Code Level Safety Analysis

*(Or "**IS IT SAFE**?")*

Ken Wong

University of British Columbia

Software Safety

- *"Is it* ***SAFE?****"*
- *Certification of critical systems*
  - *ISESS Safety Workshop - FDA, FAA, ...*
  - *Software safety standards - IEC 1508, ...*
- *Industrial critical system engineering*
  - *e.g., Praxis Critical Systems, UK*
  - *Role of formalWARE*

Software Safety Verification

- *Safety process*
  - *Identify, analyze and control hazards*
- ***"<u>IS</u>** it safe?"*
  - *Safety vs correctness/reliability*
  - *Demonstrate absence of hazards*
- *Safety verification methods*
  - *Dynamic analysis, Static analysis*

Long Thin Slice Problem

- *Critical code not isolated*
  - *Data flow from inputs to hazardous outputs*
  - *“Long thin slice” of hazard-related code*
- *Industry Example: CAATS*
  - *OO architecture - example of problem*
  - *Safety program - context for solution*
- *“Is **IT** Safe?”*

Safety Code Analysis Method

- *Create model of hazard-related code*
  - *Flatten, Fillet, Fragment, Filter and Represent (Formalize)*
  - *Understanding code and relation to hazard*
- *Reason about safety*
  - *Argue for the absence of hazards*
  - *Validate model*

Results and Current Status

- ***"<u>It is safe</u>(?)"***
- *Methodology - long thin slice problem*
  - *Framework - other techniques*
- *Results and future work*
  - *Model of the long thin slice*
  - *Rigorous (formal?) safety argument?*
  - *Model validation (SPARK, JK Nair)?*