On the Future of Generators

The Benefits of Generators for Reuse

James M. Neighbors
Bayfront Technologies, Inc.
neighbrs@netcom.com

Code Library Problems

- Code libraries burden each programmer using the library to determine code interactions.
- Code libraries work. Issue is ease of use and flexibility, not capability.
- OO techniques not anxious to address "When Objects Collide".
- Composition is our most important product.
- "Code corrosion" limits reuse lifetimes.

Generator Solutions

- Model of a domain-specific language (e.g., SQL, HTML, VHDL, but OpenGL, GUIs as protocols)
- Compilation of input description checks and limits composition of reusable parts.
- Input description is independent of shifting environment.
- Generator is an educational tool. Has a model of specification and implementation.

Generator Drawbacks

- Step out of the problem domain and die.
- Maintain generated code and die.
- Requires an understanding (Domain Analysis) to make.
- Effort looks orthogonal to development. (i.e., your not hacking code)

Generator Benefits

- Refinement - *anyone remember that?*
- Various implementations (e.g., inline, threaded code, threaded code interpreter)
- Various refinement goals (e.g., code, simulations, diagrams)
- Real optimization above the code level. (e.g., protocol state removal)
- Problem domain specific analysis of the input description. (e.g., protocol deadlocking)