

Creating Python Bindings for Large C++ Frameworks

Tamer Fahmy

(Vienna University of Technology, Austria)

Possible approaches

- Time versus feasibility constraints:
 - Do it yourself (and suffer)
 - Using a wrapper generator and write some glue code

Available wrapper generators

- SWIG
- SIP
- Boost.Python (Pyste)
- GCC_XML

C++ a hard to wrap language?

- In/Out parameters
- Mapping of advanced features of C++ into Python (e.g. operator=)
- Operator and method overloading
- Inheritance, templates, exceptions
- Runtime System available? Use it! (autocasting)

Pivy - Python binding for Coin

- Chosen approach (why SWIG?)
- How i did it?
 - -includeall and fake_headers
 - Got the basic types right!
 - 70% of the work done
 - Using typemaps and glue code!

Pivy - Python binding for Coin

- libswigpy (SWIG runtime library)
- Bridging to other bindings (PyQt)
 - Mixing SWIG and SIP
 - Get the Pointer to the C++ object
 - Watch out for the ownership!

Pivy - Python binding for Coin

- Mixing different source languages, ObjC with C++? (PyObjC bridge)
- unsigned char *? numarray, string, list?

Conclusions and Thoughts

- C++ support improved a lot over last 2 years
- GCC_XML as a core for wrapper generators to work on?
- Common pointer exchange API standard for wrapper generators?

Thank you!

Questions?