RELAP5-3D Conversion to FORTRAN 90

Presented by
Dr. George L. Mesina

2002 RELAP5 International Users Seminar

September 4-6, 2002
Outline

• Reason for Conversion
• Impact on Code Users
• Conversion Plan
• Details of Some Aspects of the Conversion
• Summary
**Reason for Conversion**

- **Modernization, FORTRAN Language moves forward**
  - Vendors only supply FORTRAN 95 compilers now
  - FORTRAN 2000 standard is nearing completion
- **Take advantage of better language features**
  - allocate, modules, derived types, etc.
- **Simultaneously improve code development effort**
  - Readability
  - Efficiency of modifying the source code
  - Maintainability (bug fixes, shelf-life of code, etc.)
Impact on Code Users

- RELAP5-3D/Ver2.0+ unusable with old compilers
  - If Operating System does not F90 libraries, even executable RELAP5-3D may not run.
  - Source code clients need FORTRAN 90+ compiler.
- NO upper limit on problem size, when conversion done.
  - Virtually all arrays are within FA, a fixed-size array.
  - After conversion, arrays will be allocated after input processing.
- Conversion will cause changes in speed and arithmetic calculations.
Conversion Plan

- Upgrade bit packing to FORTRAN 90.
- Convert FTB database to FORTRAN 90 modules.
- Convert fixed-length common blocks to modules.
- Convert special features
  - restart-plot technology, renodalization, scan-request, et cetera.
- Modify installation procedures, resolve platform issues.
Details of Some Aspects of the Conversion

- Modernization of bit-packing
- Use of derived type
- Replacement of “comdecks” with modules
- Array notation
RELAP5-3D Bit Packing Overview

• Bit packing is a data compression technique
  – several pieces of data are stored in a single word of memory.

• Special bit pack operations and tricks:
  – MIL-SPEC: Store, modify, and retrieve a particular datum from a compressed word.
  – Logical operators: for single bits.
  – Equivalence between reals and integers is used to apply special operator for one data type to the other.
  – These are used separately and in combinations.
FORTRAN 90 Bit packing

- Numerous, standard FORTRAN 90 bit operators
- Easier to read, understand, work with than MIL-SPEC and logical operators.
- FORTRAN 90 bit operators will replace MIL-SPEC and combined logical operators.
Conversion of Bit Packing

• Pervasive change will affect approximately 1400 files in the following directories:
  – envrl, fluids, matpro, relap, rgui, scdap

• Bit-packing broken into 3 major categories with about 30 subcategories or kinds
  – Conversion is complicated by changes in order and use of variable and arrays as arguments.

• A program converts one kind of bit packing in one file.
• One kind of conversion is performed on a group of 10 subroutines, then test cases are run.
Derived Types

- Derived types are user-defined data types that combine data of standard types as fields.
- Example: Derived type: Volume Data
  - Each field represents 1 array of the VOLDAT comdeck.
  - Field 1 is “p” (pressure); field 2 is tempg, et cetera.
  - Fields may have integer, logical, or even character data.
- Array, VL, is declared to be of derived type Volume Data.
  - VL(k) contains all data for volume k in its fields.
  - VL(k)%tempg = temperature of the gas in volume k
- Conversion to derived types is being programmed.
Modules

- A FORTRAN 90 module
  - global data and subprograms
  - is accessible only to subprograms that “USE” it.
- Modules eliminate data overwrites due to indexing errors.
- Modules are have greater capabilities than comdecks.
- Compiled modules are “outside” the program in a separate file.
  - Modules can be attached to many different executable programs.
RELAP5-3D Modules

- Standard format: 3 Sections
  - Documentation: Description and Data Dictionary
  - Declarations (Type, dimension, access, pointer, allocatable, etc.)
  - Module Subroutines
- Conversion of a comdeck
  - Some of the comdecks have a very complex memory layout (e.g. heat structures)
  - Use of derived types & pointers can reduce complexity
RELAP5-3D Modules (continued)

- Arrays conversion to derived types.
- Every subroutine that accesses a converted array has to be changed simultaneously.
  - Indexing with variables & arrays and referencing via FA-array makes it difficult.
- Method: create module with transfer routines; temporarily keep comdeck.
  - Convert one subroutine to access derived type
RELAP5-3D Modules (continued)

- Transfer from FA to module at beginning of routine; transfer back at end. Test!
- When every subroutine is converted, the comdeck and transfers can be eliminated.
- Must be done individually for each comdeck. There are about one hundred.
- A suite of programs is being written to perform much of the conversion automatically
Array Notation

- Allows operations on whole arrays or parts of them.
- Examples:
  - Vector(INDEX1:INDEX2) = 0.0
  - zeroes out Vector from INDEX1 to INDEX2.
  - A = B * C
  - Multiplies MxN matrices B and C to get A.
- Compiler is free to optimize array operations when used. Can invoke intrinsic functions.
- Will cause RELAP5-3D to run faster on some platforms, possibly slower on others.
Summary

• Conversion to FORTRAN 90 will impact users.
  – Requires a FORTRAN 90 compiler or beyond.
  – May affect calculations and speed.
  – Will eliminate input model size restrictions.
  – The restart-plot file will change.
• Conversion will make the source code easier to work with, reducing bugs and maintenance costs.
• Conversion extends usable lifetime of RELAP5-3D.
• Some aspects of the conversion were covered.