

# Maintenance, Sharing, and Re-Use of Resources: Lexica, Test Suites, and Others

Francis Bond

Nanyang Technological University

2016-06-18

# The Three Programming Virtues

- **Laziness**: The quality that makes you go to great effort to reduce overall energy expenditure.
- **Impatience**: The anger you feel when the computer is being lazy.
- **Hubris**: The quality that makes you write (and maintain) programs that other people won't want to say bad things about.

Larry Wall (1996)  
*Programming Perl*, O'Reilly

# Can we be more virtuous?

- Can we improve our tools?
  - ▶ We leverage work on one (DELPH-IN) grammar across all grammars, can we do this across frameworks?
- Can we share resources?
  - ▶ If our description is at a comparable level and correct, then it should be vaguely similar — we found this for the Japanese LFG and HPSG grammars: we could share lexicons
  - ▶ We can compare using common test suites
    - ★ We used the LFG test-suite for Indonesian
    - ★ We got Chinese test-suites from Stefan Mueller, Yusuke Miyao, ...
  - ▶ Can we share post-processors/applications (common tasks)?
- Can we improve maintenance by e.g. comparing different versions or approaches?

- It's hard to get funding for maintenance/expansion
  - ▶ DELPH-IN is to some extent an attempt to solve this problem
- But it is really important
- I think we are doing ok (lots of nice examples at this meeting of code reuse and sharing)
- Can we make it even better?
- Can HPSG and LFG help each other (more)?

I think we can become (even) more virtuous  
any concrete suggestions/ideas