INL/CON-06-01065
PREPRINT

Restructuring RELAP5-3D for Next Generation Nuclear Plant Analysis

American Nuclear Society 2006 Annual Meeting

Donna Post Guillen
George L. Mesina
Joshua M. Hykes

June 2006

Idaho National Laboratory

Restructuring RELAP5-3D for Next Generation Nuclear Plant Analysis

Donna Post Guillen[1], George L. Mesina[1] and Joshua M. Hykes[2]

[1]*Idaho National Laboratory, P.O. Box 1625, Idaho Falls, ID 83415*
[2]*Pennsylvania State University, 0355 Atherton Hall, University Park, PA 16802*

INTRODUCTION

RELAP5-3D is used worldwide for analyzing nuclear reactors under both operational transients and postulated accident conditions. Development of the RELAP code series began in the 1960's and since that time the code has been continuously improved, enhanced, verified and validated [1]. Since RELAP5-3D was selected as the reference code for Next Generation Nuclear Plant (NGNP) systems analysis [2], it is necessary to modernize the code to support the development of NGNP and other Generation IV reactor design concepts. This paper discusses both the reengineering of RELAP5-3D into structured code and the additions of physical models necessary to model Gen IV plants.

GEN IV IMPROVEMENTS

The Gen IV program examined one hundred reactor concepts and reduced this to six finalists: Super Critical Water Reactor (SCWR), Very High Temperature Reactor (VHTR), Gas Fast Reactor (GFR), Lead Fast Reactor (LFR), Sodium Fast Reactor (SFR) and Molten Salt Reactor (MSR). Each has special modeling needs [3] that require new or upgraded physical models in RELAP5-3D (a.k.a. ATHENA-3D). Until one final concept is selected, improvements for all six design concepts are being made. As such, in the past four years, RELAP5-3D has undergone much modification including enhancement of the turbine model and addition of new models for constitutive relations, mechanical interactions, gap conductance, compressors, and air diffusion [3].

Continuing research and development on the Gen IV project require ongoing maintenance and development of the RELAP5-3D code. Architectural modernization [4] is ongoing in the forms of rewriting in a modern programming language, code restructuring, database reengineering, and user interface improvement

For prismatic design gas reactors, it is important to model turbulence in the lower plenum. Coding that allows RELAP5-3D to couple via PVM with a computational fluid dynamics (CFD) code, such as FLUENT, has been added. The turbulent single-phase gas flow is modeled in detail with the CFD code while the rest of the plant is modeled by RELAP5-3D; each code takes input from the other as a boundary condition, performs its calculations, and feeds data back to the other code at every time step.

For each reactor concept, appropriate working fluids were either added or upgraded [5-7]. For the VHTR, GFR, SCWR and LFR, upgrades were made to the turbine model and various correlations [5]. For SCWR, the steam tables were enlarged, refined near the critical point, discontinuities in transport properties from sub- to supercritical thermodynamic states were eliminated, and upgrades to the calculation of meta-stable states were made [10]. Code upgrades to facilitate Gen IV modeling are ongoing.

RESTRUCTURING

The purpose of restructuring legacy software is to make the logic paths clearer; thus, the code is more legible and less costly to develop and maintain. This is important when numerous models are being added or modified for Gen IV modeling needs. "Restructuring" here refers to reorganizing the code according to Dijkstra's structured programming paradigm [8]. Structured programs can be broken into sub-sections or blocks, each with one point of entry and one of exit, such that control passes downward from one block to the next with no unconditional branches to higher levels of the structure [9,10].

A commercial restructuring tool, FOR_STRUCT® v2.2 [11], is used to transform unstructured RELAP5-3D FORTRAN 77 routines into structured code. The restructuring effort is encumbered by several issues [12]. FOR_STRUCT® v2.2 cannot process files with coding outside the FORTRAN 77 standard, does not recognize pre-compiler directives, cannot fully restructure large or complex subroutines, and runs on MS-Windows only.

Handling FOR_STRUCT® Limitations

The first two limitations are handled by pre-processing the file to create usable FOR_STRUCT® input and, after restructuring, post-processing to undo the pre-processing. Powerful UNIX scripts do the pre- and post-processing. The pre-processed file automatically transfers to Windows, where FOR_STRUCT® restructures it, and then automatically transfers back to the UNIX workstation for post-processing. Finally, although FOR_STRUCT® can only partially restructure large or complex files, it was found that applying FOR_STRUCT® to its own output produced further improvements.

Experimentation revealed three iterations to be sufficient for most routines.

RELAP5-3D has both platform-specific coding and FORTRAN 90 coding that FOR_STRUCT® cannot process. Pre-processing makes non-standard statements into comments and converts FORTRAN 90 derived type array references to unique FORTRAN 77 variable names

RELAP5-3D was designed to be customized for various users' needs via the use of conditional coding implemented through C-language Pre-Processor (CPP) directives. A pre-compiler directive is pre-processed by first making a comment copy of the directive and then removing the actual directive by running CPP with the directive's flag activated. The comment copy moves with the restructured code. Post-processing uses the comment to recreate the original directive.

Restructuring Complexities

The complexity of the restructuring task increases with the number of different pre-compiler flags, nesting and branching of pre-compiler directives, and file size. The files were sorted from least to most complex and restructuring began with the least complex. As each new restructuring issue was encountered, the process was expanded to handle it. For example, each flag that has a branch necessitates two applications of FOR_STRUCT®, one with the flag off and with it active. The output files are restructured differently and recombined carefully. If there are B different flags with nesting, there are 2^B combinations of active and passive flags to process and recombine.

RESULTS

A process for restructuring RELAP5-3D has been developed and implemented. The number of logic jumps in the restructured files, measured by the reduction in number of GO TO statements and line labels, has been greatly reduced compared to that of the original files (see Table 1). All restructured files have been extensively tested to ensure that they produce output identical to the original files.

REFERENCES

1. RELAP5-3D Code Development Team, "RELAP5-3D Code Manual," INEEL-EXT-98-00834, Revision 2.3, Idaho National Laboratory, Idaho Falls, ID, USA, http://www.inel.gov/relap5/r5manuals.htm, April 2005.
2. Schultz, R.D., et al., "Very High Temperature Reactor Design Methods Development and Validation Research & Development Program Plan," INEEL/EXT-04-02293, Rev. 2, Idaho National Laboratory, September 2005, p. 83.
3. Johnsen,, G. W., et al, "ATHENA-3D for Generation IV Reactor Analyses," BE2004, AND Winter Meeting, Washington D.C., Nov. 2004.
4. Mesina, G.L., "Architectural Advancements in RELAP5-3D," 2005 Transactions of the American Nuclear Society, Vol. 93, November 2005.
5. Riemke, R. A., Davis, C. B., Schultz, R. R., "RELAP5-3D Code for Supercritical-Pressure, Light-Water-Cooled Reactors," 11th International Conference on Nuclear Energy, Tokyo, Japan, April, 2003.
6. Davis, C. B., and Shieh, A. S., "Overview of the Use of ATHENA for Thermal-Hydraulic Analysis of Systems with Lead-Bismuth Coolant," 8th International Conference on Nuclear Engineering, Baltimore, MD, USA, April, 2000.
7. Davis, C. B., "Implementation of Molten Salt Properties into RELAP5-3D/ATHENA, INEEL/EXT-05-02658, Idaho Falls, ID, USA, January, 2005.
8. Dijkstra, E.W., "Go To Statement Considered Harmful," *Communications of the* Association for Computing Machinery, Inc., Vol. 11, No. 3, pp. 147-148, March 1968.
9. Friedman, F.L., and Koffman, E.B., Problem Solving and Structured Programming in FORTRAN 77, Addison-Wesley Publishing Company Inc., Reading, MA, 1987.
10. "The Definition of Structured Programming," General Services Administration, Federal Standard 1037C, Telecom Glossary, 2000.
11. Cobalt Blue, Inc., "FOR_STRUCT®, Your FORTRAN Structuring Solution," 11585 Jones Bridge Rd, Suite #420-306, Alpharetta, GA, 30005, USA, 1997.
12. Mesina, G.L., and Hykes, J.M., "Restructuring of RELAP5-3D," 2005 International RELAP5 Users Seminar, Jackson Hole, WY, September 7-9, 2005.

TABLES

Table 1. Restructuring Measures.

Measure	Before	After	Ratio
Average # of GOTOs	8.8	5.3	1.66
Maximum # of GOTOs	213	99	2.15
Average # of Labels	22.0	10.2	2.16
Maximum # of Labels	210	43	4.88
Files Restructured	449 out of 556		