Various Thoughts Regarding A Generalized Nuclear Data Format

Morgan C White

Los Alamos National Laboratory

Abstract

The successor to the ENDF data format has been discussed many times. Why do we feel that now is the right time or that this time might be different? The following slides offer my personal observations regarding what a generalized nuclear data format might be. At this stage I hold only two concepts sacrosanct. We should minimize the barrier to entry (simply reading data) by using modern, standardized tools (e.g. XML or HDF). Our focus should be on defining the nuclear data *language*, i.e. the structure used for storage and exchange. I believe that now is the right time to do this. As budgets continue to decrease, we need to find better ways to enhance collaborations and reduce the overall cost of improving nuclear data. Taking advantage of the the mature tool set available for handling data makes sense and will open the nuclear data world to a broader community.

Session 0:
Lessons Learned

- **It is a right of passage to learn the ENDF format and write a parser**
  - This is a colossal waste of our resources as we all squander precious time wrestling with file I/O instead of spending time on physical understanding
  - We cannot afford this extravagance in future (nor could we in past)
- **Current ENDF processing codes combine two complicated tasks**
  - The data I/O that is necessary to access the parameters is tricky at best
  - The algorithms to manipulate and use the data that are the meat of the task
  - **This results in code that is overly complex and difficult to decipher**
    - **This problem only grows worse as we seek to modernize and extend capabilities**
- **ENDF-6 can be extended to store any conceivable data set**
  - **Can is not should**
- **Data abstraction seeks to reduce and factor out details**
  - This allows the focus to be on concepts (***physics***), not the form

Session 1:
Purpose of the New Data Structure

- **XML was created to structure, store, and transport information**
  - Stop reinventing the wheel and integrate with standard tools where possible
  - XML (or similar meta languages) provide tools to define a nuclear data language
    - It is not XML that is important, it is abstracting the data into better structures
- **Data abstraction seeks to reduce and factor out details**
  - XML, HDF and other tools provide building blocks that enable data abstraction
  - This allows the focus to be on concepts (***physics***), not the form
    - When was the last time you heard another community argue how to achieve 8 sigfigs
    - Or whether we really need to keep a 5 digit line number left over from punch cards
- **Our focus should be to structure, store and exchange nuclear data**
  - ENDF-6, ENSDF, EXFOR, RIPL… provide the lessons learned, time to move on

Like Rodin's Fallen Caryatid, the ENDF format is being crushed by a load it was never designed to carry.

Session 2:
Nuclear Data System Overview

- **The ENDF format is a story of the good, the bad and the ugly**
  - Best testament to the ENDF format is that it has flourished over 5 decades
  - We must heed the lessons learned, good and bad, in going forward
- **The story of ENDF is one of expert judgment and pragmatic choices**
  - Some may argue that storage of the evaluation code parameter is adequate
  - However, expert judgment is often **necessary** due to model deficiencies
    - e.g. the EMPIRE evaluation code allows point-by-point *adjustments*
- **A new format should go further to be inclusive and broadly applicable**
  - There is an increased desire for interoperability of applications
  - Using common structures offers further enhancements in the use of our data
    - e.g. co-plotting evaluated and experimental cross section data
    - e.g. sharing nuclear structure data between RIPL and application sampling
    - e.g. storing 'master' and derived (processed or simplified) data in a single container
  - Eventually (not all at once) such a system should replace ENDF, ENSDF, EXFOR, RIPL …
- **We must support translation back-and-forth to ENDF-6 for some time**

Session 3:
Benefits & Requirements for Data Evaluation and Processing

- **Design structures that can accommodate multiple representations**
  - And hold multiple interpretations of them simultaneously
- **Single container can contain code inputs, evaluated and derived data**
  - Easy to tag and distinguish *approved* versus derived data
- **Significantly improve the traceability and transparency**
  - Annotations integral to the data can better document the who, what, when, why…
- **Significantly improve the version control and inter-comparisons**
  - A recurrent user complaint is the inability to compare or use others data
  - Easy inter-comparison also enables better version control (i.e. has it changed?)
- **Like all good EMPIRES, the goal is global domination**

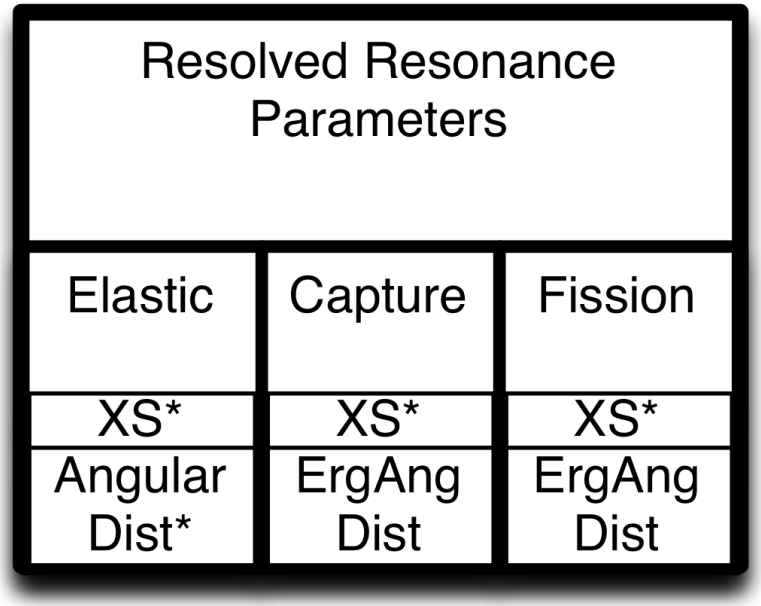
Session 4:
Format Perspective and Organization

- **Focus on defining the data structures**
  - It is time to rethink, using the lessons learned, how best to organize our data
  - Implementation in XML, HDF or another form should be a secondary objective
- **Allow for grouping by the source of the evaluation**
  - The resonance parameters natural define a collection of reactions
  - Nested data allow easier representation of reactions to discrete levels
- **Allow for more than one representation across more than one level**
  - Define the “master” representation and provide history (linkages) of derived data
  - Imagine saving the evaluation code input used to generate the data
    - Improving transparency and history of data evaluation
    - And yet still having the freedom to tweak data where needed
  - This naturally supports including derived (processed) data in the same file
  - Data for applications can be tailored to with trade-offs for speed versus fidelity

Notes on Format Perspective and Organization

This proposal is a significant break from the ENDF legacy. Oddly, it comes from observations of the use of our data in the MCNPX code. This application code uses multiple data tables and event generators to define the interactions for a particle over a range of energies. For example, neutrons colliding with hydrogen can use the FLUKA event generator at energies above 800 MeV, the CEM event generator from 150 to 800 MeV, the ACE 1001.70c data table from 10 eV to 150 MeV and the ACE lwtr.10t data table from 1E-5 to 10 eV. A depiction of what this layered approach might look like as a data abstraction is given on the following slides. Its key advantage over the legacy approach is to cluster related information as closely as possible within the structure. Recent studies have shown the importance of emission distributions, particularly the elastic and inelastic reactions. When data are disparate within the structure, e.g. MF3 / MF4, it is easy to update the cross section and leave the old emission data. The proposed structure discourages this by making such an update more transparent. It also offers the opportunity to store and use data at a level best suited to a given application.

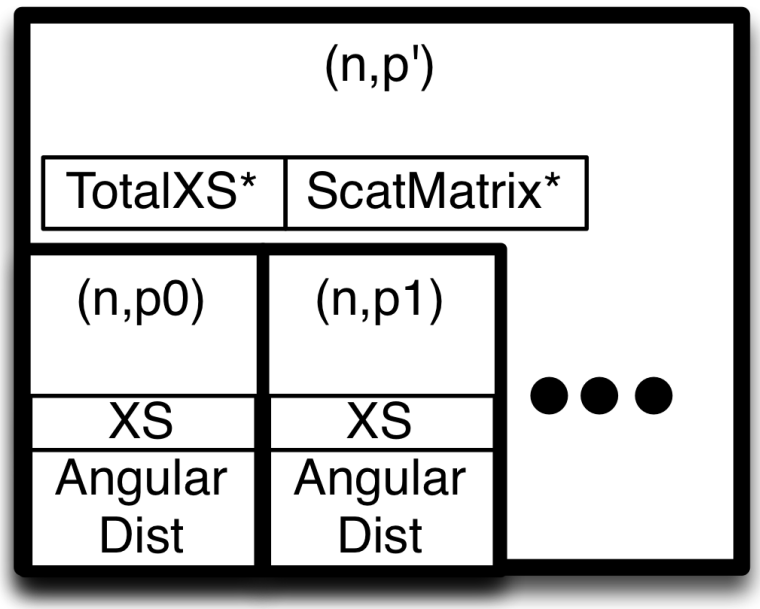
Potential Data Containers
Resonance Range Data



- **RR parameters are *master* data**
- **All XS data are *derived***
- **Elastic angular distribution are *derived***
- **Capture and fission contain embedded *master* energy-angular distribution**

A nested data structure can be used to store resonance parameters and their derived data yet still allow embedded master data, e.g. fission PFNS, where needed

Potential Data Containers
Scattering to Discrete Levels



- **Explicit data for each level can be given for *any* reaction**
- **Partial cross section and angular data can be marked as primary**
  - Total cross section and scattering matrix are derived
- **Applications can use data as needed**
  - Continuous-energy neutron transport code only needs total (n,p)
  - Continuous-energy proton transport code needs partials
  - Sn transport needs total grouped data

Session 5:
Requirements for Basic Data Structures

Complexity built from simplicity

- **The strength of the ENDF format has been its simplicity**
  - Underneath the hood, there are only **5** basic elements
    - **TEXT**, **CONT** (2 real, 4 int), **LIST** (float vector), **TAB1** (xy vector with interpolation), and **TAB2** (interpolation) records
  - Everything else is built upon these foundation elements
    - The TAB1 is the bread-and-butter widely used in all parts of the format
- **REQ: Minimize the number of basic elements**
  - Build complexity using hierarchical constructions of these basic elements
  - KISS – Keep It Simple Stupid
  - Avoid exceptions (if only we had…)
- **Given that this problem has been studied and implemented many times (program languages, operating systems, data storage languages, …) consider these sources rather than reinventing the wheel**
  - Many XML ***languages*** already define basic elements we will need

Purpose and Goals for SG38

Vision

To define the international standard format for storing and exchanging all aspects of nuclear data for the 21st century.

Nirvana is not always what you expected