

A New Format for Handling Nuclear Data

Sang In BAK,¹ René BRUN,² Federico CARMINATI,^{2,*} Jong Seo CHAI,¹ Andrei GHEATA,² Mihaela GHEATA,²
Seung-Woo HONG,¹ Yacine KADI,^{1,2} Vijay MANCHANDA,^{1,3} Tae-Sun PARK¹ and Claudio TENREIRO^{1,4}

¹ *Sungkyunkwan University, Suwon, Korea*

² *CERN, Geneva, Switzerland*

³ *BARC, Mumbai, India*

⁴ *Talca University, Santiago, Chile*

(Received 26 April 2010)

The ASCII ENDF format for nuclear data has been used for four decades. It is practical for human inspection and portability, but it is not very effective for manipulating and displaying the data or for using them in Monte-Carlo applications. In this paper we present a prototype of a nuclear data manipulation package (TNudy) based on the ROOT system (<http://root.cern.ch>). The ROOT object-oriented C++ framework is the de-facto standard in high energy and nuclear physics since ten years. Starting from the ENDF format, the data is stored in machine-portable binary format. Root files also offer a powerful direct access capability to their different sections and compressibility upon writing, minimising the disk occupancy. ROOT offers a complete library of visualisation and mathematical routines and the Virtual Monte-Carlo system, which allows running different transport Monte-Carlo (Geant 4, Geant 3) with common scoring and geometry modellers, which comes as part of ROOT. ROOT contains isotope decay data and the possibility to describe the evolution of isotopic vectors via Bateman equations. The addition of the ENDF information to Root will allow the development of a transport code for low energy neutrons and other combination of projectile target, either stand-alone as part of the Root system or in combination with some of the other Monte-Carlo systems within the framework of the Virtual Monte-Carlo.

PACS numbers: 29.85.+c

Keywords: ND2010, Nuclear data, ENDF, ROOT, Monte-Carlo, Geant3, Geant4

DOI: 10.3938/jkps.59.1111

I. INTRODUCTION

The Evaluated Nuclear Data Format (ENDF) [1] dates back more than 40 years. This is quite a unique story in the Computing world, as this quite simple ASCII format has survived all the revolutions of the Information Technology (IT) world and is still used today in spite of the many evident draw-backs of the alphanumeric format. In this paper we present the work we are receiving in the framework of a collaboration between our group at the European Laboratory for Particle Physics (CERN, Geneva, Switzerland) and the Emerging Nuclear Technology Group at the Department of Energy Science (DOES)¹ of Sungkyunkwan University aimed at developing a new format for the nuclear data.

II. THE ENDF FORMAT

The ENDF format was originally developed to describe the United State compilation of nuclear data eval-

uations known as ENDF/B. At the time of the compilation of the sixth revision (ENFD/B-VI), it was observed that many other nuclear data compilations were using the same format, therefore it was decided to disentangle the format from the actual revision of the nuclear data and to identify it as ENDF-6 format. This format, which has become a de-facto standard, is managed by the US Cross Section Evaluation Working Group (CSEWG), and the format specifications are published by the National Nuclear Data Centre at the Brookhaven National Laboratory.

Most of the nuclear data available today from several national evaluations agencies are published in ENDF-6 format, including evaluations such as JEF (Europe), JENDL (Japan), BROND (Russia) and CENDL (China). The information contained in the evaluated files are interaction and production cross sections, integral and differential, but also decay and fission products yields, gamma production spectra and covariance matrices. The data are stored in a plain sequential ASCII file, where floating point numbers are stored with 7 digits precision in a record oriented structure. An important change that appeared recently was the introduction of several projectiles beyond neutrons and the extension of the upper energy limits from the “historical” 20 MeV (or

*speaker; E-mail: Federico.Carminati@cern.ch

¹ An international group established as a World Class University program supported by the Ministry of Education, Science and Technology

better 19.6) to 200 MeV and beyond.

Before mentioning several draw-backs of this format, some obvious and some less, which have prompted this work, let us describe why this is indeed a very good format and why it survived for so long. ASCII (text) formats have the advantage to be absolutely machine portable, particularly after the disappearance of the alternate text format supported by IBM (EBCDIC). The format provides both the data and the recipe on how to handle them. It is easy to document and allows the “human inspection” albeit some exercise and experience are needed. The fact that the text format is slower to read and takes up a lot of disk space compared with binary format could be less of a problem due to substantial increase in processor speed and decrease in storage price. This trend is however offset by the increase in the energy range and granularity of the nuclear data.

The obvious drawback of this format is indeed the size taken by text representation of numbers and the time it takes to read it. Single sections cannot be accessed directly and the file has to be read and partially interpreted sequentially to extract a single section. The fixed floating point format only offers seven digits precision, even with the special trick of omitting the exponent character. By the way, this makes it almost impossible to read the data from C or C++, allowing only one language to read, FORTRAN.

The question of the precision is indeed a serious one. While sometimes it is argued that cross sections and other quantities are known with a precision higher than 10^{-5} , one has to remember that the parameters stored in the evaluated files are sometimes results of fits, often of polynomials or transcendental functions, where a precision of five or six decimal digits can make substantial difference in the final results.

Another obvious drawback is the “flat” structure of the file that does not allow one to read just one section without having to read, and partially interpreting, all the lines before.

The major limitation, however, is perhaps in the fact that the data contained in the evaluation files are useless without the “recipe” which is contained in the manual. Every new application that intends to access the information has to write its own decoding routines, which is both time expensive and error prone. This limitation is not specific to the ENDF files, and indeed is one of the main reasons that have prompted the move from procedural languages to Object Oriented ones.

The aim of the project presented here is to propose a new format for the ENDF data which retains most of the advantages of the old system, and is compatible with it, while moving these data in the world of Object Oriented languages.

III. THE ROOT SYSTEM

The High Energy Physics (HEP) experiments that just entered operation at the Large Hadron Collider

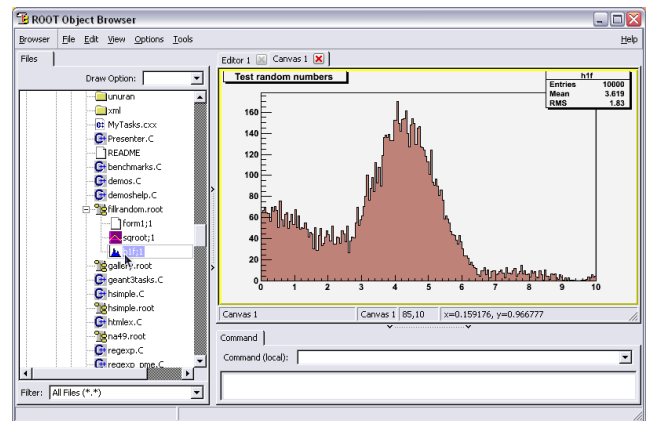


Fig. 1. (Color online) Example of interactive ROOT browser.

(LHC [2]) of the European Organisation for Nuclear Research (CERN [3], Geneva, Switzerland), will produce ~ 10 Peta-Bytes (PB, 10^{15} Bytes) of data per year for an estimated 20 years of operation. The computing requirements originating by the processing and analysis of this very large amount of data have spurred a large number of developments in Information Technology, among which the largest Grid one is in operation in the world (World LHC Computing Grid, WLCG [4]).

Another notable one amongst these development is the Root system [5]. Root is an object oriented framework for data analysis based on Object Oriented technology and written in C++. Its development started fifteen years ago and it is the de-fact standard for data manipulation and analysis in HEP. ROOT is actively maintained on all major combination of hardware and OS, including Mac OS X and Microsoft Windows.

Root is an Open Source system (LGPL license [6]) written in C++ and composed by a large number of modular libraries depending on a core system which provides introspection (Run Time Type Information, RTTI) and I/O. The introspection is achieved via a custom-developed C++ interpreter which builds run-time accessible class dictionaries. Classes are written to files with full information about their content and structure, and a “schema evolution” system allows reading different versions of a class with respect to the one that is currently compiled. The same system allows the usage of C++ as scripting language. The RTTI information is accessible interactively and even graphically via a Graphics User Interface (GUI) class browser (see Fig. 1).

The interpreter (CINT) is quite robust and it allows the interactive execution of very large pieces of C++ code (it can interpret itself!). Data are automatically compressed on writing and uncompressed on reading. “Array” like data structures can be streamed “data member”-wise, achieving higher compression and allowing selective reading. Files are written in machine independent mode, and all conversions and endiannes issues are automatically treated by the system.

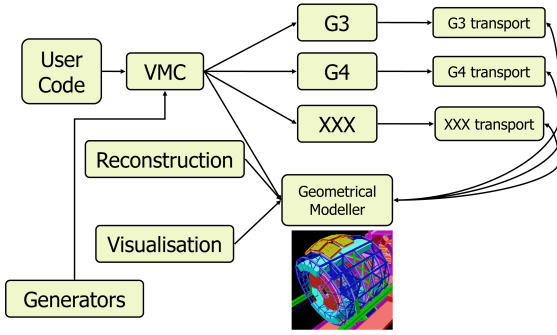


Fig. 2. (Color online) The structure of the ROOT Virtual Monte-Carlo.

IV. THE VIRTUAL MONTE-CARLO

Transport Monte-Carlos are essential in HEP to design the detector and to evaluate background and efficiencies when detecting particle collisions. The need to evaluate the systematic error introduced by the Monte-Carlo brings naturally to simulate the detector with more than one transport code. This is however made difficult by the necessity to describe the same geometry, which can be extremely complex, with the different geometrical modellers of the transport codes used, and also to rewrite several times the scoring routines, which in case of large detectors can be rather complex.

To help in this task, the so-called Virtual Monte-Carlo (VMC [9]) has been developed, which allows to run different particle transport simulation Monte-Carlo without changing the user code and therefore the input and output format as well as the geometry and detector response definition. The ROOT classes of the VMC category provide a set of interfaces which completely decouple the dependencies between the user code and the concrete Monte-Carlo used, see Fig. 2.

Actual implementations are provided for the Geant 3 [7] and Geant 4 [8] Monte-Carlo transport codes in use in HEP. Other codes could of course be added.

V. THE TNUDY PROJECT

The main idea of this project is to provide a library for handling ENDF-6 files. The first step is to provide a set of C++ classes mimicking the structure of the ENDF-6 format but in Object Oriented format as a plug-in library to the ROOT system. This includes the ability to read ENDF-6 files and store them in ROOT binary format (RENDF). It will allow us to store the ENDF-6 information in compact binary format and to access different nuclides separately without reading the whole file. The option will be given also to write an ENDF-6 file from the compact ROOT format. This very simple interface aims at getting the data “into” the ROOT system and to provide easy access to the information contained.

The immediate advantage will be that the whole rich set of ROOT facilities for data manipulation and visualisation will be available to manipulate and inspect the

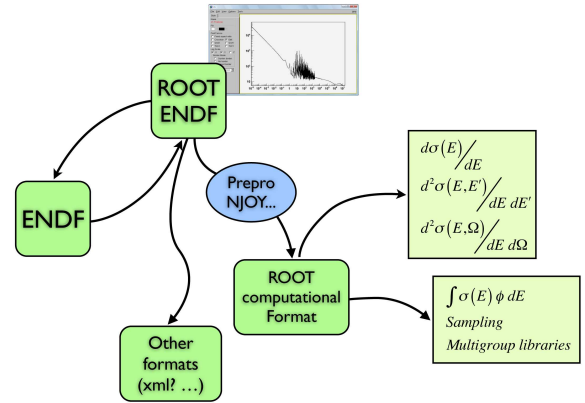


Fig. 3. (Color online) Structure of the TNudy dataflow.

nuclear data. Standard operations such as the interpolation of a function provided in the data (*e.g.*, a cross section) can be provided by the interface. The Object Oriented ability to provide abstraction will provide access to different representation methods contained in the ENDF. For instance, the spectrum of secondary particles can be represented via one of several functions or a grid. TNudy will offer a seamless access to the function, calling automatically the appropriate evaluation method, function or interpolation. This functionality is an important building block for the development of applications that could, starting from this format, create “reconstructed” information, in a way similar to what PREPRO [10] or NJOY [11] can do.

In the second part of the project, a set of classes is designed and implemented to store and access reconstructed point-like cross sections and secondary spectra in an efficient format for transport programs. The input will be taken from the files processed by the PREPRO suite. Sampling of cross sections for materials and mixtures is provided. Methods to sample secondary distributions (angle and spectra), fission neutron multiplicity and fission fragment yields are also part of the future library. The package also includes methods for spectra convolution with cross sections, to obtain reaction rates, “on the flight” Doppler broadening of cross sections, and cross section integrals.

This library can be integrated into the VMC to offer transport capabilities for low-energy neutrons to existing Monte-Carlo transport codes or even standalone. Similar classes will also be added for multi-group representation.

The success of this approach depends on the degree at which other groups will “buy” into it. Ideally, if some of the experts in the fields will decide to integrate their manipulation codes to create reconstructed cross-sections or special formats into this library, we will obtain a highly integrated code system.

At the price of installing the Root suite, different transport codes will not need any more to develop and maintain their own subsystem to read and sample the

ENDF data. As we have witnessed already several times during the Root evolution, the centralisation of procedures and formats will bring a marked improvement in the quality and functionality of the code, as several experts will contribute to the same code, allowing a faster development and a more thorough debugging.

The crucial point is that the approach proposed here will link the Nuclear Data to a specific software suite. All programmes accessing them will then depend on the Root software, and their usage will be possible only after the installation of the Root software. This may be seen as a limitation, but eventually a lot of benefits will follow in terms of synergy, and we believe that this would be a positive move.

As a lookout for the long-term future of this new format, the Root code has been in use for the last 15 years and the projected lifetime of the LHC experiments is in excess of 20 years. Moreover, other experiments, like the projects at the International Linear Accelerator, are starting to use Root as the base of their framework, extending the foreseeable lifetime of the system.

If this new format is accepted and used by many users, the format is “nothing else” than a set of classes which can be read and adopted in another framework or reimplemented.

As far as the backward compatibility is concerned, the TNudy library is able to produce and write ENDF-6 format, so that all the information can be recovered. The library itself exports a C++ Application Programming Interface (API). From our long experience in interfacing other languages to C++, it is certainly possible to offer access from FORTRAN, python and java, but also from other languages if need be.

The integration of the Nuclear Data with powerful data analysis and display methods open other interesting possibilities. Experimental data can also be stored in Root format. This is what the Root I/O is all about, having been developed for the LHC data. This would offer the possibility to develop a set of data-evaluation comparison tools and to integrate also evaluation codes into the same framework, laying the groundwork for a complete Nuclear Data evaluation and formatting platform. With a new generation of high-intensity experiments, the amount of experimental data produced will grow very rapidly, and the adoption of the Root format, which has been expressly developed for handling a very large set of experimental data, could help anticipating the solution to a problem that will certainly arise.

At the moment, we have completed the coding of the Root ENDF format (RENDF), and we are in the stage of validating. The next step will be to prepare a library to store the different sections of reconstructed pointlike information optimised for Monte-Carlo usage, with fast access to the quantities and sampling facilities. The input data will be taken from the ENDF files processed via

the PREPRO suite. This will allow us to write a simple transport code integrated into the Root VMC and to run some simple validation tests. We expect this phase to be finished by the end of the year 2010.

VI. CONCLUSIONS

A new format is proposed for the Nuclear Data. The new format is accompanied by a set of methods giving access to the data integrated in the Root system developed for the LHC experiments at CERN. Methods to access pointlike cross sections and spectra will be developed, and a simple transport Monte-Carlo will be developed and validated. This library will also offer the possibility to develop integrated data transformation methods, store experimental results and develop integrated evaluation methods. The work is being done in collaboration with the CERN Root team and the Department Of Energy Science of the Sungkyunkwan University (Republic of Korea). A first prototype is expected by the end of the year 2010.

ACKNOWLEDGMENTS

This research was supported in part by World Class University program (R31-2008-000-10029-0) through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology.

REFERENCES

- [1] Cross Section Evaluation Working Group, ENDF-102, BNL, 2009.
- [2] The LHC Project; <http://lhc.web.cern.ch/lhc>.
- [3] European Organization for Nuclear Research; <http://www.cern.ch>.
- [4] Worldwide LHC Computing Grid; <http://lcg.web.cern.ch/LCG>.
- [5] ROOT; <http://root.cern.ch>.
- [6] The Gnu Public License; <http://www.gnu.org/licenses/quick-guide-gplv3.html>.
- [7] R. Brun *et al.*, *The Geant 3 transport code*; http://wwwasdoc.web.cern.ch/wwwasdoc/geant_html3/geantall.html.
- [8] The Geant4 Collaboration, *Geant4, a toolkit for the simulation of the passage of particles through matter*; <http://geant4.cern.ch>.
- [9] VMC — ROOT; <http://root.cern.ch/drupal/content/vmc>; I. Hrivnacova *et al.*, *The Virtual Monte Carlo: Computing in High Energy and Nuclear Physics* (La Jolla, California, 2003).
- [10] PREPRO2000; <http://www-nds.iaea.or.at/ndspub/endl/prepro>.
- [11] NJOY 99 Nuclear Data Processing System; <http://t2.lanl.gov/codes/njoy99>.