Status Report on WPEC SG38 Format Development

CSEWG Formats and Processing Committee

November 3, 2015

Caleb M Mattoon

Lawrence Livermore National Laboratory

LLNL-PRES-XXXXXX

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

WPEC Subgroup 38: design a new format for storing nuclear data, and help start the transition from ENDF-6 to the new format.

- Proposed 2012 by D. McNabb (LLNL), meetings twice/year since.
- Main deliverable: documentation on the *requirements* and *specifications* for the new hierarchy. Includes
  - outlining how data are evaluated, processed and used,
  - defining initial data types that new format must support, and
  - specifying how data will be organized.

Draft SG38 requirements documents are complete, now serve as guides for writing specifications

- Three draft requirements documents:
  - General-purpose data containers (analogous to TAB1, TAB2, etc. containers in ENDF-6),
  - Top-level hierarchy (how data are organized inside an evaluation and/or library), and
  - Particle properties (for storing reaction-independent quantities like spins, masses, excited states and decay data)
- Requirements still subject to change as SG38 refines specification docs

Specifications documents are partially complete.

- Particle library specs appear to be complete
    - Sufficient for reaction evaluations, but likely not for ENSDF-style database
- General data containers and top-level hierarchy: still disagreement about how best to proceed (more details coming up)

SG38 seeks to balance potentially competing goals:

- ‘Avoid redundancy (and potential for discrepancies) by storing data only once’

  vs

- ‘Store related data together, organized to make sense from a physics point of view’

Best solution: use explicit links to associate data in different parts of a file (or files)

SG38 seeks to balance potentially competing goals:

- ‘Make a more flexible format, adding new types of data as evaluator capabilities and user needs evolve’

vs

- ‘Simplify how data are stored, using a consistent layout and a small set of general-purpose containers rather than many special cases’

Best solution: still TBD

LLNL is also working on Generalized Nuclear Data (GND), parallel effort that will merge with SG38 recommendations

- GND project started a few years before SG38.
  - First conceived as a way to meet needs of LLNL users
  - Concept grew to include easier data sharing between institutions, and then to new international standard.
  - Went to WPEC for international review and feedback
- FUDGE code provides GND infrastructure (including translation to and from ENDF-6)
- GND is evolving quickly in response to SG38 feedback.
  - More testing, feedback and contributions welcome!

Status of ENDF translation to GND

- For ENDF-VII.1 sub-libraries:
  - Can translate:
    - neutrons/ protons/ deuterons/ tritons/ helium3s/ gammas/ standards/ electrons/ photoat/ atomic_relax/

Evaluations “H1 + H2” and “H2 + H3” have data problems

  - Can be translated into GND-like format, these are not yet integrated into FUDGE:
    - nfy/ sfy/ thermal_scatt/
  - Currently not supported:
    - Decay/

Reactions involving same target/projectile are collected together in a ‘reactionSuite’:

```
<reactionSuite projectile=“...” target=“...” formatVersion=“...” ...>
    <styles> e.g. ‘evaluated’, ‘reconstructed’, ’processed’ </styles>
    <documentations> support ascii, html, etc.</documentations>
    <aliases>meta-stables are aliases to an excited level</aliases>
    <particles> ground state, levels, gammas ... </particles>
    <resonances> resolved and/or unresolved ... </resonances>
    <reactions>all reactions that sum to ‘total’</reactions>
    <sums>summed cross sections, multiplicities</sums>
    <productions>for radio-isotope production</productions>
    ...
</reactionSuite>
```

color code:

element   attribute   comment

Reactions involving same target/projectile are collected together in a ‘reactionSuite’:

```
<reactionSuite projectile=“...” target=“...” formatVersion=“...” ...>
    <styles> e.g. ‘evaluated’, ‘reconstructed’, ’processed’ </styles>
    <documentations> support ascii, html, etc.</documentations>
    <aliases>meta-stables are aliases to an excited level</aliases>
    <particles> ground state, levels, gammas ... </particles>
    <resonances> resolved and/or unresolved ... </resonances>
    <reactions>all reactions that sum to ‘total’</reactions>
    <sums>summed cross sections, multiplicities</sums>
    <productions>for radio-isotope production</productions>
    ...
</reactionSuite>
```

recent added layers to improve organization

color code:

element attribute comment

GND organizes data in a hierarchy. Most data are stored inside ‘reaction’ elements:

```
<reactions>
  <reaction label=“...” date=“...” ENDF_MT=“...”>
    <crossSection> ... </crossSection>
        <!-- options: XYs, regions, resonancesWithBackground, ... -->
    <outputChannel genre=“twoBody, nBody, etc.”>
        <Q> ... </Q>
        <product name=“...” label=“...” >
            <multiplicity>...</multiplicity>
            <distributions>...</distributions></product>
        <product>...</product>
        ...
    </outputChannel>
  </reaction>
</reactions>
```

color code:

element attribute comment

GND organizes data in a hierarchy. Most data are stored inside ‘reaction’ elements:

```
<reactions>
 <reaction label=“...” date=“...” ENDF_MT=“...”>
  <crossSection> ... </crossSection>
    <!-- options: XYs, regions, resonancesWithBackground, ... -->
  <outputChannel genre=“twoBody, nBody, etc.”>
    <Q> ... </Q>
    <product name=“...” label=“...” >
      <multiplicity>...</multiplicity>
      <distributions>...</distributions></product>
    <product>...</product>
    ...
  </outputChannel>
 </reaction>
</reactions>
```

No more ‘nativeData’

color code:

element attribute comment

GND organizes data in a hierarchy. Most data are stored inside ‘reaction’ elements:

```
<reactions>
 <reaction label=“...” date=“...” ENDF_MT=“...”>
  <crossSection> ... </crossSection>
    <!-- options: XYs, regions, resonancesWithBackground, ... -->
  <outputChannel genre=“twoBody, nBody, etc.”>
    <Q> ... </Q>
    <product name=“...” label=“...” >
      <multiplicity>...</multiplicity>
      <distributions>...</distributions></product>
    <product>...</product>
    ...
  </outputChannel>
 </reaction>
</reactions>
```

No more ‘nativeData’

Is special markup needed for delayed neutrons, gammas, P(v), etc.?

color code:

element attribute comment

Some points where SG38 members still need to reach consensus:

- Element names: ‘reactionSuite’ vs. ‘evaluation’ vs. ‘material’ vs. ‘MAT’...
- How many special cases to support (product example from previous slide)
- Are the data containers general enough? More options needed?
  - Functions with multiple dependent values?
  - Support storing x-y data either as pairs (x_1,y_1,...x_n,y_n) or as separate vectors (x_1,...x_n) and (y_1,...y_n)?

SG38 has been extended into 2016. Main deliverable (requirements and specifications docs) will be complete, but other tasks remain

- New subgroup proposal to focus on infrastructure, API, quality assurance
- Long-term subgroup proposal to oversee the new format, respond to requested format modifications

Despite some head-butting, SG38 is converging on final specifications!

Lawrence Livermore
National Laboratory

Security question: how do we protect against malicious code in xml-formatted evaluations?

- ENDF-6 is *not* inherently safe: example of malicious .endf file (requires file be executable)

```
: 'ev:: 427      $  $Date:: 2011-10-17#$                             1 0  0    0
 5.010000+3 9.926921+0          0          0          0          0 525 1451    1
 0.000000+0 0.000000+0          0          0          0          6 525 1451    2
 ...
                                                                   525 1451  131
   mt=151  effective scattering radius = 4.129038-13 cm            525 1451  132
'; echo 'insert malicious code here'; #                            525 1451  133
exit                                  #                            525 1451  134
```

- XML files also open up security concerns, but xml comes with valuable tools (schema, DTD) for validating data
  - First step: disallow xml ‘macros’

Status of translation for other libraries (incident neutrons only):

- JEFF-3.1.2
  - 328 / 381 successfully translated
  - 371 / 381 with the ‘skipBadData’ option
- JENDL-4
  - 385 / 406 successfully translated
  - 406 / 406 with skipBadData
- We are submitting bug reports to library maintainers