21st century nuclear data – a UK perspective

Lee Morgan & Jean-Christophe Sublet

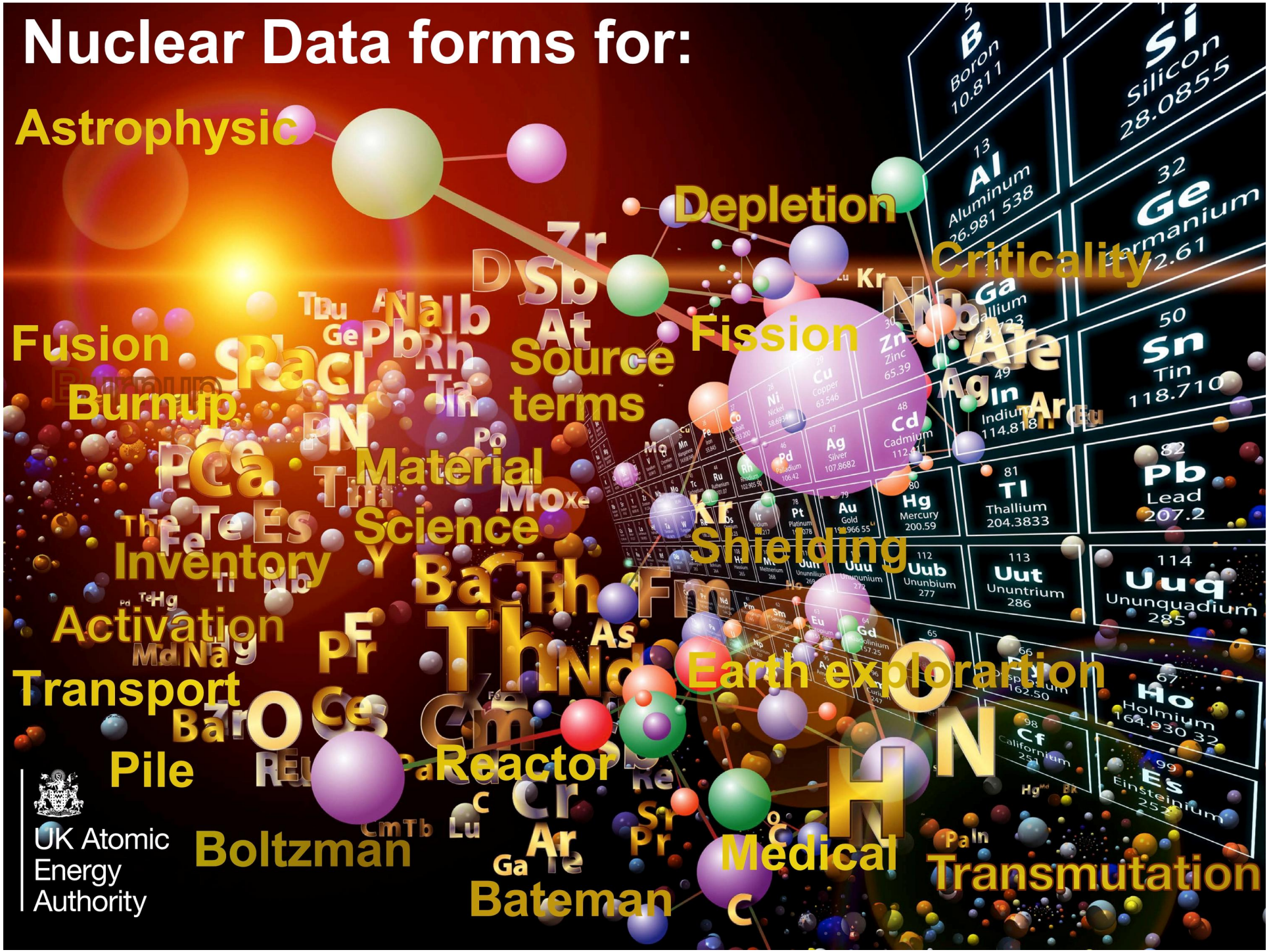
CCFE is the fusion research arm of the **United Kingdom Atomic Energy Authority.**
This work was funded by the RCUK Energy Programme [grant number EP/I501045] .

Overview

- Keeping the options open on nuclear informatics
- Delivering numerics for all nuclear sciences
- Future-proofing
- Exploring/Optimising data formats
- GND has rightly focused (so far) on *form* but we consider the *implementation* in this presentation

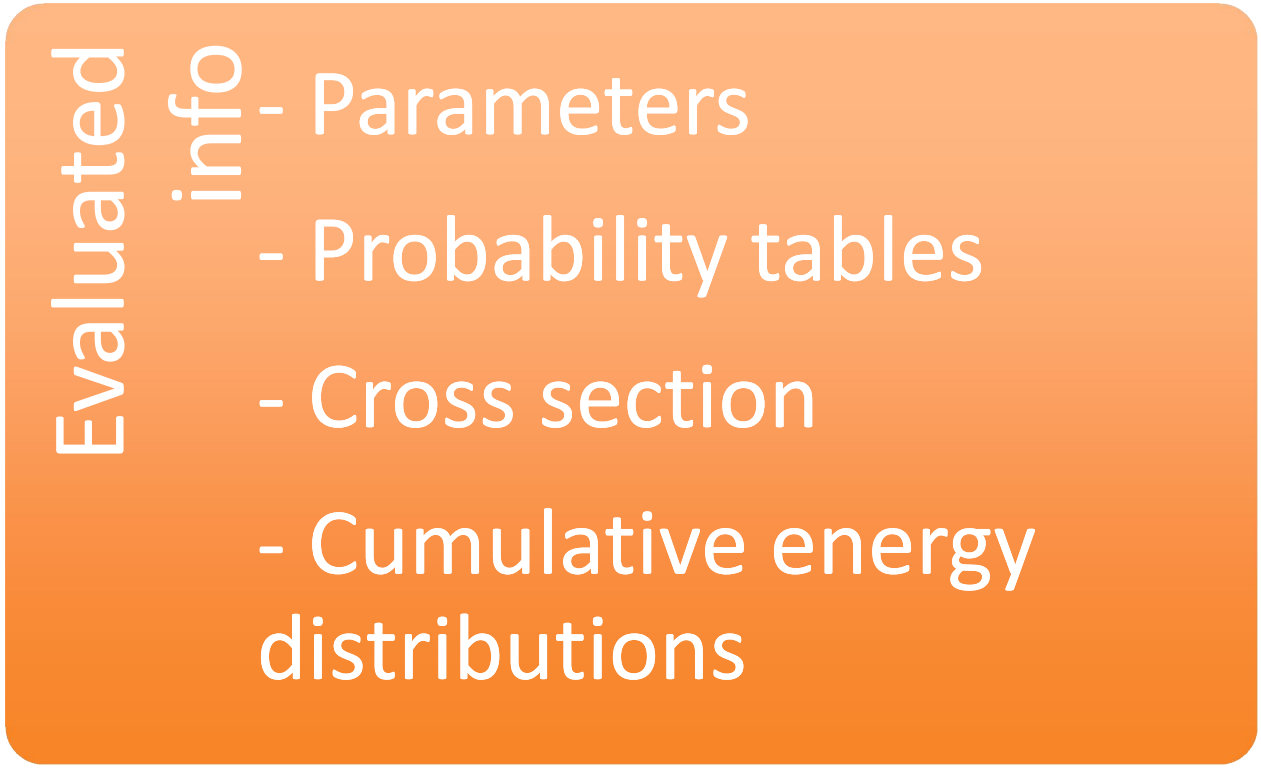
Nuclear data forms

- Nuclear data for nuclear physics are not fundamental constants, nor standard but derived physical quantities: elusive but they have to be robust
- Stems from databases; AME, RIPL, Periodic table, atlas of resonances,… models and EXFOR
- Different forms (resonance widths), format (ENDF-6) and formalism (Reich Moore), but store everything
- There are many data forms to feed the many applications, too many..
- GND need to deliver rich, interconnected type of nuclear informatics to the many multi-physics applications that needs them


Nuclear Data forms for:
Astrophysic
Depletion
Criticality
Fusion
Fission
Source terms
Burnup
Material
Science
Shielding
Inventory
Activation
Earth exploration
Transport
Pile
Reactor
Boltzman
Medical
Transmutation
Bateman
UK Atomic Energy Authority

Nuclear Data forms

- Evaluated informatics; a subset



- Already used in the EASY-II multi-particles, systems agnostic simulation tool, still applying an ENDF-6 dictionary framework

GND data storage

- "GND defines the structure needed for storing nuclear data evaluations and the type of data that needs to be stored. But unlike ENDF and ENDL, GND does not define how the data are to be stored in a file."

  Generalized Nuclear Data: A New Structure (with Supporting Infrastructure) for Handling Nuclear Data. C.M. Mattoon , B.R. Beck, N.R. Patel, N.C. Summers, G.W. Hedstrom, D.A. Brown. Nuclear Data Sheets , Volume 113, Issue 12, December 2012, Pages 3145–3171

- GND as the "lingua franca" of nuclear data languages
  - For speed and ease of use, other methodologies should be used to read and write data when processing data and reading data in to codes such as Monte Carlo Transport, Burn-up etc.

- First implementation of GND, using XML is an excellent beginning..
  - However, other storage techniques need to be explored.

XML : Pros & Cons

Pros

- Good for generic data interchange
- Human readable
- Open, international standard
- Mature technology
- Popular and well supported

Cons

- XML does not scale well
- Over-engineered and verbose – especially for large data storage
- Parsers are slow and complex

XML usage

“XML is not a database. It was never meant to be a database. It is never going to be a database. Relational databases are proven technology with more than 20 years of implementation experience. They are solid, stable, useful products. They are not going away. XML is a very useful technology for moving data between different databases or between databases and other programs. However, it is not itself a database. Don't use it like one.“

“The physical storage model of an XML document is very close to the mental model most developers have of it…..

No effort I spent on optimization. Thus, storing all your data inside XML documents instead of in a relational database is almost certainly going to kill performance on an application of any reasonable size.”

Effective XML: 50 Specific Ways to Improve Your XML by Elliotte Rusty Harold. Page 230 - 231

XML usage

"I do not suggest using a native XML database for data that does fit well into tabular structures. In practice, relational databases like Oracle, FileMaker pro, and MySQL are far more reliable, much better supported, and easier to use and administer; and they perform far better than a typical native XML database. More than anything else this reflects the relative maturity of relational databases (more than 20 years) and native XML (less than 5 years).

Effective XML: 50 Specific Ways to Improve Your XML by Elliotte Rusty Harold. Page 233

i.e.

- **XML should not be used for data storage**
- **Should be used for program to program interoperability**

Alternatives to XML

- When the ENDF format was created (from the UKNDF in the 70's) there were very few options available in terms of technology.
- Nowadays, designing the GND format, many options are available. (XML, YAML, JSON, RDBMS, TOML, NoSQL, HDF-5, Google Protocol Buffers, etc)
- In order to ensure the GND format lasts as long as its predecessor, ENDF, we shouldn't converge on a data format quickly.
- The content of the database can be decided, but the way in which is it to be stored must be tested in order to achieve the best performance now and in the future.
- The future of most software is web, cloud or server-based. Which of the data formats are web/server compliant/efficient?

Relational Database Management System (RDBMS)

- Databases scale well – There' s a good reason why large multinational companies, with massive data storage needs, use relational databases and not XML for data storage.
  - YouTube, PayPal, Google, Facebook, Twitter, eBay, CISCO, Adobe, Amazon, Walmart, Dropbox…..)

The needs of these large companies are similar to the needs of the nuclear industry.

- Maintainability
- Reliability
- Flexibility
- Low latency
- Platform agnostic

Relational Database Management System

Benefits:

- Mature technology (a lot older than XML)
- RDBMS normalization techniques enforce data integrity better than XML
- Efficient, packed data storage.
- Fast, constant time to retrieve data, wherever the data is positioned in the database.
- Standard data access via SQL (Subject Query Language)

Implications:

- If GND is to feed the RDBMS in the future, then care must be taken to ensure the hierarchal data structure associated with meta languages, such as XML, are compatible with the relational structure of databases
- The relational database design should be developed along-side the XML schema.

Relational database & XML comparison

XML

```
<particles>
   <particle name="gamma" genre="photon" mass="0 amu" transportable="true"/>
   <particle name="n" genre="nucleus" mass="1.00866491574 amu" transportable="true"/>
   <particle name="H1" genre="nucleus" mass="1.00866491574 amu" transportable="true"/>
   <particle name="H2" genre="nucleus" mass="2.01410177785 amu" transportable="true"/>
   <particle name="H3" genre="nucleus" mass="3.01604927767 amu" transportable="true"/>
   <particle name="He4" genre="nucleus" mass="4.00258411863947 amu" transportable="true"/>
   <particle name="N15" genre="nucleus" mass="14.9998559619695 amu"/>
   <particle name="N16" genre="nucleus" mass="16.0065035478781 amu">
</particles>
```

RDB

Table: particles	
particle	string
genre	enum
mass	double
mass unit	enum
transportable	bool

Conclusions

GND should:

- encourage the evaluator to go further, bridge the gap between raw and applicable forms: gives the pendf@294K
- reduce, clarify and alleviate the processing steps
- enforce a strict code of conduct
- allow hierarchical layers: parameters, pointwise(s), groupwise(s); total and partials channels
- Investigate complementary data storage methodologies, such as a relational database, on many platforms, such as the web.
- The UK Atomic Energy Authority is investing manpower to support GND
- Technologically derived libraries should guide GND form