

## Mark Harman:

Q: why not use static analysis?

A: Sure. SBSE is complementary not conflicting

Q: When ca I sue SSBSE?

A: When the space is too large for static analysis.

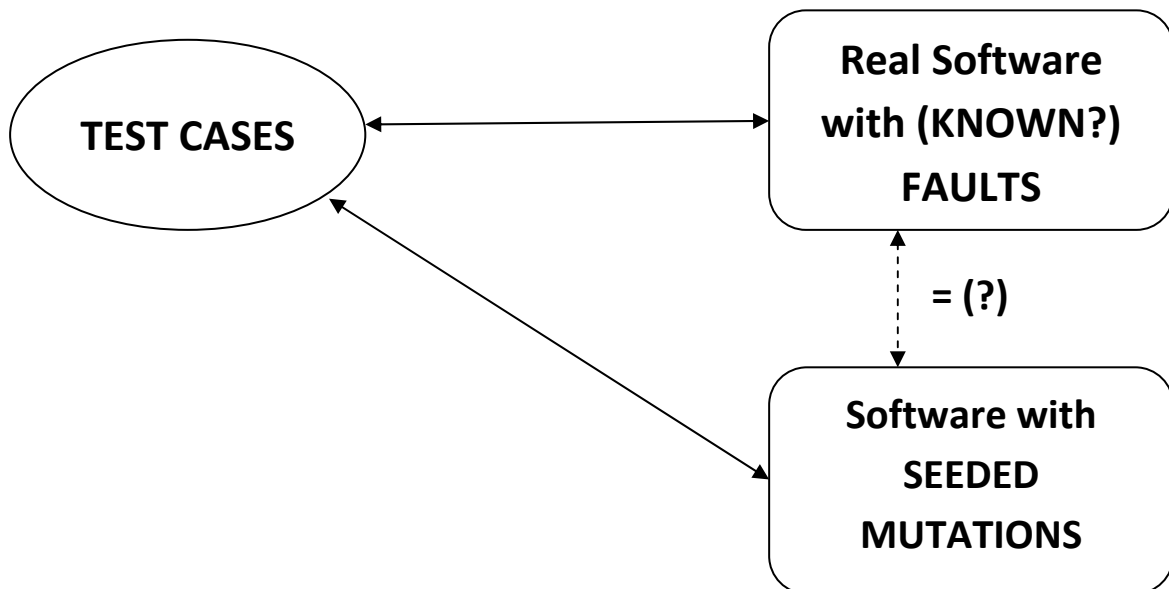
Q: When should i sue static analysis?

A: If a precise solution can be found in reasonable time (or good enough)

Model checking to prove absence of deadlock is “exhaustive search on an optimized search space that is sufficiently small”

- When it's too large we can't use exhaustive... but the space is still interesting & important.

## Simon Poulding



## Mark Harman

SBST: Search Based Software Testing

... a plea for Multi Objective SBST

Testing is not just about coverage

In fact it is seldom about a single objective!

Multiple objectives:

1. Coverage
  2. Time to execute
  3. Length of trace
  4. Oracle cost
  5. Exercise known bugs
- .... others?

Traditional techniques often have to be reformulated to handle more than one objective.

SBSE can be extended to multiple objectives with compautue ease.

Some possibilities ...

... covering arrays with: Fewest constraints violation

Interactive → Best tcover  
Lowest oracle cost  
....

Interaction sequences that:

- Ajax →
- Maximize fault cover com
  - Minimize oracle cost
  - Longest/shortest sequence
  - Include certain paths etc.

## Wasif Afzal (questions for Kpodjeac et al)

Q: why “linear” regression?

Q: Don’t you think that C \* K metrics are software dependent?

Q: wouldn’t it be useful to test for statistical significance for  $R^2$  for RQ-1?

## Mark Harman

### Interactive Evolution

Very little work on Interactive Evolution in SSBSE

Why?

+ perhaps because of the difficulties

But this approach could be so useful

+ incorporate domain knowledge

+ capture real world assumptions

+ take account of “fuzzy, messy, ill defined human stuff”

---

In requirement problems: - Juan J. Durillos’s talk:

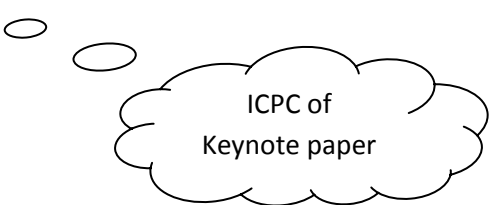
- Explore/expose un stated: dependencies, requirements, assumptions, etc.

In architectural problems Dongrun Kim’s talk:

- Capture quantitative assessments
- Explore developer cog. Models

In authorship (not direct but related...) ... Maxium Sherertalov’s talk...

- Tailored pretty printing
- Compare styles & performance



ICPC of  
Keynote paper

## Xin Yao

**Myth 1:** SSBSE relies on experimentation.

There is little theoretical & comparative work.

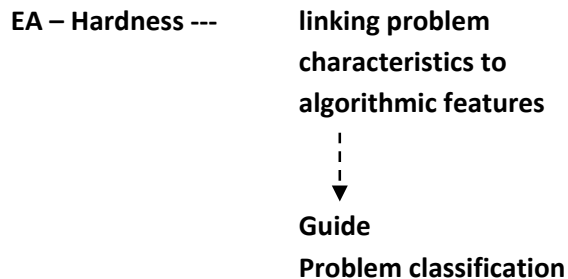
**FACT:**

1. Vigorous time complexity analysis of search algorithms for testing has appeared in several papers.
2. There are established methodology and benchmarking studies in the search community.
3. However, there are still huge gaps between theory & practice.

**Myth 2:** Search algorithms are complex/complicated

**FACT:**

1. Evolutionary algorithms are not significantly more complex than simulated annealing.
2. It is true that it is inappropriate to use complex algorithms for simple problems.
3. There has been significant progress made recently in characterising difficult problem: what makes a problem hard for a search algorithm?



**The fitness issue:**

- Hard to define precisely & accurately
- Hard to quantify

**No easy solutions, but ...**

- Co-evolution
- Interactive evolutionary computation (IEC)
- Evolutionary computation
  - o As a discovery engine
  - o Not just a problem solving tool

**Supporting what Enrique said:**

**There are many search algorithms that may be of interest to SErs:**

- **Optimisation in an uncertain environment**
- **Dynamic optimisation**  
**Robust optimisation**