Assembling Component Frameworks

THE COMPONENT RETRIEVAL LANGUAGE
What is a Component Framework?

- Set of individual software modules that perform a specific task
  - Can range from a few components to many
  - Must be extensible
- Used for various purposes
  - i.e. scientific simulations
Motivation

- Distributed Software Frameworks are hard to assemble
  - EinsteinToolkit comprised of 130 individual components
  - Very tedious to manually checkout or update
GetCactus

- Designed by Prof Allen in 2000
- Specific to Cactus Framework
- Supports only a few Version Control Systems
- Still difficult to distribute the framework
  - Users must edit the thornlist
Component Retrieval Language

- Designed to fix problems with GetCactus
- Provides unified, tool agnostic syntax
- Abstracts authentication procedures
- No longer specific to Cactus
CRL Syntax

!TARGET = $ARR
!TYPE = svn
!AUTH_URL = https://svn.cactuscode.org/arrangements/$1/$2/trunk
!URL = http://svn.cactuscode.org/arrangements/$1/$2/trunk
!CHECKOUT =
CactusArchive/ADM
CactusBase/Boundary
CactusBase/CartGrid3D
CactusBase/CoordBase
GetComponents

- Written to be very modular
  - Currently supports 5 version control systems and http/ftp downloads
  - Very easy to add more
- Can take input as local file or URL
- Manages all authentication issues
Generating component lists is still long and tedious

Don’t need all EinsteinToolkit modules to run a simulation

- Customized component lists would streamline the checkout and build process
- But how to do this?
Component Dependencies

- Dependency tracking could allow custom built simulations
- Specify one component containing data about the simulation
  - Initial values, type of simulation, etc
- Then recursively check component dependencies
Cactus Configuration Language

- CCL already provides dependency checking
  - Thorn inheritance
  - Required functions
- Can’t trace dependencies from Initial Data to Base
Cactus Configuration Language
Cactus Configuration Language

How to identify essential components for a simulation?

- Possibly with metadata
- Divide thorns into categories and check if each category is used
Drawbacks of Metadata

- Requires understanding of the physics involved
- Could easily become Cactus/ET specific again
  - Possibly layer on top of GetComponents
Acknowledgements

- Many thanks to Dr Allen, Dr Löffler, Dr Schnetter, and Dr Brandt