

# Cactus and Frameworks



Tom Goodale

LSU Center for Computation and Technology

goodale@cct.lsu.edu

[www.CactusCode.org](http://www.CactusCode.org)

Cactus Retreat

30<sup>th</sup> Apr 2004



# What Is Cactus

- Cactus is a **framework** for developing portable, modular applications, in particular, although not exclusively, high-performance simulation codes.
- Cactus is designed to allow experts in different fields to develop modules based upon their expertise and to leverage off modules developed by experts in other fields to perform their work, with minimal knowledge of the internals or operation of the other modules.

(\*) Any Unix-like machine plus Windows



# What is a Framework

- A term which is sometimes over-used, and used confusingly
  - e.g. are frameworks large-scale patterns or just another form of component.
- Some definitions
  - “ a framework is a re-usable design of all or part of a system that is represented by a set of abstract classes and the way their instances interact”
  - “ a framework is the skeleton of an application that can be customized by an application developer”
  - “ a framework is an architecture, plus an implementation, plus documentation that captures the intended use of the framework for building applications”



- “ ... a collection of abstract classes, and their associated algorithms, constitute a kind of framework into which particular applications can insert their own specialised code by constructing concrete subclasses that work together. The framework consists of the abstract classes, the operations they implement, and the expectations placed upon the concrete subclasses”
- “ A framework is an abstract design for a particular kind of application, and usually consists of a number of classes. These classes can be taken from a class library, or can be application specific.”
- “ ... a set of co-operating classes that makes up a re-usable design for a specific class of software. A framework provides architectural guidance by partitioning the design into abstract classes and defining their responsibilities and collaborations. A developer customises the framework to a particular application by subclassing and composing instances of framework classes”



# Key Ideas

- Components
  - Objects providing some functionality which can be plugged together
- Collaboration model
  - Defined ways in which the components interact
- Inversion of control
  - Traditionally a developer reuses components from a library by writing a main program that calls the components when necessary
  - In a framework the main program is reused, and the developer or user decides what is plugged into it and might make some new components that are plugged in.
    - Developers code gets called by framework code
    - The framework determines the overall structure and flow of control of the program
- Re-usability of analysis, design and code.



# What does a Framework provide ?

- A re-usable and standard context for components
  - Components generally make assumptions about their environment
  - Making components with different assumptions work together is hard
  - Framework provides standard way for components to exchange data and invoke operations on each other
- Frameworks make it easier to develop new components
  - Almost always necessary for new applications
- A framework prescribes how to decompose a problem
- Applications are built on frameworks by
  - Creating new components
  - Configuring components together
  - Modifying working examples



# Using a Framework

- Frameworks can be used in many ways
  - Black Box
    - Just connect components together and go
  - Write new components
    - Often necessary
  - Modify the framework itself
    - Hardest way
- Learning the framework
  - Normally harder than learning to use a software library as you need to learn many different things at once
  - Need good documentation
  - Need examples



# Different “Frameworks”

- CCAT
- XCAT
- CUMULVS
- DDB
- ALICE
- ESMF
- Flame
- InDEPS
- POET
- ISIS++
- Harness
- KeLP
- MpCCI
- OASIS
- Overture
- PALM
- PAWS
- POOMA
- SAMRAI
- SCIRun
- SILOON
- GrACE
- RocketMan
- ...
- Architectures
  - CCA, HLA (IEEE1516)



# Cactus and Frameworks

- It's clear that Cactus is a framework
- Cactus is primarily a framework for high-performance computing
  - Its design is optimised to enable numerical routines to interact in the most efficient manner
  - The components which come with the framework are, generally, guided by this.
- It's possible to build domain-specific framework within Cactus
  - Have done this with the “Einstein Toolkit” which has been very successful.
- Want to continue development and add more domain specific frameworks
  - Computational Fluid Dynamics
  - Climate Modelling
  - ...