 0000001)

Properties for "Registration"

- Menu
  - New product
    - Registration (code = 000001)
  - Product regmt
  - Search
    - (Category=Food)
Validation Steps Again

1. Rewrite specifications into properties.
2. Generate data for a frontend driver and a DB stub.
3. Construct a validation environment.
4. Validate the target.

Outline

1. Background and motivation for automatic web application validation
2. SWEEP: a validation tool for server-side Java applications
3. Driver/stub data generation for SWEEP
4. Concluding remarks

Data: Key for "Good" Validation

Need a corresponding data set for each property.
Data decide whether properties are checked or not.

Data Constraints

What we need to consider when generating data.

- Pre-condition of properties
- Invariant properties
  - Considered to be a kind of pre-conditions
- Type constraints
  - String/integer, length, allowed characters, etc.
- Table constraints
  - Foreign key relations, cardinality, uniqueness, etc.

Data Constraint Example in Design
Automatic Data Generation Architecture

- Use an SMT (Satisfiability Modulo Theories) solver.
  - SMT: satisfiability problem of some first order formulae
  - SMT solvers: Yices (SRI), CVC3 (NYU and U Iowa), Z3 (Microsoft Research), etc.

Problems in Data Constraint Reduction

- String encoding
  - The tool must decide many strings in a practical time.
    - A Web application generally handles much string data.

- Reduction of table size constraints
  - Constraints on the number of records are common and generally undecidable.
    - Cardinality, uniqueness, relation between tables and frontend

String Encoding (1)

- Trade off between abstraction level and performance
  - Treat every string as an array
  - Express every string with a representative integer.

String Encoding (2)

- Hybrid approach
  - Strings free from complex constraints like substring expressed with representative integer variables.
  - The others expressed as (numeric) arrays.
  - Analyze transitive closure of binary relations from data constraints to decide expressions.

Structured Constraint Reduction

- Bound table sizes
  - Reduce a constraint so that the original one holds iff the reduced one does under the bounds.

Convert Solutions into Driver/Stub Data

- Assign string values to representative integer values preserving order relations.
  - We can give strings to a transitive closure without taking care of the other closures.
**Trial with a Real Project Asset**

- **What we did**
  - Data generation for 23 properties, which specify the web application behavior around an event.
  - 13 normal cases and 10 error cases
  - Magnitude of data: 10 DB tables, half of which are master tables
- **Result**
  - The application ran successfully with the generated data.
  - Each property was evaluated at least once during validation.
  - It took about 5 minutes to generate data from each property.

**Outline**

- 1. Background and motivation for automatic web application validation
- 2. SWEEP: a validation tool for server-side Java applications
- 3. Driver/stub data generation for SWEEP
- 4. Concluding remarks

---

**Concluding Remarks**

- Validating web application by SWEEP
  - Rewrite specifications into properties.
  - Stub/driver data generation from the properties.
  - Validation environment with stubs/drivers.
  - Automatic validation based on JPF.
- Automatic stub/driver data generation
  - Extract data constraints from the properties and other design information.
  - Reduce the constraints into an SMT instance.
  - Solve the instance with an SMT solver.
  - Convert the solution into the stub/driver data.

---

**FUJITSU**

shaping tomorrow with you