

## 3. CIF DATA DEFINITION AND CLASSIFICATION

several tables, or the implied key may not have a unique value unless concatenated with other fields in the table row. For many applications this is only of academic interest; but in some subdisciplines it is important that the data model is constrained strictly to a relational one, and for those applications dictionaries built on the DDL2 formalism are more appropriate.

Of the dictionaries presented in this volume, the core, powder, modulated structures and electron density dictionaries use the DDL1 formalism and the symmetry, macromolecular and image dictionaries use the DDL2 formalism. The core dictionary uses DDL1 so that it can be used alongside other less rigorous dictionaries. The powder dictionary is one case of this, where the need to tabulate and merge extensive lists of raw or processed data is not well served by a relational model. Modulated structures are also best served by a data model that is not rigorously relational. The macromolecular dictionary uses DDL2 because many of the major database applications in macromolecular crystallography are relational in nature, but in consequence it contains a copy of the core data items re-expressed in DDL2 formalism. The image dictionary is in DDL2 because it was designed to operate closely alongside the macromolecular dictionary. The symmetry dictionary is an interesting case. It was constructed in DDL2 format as an exercise in supplying an extension dictionary immediately suitable for direct incorporation into other DDL2-based dictionaries and also suitable for transformation to the simpler DDL1 formalism as necessary to complement existing DDL1 dictionaries.

While the main difference between DDL1 and DDL2 lies in the rigour with which relational data structures are enforced, DDL2 also offers a larger set of attributes for specifying hierarchical relationships between data names and for typing data values, and in consequence a complete DDL2-based dictionary is richer (and correspondingly more complex to construct) than an equivalent DDL1 description.

There may be no obvious reason for selecting one formalism over the other when planning a new data dictionary, and prospective authors must give considerable thought to the merits of both formalisms. However, once the choice has been made, the structure of the dictionary and its component definitions is profoundly affected. The constructions of the two types of dictionary are discussed separately in Sections 3.1.5 and 3.1.6 below.

### 3.1.5. Constructing a DDL1 dictionary

Dictionaries constructed according to DDL1 have quite a simple structure. The structure is summarized in this section; Sections 3.1.5.1–3.1.5.4 provide more detail. Each definition is encapsulated within its own data block. Fig. 3.1.5.1 outlines the contents of the core CIF dictionary. The order of the data blocks has no significance, but it is common practice to start the file with the data block that describes the name, version and revision history of the dictionary itself and then to arrange data blocks in alphabetical order, sorted first on category then on names within a category. This practice is not always followed – for example, the powder dictionary is ordered by theme. The choice of order in a dictionary is only used for presentation and dictionary parsers should not assume or rely on any order of data blocks.

The name of a data block is usually constructed from the name of the data item it describes, *e.g.* `data_refl_n_phase_meas`. Where the data block describes an entire category instead of a single data item, the category name is followed by matching square brackets, which may contain an alphabetic code representing the dictionary name if it is an extension to the core dictionary (*e.g.*

```

data_on_this_dictionary
  _dictionary_name      cif_core.dic
  _dictionary_version   2.3.1
  _dictionary_update    2005-06-27
  _dictionary_history
;
  1991-05-27  Created from CIF Dictionary text. SRH
  . . .
;
                                     (a)

data_atom_site_[]
  _name                 '_atom_site_[]'
  _category              category_overview
  _type                  null
  loop_ _example
  _example_detail . . .

data_atom_site_adp_type
  _name                  '_atom_site_adp_type'
  _category              atom_site
  _type                  char
  _definition            'A standard code ...'

data_atom_site_aniso_B_
  loop_ _name            '_atom_site_aniso_B_11'
  . . .
                                     (b)

```

Fig. 3.1.5.1. Schematic structure of core CIF dictionary. (a) Dictionary identifiers. (b) Definitions of categories and data items.

`data_refl_n_[]`, `data_audit_link_ms[]`). Where the data block defines several data names, the initial common portion of the names is used with a trailing underscore (*e.g.* `data_refl_n_`).

A preliminary data block, by convention labelled with the header string `data_on_this_dictionary`, contains the dictionary identification information and revision history. The name of the dictionary itself (given by the data name `_dictionary_name`) is conventionally of the form `cif_identifier.dic`, where the *identifier* is a short code for the topic area of the dictionary (*e.g.* 'core' for the core dictionary, 'pd' for the powder dictionary, 'ms' for the modulated structures dictionary, 'rho' for the electron density dictionary).

Data names are classified by category. The `_category` attribute is a character string intended to indicate the 'natural grouping' of data items. If a data item occurs in a looped list, it must be grouped only with items from the same category. It is, however, permissible for a file to contain more than one looped list of the same category, provided that each loop has its own specific reference item identified by the `_list_reference` attribute of the data names included. Examples of this will be given below.

For each category, a data block is usually provided that contains information about the purpose of the category, generally illustrated with examples.

All other data blocks represent self-contained definitions of a single data item or a small set of closely related data items. The definition includes the physical units of and constraints on the values of the data labelled by the defined data name, and also information about relationships with other data items.

It is conventional, although not mandatory in DDL1 dictionaries, that the category name should appear as the leading component or components of a data name. For example, the data name `_exptl_crystal_colour` is a member of the core category EXPTL, while `_exptl_crystal_density_meas` is a member of the category EXPTL\_CRYSTAL and `_exptl_crystal_face_perp_dist` is a member of the category EXPTL\_CRYSTAL\_FACE. However, it will