Core Show and Tell

Pat Scott

Department of Physics, McGill University

Slides (usually) available from
http://www.physics.mcgill.ca/~patscott

Core/Model responsibilites (the fun ones)

To design, code and maintain:

- A system to allow observables and likelihoods to be defined in a modular way
- A system that allows those functions to be hooked up easily in arbitrary ways
- A system to allow arbitrary external codes to be hooked up, and their functions, etc[1] used seamlessly from within obervable/likelihood codes
- A system for reading and interpreting some text-based scan initialisation file
- A framework for defining arbitrary models and their relationships to observables, external codes and other models
- A system for passing the values of model parameters, observables and likelihoods back and forth between the scanner and observable/likelihood calculators

[1] yes, even classes

Core/Model responsibilites (the fun ones)

To design, code and maintain:

- A system to allow observables and likelihoods to be defined in a modular way – **"rollcall"**
- A system that allows those functions to be hooked up easily in arbitrary ways
- A system to allow arbitrary external codes to be hooked up, and their functions, etc[1] used seamlessly from within obervable/likelihood codes
- A system for reading and interpreting some text-based scan initialisation file
- A framework for defining arbitrary models and their relationships to observables, external codes and other models
- A system for passing the values of model parameters, observables and likelihoods back and forth between the scanner and observable/likelihood calculators

[1] yes, even classes

Core/Model responsibilites (the fun ones)

To design, code and maintain:

- A system to allow observables and likelihoods to be defined in a modular way – **"rollcall"**
- A system that allows those functions to be hooked up easily in arbitrary ways – **"rollcall" + "dependency resolver"**
- A system to allow arbitrary external codes to be hooked up, and their functions, etc[1] used seamlessly from within obervable/likelihood codes
- A system for reading and interpreting some text-based scan initialisation file
- A framework for defining arbitrary models and their relationships to observables, external codes and other models
- A system for passing the values of model parameters, observables and likelihoods back and forth between the scanner and observable/likelihood calculators

[1] yes, even classes

Core/Model responsibilites (the fun ones)

To design, code and maintain:

- A system to allow observables and likelihoods to be defined in a modular way – **"rollcall"**
- A system that allows those functions to be hooked up easily in arbitrary ways – **"rollcall" + "dependency resolver"**
- A system to allow arbitrary external codes to be hooked up, and their functions, etc[1] used seamlessly from within obervable/likelihood codes – **"backend system"**
- A system for reading and interpreting some text-based scan initialisation file
- A framework for defining arbitrary models and their relationships to observables, external codes and other models
- A system for passing the values of model parameters, observables and likelihoods back and forth between the scanner and observable/likelihood calculators

[1] yes, even classes

Core/Model responsibilites (the fun ones)

To design, code and maintain:

- A system to allow observables and likelihoods to be defined in a modular way – **"rollcall"**
- A system that allows those functions to be hooked up easily in arbitrary ways – **"rollcall" + "dependency resolver"**
- A system to allow arbitrary external codes to be hooked up, and their functions, etc[1] used seamlessly from within obervable/likelihood codes – **"backend system"**
- A system for reading and interpreting some text-based scan initialisation file – **"ini file parser"**
- A framework for defining arbitrary models and their relationships to observables, external codes and other models
- A system for passing the values of model parameters, observables and likelihoods back and forth between the scanner and observable/likelihood calculators

[1] yes, even classes

Core/Model responsibilites (the fun ones)

To design, code and maintain:

- A system to allow observables and likelihoods to be defined in a modular way – **"rollcall"**
- A system that allows those functions to be hooked up easily in arbitrary ways – **"rollcall" + "dependency resolver"**
- A system to allow arbitrary external codes to be hooked up, and their functions, etc[1] used seamlessly from within obervable/likelihood codes – **"backend system"**
- A system for reading and interpreting some text-based scan initialisation file – **"ini file parser"**
- A framework for defining arbitrary models and their relationships to observables, external codes and other models – **"ModelBit"**
- A system for passing the values of model parameters, observables and likelihoods back and forth between the scanner and observable/likelihood calculators

[1] yes, even classes

Core/Model responsibilites (the fun ones)

To design, code and maintain:

- A system to allow observables and likelihoods to be defined in a modular way – **"rollcall"**
- A system that allows those functions to be hooked up easily in arbitrary ways – **"rollcall" + "dependency resolver"**
- A system to allow arbitrary external codes to be hooked up, and their functions, etc[1] used seamlessly from within obervable/likelihood codes – **"backend system"**
- A system for reading and interpreting some text-based scan initialisation file – **"ini file parser"**
- A framework for defining arbitrary models and their relationships to observables, external codes and other models – **"ModelBit"**
- A system for passing the values of model parameters, observables and likelihoods back and forth between the scanner and observable/likelihood calculators – **"Likelihood container object" + friends**

[1] yes, even classes

Core/Model responsibilites (the boring ones)

To design, code and maintain:

- A simple central system for error/exception managing
- A simple central system for logging
- A central documentation system
- A compilation and configuration system

Core/Model responsibilites (the boring ones)

To design, code and maintain:

- A simple central system for error/exception managing
- A simple central system for logging
- A central documentation system
- A compilation and configuration system

Core/Model responsibilites (the boring ones)

To design, code and maintain:

- A simple central system for error/exception managing
- A simple central system for logging
- A central documentation system
- A compilation and configuration system

Core/Model responsibilites (the boring ones)

To design, code and maintain:

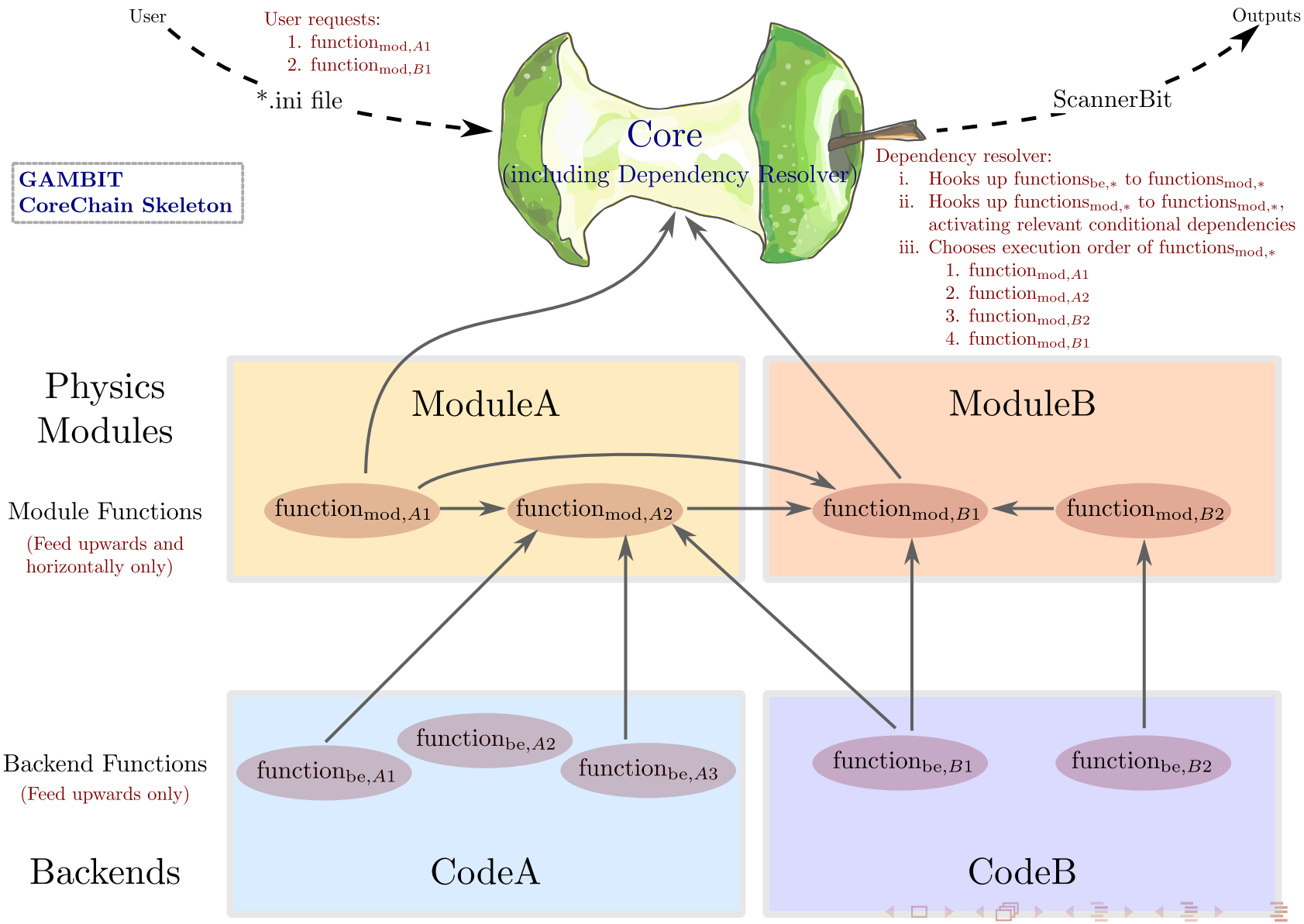
- A simple central system for error/exception managing
- A simple central system for logging
- A central documentation system – **Doxygen**
- A compilation and configuration system

Core/Model responsibilites (the boring ones)

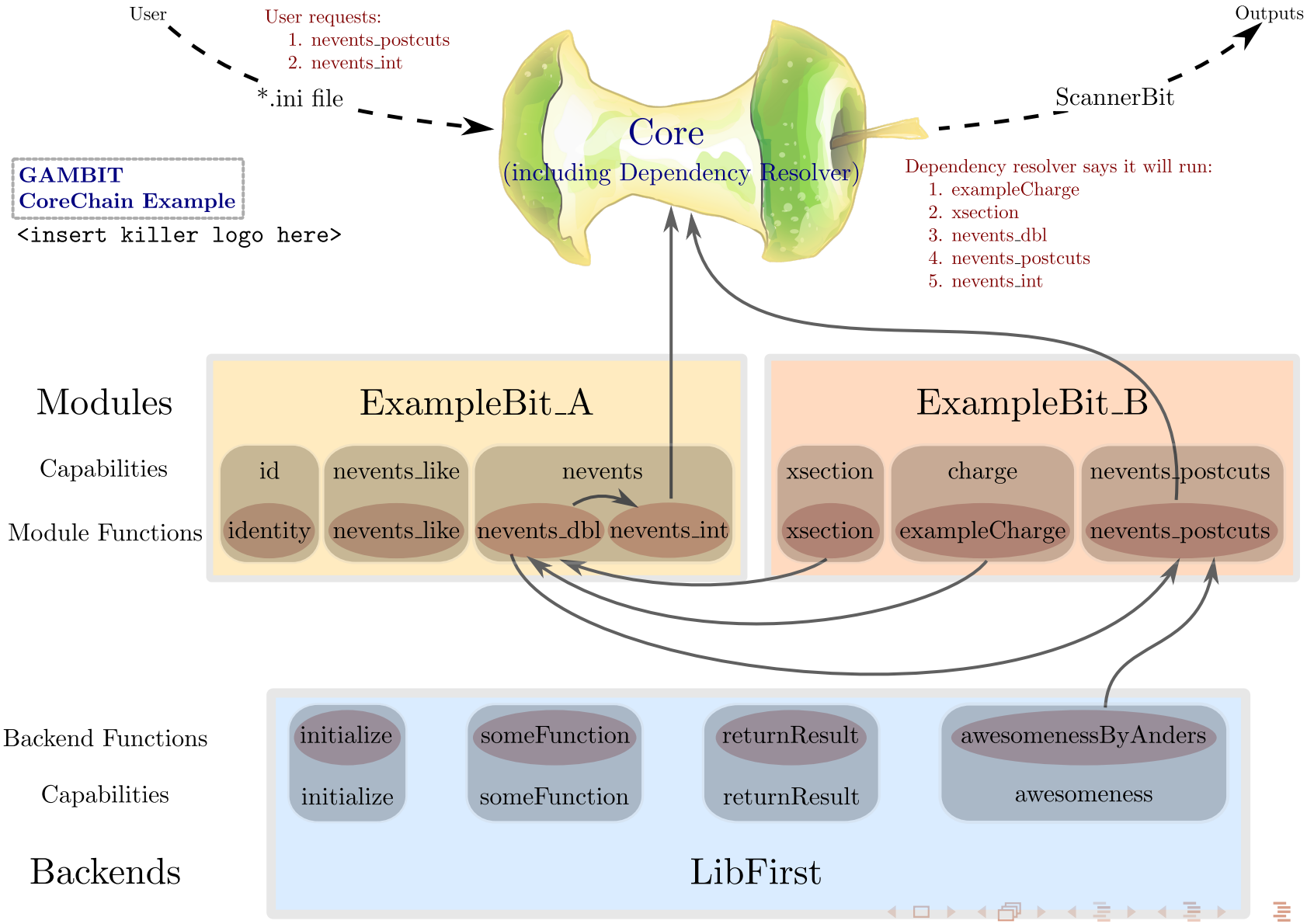
To design, code and maintain:

- A simple central system for error/exception managing
- A simple central system for logging
- A central documentation system – **Doxygen**
- A compilation and configuration system

Rollcall + backends + dependency resolution
Skeleton example, take 2



Rollcall + backends + dependency resolution
Better example, take 2



Basic files to look at

- `modules/Core/include/module_rollcall.hpp`
- `modules/Core/include/backend_rollcall.hpp`
- `modules/ExampleBit_A/include/ExampleBit_A.hpp`
- `modules/ExampleBit_A/src/ExampleBit_B.hpp`
- `modules/ExampleBit_B/include/ExampleBit_A.hpp`
- `modules/ExampleBit_B/src/ExampleBit_B.hpp`
- `modules/Backends/include/backend_libfirst.hpp`
- `modules/Backends/lib/libfirst.cpp`