

Testing Techniques based on relational symbolic execution will be used in the industry to improve the reliability of data-concerned software!

WHY CAN IT (NOT) HAPPEN?

(Michael MARCOZZI)

The market is there!

It will depend, on practical needs and scalability, on those needs!

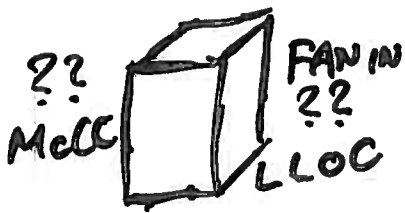
WHAT ~~OTHER~~ ARE THE MOST USEFUL DATA SOURCES
USED IN YOUR STUDIES? BUT TRACKING/MAILING LISTS/SOCIAL'S...?

CODE METROPOLIS

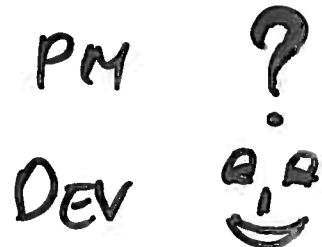
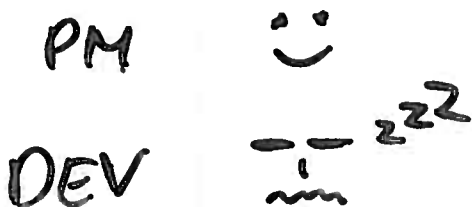
① COULD VISUALISATION UNDERSTAND INTUITIVELY?

② COULD CITY METAPHOR PROVIDE AN USEFULL VIEWPOINT AND MAINTAIN MOTIVATION?

① TOO MANY LOW LEVEL METRICS $\xleftrightarrow{???$ UNDERSTANDING INTUITIVELY? OF SYSTEM



② USEFULL $\xleftrightarrow{???$ BOTH? MOTIVATION



BAL R14, LAB1

1234: RES: ---

LAB1 - - - ST R14, SAVE

L R14, SAVE
BR R14
B B4(R14)

r14 := 1234;

call LAB1

LAB1 ()

~~dispatch~~

call 1234

LAB1 ;

dest := r14;

call dispatch

if dest = --

then call

Martin Ward

Software Migrations Ltd

De Montfort University

martin@glc.org.uk

<http://www.glc.org.uk>



Source code should be
hard to write

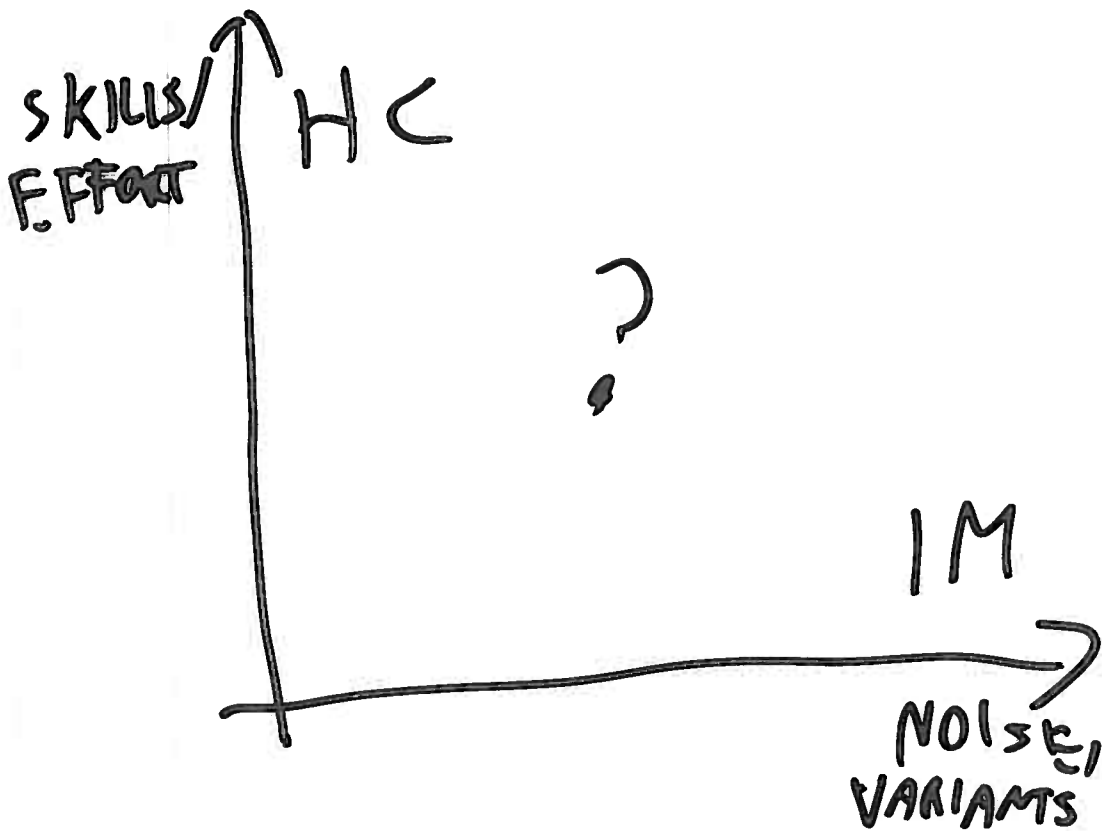
[and easy to read.]

HAND-CODED (HC)

REGEX

VS.

INFERRED MODEL (IM)



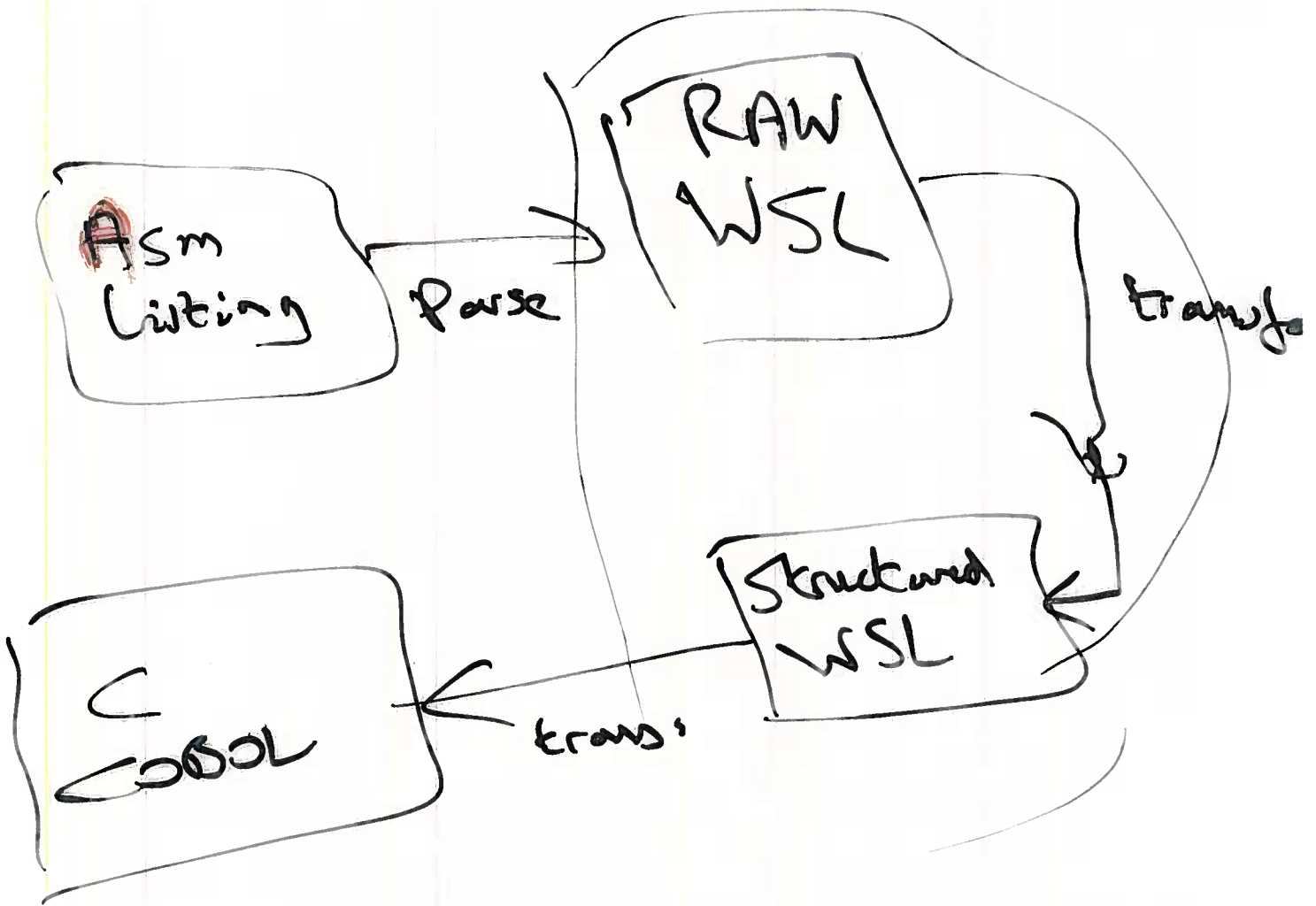
M

Controversial Statements

- 1) Fully automatic MLR is not feasible ⁱⁿ generally.
- 2) It's better to find an improvement than searching hard to find an overall solution.

Most of the developers do not run
static analysis tools within their
IDE and fix the problems before
code review. Either we need a
righter integration (like
Review Bot + fix-it) with code review

or
?



```
const char * s;
```

```
printf("s", s);
```

• CIP IS MORE MAINTAINABLE THAN ASPECTS!
TANGLING! ← SCATTERING

• SHOULD LINKERS REALLY BE LINKED?

* The Optimization of Code to cater to design on 'large code bases' was the biggest challenge. MQ optimization is an NP hard problem.

Research shows all current techniques at best get close only 60% to the actual architecture component design.

~~50/20/20/20/20~~

* All distributions are skewed,
So how do we aggregate them —
geometric means, inequality indices,
quality model aggregation techniques,
Something else?

Past

- Slicing theory is sound

Present

- No new research
- Slicing has not been commercial ^{success} ~~success~~
- Doesn't scale

Future

- Slices in support of other analysis
- Generalization of slices

Questioning the programmer

Integration & Verification of
configurable software

Interpretation of empirical data

Application and Interpretation
to apply techniques in the
industry.

Past

Evaluating SCAM tools

- * Not (convincing arguments)
 - * Compared tools
 - * Seeding errors/clones
- ↑
incomparable?

Present

- * Manual construction of benchmarks
- * Tools to compare tools

Completeness versus Accuracy

Future

- * Synthetic data generation
- * Requirements from industry (big data)
- * User studies
- * Industrial studies

Missing Link:

Academia & Industry

You are not a real researcher! ↔ You work in ivory towers!

- Problems addressed by researchers are not interesting and cannot be adopted by industry.
- Conferences are either for industry or (xor) for academia

(SOUND)
POINTER ANALYSIS
(FOR BUGFINDING)
REQUIRES
ASSUMPTIONS

Working on a sound
analyzer means not
having to choose between
finding all bugs in Juliet
and finding all bugs in
the wild.

What other fitness functions
would be useful

NOTES

2002 success: .NET & Apple's LLVM feedback → success!!
(compilers should make code analysis easier)

"sword" people are still stinging

Myths: still just as mythical

CONTROVERSIAL STATEMENTS

1. WE ARE TERRIBLE RESEARCHERS,
WE DON'T SOLVE ANY IMPORTANT QUESTIONS
2. WE DON'T NEED TO SEARCH FOR NEW PROBLEMS TO SOLVE.

* Convince compiler manufacturer
to make code analysis easier

BAD NEWS:

- GCC has been exploiting
undefined behavior more
and more aggressively
- Floating-point behavior when
compiling towards the x87
has been made deterministic

GOOD NEWS

Should language
standards have UB at
all?

Dependency cycle/fentrol objects: Tosin Daniel Oyedotan

Conversational/Future

1. Can refactoring degenerate components in dependency cycle reduce defect-proneness?
2. Complexities of components in dependency cycle
3. depend-on-cycle in relation to in-cycle components.

Fork Sim

Jeff Svajlenko

- ① How can we make synthetic test data more realistic?
- ② How to configure ForkSim?
- ③ Synthetic vs. Real data for performance evaluation.
- ④ Other domains we can apply this technique to?

1) Do ~~the~~ lexical means are enough!

for evaluating complexity?

~~identification~~

~~features~~

and identification of features?

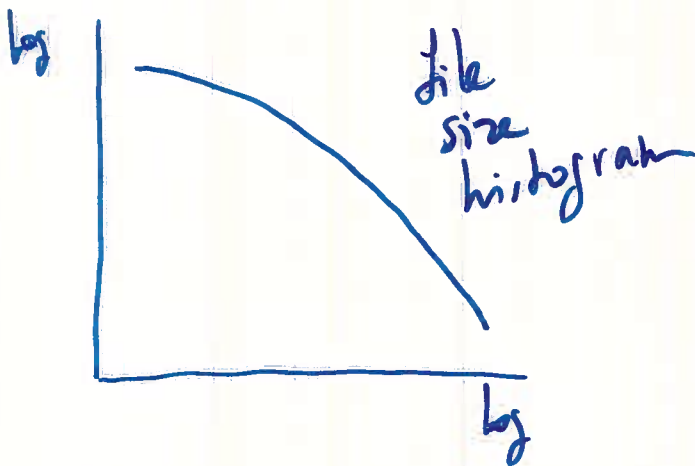
1) one case study → deep analysis,
semantic ~~and~~ consideration,
Generalize?

many case studies → automatic analysis,
miss on the details!

M.V. Tests

⇒ distributions of code features over files are often useless: the size of the files dominate the observations (long tails).

How do we model/measure statistically & what to expect when opening/looking at "arbitrarily/randomly" selected code?



JIT for FPGA?

Debugger

- 1) Multithreaded programs ?
deterministic replay ?
- 2) Execution slice ?
only execute slice ?
- 3) GUI for scalability

Larus & Ball, they didn't answer the problem
already?

optimal placement for Tracy.

90% of SE research focus on rigid properties,^{assumptions}

What about plastic properties?

Diversity, opportunistic reuse and correctness, ...

GASR

LANGUAGE RESEARCHERS



"COOL" LANGUAGE CONSTRUCTS



PEOPLE USING THIS



ANALYSIS? TOOLS? DESIGN? ...?

DEPENDENCE CLUSTERS

- Árpád Beszides

Controversial questions:

- ① What correspondence is among different d.d. definitions?
(SEA-based, slice-based, same set, dique-based, ...)
- ② Are dependence clusters good or bad?
- ③ Are linchpins good or bad?
3b) How often are they "dependence pollution"?