Popular Delusions, Crowds, and the Coming Deluge: end of the Oracle?

Robert V. Binder

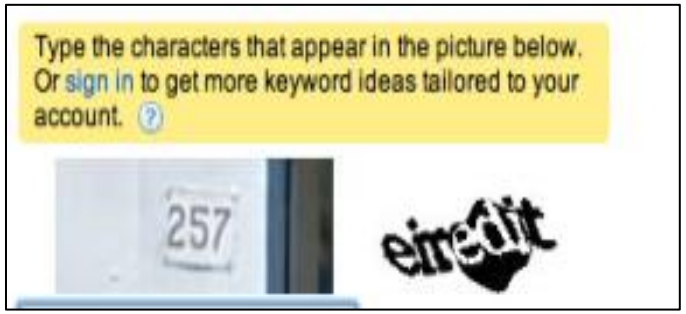
The 20th CREST Open Workshop
The Oracle Problem for Automated Software Testing
University College of London
May 21, 2012

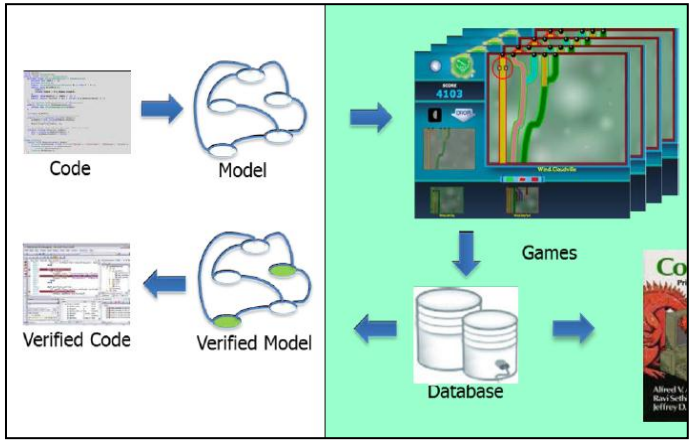
Overview

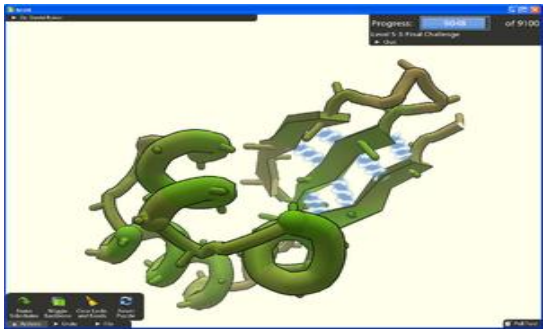
- Pragmatic Innovations
- Oracle Taxonomy
- Characterization
- Challenges

Crowd Sourced Evaluation

- Google presents street sign images in ReCAPTCHA
- Crowd Sourced Formal Verification- DARPA
  - Correctness proof as web game
  - Players devise strategies to win
- Bio: Foldit, Foldit@home, Phylo, Rosetta@home
  - “Top-ranked Foldit players can fold proteins better than a computer.”







Testing and its Discontents

- "Testing is Dead"
- Exploratory Testing
- Crowd Testing
  - MobTest
  - UTest
  - Mob4Hire

http://www.youtube.com/watch?v=X1jWe5rOu3g

mobtest

uTest
Where the best test

MOB4HIRE
The Leader in Brand Mobile Optimization

"58,159 people (mobsters) have 33473 different mobile handsets on 439 carriers in 155 countries"

What is a Test Oracle?

Any strategy that can produce a verdict from an observation of a SUT in action.

John Collier's *Priestess of Delphi.* Oil, 1891

Survey of Test Oracles

- 600+ publications
- Many strategies
  - Mostly esoteric
  - Some pragmatic
- Hard to compare
- No basis for evaluation

Test Oracle Taxonomy

Predictive

- For *selected* test inputs, predict or constrain expected result
- Expect expected and actual same, for each *test input*

Imitative

- Develop one or more facsimile systems
- Submit *any* input to SUT *and* facsimile
- Expect all outputs equivalent

Reactive

- Define output criteria
- Submit *any* input to SUT
- Expect output criteria met

Judging

- Cultivate sense of appropriate
- Submit *any* input to SUT
- Decide if response is appropriate

Any strategy that can produce a verdict from an observation of an SUT in action

Predictive Test Oracles

Strategy	Tactics	
Special Values	Sensitive Points Rejection Response	
Solved Example	Reference Table Lookup	
Design by test	Test First Design	
I-O Invariants	Range Input-output balancing Behavior	
Metamorphic Testing	Constant Step Reorder	Permute Add, drop
Regression Test	Reference Testing Capture/Replay	
Specification-based	Abstract Concrete	I-O Grammar Checker Transition system trace

Predictive Test Oracles

I-O Invariants

For specific input, expected output is within a range or a member of a set; "Sanity Check"

Range	**assert(paymentAmount < 1000000);**
Input-output balancing	**assert(total.A-in == total.B-out + total.C-out);** **assert(countNew() == countOld()+ adds - drops);**
Behavior	**assert(mode==landing && !wheels==up);**

Predictive Test Oracles

Metamorphic Testing

Output tuples are expected to meet certain properties

Constant Step	**g(x)-> 10, asssert(g(x+10)== 20);**
Reorder	**inputList=(rand(sortedList)); assert(sort(inputList)==sortedList);**
Permute	**assert(gcd(x,y) == gcd(y,x));**
Add, drop	**assert(f(a,b,c)+d==f(a,b,c,d));**

Imitative Test Oracles

Strategy	Tactic	
Neural Network	Machine Learning	
Reduced Implementation		
Executable Specification	Complied Abstraction I-O Grammar Checker	
Voting	Reference Implementation Parallel Testing	Stack Variation N-way Voting

Imitative Test Oracles

Executable Specification

An SUT specification is translated into an executable, which maps inputs to expected outputs.

Abstract	**An abstract notation (OCL, SDL, Z, VDM, …) is automatically translated to a program, that computes the output**
I-O Grammar	**For SUT specified in a BNF, use the BNF to define a reader of the output that can detect malformed output**

Imitative Test Oracles

Voting

Submit any input to the SUT and one or more facsimile systems, expect result of each is equivalent

Reference Implementation	**SUT and reference impl are black-box equivalent**
Parallel Testing	**SUT and parallel impl are black-box equivalent or nearly so**
Stack Variation	**Build SUT for different compilers and/or computers; use same input on all; compare all outputs**
N-way Voting	**Use one or more facsimile systems; broadcast same input to SUT and all facsimiles; compare all outputs**

Reactive Test Oracles

Strategy	Tactics	
Environment Monitor	Resource Utilization Timers	Abend
Output Invariants	No Change Range Behavior Format	Content Entity Relationships Parametric
Built-In Test	Assertions DBC - Built-in DBC - Pragmas	DBC - Sampling Application-specific
Parametric	Output Stream	Persistent Store
Trace Analysis	As Built	Additional
Algebraic	ADT API	SQL
Performance	Response Time Throughput	Reliability Availability

Reactive Test Oracles

Algebraic

Exploit externally observable algebraic relationships

```
assert(date.yesterday() == date.today - 1)
```

ADT	Run equivalent operation sequences, evaluate resultant state
API	Same as ADT, may require helper methods
SQL	Construct equivalent DBMS queries and updates; expect result tables and target tables identical;

Reactive Test Oracles

Trace Analysis

Parse available outputs; check conditions, relations, grammar

As Built	**Use only available app or system log files; check conditions and relations**
Instrumented	**Modify SUT to produce outputs of interest; check conditions and relations**

Judging Test Oracles

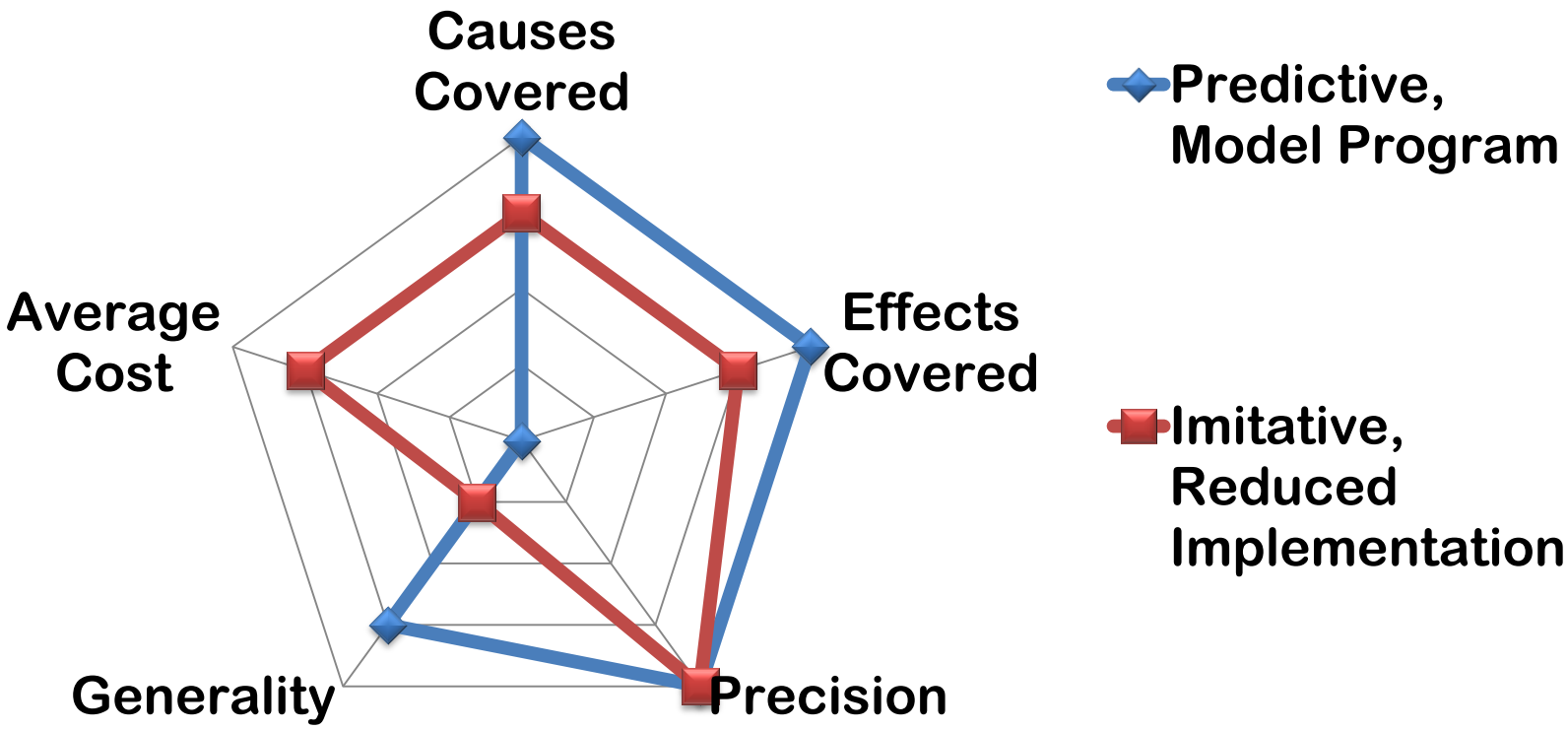
Exploratory Testing	The tester critiques the SUT while following an general interaction strategy
Ad hoc	The tester improvises interactions
Tour-based	The tester improvises interactions based a pre-defined strategy
FDA Validation Testing	The SUT is used in situ to see how well it supports realistic tasks and workflow
Beta Testing	Users interact with SUT according to idiosyncratic interest
Crowd Testing	Users selected for operational environments, modes, and configurations;
Usability Testing	Evaluate HCI for external standards
Quantitative	Compare measurements of user physiological responses to structured and unstructured interaction with the SUT
Qualitative	Study subjective like/dislike

Oracle Characterization

- What attributes or properties are useful to characterize or compare oracle types?
- Questions must be germane and answerable for all types

- Cause Coverage
- Effect Coverage
- Precision
- Point of Control/Observation
- Test Strategies supported
- Average Cost per verdict
- Antecedent
- Comparator

Example Comparison



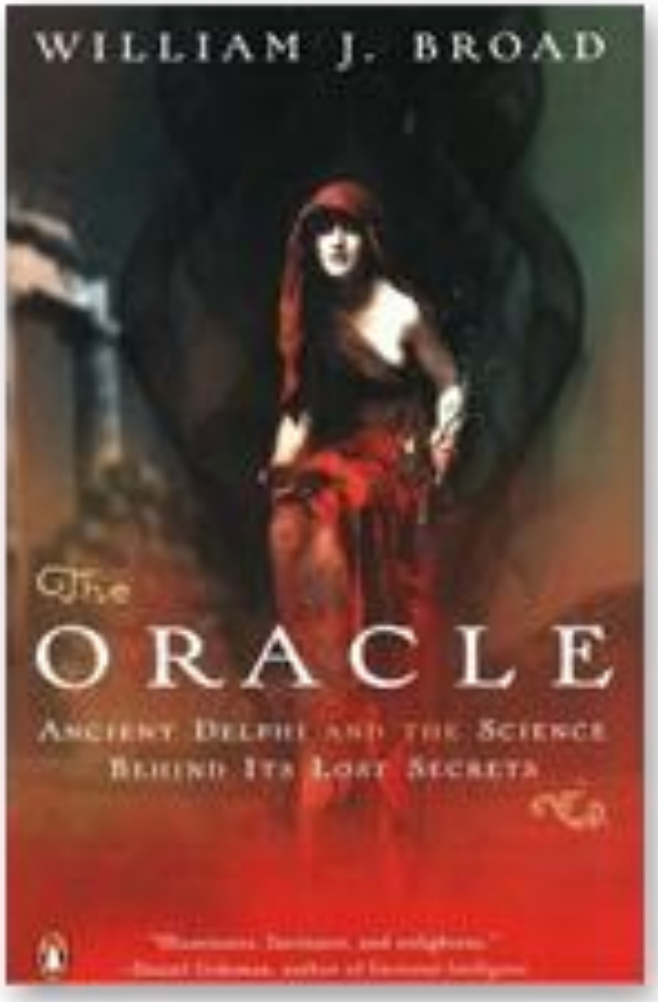
Challenges

- Scalability
- Novel interfaces
- Can judging be reduced to an expert system?
- Effective integration of automated Oracles with Crowds?

High and Sly

- Prophecy not free
- “The” oracle was many individuals
- Indeterminate questions got ambiguous answers
- Opportunistic use of natural resources (ethylene)

Recommended Reading



William J. Broad
The Oracle: Ancient Delphi and The Science Behind Its Lost Secrets (2006)