developed for other types of diffraction experiments, electron-microscopy data and for the general description of image data. The metadata concepts and tools that have been developed to support mmCIF are sufficiently general that they may be applied to the description of data in virtually any application.

The demands of structural genomics projects have driven the development of extensions to capture an increased level of experimental detail. These are available at http://mmcif.pdb.org/. Extensions have also been introduced to describe NMR, cryo-electron microscopy and all aspects of protein production. The ability to rapidly add extensions and incorporate these into the PDB data-processing system is an important feature for supporting the rapidly evolving technologies associated with high-throughput structure determinations.

The mmCIF metadata architecture is built from three levels as illustrated in Fig. 5.5.2.2 (see also Chapter 2.6). Individual data files are described at the top level (*e.g.* Fig. 5.5.2.2*a*). The contents of these data files are defined by a data dictionary (*e.g.* Fig. 5.5.2.2*b*) in the next lower level (see Chapters 3.6 and 4.5). The attributes used in this data dictionary to build data definitions are in turn defined in the dictionary description language (DDL) (*e.g.* Fig. 5.5.2.2*c*) in the lowest level (see Chapters 2.6 and 4.10).

The major syntactical constructs used by mmCIF are illustrated in the data file example of Fig. 5.5.2.2(*a*). Each data item or group of data items is preceded by an identifying keyword. Groups of related data items are organized into data categories. Two categories, CELL and ENTITY_POLY_SEQ, are shown in the example. CELL contains an individual instance describing a single set of crystallographic cell constants. ENTITY_POLY_SEQ contains a `loop_` (*i.e.* table) of instances describing a polymer residue sequence. Essentially all mmCIF data are described as a set of tabular data structures.

Each mmCIF data item is defined in a data dictionary. Data definitions are given between save-frame delimiters (*i.e.* `save_`); apart from this, the data definitions share the same simple syntax as used in data files. An example definition for a crystallographic cell constant is shown in Fig. 5.5.2.2(*b*). Many features of the cell constant are described in this definition, including data type, range restrictions, units of expression, dependent quantities, related definitions, necessity and related precision estimate. Although not shown in this example, dictionary definitions can also include parent–child relationships that have important consequences in maintaining data consistency.

The attributes of each data definition are defined in the DDL dictionary. Fig. 5.5.2.2(*c*) shows example DDL definitions describing data types. DDL definitions have the same syntax as definitions used in the data dictionary. Because the attributes of the DDL are also used in DDL definitions, this metadata architecture is described as self-defining.

The RCSB PDB distributes parsing tools that support all three levels of this metadata architecture (http://sw-tools.pdb.org/). The *CIFPARSE_OBJ* package (Tosic & Westbrook, 2000) provides high-level methods to read, write, validate and manage data from data files, dictionaries and DDLs. Data files can be validated relative to an input data dictionary, and dictionary files can be validated relative to an input DDL. *CIFPARSE_OBJ* stores information in a collection of table objects. Access methods are provided to search and manipulate the table objects. A companion package, *CIFOBJ* (Schirripa & Westbrook, 1996), provides an alternative representation of dictionary and DDL data. *CIFOBJ* organizes dictionary information into a collection of category and item-level objects. Access methods are provided for all dictionary attributes.

```
_cell.entry_id      W1QQQ
_cell.length_a      129.230
_cell.length_b       60.440
_cell.length_c       56.630
_cell.angle_alpha    90.00
_cell.angle_beta    119.05
_cell.angle_gamma    90.00
_cell.Z_PDB           4
loop_
  _entity_poly_seq.entity_id
  _entity_poly_seq.num
  _entity_poly_seq.mon_id
    1    1     ASP       1    2     ILE
    1    3     VAL       1    4     LEU
    1    5     THR       1    6     GLN
    1    7     SER       1    8     PRO
    1    9     ALA       1    10    SER
```

(*a*)

```
save__cell.length_a
    _item_description.description
; Unit-cell length a corresponding to the structure
  reported.
;
    _item.name                      '_cell.length_a'
    _item.category_id                 cell
    _item.mandatory_code              no
    _item_aliases.alias_name        '_cell_length_a'
    _item_aliases.dictionary          cif_core.dic
    _item_aliases.version             2.0.1
     loop_
    _item_dependent.dependent_name
                                    '_cell.length_b'
                                    '_cell.length_c'
     loop_
    _item_range.maximum
    _item_range.minimum              .    0.0
                                    0.0   0.0
    _item_related.related_name      '_cell.length_a_esd'
    _item_related.function_code       associated_esd
    _item_sub_category.id             cell_length
    _item_type.code                   float
    _item_type_conditions.code        esd
    _item_units.code                  angstroms
    save_
```

(*b*)

```
save_ITEM_TYPE_LIST
    _category.description
;   Attributes which define each type code.
;
    _category.id                    item_type_list
    _category.mandatory_code        no
    _category_key.name            '_item_type_list.code'
     loop_
    _category_group.id              'ddl_group'
                                    'item_group'
     save_

save__item_type_list.code
    _item_description.description
;    The codes specifying the nature of the data value.
;
     loop_
    _item.name
    _item.category_id
    _item.mandatory_code
         '_item_type_list.code'   item_type_list      yes
         '_item_type.code'        item_type           yes

    _item_type.code                 code
    _item_linked.child_name       '_item_type.code'
    _item_linked.parent_name      '_item_type_list.code'

     save_
```

(*c*)

Fig. 5.5.2.2. Files at different levels of the mmCIF metadata architecture. (*a*) mmCIF data file excerpt. (*b*) Example mmCIF data dictionary definition. (*c*) Example DDL dictionary attribute definition.