LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-743637

Designing a New Structure for Storing Nuclear Data

C. M. Mattoon, B. R. Beck

December 21, 2017

European Physics Journal A

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

EPJ manuscript No.
(will be inserted by the editor)

LLNL-JRNL-743637

Designing a New Structure for Storing Nuclear Data

Progress of the Working Party for Evaluation Cooperation Subgroup #38

C.M. Mattoon[1] and B.R. Beck[1]

Lawrence Livermore National Laboratory, 7000 East Ave L-059, Livermore CA 94550

Received: date / Revised version: date

Abstract. An international effort is underway to design a new structure for storing and using nuclear reaction data, with the goal of eventually replacing the current standard, ENDF-6 [1]. This effort, organized by the Working Party for Evaluation Cooperation, was initiated in 2012 and has resulted in a list of requirements and specifications for how the proposed new structure shall perform. The new structure will take advantage of new developments in computational tools, using a nested hierarchy to store data. The structure can be stored in text form (such as an XML file) for human-readability and data sharing, or it can be stored in binary to optimize data access. In this paper, we present the progress towards completing the requirements, specifications and implementation of the new structure.

PACS. 29.87.+g Nuclear data compilation – 29.50.+v Computer interfaces

1 Introduction

The Evaluated Nuclear Data Format (now on version 6, abbreviated ENDF-6) has been used for nearly 50 years to store and exchange evaluated nuclear data. It serves as a standard, computer-readable form for nuclear scientists and engineers to provide recommended nuclear reaction data for use in nuclear engineering applications in power, national security and criticality safety. While ENDF-6 continues to play an important role in nuclear physics and engineering, it is showing signs of age. ENDF-6 was originally designed to be stored on 80-column punch cards, with 14 characters out of each line reserved for identifiers such as line numbers. Numbers are stored in fixed-width fields, limiting the number of significant digits that can be used to represent data. Documentation is allowed, but can only appear at the start of each file (that is, only once per projectile/target combination) and can only be in plain text. ENDF-6 files are difficult for humans to read, and frequently contain subtle errors that can remain hidden for years, partly due to the complexity of the format. Most importantly, a new generation of scientists and engineers are increasingly disinterested in learning the details of a legacy format like ENDF-6, preferring instead to work with modern software practices and tools (like XML) with which they are more familiar and that they find more useful. At a time when the nuclear data community is losing expertise to retirements, it needs to be willing to adopt modern practices in order to attract a new generation of talent. For all these reasons, a modern replacement for ENDF-6 has become an important priority.

Send offprint requests to:

The Working Party on Evaluation Cooperation [2] subgroup #38 (SG38) was formed in 2012 to oversee the design and implementation of a modern replacement for ENDF-6. The replacement should take advantage of the many advances in computer technology since ENDF-6 was created, and should be flexible, extensible, and both human- and computer-readable.

In this paper we will present on the progress that has been made by SG38 towards defining a new standard for storing nuclear data, and on the open challenges and questions that still need to be addressed before the new standard will begin to be fully utilized. We also present on the status of the Generalized Nuclear Data structure (or GND), which was developed at Lawrence Livermore National Laboratory. GND is being updated based on feedback from SG38, and is expected to evolve into the new standard format.

2 Requirements

The task of SG38 began with defining the list of fundamental *requirements* that a new structure must meet in order to fulfill its purpose as a replacement for ENDF-6. Over the course of six bi-annual SG38 meetings, the list of requirements has been refined through feedback and review. The latest version has been collected into a requirements document [3] that presents a high-level overview of how evaluated nuclear data are likely to be organized and used. The following list highlights some of the requirements outlined in that document:

– Data shall be stored in a hierarchical structure.

– The hierarchy should reflect our understanding of nuclear reactions and decays, clearly and uniquely specifying all such data.
– The hierarchy shall support meta-data that unambiguously describe how the data are to be interpreted.
– Details that refer to the same data should be stored as closely together as feasible.
– The hierarchy should use general-purpose data containers suitable for re-use across all application space. These data containers should be easy to use with commonly used computer languages and numerical libraries.
– Data containers should be designed to be consistent with object-oriented programming. This includes nesting data containers when it makes sense instead of defining a new sub-container.
– The hierarchy should eliminate redundancy where possible.
– For backwards-compatibility, all valid ENDF-6 data in the currently supported ENDF-102 Version 6 format manual shall continue to be supported by the new format.

These requirements apply primarily to the top-level hierarchy for organizing nuclear data. Two other 'specifications' documents have also been prepared. One of these presents requirements and specifications for storing properties of particles that are reaction-independent (such as mass, spin, parity, etc.). The other document describes the low-level data containers that will be used throughout the new database. These two documents are described further in sections 4 and 5 of this document.

The main reason the ENDF-6 format was defined was to allow nuclear data developers and users to share evaluated nuclear data. To support the sharing for processed data, evaluated data converted into a representation (e.g., grouping data for deterministic transport) more suitable for transport codes, the replacement for ENDF-6 shall also support the storing of processed nuclear data.

3 Proposed Hierarchy for Organizing Data

As the requirements defined by SG38 have evolved, the authors have also been modifying the Generalized Nuclear Data (or GND) structure in order to meet those requirements [4]. GND is being developed primarily by contributors at Lawrence Livermore National Laboratory and Brookhaven National Laboratory. The project pre-dates SG38, and was conceived as a way to organize nuclear data in a nested hierarchy that is designed to be understandable by users familiar with nuclear reaction physics, and that is easy to integrate into applications such as transport codes.

The overall hierarchy of a GND nuclear data file is summarized in figure 1. The top-level 'reactionSuite' element is similar to the MAT in an ENDF-6 file: it consists of a target and projectile (each stored as meta-data inside the reactionSuite element), and descriptions of each of the reactions that can occur during the interaction of that target/projectile. Each reaction is stored in a 'reaction' element that contains the cross section and an 'outputChannel' with a list of reaction products. If any reaction product is subject to decay, an additional 'decayChannel' can be added inside that product.

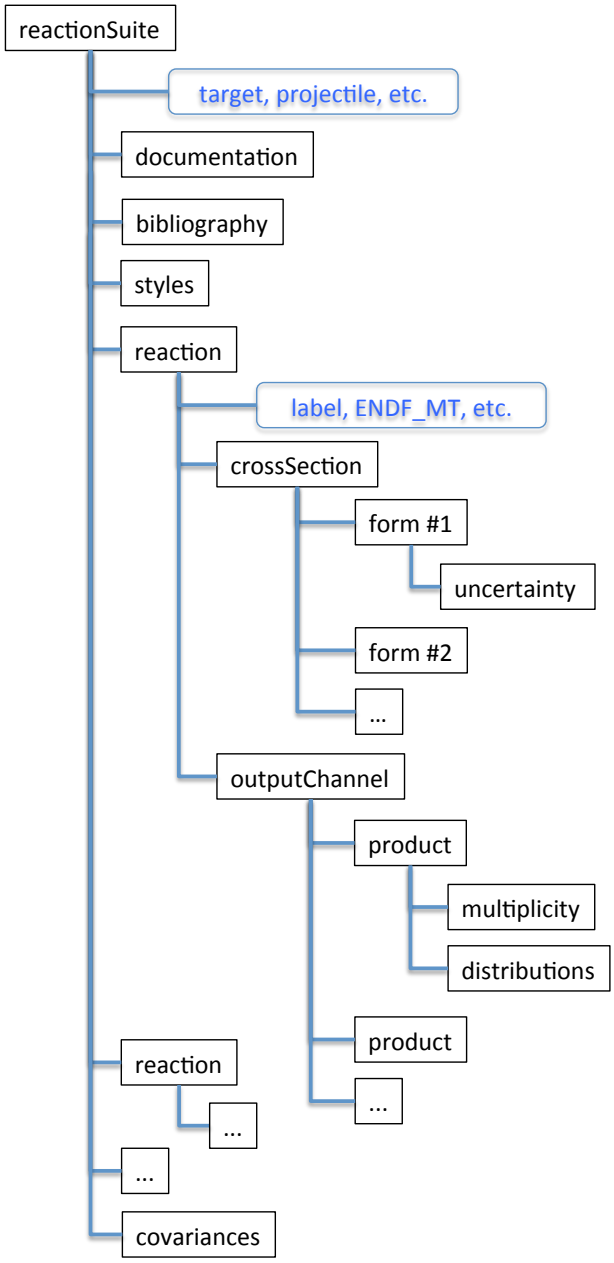

Fig. 1. A simplified overview of how data are organized inside the GND 'reactionSuite'. Elements are listed in black, with associated metadata listed in blue. The nested hierarchy includes multiple 'reaction' elements, each of which contains a cross section and a list of outgoing products with their associated multiplicities, distributions and possible other data. Uncertainties and/or covariances can be stored together with their associated data, or grouped together within the 'covariances' section at the end of the file.

Like ENDF-6, GND includes room for documentation and bibliography information, since documentation is a vi-

tal component of a nuclear data evaluation, helping users understand the ingredients that went into the evaluation as well as the potential problems they should be aware of before using it in their applications. One major change in GND compared with ENDF-6 is the ability to attach documentation not only at the top level of the file, but also inside specific reactions, cross sections, distributions, etc. Although these additional documentation elements are omitted from figure 1 to keep the overview simple, they will allow the evaluator to give detailed documentation for each part of the evaluation and to keep relevant documentation close to the data it describes.

Documentation in GND may be marked up using xhtml or latex, allowing the evaluator more freedom to include equations, html tables and other features. We note, however, that the documentation is *not* meant to be machine-readable in the sense that it cannot contain important details that change the interpretation or use of the evaluation. This choice was made partly in response to the ENDF-6 LIP flag (described on page 125 of the ENDF-6 manual [1]). According to the manual, the LIP flag can mean different things in different evaluations, in some cases modifying the 'MT' reaction identifiers to mean different reactions. The user must look inside the text documentation to determine what the LIP flag means in any file. Since parsing text documentation to extract the meaning of the flag is very difficult to automate, this flag effectively means that application codes cannot reliably interpret the intent of the evaluator without manual intervention from the user.

In GND, the actual data must be unambiguous, not subject to interpretation based on details inside the documentation. The structure accomplishes this goal partly by adding more specific meta-data and more flexibility (so that, for example, reactions that would require non-standard definitions of the LIP flag in an ENDF-6 file can be described clearly in GND). GND also supports adding extra information about how data were derived and how they are meant to be used, inside the 'styles' element.

3.1 Using GND 'styles' to store multiple types of data in an evaluation

One of the goals outlined by SG38 is to support storing multiple representations of data within the same evaluation, along with details of how each representation was created. For example, for a reaction cross section an evaluation may store not only the original representation generated by the evaluator, but also Doppler broadened forms representing the same reaction cross section for different material temperatures. It may also be converted to a group-averaged form for use in multi-group transport simulation codes.

Figure 1 shows two different forms ('form #1' and 'form #2') of the cross section for the same reaction. Since multiple forms can be stored simultaneously, GND introduces the 'styles' and 'style' elements in order to differentiate between all the different forms of data. Each style corresponds to one type of data, such as 'evaluated', 'linearized', 'heated to 600 K', 'grouped onto N bins', etc. The style is defined inside the 'styles' section near the beginning of a GND file. The definition of each style includes details about how it was generated, what other style(s), if any, it is derived from, what code(s) were used during the derivation, etc. Each style also includes a unique label that can be referred to from any other point in the file.

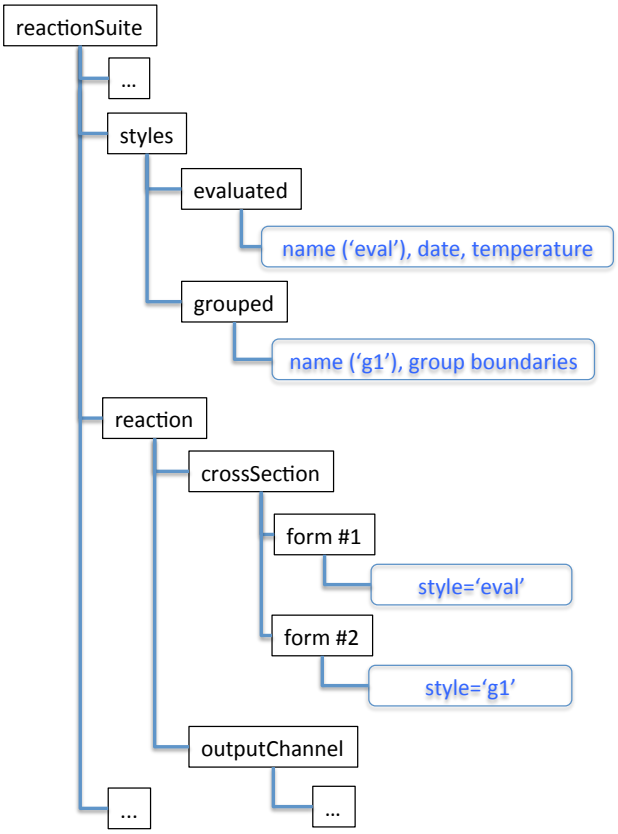

Fig. 2. Example of how 'styles' can be used to store multiple types of data, including evaluated and processed versions, within the same hierarchy. An application code looking for a certain style of data would search for a specific 'style' attribute to determine which form to use.

Figure 2 illustrates how styles are used to store more than one type of data in an evaluation. While the example only shows multiple forms for the cross section data, styles can also be used to store different forms of multiplicity data, distribution data, etc. Users may wish to keep all styles (evaluated and processed) inside the same file, or for performance and space reasons they may wish to create an application-specific library that contains only the processed forms for faster loading.

3.2 Uncertainties/Covariances

Nuclear data covariances have been the focus of increased attention over the past decade. They provide an essential ingredient for uncertainty quantification efforts that

measure the impact of nuclear data uncertainties on applications like criticality safety, stockpile stewardship and global security. While older evaluated nuclear data libraries often omitted uncertainty estimates for most cross sections, library managers are increasingly pushing to include uncertainty estimates for all quantities in the database.

The ENDF-6 format supports storing covariance matrices for cross sections, average multiplicities, angular and energy distributions and resonance parameters. In order to maintain backwards-compatibility, GND also supports nearly all of these options.[1]

GND needs to not only support all types of covariance data that can be stored in ENDF-6 files. It also needs to be flexible enough to quickly add support for other types of covariance. In particular, the nuclear data community is becoming more interested in supporting covariances on double-differential particle distributions, as well as covariances for fission product yields and for thermal neutron scattering data. ENDF-6 currently makes no provision for storing these types of covariance matrix (which can become very high-dimensional). Adding these options to ENDF-6 is expected to be a difficult task, but adding support in GND should be relatively straightforward, in part because GND already includes support for N-dimensional arrays while ENDF-6 only currently supports 2-dimensional arrays (for more detail see section 5).

GND also facilitates storing covariances more compactly. Often, performing an eigenvalue decomposition on a nuclear data covariance matrix produces just a few dominant eigenvalues along with many that are nearly negligible. In these cases, the covariance can be represented to good precision by storing only the principal eigenvalues along with their eigenvectors. For example, if an $M \times M$ covariance matrix C has N eigenvalues that are much larger than the remaining $M - N$ values, only those N eigenvalues and eigenvectors need to be stored. The full matrix can be approximated to good precision: $C \approx \Lambda \cdot \lambda \cdot \Lambda^T$, where Λ is the $M \times N$ matrix composed of the principal eigenvectors and λ is the $N \times N$ matrix with eigenvalues along the diagonal. Support for storing covariances in this fashion can easily be added to GND.

Covariance matrices may appear in several different places within a GND file. As figure 1 shows, the file can contain a 'covariances' section, where all kinds of covariance matrices are stored. If a covariance pertains to a single element (such as a reaction cross section), GND also supports storing it inside that element, either as a full covariance matrix or as a link to the appropriate section inside 'covariances'. Covariances in nuclear data often link two different types of data, such as cross sections for two different reactions or even cross sections for two different target/projectile combinations. In that case the covariance should be stored inside the 'covariances' section, with links pointing to the two data that are being co-varied (links are represented using the xml 'xlink' syntax). The covariances section can be stored inside a reactionSuite, or it may be stored in an external file (particularly if it contains covariances linking data from two different reactionSuite files).

[1] The only exception right now are covariances for LRF=7 resonance evaluations, which until recently have not appeared in any evaluations. Recently, new candidate evaluations have started to include LRF=7 covariances, so adding support for them is now a priority for GND.

4 Properties of Particles

While libraries like ENDF-VII.1, JEFF-3.2 and JENDL-4 are primarily meant for storing nuclear reaction data such as cross sections and distributions for outgoing particles, they also require particle and nuclear structure information to fully describe the reactions and products. Important structure information includes particle masses, spins, excitation energies of nuclear states, and decay properties. Part of the task of SG38 is to define a hierarchy for storing these data, that can be integrated into the reactionSuite hierarchy. This particle database is intended to help simplify how reaction products are stored: rather than having to list properties like the mass every time a particle appears as a reaction product, the hierarchy instead refers to a particle 'id' that can be used to look up the product in the particle database.

Requirements and specifications for a particle database have been collected by SG38 into a document that is available from the SG38 website [3].

5 General-Purpose Data Containers

One of the strengths of the ENDF-6 database structure, that allowed it to be useful for over 50 years, was its reliance on a small set of flexible, general-purpose data containers. These include TAB1, a container for storing 1-dimensional data such as cross sections or average product multiplicities as a list of (x,y) points with interpolation rules. They also include TAB2, which stores higher-dimensional data such as angular distributions $P(\mu|E)$ (where E is incident energy and μ is the scattering cosine), or double-differential distributions for outgoing energy and angle, $P(E', \mu|E)$. In both cases, TAB2 stores a nested list of lower-dimensional curves along with interpolation rules.

SG38 is adopting the same strategy by designing general-purpose low-level data containers that can be reused to store all kinds of data. Data containers supported by GND are outlined in table 1. Some containers are similar to ENDF-6 options, but with some differences. For example, while ENDF-6 supports discontinuities within a single interpolation region, GND only permits them at the boundary between two regions.

6 Infrastructure Needs

A new data storage hierarchy by itself is not enough to replace the ENDF-6 standard. It must be accompanied by a robust code infrastructure to support generating new data, visualization, checking for problems in data formatting and in physics content, and for processing data and

Table 1. General purpose data containers in GND

Name	Description	Closest ENDF-6 equivalent
text	text with optional markup	TEXT
values	list of data	LIST
axes	labels and units for each axis	
XYs	1-d function	TAB1
multiD_XYs	N-d function	TAB2
regions	function with discontinuities and/or multiple interpolation regions	TAB1/TAB2
series	polynomial expansion	
array	N-dimensional array. Can compress sparse data	
gridded	array along with a grid that defines the meaning of each array element	
table	unlike 'array', may contain mixed data types or empty elements	

using them in application codes. For ENDF-6, this infrastructure already exists, so modernizing or replacing it will be one of the major challenges hindering the adoption of GND or any other new structure.

Fortunately, some progress has already been made towards building up this new infrastructure. This includes the code FUDGE, developed alongside GND at Lawrence Livermore National Laboratory [5]. FUDGE can read and write GND files, reconstruct cross sections and angular distributions from resonance parameters, Doppler broaden data to a desired temperature, generate plots, and check the physics content of GND files. A new processing code for generating transfer matrices for use in deterministic transport codes has also been developed and is being integrated into FUDGE.

FUDGE also supports translating ENDF-6 files to and from GND. This capability is expected to be very important in the near term, when both formats are being used to store and share data. The translation code has been tested on multiple ENDF-6 formatted libraries, and has helped uncover bugs both in those libraries and in FUDGE.

Other code support teams have already committed to updating their codes to handle the new structure once the specifications from SG38 are complete. These include the NJOY code from Los Alamos National Laboratory [6] and the AMPX code from Oak Ridge National Laboratory [7].

Progress is also being made towards defining a standard Application Programming Interface or API for reading and sampling data from GND files. This effort has resulted in the 'General Interaction Data Interface' or GIDI, a c++ library that is being integrated into Monte Carlo transport codes at LLNL.

7 Conclusions

WPEC subgroup #38 has undertaken the somewhat daunting task of replacing a 50-year old standard format that is used to supply nuclear data for a wide range of nuclear applications including energy, national security, medical isotope production and basic science research. Although this task will require substantial time and effort, the members of SG38 consider it to be an important step towards modernizing the infrastructure for storing and handling nuclear data, and ensuring the quality of future nuclear data libraries. Once SG38 completes its task of spelling out the requirements and specifications, the GND project will be updated to reflect the new requirements.

Acknowledgments: WPEC is under the auspices of the 31 OECD/NEA databank member countries: `http://www.oecd-nea.org/nea/mcnea.html`

This work was performed under the auspices of Department of Energy contract No. DE-AC52-07NA27344 (Lawrence Livermore National Laboratory). The project was partly funded through the Nuclear Data Program Initiative of the American Recovery and Reinvestment Act (ARRA).

References

1. "ENDF-6 Formats Manual: Data Formats and Procedures for the Evaluated Nuclear Data Files ENDF/B-VI and ENDF/B-VII", ed. A. Trkov and D. Brown, BNL Report BNL-90365-2009 Rev.2, CSEWG Document ENDF-102 (2011). See: `http://www.nndc.bnl.gov/csewg/docs/endf-manual.pdf`
2. Working Party on International Nuclear Data Evaluation Co-operation (WPEC). See: `http://www.oecd-nea.org/science/wpec/`.
3. OECD/NEA/WPEC SubGroup 38, "Requirements for a next generation nuclear data format". See: `https://www.oecd-nea.org/science/wpec/sg38/`
4. C.M. Mattoon *et al.*, "Generalized Nuclear Data: a New Structure (with Supporting Infrastructure) for Handling Nuclear Data", NUCL. DATA SHEETS **113**, 3145 (2012).
5. B.R. Beck, "FUDGE: A Program for Performing Nuclear Data Testing and Sensitivity Studies", AIP Conf. Proc. **769**, 503 (2004).
6. R.E. MacFarlane, A.C. Kahler, "Methods for Processing ENDF/B-VII with NJOY", Nuclear Data Sheets, **111**, Issue 12, 2739 (2010). See: `http://t2.lanl.gov/codes/njoy99/`
7. N.M. Greene, "The AMPX-2000 Operating System for Producing Continuous Energy and Multi-Group Cross Sections from Basic Data Libraries Using the ENDF/B-6 Formats", Proceedings of the International Conference on Nuclear Data for Science and Technology (ND2001), Oct. 7-12, 2001, Tsukuba, Japan.