Lessons from ENDF, EXFOR and other formats

Viktor Zerkin

International Atomic Energy Agency, Nuclear Data Section

WPEC Subgroup-38 Workshop 2012

OECD Nuclear Energy Agency, Issy-les-Moulineuax, France

Paris, 29-30 November 2012

Topics:

1. EXFOR vs. ENDF summary of differences

   EXFOR
   - Structure of EXFOR file and logic of compilation
   - NDS EXFOR system overview

2. Lessons from EXFOR

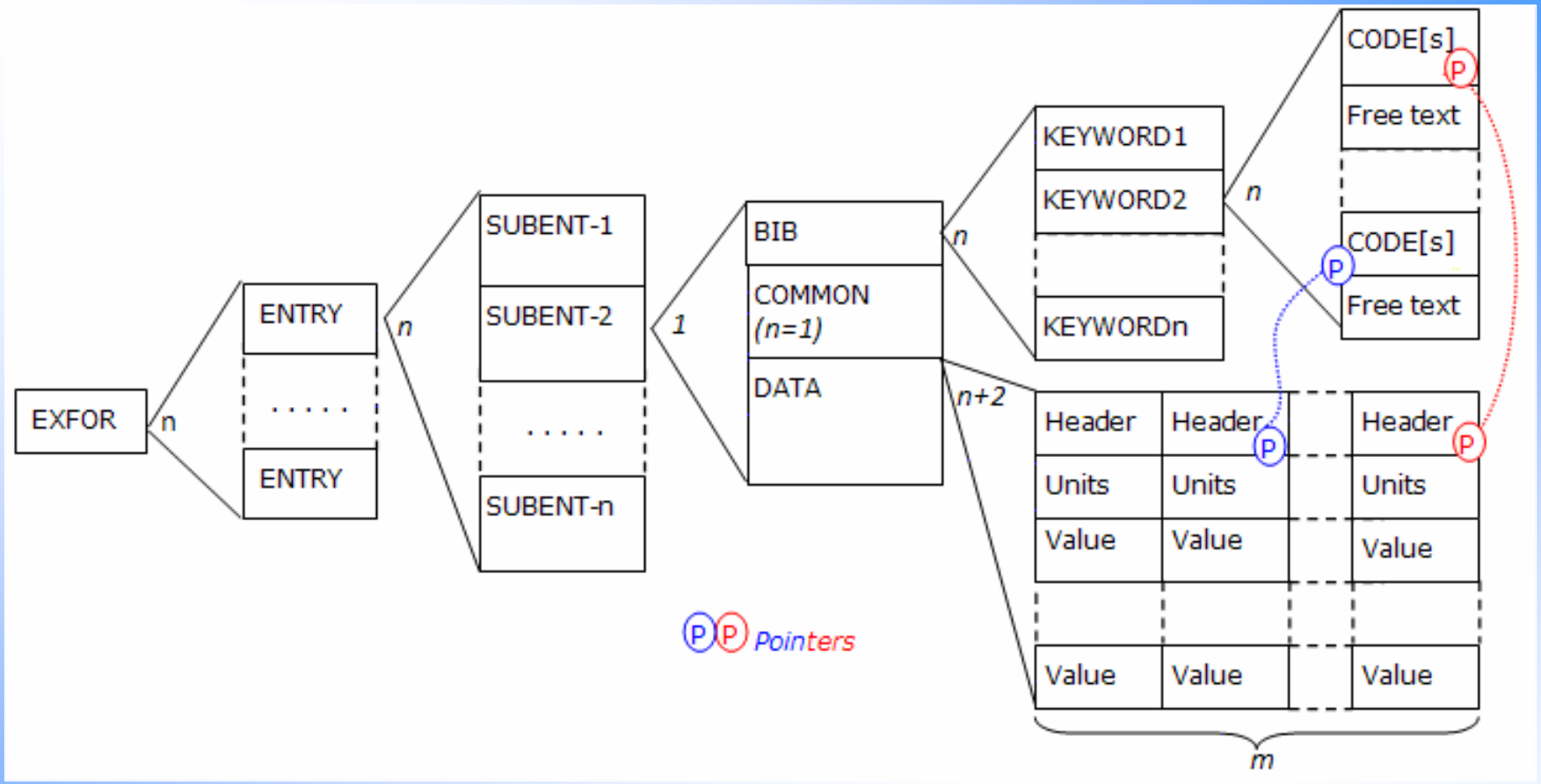
   ENDF
   - How ENDF format/files are used
   - Tasks of NDS ENDF system
   - NDS ENDF system overview

3. Lessons from ENDF
4. Lessons from CINDA
5. Lessons from DOC and DOCX
6. Lessons from HTML
7. Concluding remarks

1. EXFOR vs. ENDF: summary of differences

	EXFOR	ENDF-6
Committee	**NRDC** (International)	**CSEWG** (USA)
Publication files	Regulated by formal protocol Exchange between Centers	No international co-ordination Issuing libraries on national level
	Human readable files	Computational format (easy for Fortran)
Files	ASCII-7, EOL: CR/LF, LF, Line: 80 columns, fixed width real numbers (11 columns)	
Contents	Files, Manual, Dictionaries	Files, Manual, no concept of Dictionaries
Libraries	One common library EXFOR No other libraries in EXFOR format	Many (43 in the IAEA-NDS database) No common library
Produced	By human (manually)	Generated by programs
Easy	To compile data (structure=paper)	To read data by Fortran program
Efforts	to produce computational output	to produce interpreted output, to understand (interpret) simple reading text (human), to edit data manually
Data	Header/Units/Values (Formal description: reaction -> dictionary)	Values only in basic units (description: manual)
Arrays	Common1/CommonN/Data a[n][m] with a_{ij}=<null>	LIST, TAB1, TAB2, etc. a[n]{b[m_n]} a_{ij} is always defined
Basic hierarchy/ problems	Trans/Entry/Subentry/Common/Data (Vector-Common)	NSUB/MAT/MF/MT Mixture: MF10, MF40, MF33 No concept of product in the basic hierarchy (implemented in subsection, sub-sections)
Structures/ Coding	Nested text Keyword/Code param. by position	MAT/MF/MT/ZA/Numerical flags Numbers on fixed positions (manual)
Comments	Free text; can be single or multi-line	Only in MF1/MT451 No concept of comment of a dataset (read array by one Fortran operator '6e11.0')
Cross usage	C4 => evaluating software	MF3,33 to EXFOR data renormalization

Structure of EXFOR file and logic of compilation



This structure does not really need essential revision, because it is very close to the structure of information in the traditional articles (bibliography info, common parameters, data tables), and therefore it makes compilation process natural (helping to minimize number of errors) and simplifies cross checking process by other compilers.

Logic of EXFOR file, concept of Keywords and Codes

```
TRANS → {ENTRY} → {SUBENT} → BIB      → {KEYWORD → {({CODE}) free text }}
                             COMMON   → {HEADER UNITS value}
                             DATA     → {HEADER UNITS {values}}
```

Logical nesting of information blocks in EXFOR file

```
TRANS   ID
    ENTRY   ID
        SUBENT   ID
            BIB
                KEYWORD1    (CODE1, CODE2, ...) Free text
                                . . .
                            (CODE3) Free text
                KEYWORDn    Free text
            ENDBIB
            COMMON
                . . .
            ENDCOMMON
        ENDSUBENT
        SUBENT   ID
            BIB
                KEYWORD1    (CODE1, CODE2, ...) Free text
                                . . .
                KEYWORDn    Free text
            ENDBIB
            DATA
                . . .
            ENDDATA
        ENDSUBENT
        SUBENT   ID
            . . .
        ENDSUBENT
    ENDENTRY
    ENTRY   ID
        . . .
    ENDENTRY
ENDTRANS
```

Basic rules

1. keywords start from position 1 in BIB section
2. no more than 1 keyword of a given type is allowed in a BIB section
3. codes start from the symbol “(“ in position 12
4. positions 1 to 10 under the current keyword in the following lines must be blank
5. position 11 can be used for Pointers
6. codes inside () can have several values
7. code can have parameters
8. etc. . . .

EXFOR Logic

```
ENTRY            41323   20050902
  SUBENT        41323001   20050902
    BIB                 7         12
      INSTITUTE  (4RUSMIF)
      REFERENCE  (J,AE,50,(5),350,198105) MAIN REFERENCE, DATA ARE GIVEN
                 (J,SJA,50,325,1981)  ENGLISH TRANSLATION
      AUTHOR     (V.E.ZHITAREV,A.M.MOTORIN,S.B.STEPANOV)
      TITLE      .INTERACTION CROSS SECTIONS OF CERTAIN METALS
                  WITH COLD NEUTRONS
      FACILITY   (REAC)
      ERR-ANALYS (EN-ERR)   WAVE-LENGTH RESOLUTION DELTA-LAMBDA/LAMBDA
                            TIMES 100 (IN PERCENT)
      HISTORY    (19981121C)  + +  COMPILED AT THE CJD + +
                 (20050902A)  . .  Corrected at the CJD + +
                               Data-heading "EN" changed to "WVE-LN"
    ENDBIB              12
    COMMON               3          3
      EN-ERR     TEMP       TEMP-ERR
      PER-CENT   DEG-C      DEG-C
         3.        22.        3.
      ENDCOMMON           3
    ENDSUBENT           19
  SUBENT        41323002   20050902
    BIB                 5          8
      REACTION   (13-AL-27(N,TOT),,SIG)
      SAMPLE     .ALUMINIUM MONOCRYSTAL, PURITY 99.99 PC, THICKNESS
                  96 MM, DENSITY 2.70 GRAM/CM3 AND
                  MACROCRISTALLINE ALUMINIUM, PURITY 99.99 PC,
                  THICKNESS 50 MM, DENSITY 2.70 GRAM/CM3
      ERR-ANALYS (DATA-ERR) NO INFORMATION GIVEN
      STATUS     (TABLE)  DATA ARE TAKEN FROM TABLE 1 OF MAIN REF.
      HISTORY    (19981121T)  + + CONVERTED FROM SUBENT 88023002
    ENDBIB              8
    NOCOMMON            0          0
    DATA                3          8
      WVE-LN     DATA       DATA-ERR
      ANGSTROM   B          B
       1.3000E+01 1.9300E+00 1.3000E-01
       1.4000E+01 2.1200E+00 9.0000E-02
       1.5000E+01 2.2500E+00 8.0000E-02
       1.6000E+01 2.3800E+00 7.0000E-02
       1.7000E+01 2.5400E+00 6.0000E-02
       1.8000E+01 2.6100E+00 6.0000E-02
       1.9000E+01 2.8200E+00 8.0000E-02
       2.0000E+01 3.1500E+00 6.0000E-02
    ENDDATA            10
  ENDSUBENT          23
ENDENTRY            2
```

Concept of EXFOR Codes

Meaning of a parameter is usually defined by its position in the code

REFERENCE (J,AE,50,(5),350,8105)

How to understand meaning of parameters:

- read EXFOR manual,
- search in EXFOR Dictionaries

Meaning:

Reference	
J	Journal
AE	Atomnaya Energiya (Russia)
50	Volume 50
(5)	Issue 5
350	Page 350
8105	May 1981

Alternatively, parameters could be defined by the pair *{name = value}*, than position is not important anymore, easier to make parameters optional. Modified EXFOR:

REFERENCE (type=J, code=AE, vol=50, issue=5, page=350, date=8105)

How to understand meaning of parameters:

- intuitively (otherwise - read EXFOR manual)
- search in EXFOR Dictionaries

XML:

```
<REFERENCE Type="J" Ref="J,AE" Vol="50" Issue="5" page="350" Year="1981" Code="J,AE,50,(5),350,8105"
expansion="Jour: Atomnaya Energiya, Vol.50, Issue.5, p.350 (1981), Russia"></REFERENCE>
```

Implemented in NDS Web Retrieval system (X4.xml)

EXFOR format problems/solutions/lessons

1. Reaction-code longer then 55 symbols (SF7)
   Solved by introducing new rules in and codes in Dictionary
2. Defining links to Web Journals from Reference code
   Solved by introducing new Dictionary

---

3. Coding of DOI and NSR-Keyno for References
   Solved by introducing new rules in Free text
4. Coding of Covariance data
   Solved by introducing new rules in Free text

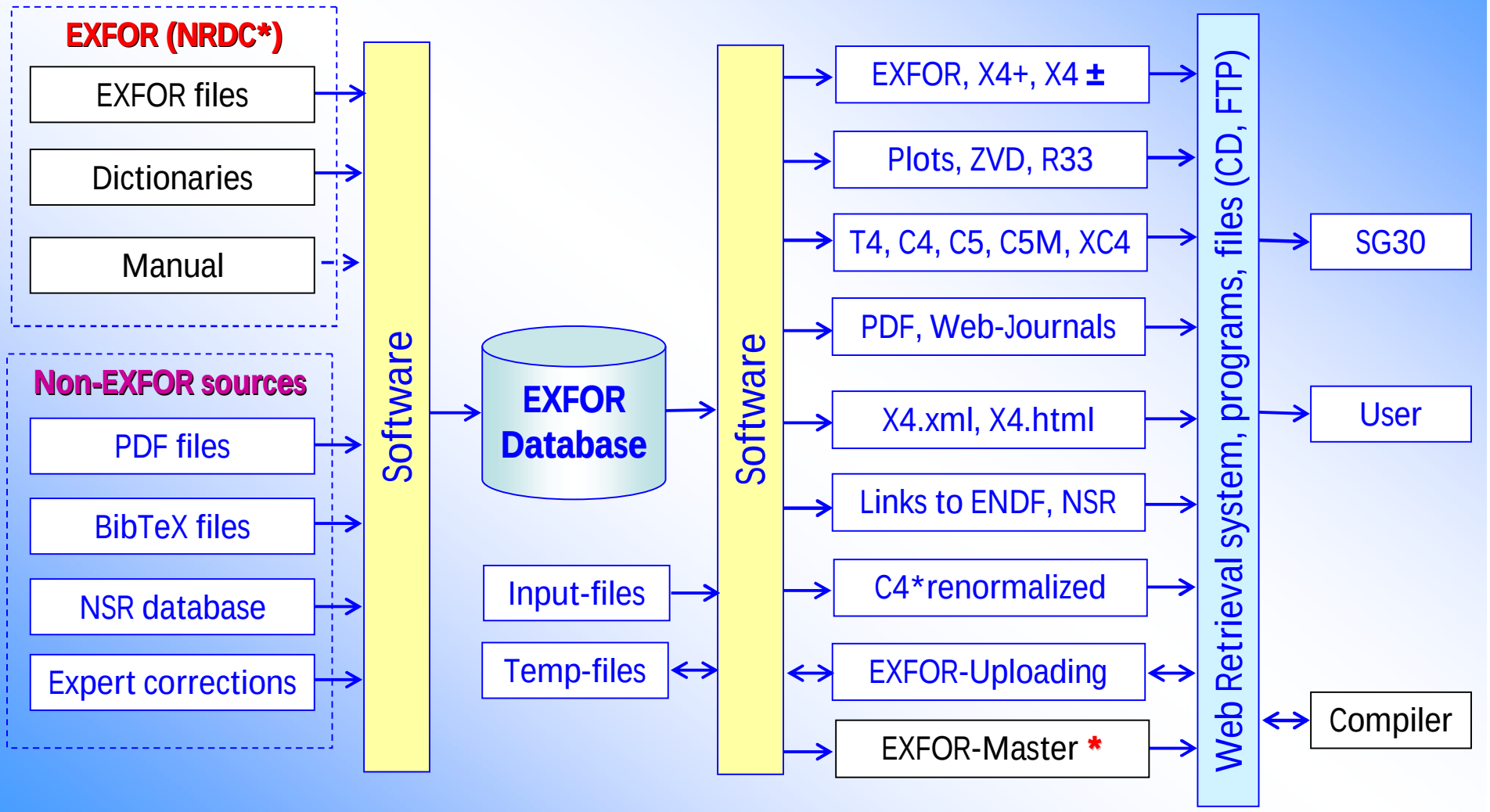
---

5. Storage of Title/Authors of all references
   Solved for Web users in NDS EXFOR Database by import from NSR and Web
6. Storage of original publications - PDF files (binary files)
   Solved in NDS EXFOR Database, available for Web users

---

7. Compatibility with ENDF
   Solved by software with additional dictionaries (x4toc4, x4toc5, etc.)
8. Coding: connection between codes in EXFOR file and dictionaries
   Solved for compilers by software (EXFOR-Editor)
   Solved for Web users by EXFOR Retrieval system in Interpreted output
9. Output formats for users
   Solved by software depending of user needs in output formats:
   Web: X4+, X4±, T4, R33, ZVD, C4, XC4, C5, C5M, X4.xml, X4.html, X4out, X4out.xml
   For SG30 members: XC4, XC5
   For Endver/GUI users and Mirror sites: NDS EXFOR Database (MySQL)
10. EXFOR data renormalization
    Solved by software under Web Retrieval system using NDS EXFOR Database

NDS EXFOR System Overview



Experience from EXFOR system

1. Dictionaries make system more stable (codes are independent from data and programs)
2. Last extensions of EXFOR format were done via Free text
3. Extensions of EXFOR system were done using external files
    1. *import from NSR (Title, Authors, DOI, Keyno)*
    2. *import from Web (PDF, BibTeX) and photocopies (PDF)*
    3. *generated by Auto-renormalization system, provided by experts (.Renorm)*
4. EXFOR relational Database contains more than basic EXFOR files:
    1. *includes EXFOR files and dictionaries as initial data source*
    2. *explicitly presents parameters (hidden information regulated by Manual)*
    3. *container for various extensions of EXFOR system*
    4. *provides effective storage and search*
5. EXFOR Software implements all functionality of EXFOR System
    1. *extracts and stores hidden information regulated by Manual*
    2. *implements connection to other databases*
    3. *implements maintenance procedures (load, update, backup, mirroring)*
    4. *provides data to user in output formats (including XML and renormalized data)*

2. Lessons from EXFOR

Components of NDS EXFOR System:

1.EXFOR Original Files (Master, TRANS)

2.EXFOR Manual

3.EXFOR Dictionaries

4.EXFOR Software*

5.External Files

6.EXFOR Database

7.EXFOR Output Formats

System design, database structure and software make NDS system flexible, robust, extendable.

*EXFOR + ENDF programs on Java and C: ~200,000 lines

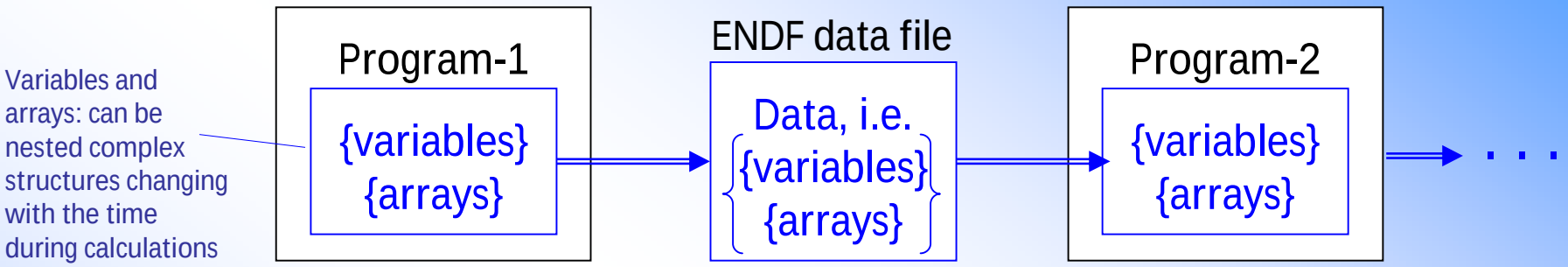
Conclusion

Many tasks can be solved outside of EXFOR format/files

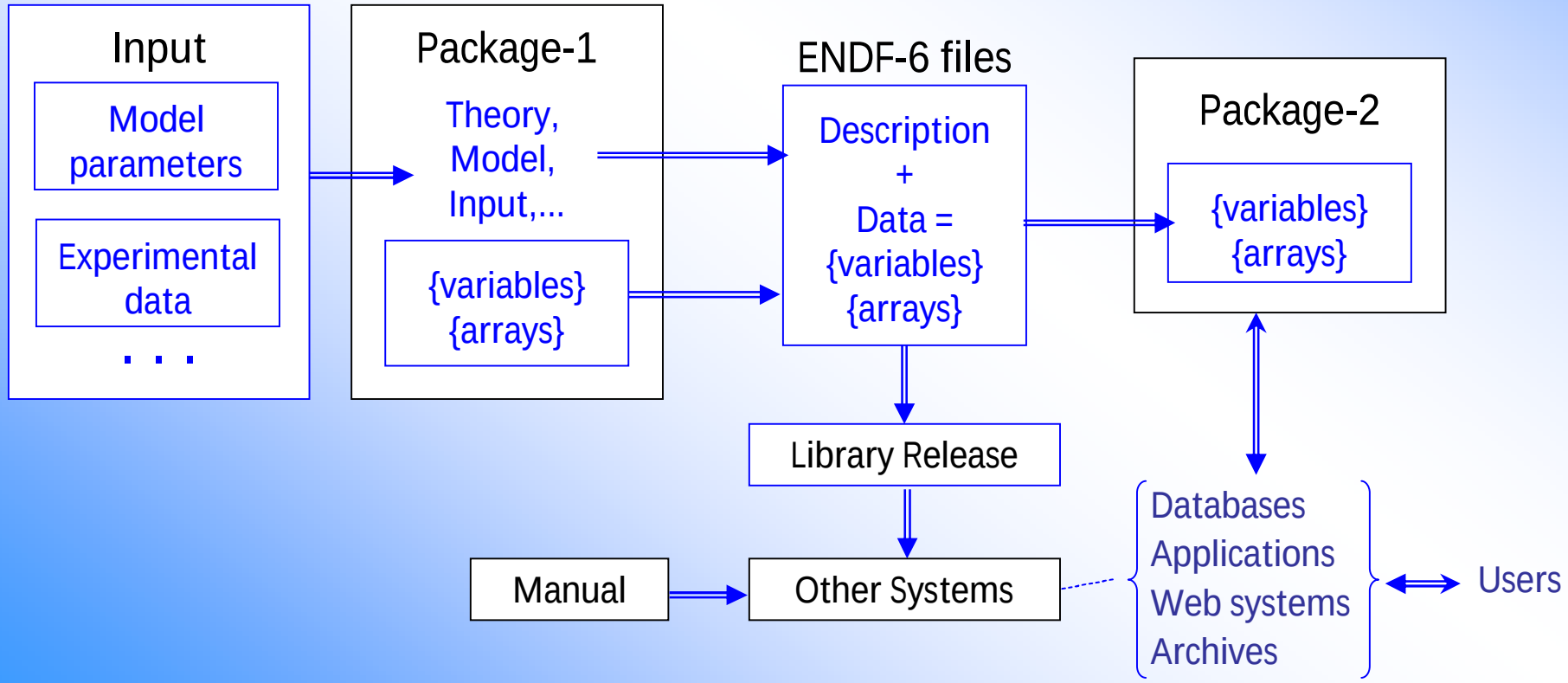
EXFOR format can be extended by data for optional programming

How (what for) ENDF format/files are used

Basic purpose: transfer data between programs for further calculations



Schema of data flow today

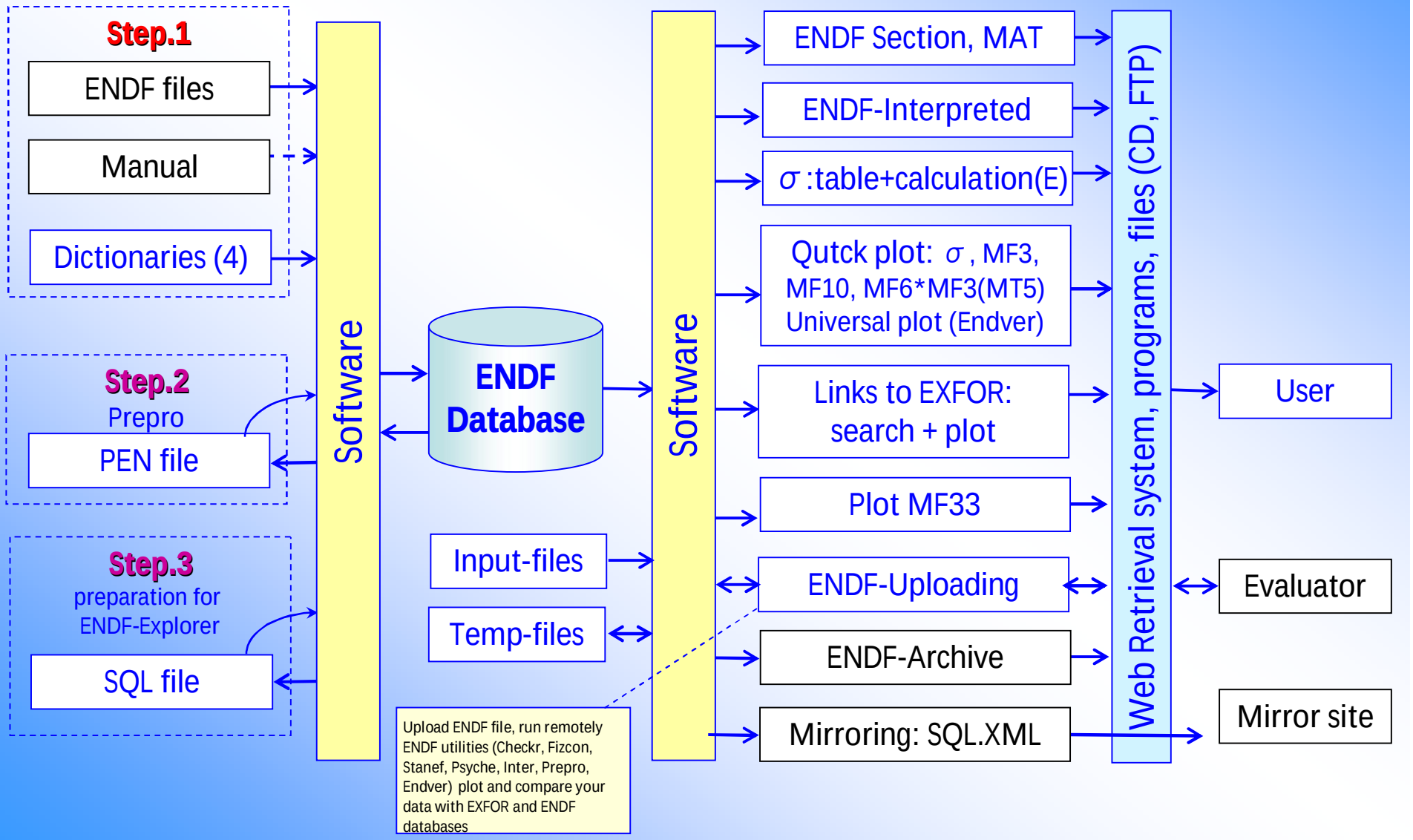


Tasks of NDS ENDF system (2004-2012)

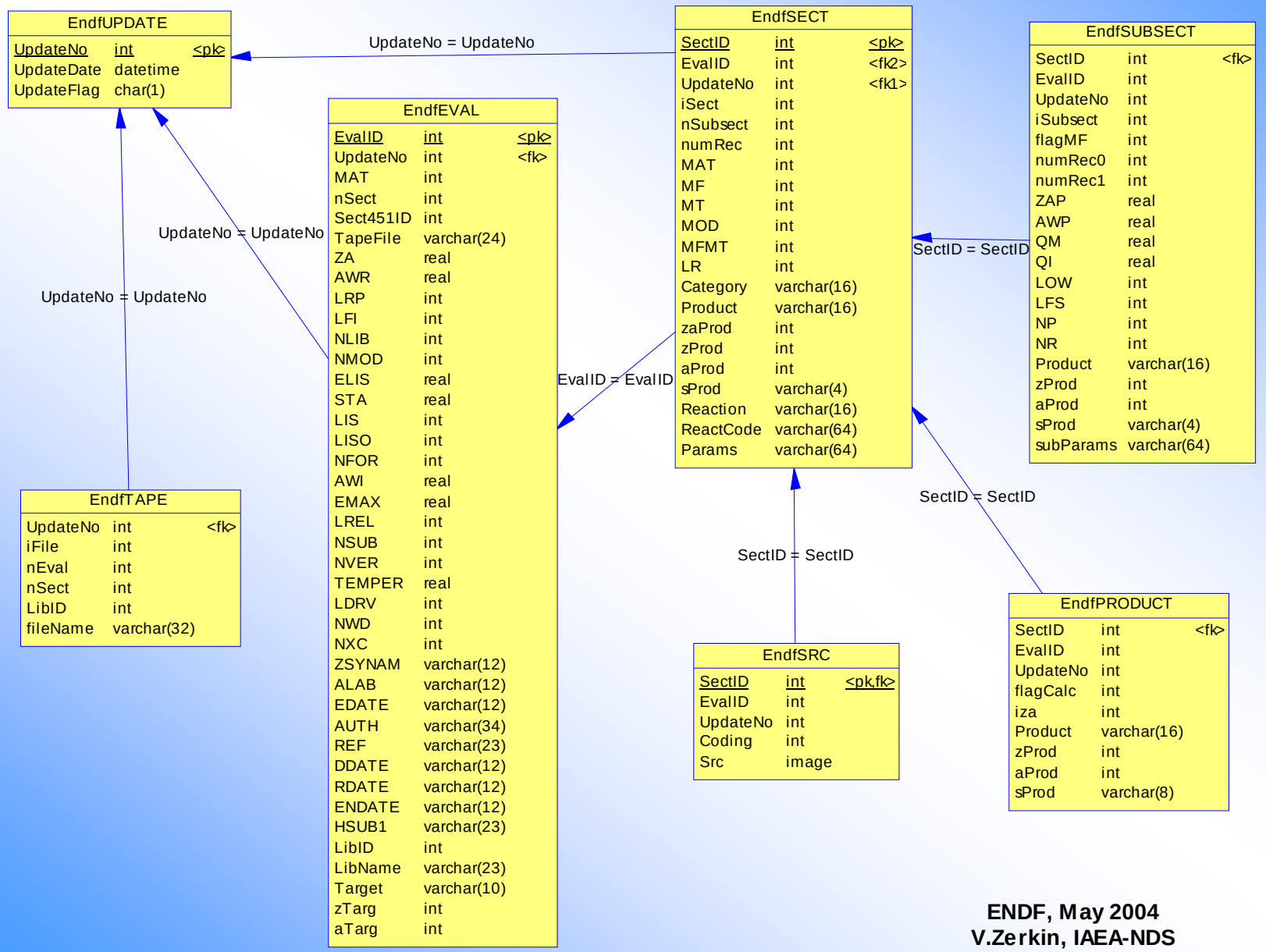
1. Create relational database compatible with ENDF-6 data structures
2. Provide Web user interface to the database: direct and sequential search of data with requests and data presentation easy for use and interpret by human
3. Provide universal search and retrieval of data using MF, MT, and also by Target, Reaction, Quantity, Product (coded in subsections), etc.
4. Provide plotting of simple and complex quantities (using PREPRO and ENDVER)
5. Provide search of similar data in EXFOR database and common plotting
6. Store all ENDF libraries available for the IAEA: major/national, specialized, derived and archived (now 43 libraries)

7. Provide possibility for export mirroring database and Web interface (to India)
8. Plotting of covariance matrices including 3-D animated, MT-MT and MAT-MAT (2010)
9. Implement uploading of users ENDF file for remote Web server calculations using ENDF Utility codes and Prepro, comparison with NDS ENDF and EXFOR databases, etc. (2010)
10. Convert EAF-2010/MF33 to MF40, plot MF6/MT5/Law=0*MF5, plot MF10, etc.

NDS ENDF System Overview

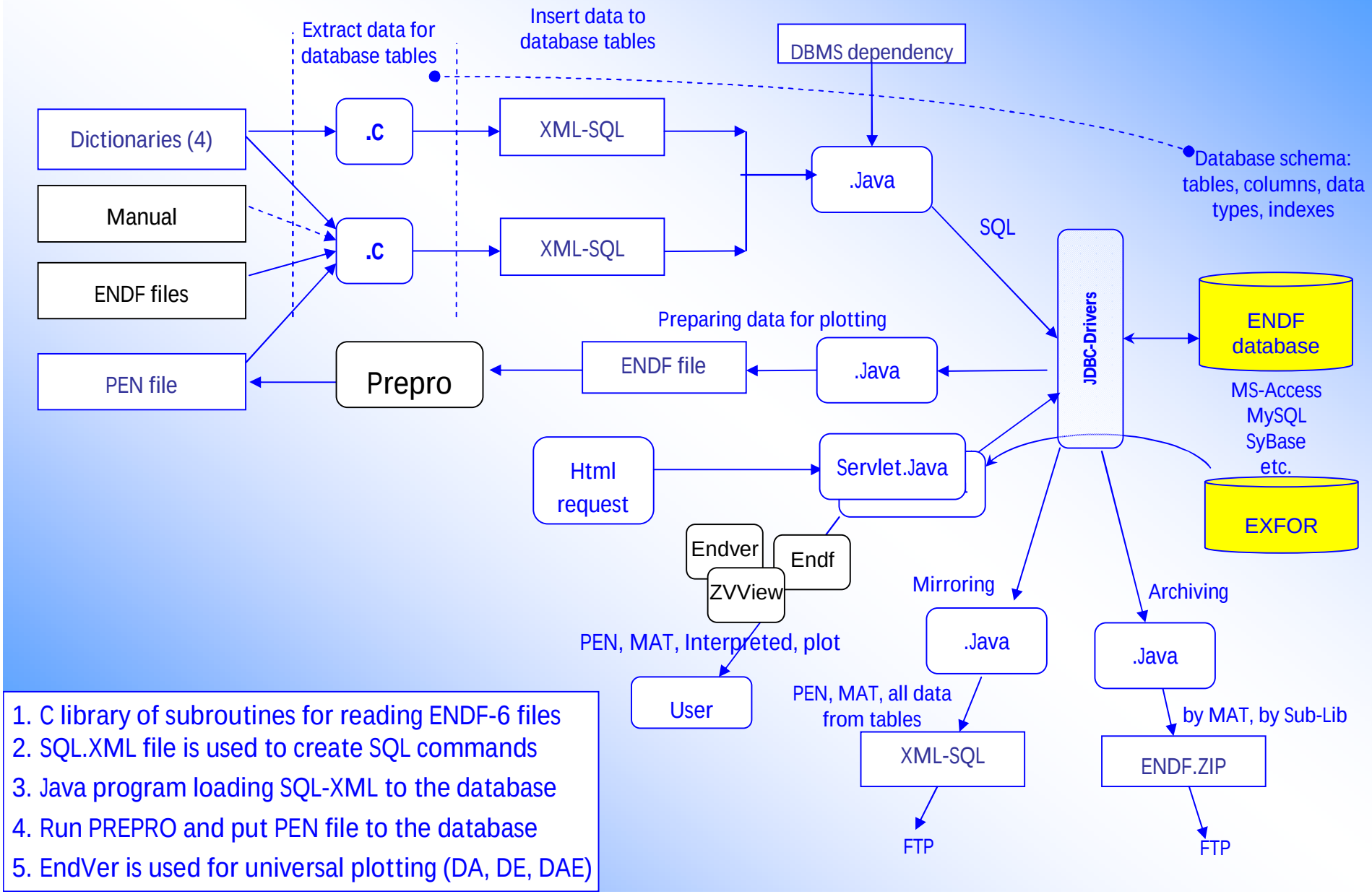


ENDF database schema (as of 2004)



ENDF, May 2004
V.Zerkin, IAEA-NDS

Loading ENDF database, Web interface, archiving, mirroring



1. C library of subroutines for reading ENDF-6 files
2. SQL.XML file is used to create SQL commands
3. Java program loading SQL-XML to the database
4. Run PREPRO and put PEN file to the database
5. EndVer is used for universal plotting (DA, DE, DAE)

ENDF notes

1. No concept of Dictionaries: all in the manual.

   Extract numerical data and convert them to human readable information using EXFOR-like dictionaries for NSUB, MAT, MF, MT, LibIDs making:

   NSUB + Dict -> Incident particle
   ZA + LISO + Dict -> Target
   MF + Dict -> Quantity
   MT + Dict -> Reaction
   Sub-sect ZAP + Dict -> Product

   O-16(N,N`)O-16-L6(A)C-12,SIG MF=3 MT=56 LR=22 QM=-7.161e+6 QI=-9.597e+6
   JENDL/HE-2004 E=3000MeV Lab=KYUSHU Date=REV1- S.KUNIEDA, N.SHIGYO, K.ISHIBASHI
   W-186(P,X)TA-183 Lines:199877-199894 ZAP=73183 AWP=181.38 LIP=0 LAW=0 MF=6 MT=5

2. Why MF10, MF40? Because initially there was no concept of product in hierarchy?
3. No concept of comment in datasets (only in MF1/MT451)
4. There is almost no possibility for optional programming (only on the level of MT)
5. Database structure makes ENDF “human readable” up to the level of sub-sect
6. Relational database structure allows to combine data in any order (no fixed hierarchy as it is in XML)
7. SQL-XML for mirroring (replication) contains complete information for database allowing to have full functionality of Web retrieval system (including ENDF, PEN)

3. Lessons from ENDF

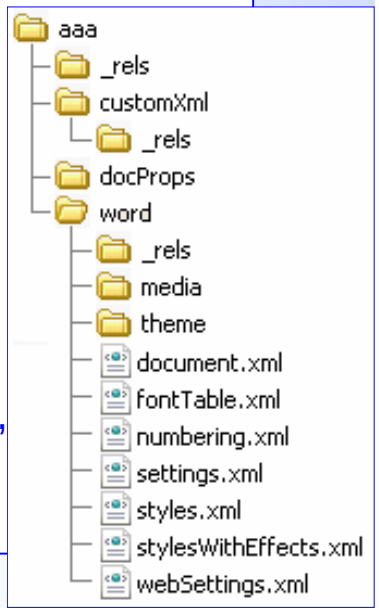
1. Dictionaries separate numerical data and flags from software (presenting them in “human” readable text)
2. ENDF-6 is very precise, programs reading files and extracting data to the database can be written quickly
3. ENDS relational Database stores named parameters and data extracted from ENDF and PEN files suitable for search and retrievals.
4. ENDF Software implements all functionality of ENDF system including data presentation (interpreted ENDF data, plots, simple column output) and connection to EXFOR System
5. ENDF Java package helps to solve unpleasant problems (reading MF33, MF6*MF3, conversion nonstandard MF33 to MF40, reading MF3 from ENDF-5, -4 to extract Standard cross-sections, etc.)
6. NDS ENDF System is functioning 8 years at IAEA and NNDC, we should have serious reasons to change it

4. Lessons from CINDA

1. Simultaneously:
    1. Change format to include charged particles and photonuclear
    2. To import data from EXFOR increasing size in 3 times
    3. Migrate (rewrite) all software from VMS/Fortran to Unix/Windows including database maintenance, editor, retrieval systems, printing book
    4. Replace people (new generation of compilers and programmers)
    5. Moratorium until all parts of the system will be ready (2+ years to agree on format + software)
2. Blocking system, unnatural hierarchy of the data, difficult to formulate clear rules and explain to new compilers. “Lets at first implement CINDA-2001, than revise basic approach…” – failed.
3. Function of EXFOR compilation control has gone to another database, function ‘show EXFOR prelim data’ does not exist…
4. CINDA now exists only in automatic regime of extensions – it does not exist as independent database. May be it is not bad… (?)

5. Lessons from DOC and DOCX

1. Nobody from end-users know DOC format, all manipulations are done via software interface (MS-Word, copy/paste)
   Our case (nuclear data formats/files)
   1. Having powerful EXFOR Editor compiler can forget about counting positions in EXFOR file, memorizing dictionary numbers, codes etc.
   2. Having interpreted output files (in several formats) end-user should not know structure and coding rules in EXFOR and ENDF files.
   3. Having API program libraries (software interface to the EXFOR and ENDF data) application programmer should not know anymore details of formats and implementations
2. DOCX was introduced as development of DOC available for other software development companies (open?). Essentially DOC was converted to XML.
3. But it is not one XML file. If we rename file.docx to file.zip we see a directory structure with XML, GIF, PNG files
   Our case
   - EXFOR: we have PDF, BibTeX files, Excel and e-mail files from authors, old versions of Entries. Many files in various formats naturally organized to directory structure. We can start proper archiving and exchange, build scripts to run some programs.
   - ENDF: we can introduce a directory structure with several types of files: one of them will be ENDF, others: PEN, XML describing data/structure/s, XML describing list of experimental data used for evaluation, list of data corrections, file with parameters, etc. All these files (with naming conventions) can be zipped and used as “Evaluated Material” to be accessed via software interface.



5. Lesson from HTML

1. Everybody from end-users can know HTML format and created Web pages using any text editor
2. Structure of HTML <tag attr="value">text</tag> is not strict. It allows mistakes and (!) optional programming: Web-browser can ignore (not interpret) some attributes, which it "does not understand", every browser can introduce their own attributes or values; there is a committee officially approves format.

   Our case (examples)

   1. Coding of DOI, NSR-Keyno, Covariance is done for "optional programming"
   2. ENDF Descriptive Section (1/451) contains used parameters, tables, references to experimental datasets, etc. This information is understandable for reader but not structured for programming
   3. "My EXFOR.XML" has both attributes EXFOR-Code and Code expansion from the dictionary. First must be strictly correct and being by programs, second is just note for human-user.

About "unfixable" problems in EXFOR

Unfixable problems in both EXFOR and ENDF formats

- ✘ 1. Rigid structure not lending itself for easy extensions
- ✘ 2. Whatever is not in the format can't be stored (downstream codes would crash!)
- ✘ 3. Fixed number of digits (in ENDF, this is a big problem with covariances; EXFOR has few covariances)
- ✔ 4. Dropping 'E' in the exponent is not well received by modern languages
- ✘ 5. **Impossible to read by an average homo sapiens**
- ✔ 6. Complex coding needed to read, no standard software support
- ✘ 7. Does not integrate with other libraries (EXFOR, ENSDF, ENDF)

Brookhaven Science Associates — *Slide based on slide from M. Herman* — 4 — BROOKHAVEN NATIONAL LABORATORY

Comments:

1. True for ENDF only
2. False for EXFOR: selective processing for keywords/codes by most of EXFOR codes
3. True only for COMMON/DATA sections; most of covariance data are stored in free text
4. Not complex (unpleasant)
5. True for ENDF only
6. True
7. EXFOR-ENDF are integrated (e.g. in X4TOC4)

What would be damaged if we change format?

Comprehensive programs reading EXFOR files

1. **X4TOC4 (Fortran-77)**
2. **CHEX (Fortran-77)**
3. **Mk_tabs (C)**
4. **X4java (Java)**
5. **Janis (Java)**
6. **X4i**
7. **Japan? China?**
8. **What else?**

7. Concluding remarks

1) Many problems of EXFOR and ENDF were solved by **Software**: EXFOR Editor, Relational Database, Interpreted output, PDF, etc.
2) Many problems which look “unfixable” can be fixed on **Software** level
3) Nuclear data community collaborating in **Software** (working with formats) on the level of single program and packages (exe-level). Modernizing ENDF format will need deeper cooperation - at least in data structures and API (object-level).

Thank you