



Preliminary Investigations on Intelligent Modeling of UML Scenarios



West Virginia University.

Tim Menzies (WVU)

8/1/08



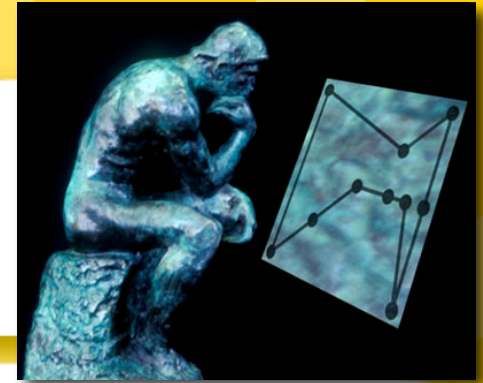
Problem



- ➔ How to understand “adversity”?
- ➔ UML modeling off-nominal “rainy day scenarios”
 - Essential for mission assurance and V&V and IV&V of mission- and safety- critical components
- ➔ Too many adverse off-nominal situations:
 - E.g. 20 things to go right,
 - 2^{20} (over a million) ways that case can go off-nominal.
 - If temporal orderings matter,
 - Then $20! (20*19*18*17*....*3*2*1) > 10^{18}$



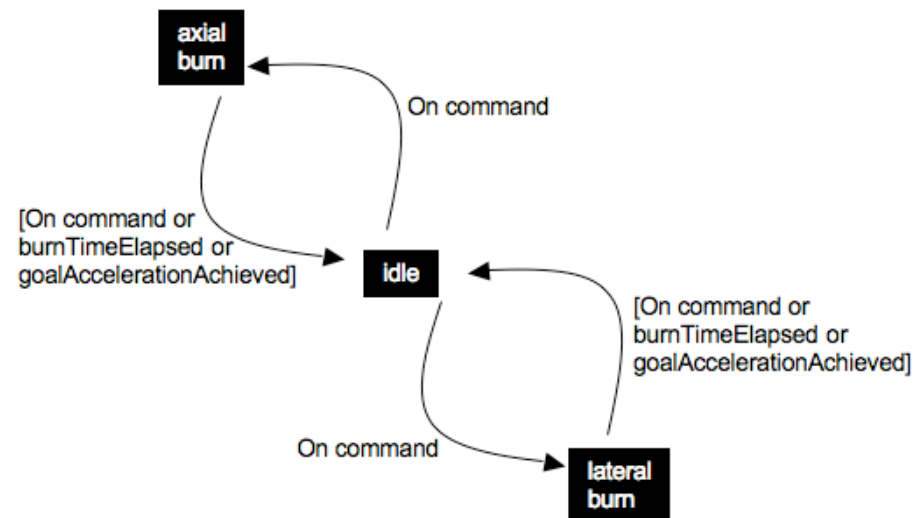
Approach



- ⇒ Intelligently sample potential set of scenarios variants.
 - Prioritization of possibilities
 - relevant → dull
 - Not model checking
 - not covering of all parts of the state space
- ⇒ Joint exploration of space of adverse conditions
 - By humans and AI

Approach (details)

- ➔ Import XML of real-world NASA UML models (JUNO)
 - Thanks Wes Deadrick
- ➔ Constrained scenario generation
 - Loop through the model
 - Note the repeated structures
 - Learn cliches: predictors for next state given N prior states
- ➔ Constrained variant generation
 - From the frequency cliches
 - Cluster
 - Show users random samples from each cluster
 - Rank samples by “interesting” of “dull”



Relevance to NASA

- ➔ Even experienced engineers bias their conclusions based on pre-conceptions.
 - Columbia ice strike,
 - Challenger launch temperature
- ➔ Can lead to “selective looking”
 - Missing events which, in retrospect, were obvious
 - E.g. 46% of 192 subjects asked to count number of passes did not notice the big hairy gorilla walking into the basketball game
- ➔ Selective looking is dangerous
 - Isn't what we don't know that gives us trouble, it's what we know that ain't so.
-- Will Rogers



Gorillas in our midst: sustained inattention blindness for dynamic events Perception, 1999, volume 28, pages 1059 - 1074 Daniel J Simons, Christopher F Chabris



Accomplishments

- ➔ Import XML of real-world NASA UML models (JUNO)
 - Thanks Wes Deadrick
- ➔ Execution engine of those UML models
- ➔ Constraint modeling language
- ➔ Data miner that can find options to select preferred runs.
 - “Preferred” defined w.r.t. constraints
 - So model checkers tell you what can be wrong
 - This tool tells you what to change such that the odds of doing wrong is decreased / increased
 - Not “what is” but “what can be changed”

```
xmlresult1.xml
<show name="Law &amp; Order">
  <episode>
    <season>13</season>
    <number>22</number>
    <title>Sheltered</title>
    <airdate>2003-05-14-05:00</airdate>
    <synopsis>
      The NYPD detectives hunt down a sniper who kills his
      victims in broad daylight.
    </synopsis>
  </episode>
  <episode>
    <season>13</season>
    <number>23</number>
    <title>Couples</title>
    <airdate>2003-05-21-05:00</airdate>
    <synopsis>
      The detectives catch four murders and a kidnapping on
      the same day.
    </synopsis>
  </episode>
  <episode>
    <season>13</season>
    <number>24</number>
    <title>Smoke</title>
    <airdate>2003-05-21-05:00</airdate>
    <synopsis>
      An eccentric comedian is under suspicion of murdering
      his baby son by dangling him over a ledge.
    </synopsis>
  </episode>
</show>
```

(NASA XML not shown for reasons of confidentiality)



Next Steps



- ➔ Scale up
 - Our learner needs more speed
- ➔ Technical possibility
 - Not supervised learning
 - But clustering
- ➔ GUI
 - So more users can use these tools.
- ➔ User studies