Does Conventional software need conventions?

Brian Luzum
Merri Sue Carter
Importance of Conventional Software

- Where the theory meets the practical
- Unambiguously defines the model
- Allows all users of conventions access to the same information – Within limits of the hardware, operating system, compiler, etc.
- Note that this can be influenced by the code
Current Situation

- Software (subroutines) are provided to the IERS Conventions by colleagues
  - Usually experts in the field
  - Software written to fit personal needs/tastes/styles
  - Validation of software performed

• Usually
Software Practices

• Proper Design
• Test cases/unit testing
  – Product Quality/Correctness
• Code to design
  – Completeness/Maintainable
• Sufficient error trapping
  – Improve diagnostics and security
• Documentation
• Acceptance tests
• Upgrades managed through version control
Current Situation from a software engineering perspective

- Code is usually written in Fortran
- Often does not come with test case
- Consistency is not ensured because software is not designed to be integrated with other modules
- Limited or no error trapping
- Documentation of the code is variable
- Informal acceptance criteria
- Version control is not well defined
What languages should be supported?

• Currently all code is written in Fortran (no standardization between 4/77/90/95/2003)
  – Reasonable for EO scientific community
  – Reasonable for EO user community?
• Other language possibilities
  – C
  – C++
  – Python
  – Others?
Test Cases

- Should we provide test cases for each of the routines?
  - If so, how extensive?
  - Are there any implications for software acceptance?
Standardization of Code

• Does the coding need to be standardized?
  – SOFA produces standard code

• Sufficient error trapping?
  – Provide meaningful error messages
• Are there standard things that belong in the comments?
  – Authorship/dates
  – Purpose
  – Variable definitions
  – Units

• Need for additional documentation?
Acceptance Criteria

• Current approval through the same process as other parts of the Conventions
  – Is this sufficient?
  – Is a more well-defined method necessary?
Version Control

- Currently no version control
- Should version control be instituted?
  - If so, how?
    - Manually
      - Minimum standard metadata
    - Through software
      - Subversion
      - Others?
Example of consistency concerns

- Simon et al. (1994) fundamental arguments occur independently in
  - IAU2000A/IAU2000B
  - PMsdnut
  - hardisp
  - Zonal tide routine

- Have the implementations been checked for consistency?
• Ultimately, this becomes a question of resources
  – Is any improvement in the usability of the product worth the cost of implementation?
• Depends on the scope of implementation
Summary

• Need to determine
  – Scope of usage
  – Scope of obligation
  – What languages to support
  – What standardization to implement
  – Necessity of test cases
  – Necessity of version control
  – Acceptance criteria
  – Level of support