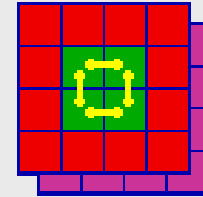




ATLAS Level-1 Calorimeter Trigger



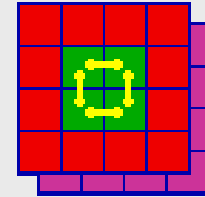
Modules and Sub Modules

Dividing the Parts

Bruce.M. Barnett, Rutherford Appleton Laboratory
ATLAS Level-1 Calorimeter Trigger Collaboration Meeting.
November 7-10, 2001



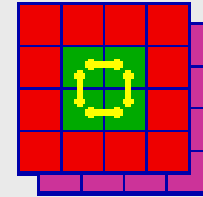
Overview



- A Basic Definition
- Functional Units
- A Hardware Definition Language
- On Two World Views
- Relationships
- Convergence



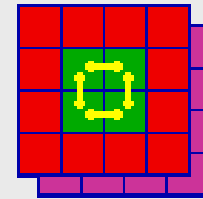
A Basic Definition



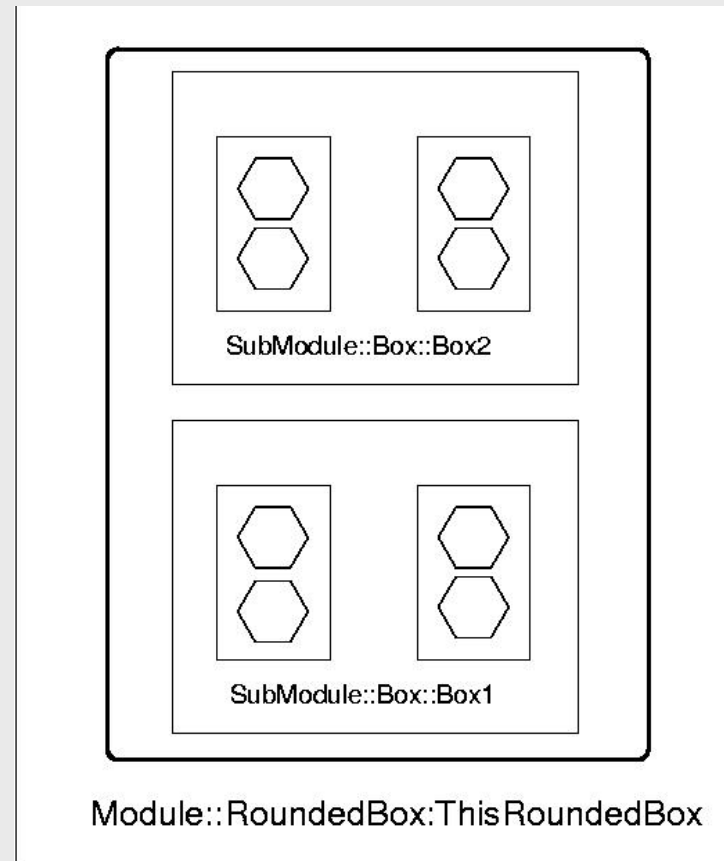
- **module / n.**
 - a standardized part or independent unit used in construction, esp. of furniture, a building, or an electronic system.
 - an independent self-contained unit of a spacecraft (lunar module).
 - a unit or period of training or education.
 - a a standard or unit of measurement. b Archit. a unit of length for expressing proportions, e.g. the semidiameter of a column at the base.
 - [French module or Latin modulus: see modulus]
- **modulus / n. (pl. moduli /) Math.**
 - a the magnitude of a real number without regard to its sign. b the positive square root of the sum of the squares of the real and imaginary parts of a complex number.
 - a constant factor or ratio.
 - (in number theory) a number used as a divisor for considering numbers in sets giving the same remainder when divided by it.
 - a constant indicating the relation between a physical effect and the force producing it.
 - [Latin, = measure, **diminutive of modus**]



Functional Units

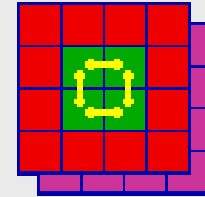


- Functional units
 - Contain other Functional Units
 - And are each an instance of a given class
- SubModule
 - of type RoundedBox has
 - 2 Submodules of type box
 - Box1 and Box2
 - Box1





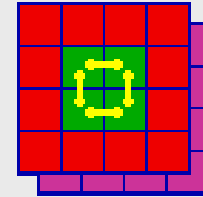
A Hardware Definition Language



- Parts Files:
 - Define how some Module named ThisRoundedBox (without specifying that Module is a RoundedBox) contains Register Hexagon Hexagon1 (without specifying that Hexagon is contained in a Submodule of class Box named Box1)
 - Define hardware relationship (offset/address) of Hexagon1 from base of ThisRoundedBox
 - even though we're interested in the offset relative to Box1
- Conf Files:
 - Define details of how the smallest piece is mapped into its atomic constituents (bits)



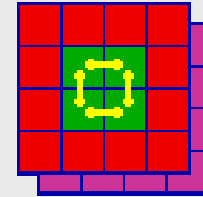
On Two World Views



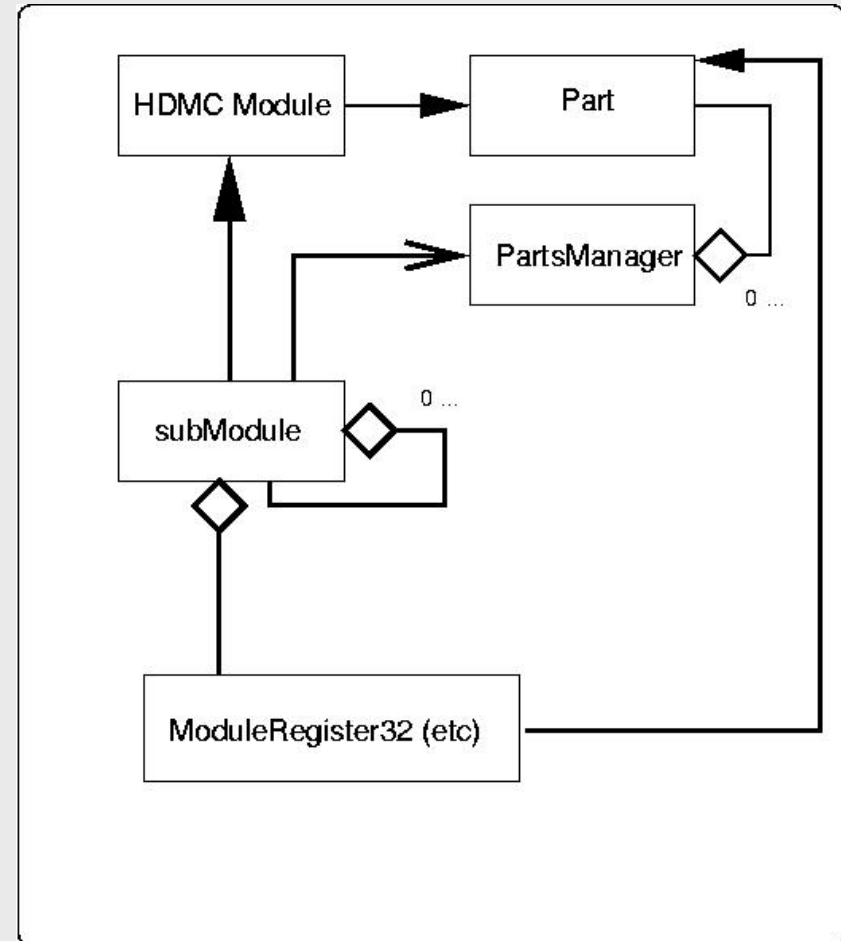
- Diagnostic View
 - Philosophy: scripting, database.
 - Uses file based hardware definition to create a system out of modules.
 - GUI Based
 - Should be able to represent all levels of abstraction that are relevant to the system (... and we're not)
- DAQ View
 - Philosophy: Compiled, dynamic library.
 - Should use the same hardware definition to create BASIC C++ system components.
 - Regbuild (conf files)
 - modbuild (parts files)
 - Actual (Module Services) methods for these can be filled in later and *associated* with the basic components.



Relationships



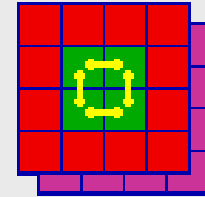
- But the relationships between the pieces (classes) are the same in both world views





Convergence

(Digital or Otherwise)



- Rationalisation of database approach:
 - Parts files ... XML
 - There are Parts files and Parts files:
 - Some need to contain basic crate/bus definitions.
 - Others need to contain specification of what submodules each specific module contain.
- Submodule code generator
 - **syntax** and **implementation**