Australian Government

Australian Nuclear Science and Technology Organisation

Recent CIF developments: DDLm and dREL

James Hester

CIF: Crystallographic Information *Framework*

- Consisting of:
  - A syntax specification for CIF data files
  - Sets of definitions (written in the same syntax) called “domain dictionaries”
  - Definitions for the attributes used in the domain dictionaries

CIF syntax

```
#####################################################
#  Example CIF based on PDB entry 432d
#####################################################

data_432D
loop_
_audit_author.name
'Vlieghe, D.'
'Van Meervelt, L.'

_cell.length_a                   25.616
_cell.length_b                   36.563
_cell.length_c                   52.961
_cell.angle_alpha                90.00
_cell.angle_beta                 90.00
_cell.angle_gamma                90.00
_cell.volume                     49603.2

_symmetry.space_group_name_H-M   'P 21 21 21 '

loop_
_chem_comp.id
_chem_comp.mon_nstd_flag
_chem_comp.formula
_chem_comp.name
_chem_comp.mon_nstd_details
  A yes 'C10 H14 N5 O7 P1'        "Adenosine"              .
  C yes 'C9 H14 N3 O8 P1'         "Cytidine"               .
  G yes 'C10 H14 N5 O8 P1'        "Guanosine"              .
  T yes 'C10 H15 N2 O8 P1'        "Thymidine"              .
```

CIF syntax

```
#####################################################
#  Example CIF based on PDB entry 432d
#####################################################

data_432D
loop_
_audit_author.name
'Vlieghe, D.'
'Van Meervelt, L.'

_cell.length_a                   25.616
_cell.length_b                   36.563
_cell.length_c                   52.961
_cell.angle_alpha                90.00
_cell.angle_beta                 90.00
_cell.angle_gamma                90.00
_cell.volume                     49603.2

_symmetry.space_group_name_H-M    'P 21 21 21 '

loop_
_chem_comp.id
_chem_comp.mon_nstd_flag
_chem_comp.formula
_chem_comp.name
_chem_comp.mon_nstd_details
  A yes 'C10 H14 N5 O7 P1'         "Adenosine"                .
  C yes 'C9 H14 N3 O8 P1'          "Cytidine"                 .
  G yes 'C10 H14 N5 O8 P1'         "Guanosine"                .
  T yes 'C10 H15 N2 O8 P1'         "Thymidine"                .
```

Domain dictionary

```
save_symmetry.space_group_name_H-M
    _item_description.description
;               Hermann-Mauguin space-group symbol. Note that the
                Hermann-Mauguin symbol does not necessarily contain complete
                information about the symmetry and the space-group origin. If
                used, always supply the FULL symbol from International Tables
                for Crystallography Vol. A (2002) and indicate the origin and
                the setting if it is not implicit. If there is any doubt that
                the equivalent positions can be uniquely deduced from this
                symbol, specify the  _symmetry_equiv.pos_as_xyz or
                _symmetry.space_group_name_Hall  data items as well. Leave
                spaces between symbols referring to
                different axes.
;
    _item.name                  '_symmetry.space_group_name_H-M'
    _item.category_id             symmetry
    _item.mandatory_code          no
    _item_aliases.alias_name    '_symmetry_space_group_name_H-M'
    _item_aliases.dictionary      cif_core.dic
    _item_aliases.version         2.0.1
    _item_type.code               line
    loop_
    _item_examples.case           'P 1 21/m 1'
                                  'P 2/n 2/n 2/n (origin at -1)'
                                  'R -3 2/m'
     save_
```

- Definitions are human and machine readable
- Set of allowable attributes called “Data definition language” (DDL)
- Dictionaries may be combined
- Core dictionary: 700 definitions
- Macromolecular: 1850 definitions

Domain dictionary

```
save__symmetry.space_group_name_H-M
    _item_description.description
;               Hermann-Mauguin space-group symbol. Note that the
                Hermann-Mauguin symbol does not necessarily contain complete
                information about the symmetry and the space-group origin. If
                used, always supply the FULL symbol from International Tables
                for Crystallography Vol. A (2002) and indicate the origin and
                the setting if it is not implicit. If there is any doubt that
                the equivalent positions can be uniquely deduced from this
                symbol, specify the  _symmetry_equiv.pos_as_xyz or
                _symmetry.space_group_name_Hall  data items as well. Leave
                spaces between symbols referring to
                different axes.
;
    _item.name                  '_symmetry.space_group_name_H-M'
    _item.category_id             symmetry
    _item.mandatory_code          no
    _item_aliases.alias_name    '_symmetry_space_group_name_H-M'
    _item_aliases.dictionary      cif_core.dic
    _item_aliases.version         2.0.1
    _item_type.code               line
    loop_
    _item_examples.case           'P 1 21/m 1'
                                  'P 2/n 2/n 2/n (origin at -1)'
                                  'R -3 2/m'
     save_
```

- Definitions are human and machine readable
- Set of allowable attributes called “Data definition language” (DDL)
- Dictionaries may be combined
- Core dictionary: 700 definitions
- Macromolecular: 1850 definitions

DDL dictionary

```
save_item.mandatory_code

    _item_description.name              '_item.mandatory_code'
    _item_description.description
;
     Signals if the defined item is mandatory for the proper description
     of its category.
;
    _item.name                          '_item.mandatory_code'
    _item.category_id                     item
    _item.mandatory_code                  yes
    _item_type.name                     '_item.mandatory_code'
    _item_type.code                       code
     loop_
    _item_enumeration.name
    _item_enumeration.value
    _item_enumeration.detail
       '_item.mandatory_code'
         yes      'required item in this category'
       '_item.mandatory_code'
         no       'optional item in this category'
       '_item.mandatory_code'
         implicit 'required item but may be determined from context'
     save_
```

- Defined using its own attributes
- DDL1: 27 attributes
- DDL2: 60 attributes
- About half are machine-significant

Issues and opportunities

- Two DDLs developed in tandem: DDL1 and DDL2, need to merge
- Considerable repetition involved in dictionary writing
- Relationships between data items not machine-readable

Enter…DDLm

- Easier dictionary writing
  - Definition templates
  - Tabular data (e.g. scattering lengths) in external, separately updateable file
  - Build domain dictionary out of separate chunks (categories, data items, whole dictionaries)
  - Use of category hierarchies for attribute inheritance (cf programming classes)

Enter…DDLm

- Complexity has not increased: 62 attributes compared to 60 (DDL2)

Enter…dREL

- A symbolic language for derivation of a data item value from other values in the dictionary

```
mass = 0.
Loop t as atom_type  {
                mass += t.number_in_cell * t.atomic_mass
 }
_cell.atomic_mass = mass
```

```
 With t as atom_type
cnt =  0.
Loop a  as  atom_site  {
   if ( a.type_symbol == t.symbol ) {
        cnt +=  a.occupancy * a.symmetry_multiplicity
  }
}
_atom_type.number_in_cell =  cnt
```

dREL in context

```
save_cell.atomic_mass
    _definition.id              '_cell.atomic_mass'
    _definition.update           2006-06-20
    _description.text
;
     Atomic mass of the contents of the unit cell. This calculated
     from the atom sites present in the ATOM_TYPE list, rather than
     the ATOM_SITE lists of atoms in the refined model.
;
    _description.common         'Cell Atomic Mass'
    _name.category_id            cell
    _name.object_id              atomic_mass
    _type.purpose                Assigned
    _type.container              Single
    _type.contents               Real
    _enumeration.range           0.:
    _units.code                  daltons
     loop_
    _method.purpose
    _method.expression
     Evaluation
;
    mass = 0.

    Loop t as atom_type  {

                   mass += t.number_in_cell * t.atomic_mass
    }
      _cell.atomic_mass = mass
;
     save_
```

But why dREL?

- A dREL method's task is simple: using other datafile values, return a derived value. We do not need:
  - I/O
  - Ability to define classes, other complex objects (types, templates, methods)
  - Exceptions
- A sufficiently simple symbolic syntax allows automatic error propagation and derivatives

But why dREL?

- dREL lives in an abstract context (the dictionary): concrete implementations are completely free to use their own approaches and languages to...
  - Access the data file
    - Packets by key, looping packets
  - Access dictionary information (keys)
  - Manage execution
- Methods are compact
  - dREL is a “little language”, tailored for purpose

Compactness

```
With p as position
       Loop g as geom {
       If (g.type == "point") {
             PointList += Tuple(g.vertex1_id,p[g.vertex1_id].vector_xyz)
       }
```

Compactness

```
With p as position
       Loop g as geom {
       If (g.type == "point") {
             PointList += Tuple(g.vertex1_id,p[g.vertex1_id].vector_xyz)
       }
```

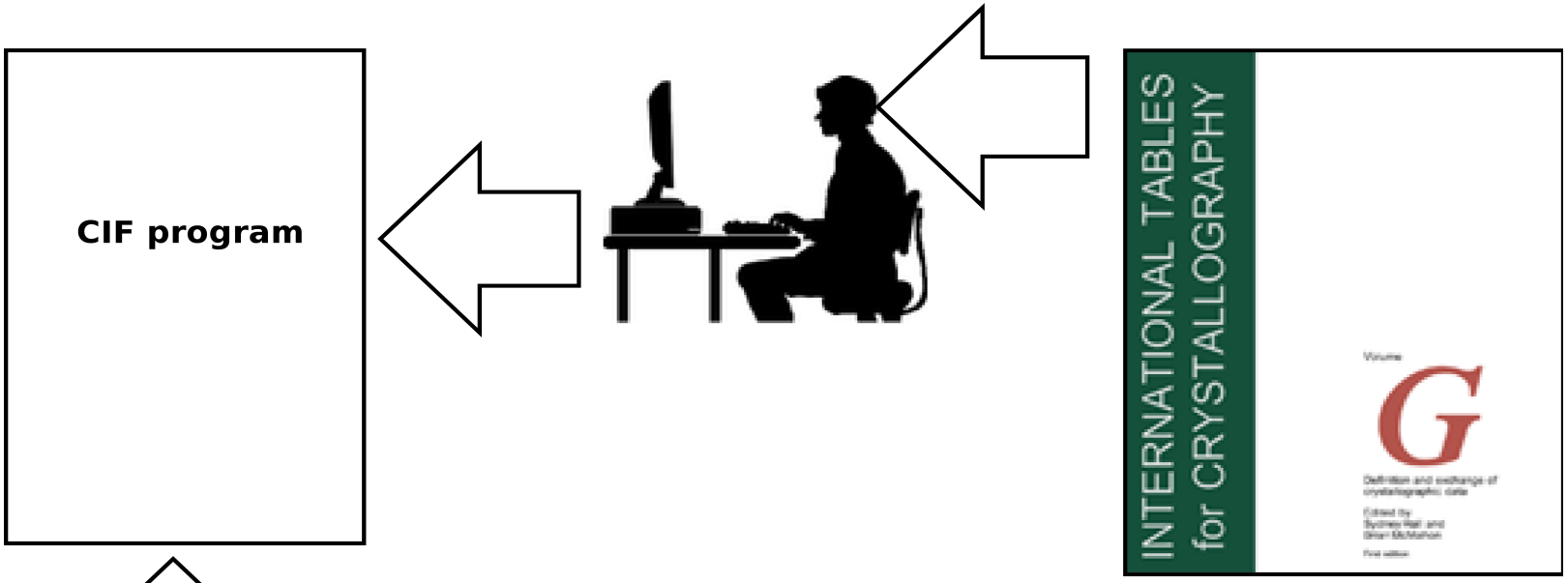
Retrofitting to “legacy” context (Python CIF module):

```
for __pi2 in range(len(ciffile[__pycitems[0]])):
        if ( ciffile["_geom.type"][__pi2] == "point" ):
            PointList += StarFile.StarTuple(ciffile["_geom.vertex1_id"][__pi2] ,
 ciffile.GetLoop(self["position"]["_category_key.generic"]).GetKeyedPacket(self[
"position"]["_category_key.generic"],ciffile["_geom.vertex1_id"][__pi2])["_posit
ion.vector_xyz"])
```

Whatever the environment, need to consider:

- [key] notation
- Packet access
- dREL syntax (lists, tuples,tables)

Using CIF dictionaries: typical practice



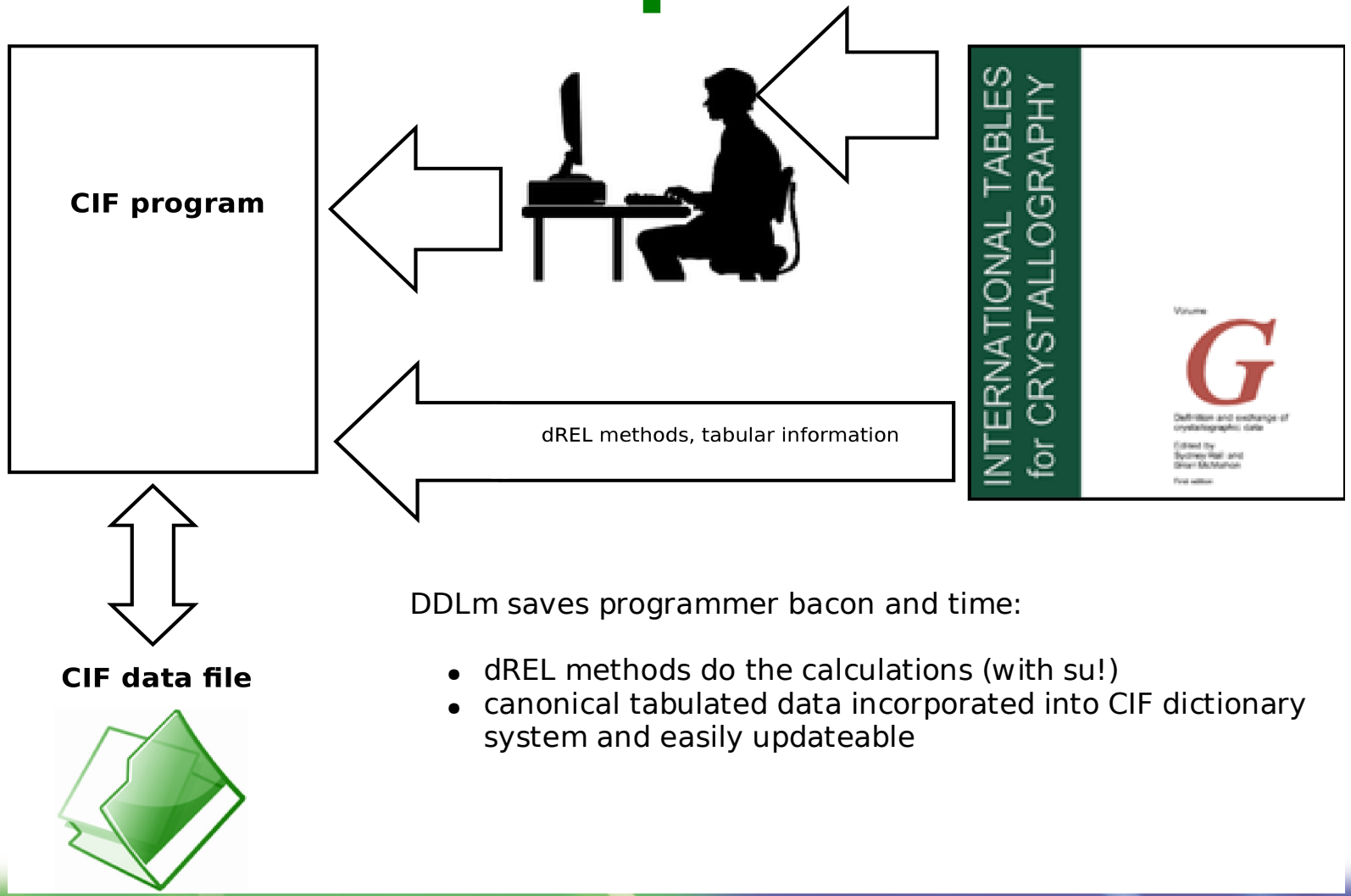
Programs need to be told:

- linkage between item names and program values
- when numeric/string data are expected
- when multiple values are possible/required

None of this information is subject to change between writing and execution time, therefore it is sufficient to code it statically into the program

The human-readable aspect of CIF dictionaries is overwhelmingly important!

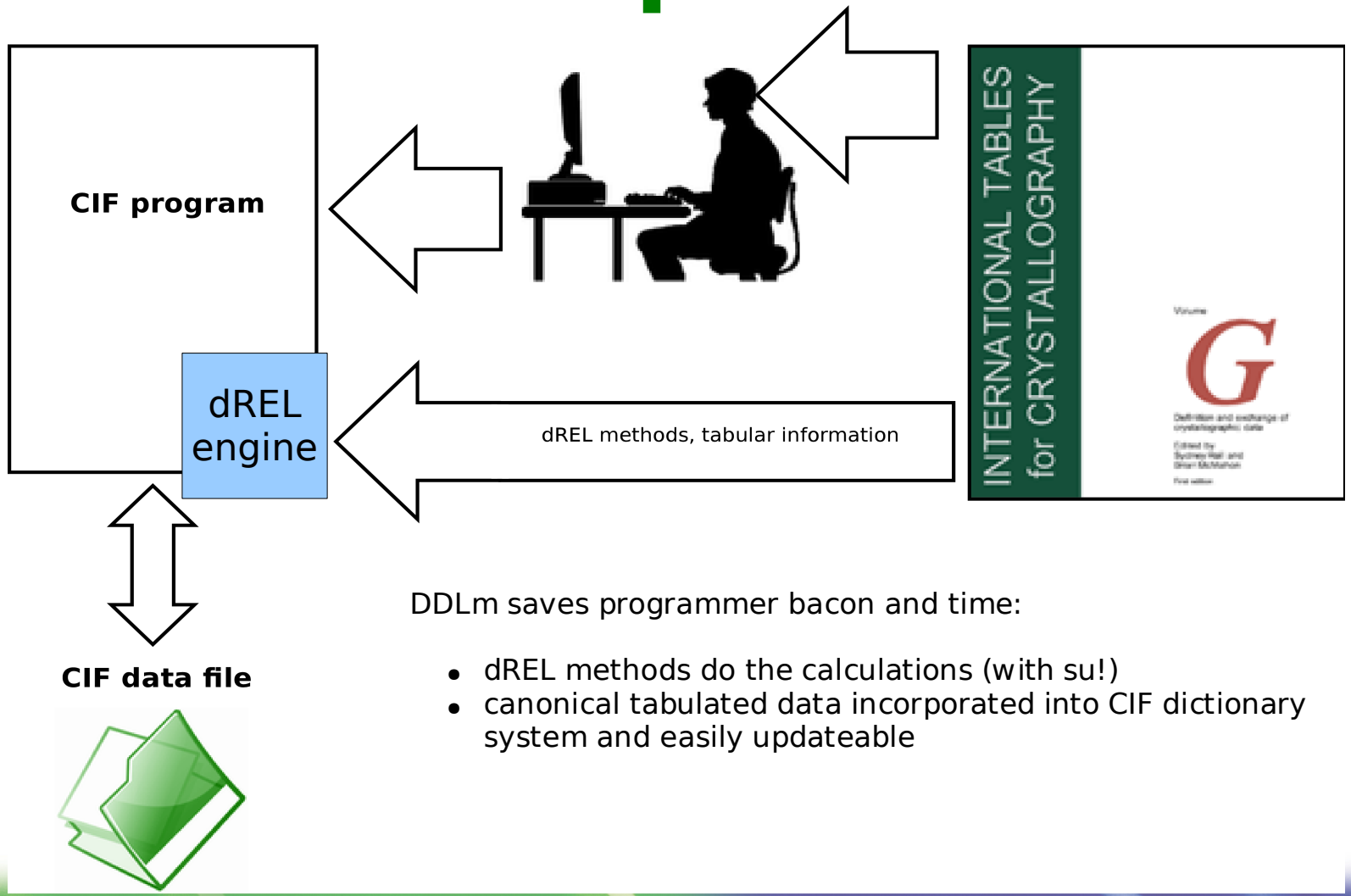
Using CIF dictionaries: DDLm potential



DDLm saves programmer bacon and time:

- dREL methods do the calculations (with su!)
- canonical tabulated data incorporated into CIF dictionary system and easily updateable

Using CIF dictionaries: DDLm potential



DDLm saves programmer bacon and time:

- dREL methods do the calculations (with su!)
- canonical tabulated data incorporated into CIF dictionary system and easily updateable

Given a dREL engine…

- CIF software need only generate those data items which are non-derivable from primitive measurements (eg results of least-squares refinements)
- Some prototyping calculations may be best performed in dREL
  1. Write new dictionary definition with dREL algorithm
  2. Point dREL engine at new dictionary and test datafile
- Derivatives are available for least-squares

Data file changes

The value of a data item may now be a (possibly nested) bracketed expression using “[]”, “()” or “{}”

```
_geom_bond.id     (('O1','1_555'),('O2',2_345))
```

The following examples from CIF files submitted to Acta C this year would be syntactically incorrect if they claimed DDLm conformance:

```
_publ_contact_author_fax      (+52)(81)82204924

_chemical_name_systematic      (2S,3S)-2-(N,N-dibenzylamino)butane-1,3-diol
```

Current status

- DDLm and dREL officially adopted by COMCIFS as 'alpha' status in August
- Formal syntax definition, prototype dREL engine due out by end of year
- Work proceeding on producing DDLm versions of current dictionaries

CIF References

- CIF: International Tables Vol G. http://it.iucr.org/G
- CIF, DDLm and dREL: http://ww1.iucr.org/cif