

Pictures at QRS 2015

Day 2 (August 4)











Marriott
VANCOUVER AIRPORT



HRML: a hybrid relational modelling
language

He Jifeng







Hybrid Systems

- Systems are composed by continuous physical component and discrete control component
- The system state evolves over time according to its discrete and continuous dynamics.
 - For discrete dynamics, it changes state instantaneously and discontinuously.
 - During continuous transitions, its state is a continuous function of continuous time and varies according to a differential equation.
- Modelers mix discrete time reactive systems with continuous time ones.







lenovo

Marriott
VANCOUVER AIRPORT



lenovo

Marriott



Parallel by merge

Let P and Q be hybrid relations with $x' \in \text{outa}P \cap \text{outa}Q$. We define their parallel composition equipped with the merge mechanism M , denoted by $P \parallel_M Q$, as follows:

$$P \parallel_M Q =_g \exists m, n : \text{Val} \bullet$$

$$(P[m/x'] \wedge Q[n/x'] \wedge (x' = (m \text{ op } n)))$$

$$\text{ina} =_g \text{ina}P \cup \text{ina}Q$$

$$\text{outa} =_g \text{outa}P \cup \text{outa}Q$$

$$\text{cona} =_g \text{cona}P \cup \text{cona}Q$$







Closure of healthy hybrid relations

Theorem

(1) $H(P) \cap H(Q) = H(P \cap Q)$

(2) $H(P) \triangleleft b \triangleright H(Q) = H(P \triangleleft b \triangleright Q)$

(3) $H(P); H(Q) = H(P; H(Q))$

(4) If P and Q lie in the complete lattice L , then
 $(P \parallel_M Q)$

where the merge mechanism

HRML: a hybrid relational modelling language

$AP ::= \text{skip} \mid \text{chaos} \mid \text{stop} \mid x := e \mid !s \mid \text{delay}(\delta)$

$EQ ::= R(v, i) \mid EQ \text{init } v_0 \mid EQ \parallel EQ$

$P ::= AP \mid P \sqcap P \mid P; P \mid P \triangleleft b(x) \triangleright P \mid P \parallel P \mid$

$EQ \text{until } g \mid \text{when}(G) \mid \mu X \bullet P(X)$

$\text{timer } c \bullet P \mid \text{signal } s \bullet P$

$g ::= \text{skip} \mid s \mid \text{test} \mid g \cdot g \mid g + g$

$\text{test} ::= \text{true} \mid v \geq e \mid v \leq e \mid \text{test} \wedge \text{test} \mid \text{test} \vee \text{test}$

$G ::= g \& P \mid G \mid G$

EXIT

Marriott
VANCOUVER AIRPORT



Marriott
VANCOUVER AIRPORT



Marriott
VANCOUVER AIRPORT

Adjust the sampling rate

The following theorem is used to reduce the sampling rate of the controller by estimating the change speed of physical state

Theorem Let $l < m < n$.

If $R \supseteq (0 < \dot{x} \leq \epsilon)$ and $\delta < (n - m)/\epsilon$, then

$(R \text{ init } \text{ until } \{v \geq n\})$

= signal $s, v \bullet \left(\begin{array}{l} (R \text{ init } \text{ until } \epsilon) \\ \text{delay}(\delta); \\ \text{when}(\{v \geq n\} \& \{s\}) \{ \{v \leq l\} \} \end{array} \right)$

EXIT

A man with glasses, wearing a dark jacket over a light-colored patterned shirt and a dark belt with a silver buckle, stands behind a podium. He is gesturing with his right hand. The podium features the Marriott logo and the text 'VANCOUVER AIRPORT'. A laptop with the 'lenovo' logo is on the podium. The background is a plain wall with vertical panels.

Marriott
VANCOUVER AIRPORT

lenovo

Jifeng He
East Coast Manager
Community



Marriott
VANCOUVER AIRPORT

The f
cont
Theo
If R
(Rin
- s

QRS 2015, Vancouver

Integrating Specification Animation with Specification-Based Program Testing and Inspection for Software Quality Assurance

Shaoying Liu

Department of Computer Science
Faculty of Computer and Information Sciences

 Hosei University, Japan

Email: slu@hosei.ac.jp

HP: <http://ios.k.hosei.ac.jp/~slu/>

This work is supported by JPS KAKENHI Grant Number 26430000



Overview

1. Challenges to Software Quality Assurance
2. Our Solution
3. Specification Animation
4. Specification-Based Program Testing and Inspection
5. Open Problems
6. Conclusions
7. Future Work



1. Challenges to Software Quality Assurance

- The scale and complexity of software development projects
 - The scale of documentation
 - The complexity of documentation
 - The complexity of situations (e.g. requirements changing, people moving, client complaining, manager worrying, and developer fighting)





1. Challenges to Software Quality Assurance

- The scale and complexity of software development projects
- The complexity of documentation
- The complexity of situations (e.g., requirements changing, people moving, client complaining, manager worrying, and developer fighting)

1. Challenges to Software Quality Assurance

- The scale and complexity of software development projects
 - The scale of documentation
 - The complexity of documentation
- The complexity of situations (e.g., requirements changing, people moving, client complaining, manager worrying, and developer fighting)



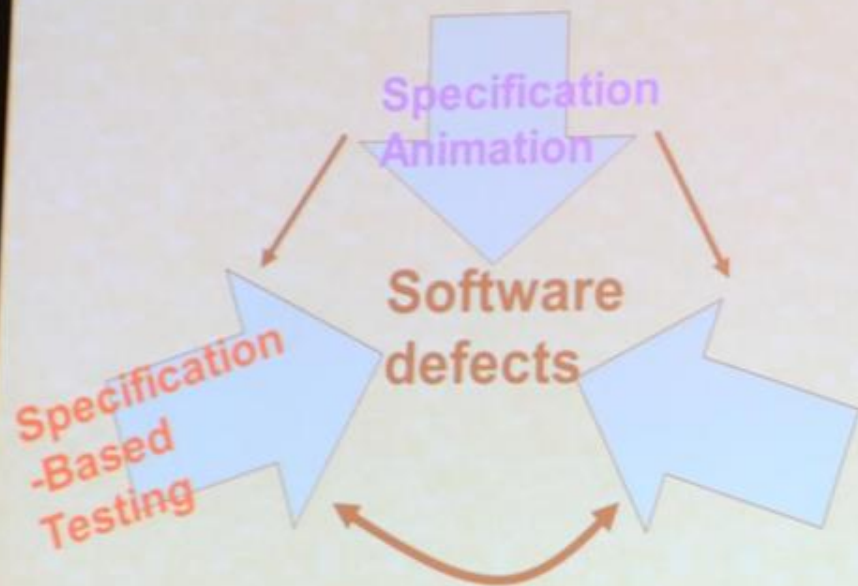


➤ Deficiencies of techniques available for use

- ❑ **Formal proof of correctness:** ideal but tedious, ineffective (for faulty programs), requiring skills (loop invariants), error-prone, and time consuming.
- ❑ **Model checking:** needs **appropriate abstraction** of a real system to a FSM model and faces the **state explosion** problem (two state space explosions for software: initial state space and program state space).
- ❑ **Testing:** can tell the existence of bugs, but cannot tell their absence in general. **Nevertheless, it is a common practice in industry.**
- ❑ **Review and inspection:** easy to carry out, but heavily depend on human judgment, ability, and experience



2. Our Solution



3. Specification Animation

Specification animation is a technique for dynamic and visualized demonstration of the system behaviors defined in the specification.

Three expected effects: improving understanding of requirements or designs, strengthening communication, and verifying/validating specifications.

Specification
(textural,
graphical)

Specification
Animation

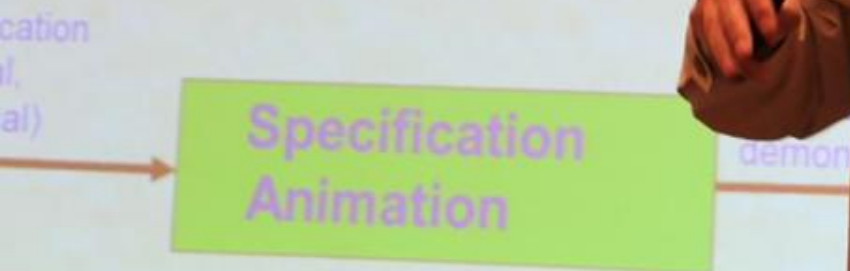
Dynamic
visualized
demonstration



Specification Animation

Specification animation is a technique for dynamic and visualized demonstration of the system behaviors defined in the specification.

Expected effects: improving understanding of requirements or designs, strengthening communication, and verifying/validating specifications.





Overview

1. Challenges to Software Quality Assurance
2. Our Solution
3. Specification Animation
4. Specification-Based Program Testing and Inspection
5. Open Problems
6. Conclusions
7. Future Work









EXIT

Advanc
for

Re

the organ
to bring re

Join the

International
clinical info
relevant pub
contributing
networking
world



uni.lu
SNT
University of Luxembourg, Microsoft Research, Tencent

Potential Component Leaks in Android Apps

An Investigation into a new Feature Set for Malware Detection

Li Li
SnT, University of Luxembourg

Joint work with

- Kevin Allix, Daoyuan Li, Tegawende Bressande, Jacques Klein, SnT
- Alexandre Bartel, TU Darmstadt

uni.lu SNT Technische Universität Darmstadt





EXIT



Android

Leaks

Android browser flaw found to le:



The vulnerability enables a hacker to run JavaScript from a website to steal data from web pages open in other browser tabs.

Dozens of Popular Android Apps Leak Sensitive User Data

A group of researchers from the [University of Luxembourg](#) have uncovered vulnerabilities in several popular Android apps, including Instagram, Vine, iXCloud and more. The bugs could expose the sensitive information of some 300 million users that have installed the affected applications on their [Android mobile devices](#).



SnT, University of Luxembourg

EXIT



Bernhard
Garn
SEA Research



Bernhard
Garn
GSA Research

Scen

cure
Research.org


Ou

Em

veo

VANC





An Effective Approach to Continuous User Authentication for Touch Screen Smart Devices

Arun Balaji Buduru and Stephen S. Yau
Information Assurance Center
School of Computing, Informatics, and Decision Systems Eng.
Arizona State University

ASU Information Assurance Center
ARIZONA STATE UNIVERSITY

School of ICDSE
100 E. Taylor School of Engineering



Current State of the Art



- Multi-modal continuous authentication techniques use factors, such as fingerprints, iris and face recognition to continuously authenticate the legitimate user
 - Muncaster and Turk presented an approach to performing continuous, score-level multi-modal authentication based on a weighted sum of scores from each modality
 - A continuous multi-modal biometrics system using a hidden Markov model (HMM) was developed by Sim, et al
 - Shi, et el, used multimodal inputs, such as voice, location, multi-touch and motion, to perform continuous user authentication
- These techniques are inherently infeasible due to the requirements of additional hardware in low-cost touch-screen smart devices, and frequent conscious user interactions



rt (cont.)

naïve Bayes
accurate
gorithms, and
not user

ed as a trusted

School of ECE
and Computer Science of Engineering

EXIT



MAPPING MODEL COVERAGE TO CODE COVERAGE

- ❖ A CEC contains Simulink blocks and conditionals, and
- ❖ A Simulink CEC is a 4-tuple $E = \{D, G, B, V\}$,
 - D is a finite set of decisions determined by the Predicate Blocks of each conditional subsystem;
 - G is a finite set of decisions and conditions formed by functional blocks;
 - B is the set of decisions and conditions that are formed by functional blocks;
 - V is the set of values associated with the conditions and decisions



SVM Score distribution

Training Set Normal Model - SVM Score



Training Set Malicious Model - SVM Score



EXIT



Detection Logic

algorithm

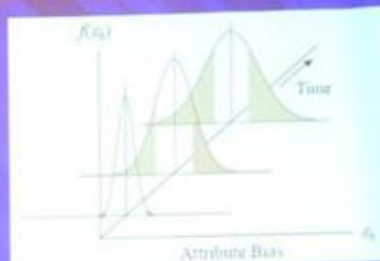
```
anomaly = normal_model_score  
malicious = malicious_model_score  
if anomaly < n, && malicious < m,  
    predict benign  
else if anomaly > n, && malicious > m,  
    predict malicious  
else  
    predict ambiguous
```



1. Introduction

Reliability of the real time system is depend on the distribution of uncertainty (attribute bias).

Uncertainty analysis is used to investigate the reliability of RTS with uncertainty variables for decision-making program.



EXIT















































XML Bomb/Billion Laughs (BIL)

- DoS attack on XML parser using well-formed XML

- Exponential entity expansion



- **Impact**

- Higher memory & CPU Consumption
- DoS on parsing system

```
<?xml version="1.0"?>
<!DOCTYPE xmlBOMB [
  <!ENTITY a "Random Text">
  <!ENTITY b "&a; &a; &a; &a; &a; &a;">
  <!ENTITY c "&b; &b; &b; &b; &b; &b;">
  <!ENTITY d "&c; &c; &c; &c; &c; &c;">
]>
<xmlBOMB>&d;</xmlBOMB>
```





Sadeeq Jan
University of
Luxembourg

Background

Dynamic Atomicity Violation Detectors

- ◆ Examples
 - CTrigger [ASPLOS 2009, TSE 2012]
 - AtomFuzzer [FSE 2008]
 - AssetFuzzer [ICSE 2010]
 - Maple [OOPSLA 2012]
- ◆ Two-phase strategy is often used
 - Prediction phase
 - Confirmation phase



My Research

- Building reliable and secure software applications
- Compiler & runtime techniques for error resilience
 - Partitioning data for differential resilience [ASPLOS'11]
 - Error detection in different programs [DSN'12][DSN'13]
 - Fault Injection techniques and tools [DSN'14][ISPASS'14]
- This tutorial
 - Reliability of modern web applications (Part 1)
 - Tools for building robust web applications (Part 2)

© 2014 ACM. All rights reserved. 11/14

EXIT



KIT



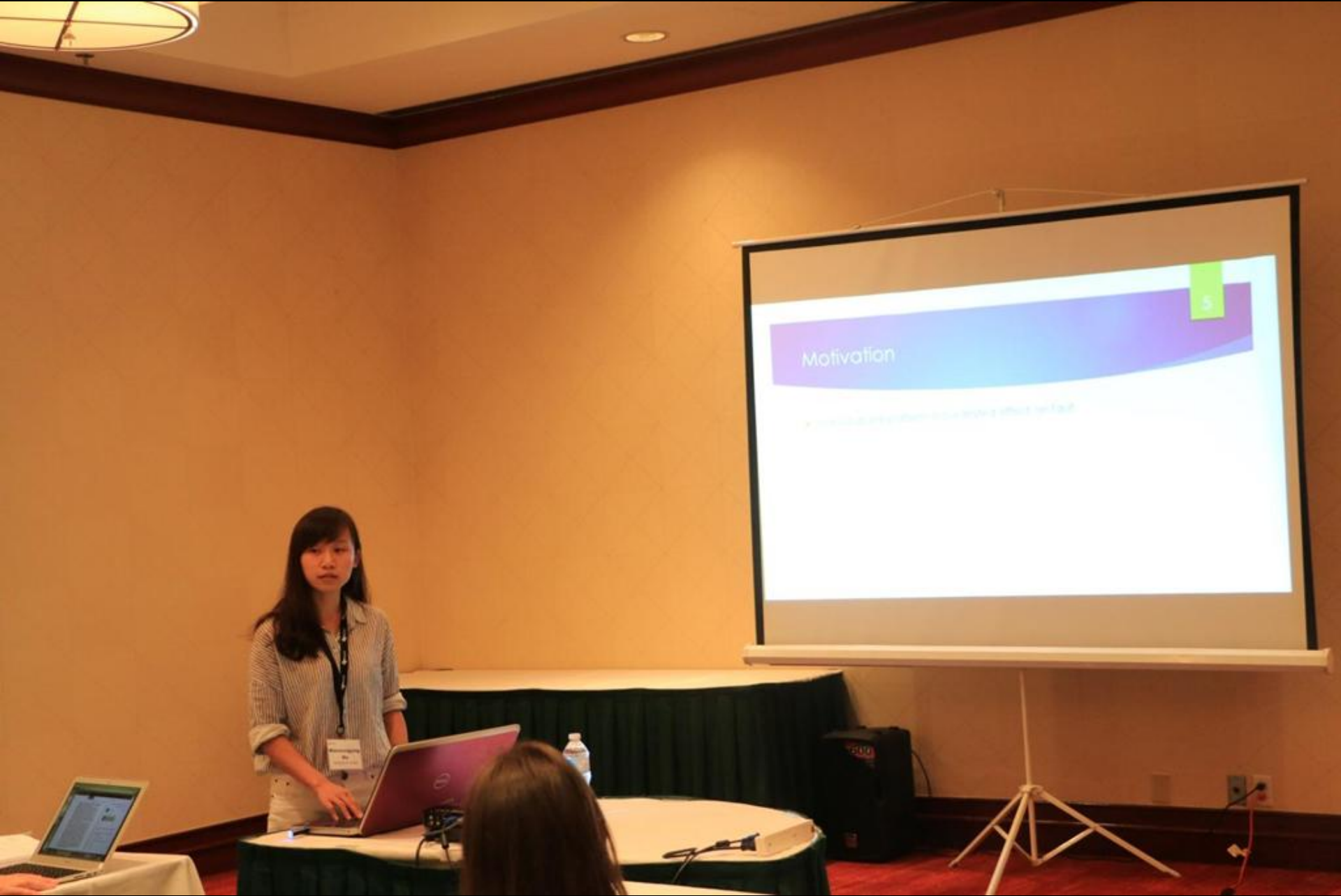
Karthik
Pattabiraman





Motivation

...and its effect on fault





Running Example

Cell	G4	H4	J2	J3
E7		•		•
E8				
G2	•	•	•	•
G3	•	•		•
G4	•			
H2			•	
H3		•		•
H4		•		
J2			•	
J3				•
Classification d	true	false	true	false

8

Stacy Heller and Franz Hees, "Fair Allocation in the Light of Quality Cost Input"
<http://www.mathworks.com/matlabcentral>



Running Example

Cell	G4	H4	J2	J3	SC
E7		•		•	1.000
E8					0
G2	•	•	•	•	0.707
G3	•	•		•	0.816
G4	•				0
H2		•	•		0.500
H3		•		•	1.000
H4		•			0.707
J2			•		0
J3				•	0.707
Classification d	true	false	true	false	

8

Bojgi Keller and Franz Dohmen, "Fault localization in the light of faulty user input," <http://www.madeinusa.de/bojgi/>









Highly-impactful vs. Other bugs

9

Highly-impactful bugs



reporter experience



Other bugs



comment length



closed %



EXIT



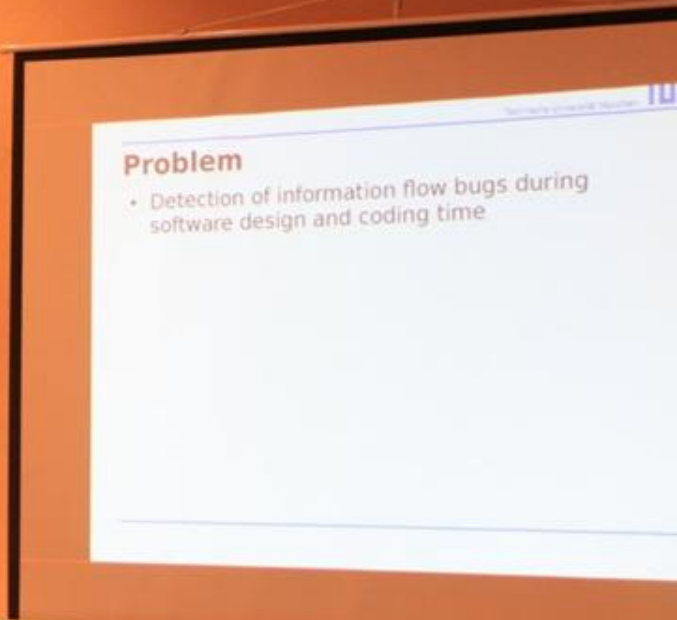












Problem

- Detection of information flow bugs during software design and coding time









Boolean Differentiation for Formalizing Myers' Cause-Effect Graph Testing Technique



Tolga Ayav, Fevzi Belli

Izmir Institute of Technology
Dept. of Computer Engineering

5th IEEE International
Workshop on Model-Based
Verification & Validation
Vancouver, August 3-5, 2015

Cause-Effect Graphs

- Cause-Effect Graphs (CEG) assists deriving tests from a given specification given in natural language.
- CEG is constructed by an experienced test engineer.
- Test cases are derived from the graph.
- Myriad case generation method from CEG is intuition-based.



Introduction



◆ In multilevel hybrid cloud, most of the access control policies often deal with user's single access behavior only

- Multiple access between malicious users may make data flow violating security policy
- malicious behavior must be prevented in the high cloud systems

Evaluations

- Both *Maven* and our technique have the same fault coverage
- In terms of "number of detected faults per test", ours shows 2.2 times better performance.

Fault Classes	Total Faults	Maven		Ours	
		Fault Coverage (Average)	Faults Detected (Average)	Fault Coverage (Average)	Faults Detected (Average)
Class 1	100	100%	100	100%	100
Class 2	200	100%	200	100%	200
Class 3	300	100%	300	100%	300
Class 4	400	100%	400	100%	400
Class 5	500	100%	500	100%	500
Class 6	600	100%	600	100%	600
Class 7	700	100%	700	100%	700
Class 8	800	100%	800	100%	800
Class 9	900	100%	900	100%	900
Class 10	1000	100%	1000	100%	1000
Total	5000	100%	5000	100%	5000



EXIT



EXIT







































































T. Y. Chen
Swinburne University
of Technology





















EXIT



QRS 2016
The 2016 IEEE International Conference on
Software Quality, Reliability & Security
Vienna, Austria, August 1-3, 2016
<http://paris.utdallas.edu/qrs16>

IEEE
TU
secure



QRS 2016

The 2016 IEEE International Conference on
Software Quality, Reliability & Security
Vienna, Austria, August 1-3, 2016
<http://paris.lodron.edu/qrs16>

Security & Safety



secure
FOR COMMUNITY



QRS 2016

The 2016 IEEE International Conference on
Software Quality, Reliability & Security
Vancouver, August 1-3, 2016
<http://www.dallas.edu/qrs16>

IEEE

Security & Reliability

TU

secure



QRS 2016

The 2016 IEEE International
Software Quality, Reliability,
and Security Conference

Vienna, Austria, August 14-18, 2016

<http://paris.utdallas.edu/qrs2016>

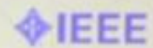


QRS 2016

The 2016 IEEE International Conference on
Software Quality, Reliability & Security

Vienna, Austria - 2-3, 2016

<http://paris.uibk.ac.at/qrs16>



Reliability Society



secure
sba-research.org



QRS 2016

The 2016 IEEE International Conference on
Software Quality, Reliability & Security
Vienna, Austria, August 1-3, 2016
<http://paris.utdallas.edu/>



IEEE
Institute of Electrical and Electronics Engineers



secure
technology







Software Quality, Reliability & Security
Vienna, Austria August 1-3, 2016
<http://paris.lip6.fr/~mhenkel/edu/qrs16>



Software Quality, Reliability & Security
Vienna, Austria August 1-3, 2016
<http://paris.lip6.fr/~mhenkel/edu/qrs16>





IEEE
Reliability Society
TU
secure



QRS 2016

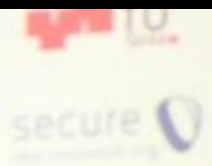
The 2016 IEEE International Conference on Software Quality
Vienna, Austria, Austria
<http://paris.utd.edu>



QRS 2016

The 2016 IEEE International Conference on
Software Quality, Reliability & Security
Vienna, Austria, August 1-3, 2016

<http://www.qrs2016.org>





QRS 2016

The 2016 IEEE International Conference on
Software Quality, Reliability & Security
Vienna, Austria, August 1-5, 2016
<http://www.qrs2016.org>

IEEE










Eric Wong
University of Texas at
Dallas

William Li
University of Texas at
Dallas

A portrait of Prof. Dr. Franz Wotawa, a man with glasses and a goatee, wearing a blue shirt. He is positioned in front of a white bookshelf filled with books. The image is presented as a video frame with a dark border.

Prof. Dr. Franz Wotawa / TU Graz
General Chair QRS 2016, Vienna, Austria











Ken Wong
Executive Director
Singapore

Min Xie
Executive Director
Singapore

