hazards of verification

Daniel Jackson, MIT

Haifa Verification Conference · October 28, 2008
warnings

the contents of this talk are

- anecdotal, not analytical
- broad, not focused
- old, not novel

It is insufficiently considered that men more often require to be reminded than informed.

--Samuel Johnson
how we got here

growth in SAT power (number of variables, data from Sharad Malik)
\[ \text{one example of why early pessimism about verification was misplaced} \]
but will verification made software safe and dependable?
• on the road ahead: much progress, but hazards too

hazards due to
• technical factors
• engineering factors
• social/managerial factors
technical factors
unsound confirmation

examples

- finite scope & unrolling [KOA, Dennis VSTTE08]
- lack of coverage [CP bug after 8 years, Holzmann]
- abstraction [binary search, Bloch]

```
L := 1; U := N
loop
  if L > U then
    P := 0; break
  endloop
  M := (L+U) div 2
  case
  X[M] < T:  L := M + 1
  X[M] = T:  P := M; break
  X[M] > T:  U := M - 1
```

Holzmann paper: http://spinroot.com/spin/Workshops/ws08/HJG08.pdf
Bentley column: Communications of the ACM; Volume 26, Issue 12 (Dec. 1983)
Also see footnote (which page?) in http://www.cs.cmu.edu/~jcr/craftprog.html
how big a bound?

minimum scope/bitwidth/unrolling to find bugs in voting code

<table>
<thead>
<tr>
<th>class</th>
<th>method</th>
<th>error</th>
<th>min bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>CandidateListMetadata</td>
<td>init</td>
<td>under</td>
<td>1 / 3 / 1</td>
</tr>
<tr>
<td>KiesKring</td>
<td>addDistrict</td>
<td>bug</td>
<td>1 / 3 / 1</td>
</tr>
<tr>
<td>VoteSet</td>
<td>addVote(String)</td>
<td>over</td>
<td>1 / 3 / 1</td>
</tr>
<tr>
<td>KiesLijst</td>
<td>clear</td>
<td>over</td>
<td>1 / 3 / 3</td>
</tr>
<tr>
<td>AuditLog</td>
<td>getCurrentTimeStamp</td>
<td>over</td>
<td>2 / 1 / 1</td>
</tr>
<tr>
<td>Candidate</td>
<td>init</td>
<td>under</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>CandidateList</td>
<td>addDistrict</td>
<td>under</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>CandidateList</td>
<td>addKiesLijst</td>
<td>over</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>CandidateList</td>
<td>init</td>
<td>over</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>KiesKring</td>
<td>addKiesLijst</td>
<td>bug</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>KiesKring</td>
<td>init</td>
<td>under</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>KiesKring</td>
<td>make</td>
<td>under</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>KiesLijst</td>
<td>addCandidate</td>
<td>over</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>KiesLijst</td>
<td>compareTo</td>
<td>bug</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>KiesLijst</td>
<td>make</td>
<td>over</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>VoteSet</td>
<td>addVote(int)</td>
<td>over</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>VoteSet</td>
<td>validateKiesKringNumber</td>
<td>over</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>VoteSet</td>
<td>validateRedundantInfo</td>
<td>over</td>
<td>2 / 3 / 1</td>
</tr>
<tr>
<td>KiesKring</td>
<td>clear</td>
<td>over</td>
<td>2 / 3 / 3</td>
</tr>
</tbody>
</table>

Table from Dennis, VSTTE08
unsound counterexamples

examples

- unsound checker finds more bugs [Xie and Aiken 2005]
- most effort on error reporting [Pincus et al, Prefix]

overconstraint

examples

- declarative models of software (Alloy, Z, VDM, B, etc)
- axioms for code verifiers
- ‘unreachable states’ in model checking
approaches

vacuity and coverage in model checking
- Beer, Ben-David, Eisner, Rodeh
- Chockler, Kupferman, Vardi
- Chechik, Devereux, Gurfinkel

coverage in Alloy
- new algorithm [Torlak, FME08]
engineering factors
are bugs in code the problem?
- Mackenzie: 3% of software fatalities due to code
- most problems in human/computer interaction

is run-time-error elimination enough?
- ‘Sorry no more bugs’ -- Greg Nelson, 1980

sad examples
- PLUGR, Afghanistan 2001
- Airbus A320, Warsaw 1993

airborne ↔ not WheelPulse ↔ disabled

ENV × MACHINE ✓ 12

see Michael Jackson, Problem Frames, Addison Wesley, 2001
an approach

dependability case for proton therapy

Robert Seater, 2008
Korean Air 747, Guam 1997: 200 killed

If the ARTS IIA minimum safe altitude warning system had been operating as initially intended, a visual and aural warning would have activated about 64 seconds before flight 801 impacted terrain --NTSB report
ignoring design

early blender patent
  · opening too small for child’s hand
  · removal of closure disconnects blade

examples
  · Therac 25: removed hardware interlock
  · voting software: immutable types
  · emergency stop: uses message queue

time to think again about
  · safety kernels and modularity
platform risk

IDE risk
\· refactoring may not preserve meaning
\· >7 such bugs open in Eclipse

language risk
\· in Java, eg: memory model, generics

operating system viruses
\· time to infection for new PC: 4 mins

configuration problems
\· DLLs, classpaths, etc

Sample Eclipse refactoring bugs, thanks to Adam Kiezun:

- [extract local] must not ignore value changes: [https://bugs.eclipse.org/bugs/show_bug.cgi?id=27740](https://bugs.eclipse.org/bugs/show_bug.cgi?id=27740)
- [inline] Inlining synchronized method should create a synchronized block: [https://bugs.eclipse.org/bugs/show_bug.cgi?id=112100](https://bugs.eclipse.org/bugs/show_bug.cgi?id=112100)
- [push down] field lets client access formerly hidden field instead [refactoring]: [https://bugs.eclipse.org/bugs/show_bug.cgi?id=235118](https://bugs.eclipse.org/bugs/show_bug.cgi?id=235118)
- [pull up] field ignores hiding of inherited field [refactoring]: [https://bugs.eclipse.org/bugs/show_bug.cgi?id=235112](https://bugs.eclipse.org/bugs/show_bug.cgi?id=235112)
- [use supertype] changes to static binding, changing program behaviour: [https://bugs.eclipse.org/bugs/show_bug.cgi?id=233796](https://bugs.eclipse.org/bugs/show_bug.cgi?id=233796)
- [generalize type] fails to see lack of overriding [Refactoring]: [https://bugs.eclipse.org/bugs/show_bug.cgi?id=233437](https://bugs.eclipse.org/bugs/show_bug.cgi?id=233437)
- [push down] method changes program semantics in presence of overloading: [https://bugs.eclipse.org/bugs/show_bug.cgi?id=234981](https://bugs.eclipse.org/bugs/show_bug.cgi?id=234981)
social/managerial factors
process

does process really matter?
bad process

Alameda County, CA, 2003

- 25% of voting machines crashed on boot
- so Diebold installed uncertified patches

Accuvote-TSx

bad process

London Ambulance, 1992

• contract awarded to lowest bidder
• report from Arthur Andersen ignored
• no independent QA, software changes on-the-fly
• no incremental deployment, no paper backup
• untested change in operations

neglecting process

NOAA weather satellite at Lockheed Martin, September 2003


“Proximate Cause: The NOAA N–PRIME satellite fell because the LMSSC operations team failed to follow procedures to properly configure the TOC, such that the 24 bolts that were needed to secure the TOC adapter plate to the TOC were not installed.”
overconfidence

Titanic, 1912

• advanced technology, ‘unsinkable’
• so enough lifeboats not needed

on Titanic: http://en.wikipedia.org/wiki/RMS_Titanic
Texas A&M bonfire

- traditional began in 1928
- by 1990's, crane needed

what happened in 1999

http://www.fayengineering.com/structural.html
(thanks to Moshe Vardi for spotting this!)
### MAR knockout

- major Chicago hospital
- pharmacy database failure
- medication records lost

---

"Accidents are signals sent from deep within the system about the vulnerability and potential for disaster that lie within"

--Richard Cook and Michael O’Connor

---

Managers are now aware of the problem of entering zero into database fields and are trained to bypass a bad data field and change the value... ships do go dead in the water... People sometimes make mistakes and systems break. The trick is we have trained our crew...

-- Commanding Officer, USS Yorktown
Panama City Hospital, 2001

- Theratronic-780 with therapy planning system by Multidata
- 18 patients killed


Theratronic made by makers of Therac 25
3 Panama physicists tried for second-degree murder

- Olivia Saldaña González paid for her own defence; earns $585/month
- sentenced to four years in prison
- suit by families against Multidata rejected by Panama court

Given [the input] that was given, our system calculated the correct amount, the correct dose. It was an unexpected result. And, if [the staff in Panama] had checked, they would have found an unexpected result.

-- Mick Conley, Multidata
conclusions
implications for research

if you reward publication, you get
  • focus on logic & algorithms
  • benchmarks, not real problems
  • throwaway implementations

some good strategies
  • fund tool development [NSF infrastructure]
  • issue challenges [VSR's Mondex, Flash]
  • publish case studies [Z, Patterns]

will interdisciplinary work help?
implications for teaching

what we typically do

‣ focus on ‘respectable’ topics (eg, semantics)
‣ illustrate with small problems
‣ say hard parts are out-of-scope
‣ set formal problems that are easy to grade

instead, we might

‣ explain ‘soft’ aspects too
‣ illustrate with substantial case studies
‣ address the hard parts
‣ set informal, open-ended problems
thank you!