Acta Crystallographica Section B
Structural Science
ISSN 0108-7681

CIF: the computer language of crystallography

I. David Brown[a]* and Brian McMahon[b]

[a]Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada L8S 4M1, and [b]International Union of Crystallography, 5 Abbey Square, Chester CH1 2HU, England

Correspondence e-mail: idbrown@mcmaster.ca

Received 16 November 2001
Accepted 21 February 2002

The Crystallographic Information File (CIF) was adopted in 1990 by the International Union of Crystallography as a file structure for the archiving and distribution of crystallographic information. The CIF standard is now well established and is in regular use for reporting crystal structure determinations to *Acta Crystallographica* and other journals. The structure of CIF is flexible and extensible and is compatible with other evolving standards. It is well suited to relational and object-oriented models, and is being adopted by the crystallographic databases. This paper reviews the development of CIF and describes its salient features. Future extension of the standard to include implementation of methods will allow CIF to exploit the potential of advanced information-handling software.

1. Introduction

It is more than fifty years since crystallographers started using computers to solve and refine crystal structures. Computers allowed them to determine structures that were too complex to be handled by the simple mechanical calculators that were then available. In each subsequent generation, they have taken advantage of the burgeoning power of computers to increase both the size of the structures they study and the accuracy of the reports they publish. They have, however, been slower to realise the potential of computers for information handling; that is, for archiving and manipulating this information. Olga Kennard pointed the way in the 1960s by establishing the first electronic crystallographic database (Allen *et al.*, 1979; Allen, 2002; Bruno *et al.*, 2002), but it took a further decade before the community discovered how to use this database to gain new insights into structural chemistry.

By 1980, the automation of crystal structure determinations was resulting in reports that contained such large tables of coordinates that publication from typewritten manuscripts was becoming awkward and unreliable – hardly a paper appeared in print without at least one numerical error. The result was frustration among those trying to use the coordinates for further calculations. A new process was needed, one in which the numerical information generated by the computer-controlled diffractometers could be transferred directly to the journals and databases without the need for retyping. While such transfer was easy to accomplish in principle, it was complicated by the fact that each crystallographic computing package used a different output file structure and

none made provision for all the information required in a full structure report. The crystallographic community needed a common file structure that all crystallographic applications would recognize. It would, of course, have to be able to transfer the numerical information, but it also had to be able to transfer the text of a paper, which implied a flexible structure with free format. A well designed file structure could also be exploited by the coming information revolution in which much of the retrieval, assembly and analysis of information on crystal chemistry would be performed automatically by computer. In short, the common file structure had to be more than just a storage place for archiving the results of crystal structure determinations, it had to be a crystallographic language that could be used by computers to explore the wealth of information on crystal chemistry that was even then accumulating in electronic databases.

The Warsaw meeting of the IUCr in 1978 saw the first proposal for a standard file structure for crystallography (Brown, 1983), but it was a further 12 years before *Acta Crystallographica* committed itself to accepting manuscripts in electronic form. The requirements of a standard crystallographic file structure suitable for crystal structure reports were not as simple as adopting a particular typesetting or word processing package such as TEX or *WordPerfect*. While the file structure had to be able to accommodate the text of a paper, it also needed to identify each item of numerical information so that it could be checked for possible errors and retrieved for analysis by a new generation of crystal chemistry programs. The file structure designed to meet these requirements was the Crystallographic Information File (CIF; Hall *et al.*, 1991). Within six years, all crystal structures reported in *Acta Crystallographica* were being submitted and archived electronically as CIFs, and the crystallographic databases were beginning to explore the potential that the relational properties of the CIF offered for structuring their archives.

The tag–value syntax of the CIF (see §2) is conceptually simple but requires the development of dictionaries to define the meaning of each of the tags that could appear in a CIF. The dictionary was originally conceived as a printed manual to be used by crystallographic programmers, but it was soon realised that the dictionary could be most easily typeset if it too were stored as a file with the same flexible structure as that of a CIF. Once on a computer, the dictionary could also be interrogated on-line, opening the possibility that an application could use the dictionary directly to help identify the different items reported in a CIF. The properties of each CIF item, such as its name (tag), whether its value is a number or text, and its allowed range of values were all defined in the dictionary and were thus directly available to the application. The dictionaries could be used, for example, to detect not only syntax errors in the CIF but also semantic errors, such as values that are physically unreasonable.

New features continue to be added to the CIF dictionaries. Future dictionaries will include 'methods'; that is, algorithms that instruct the computer how to calculate the value of a particular item from other items in the CIF. When fully developed, CIF will be more than just a file structure for archiving crystallographic information. The CIF dictionaries will be comprehensive encyclopaedias of crystallographic knowledge and CIF itself will become the computer language used to assemble and process the large quantities of information generated by X-ray and neutron diffractometers and stored in the crystallographic databases.

In modern parlance, CIF will offer a 'domain ontology' – a mechanical representation of the knowledge within a specific subject area, one that is accessible to computer manipulation. The syntax behind the file structure (described in the next section) may differ from the syntax of data and knowledge representation adopted by other disciplines, thus necessitating file conversion utilities, but the maintenance of well structured information about the tags that are used in CIFs will make it possible in future to design inter-discipline knowledge exchange mechanisms – an option that has not hitherto been available.

```
data_global
_audit_creation_date              'October 21, 1999'
_publ_contact_author_name         'Professor George Ferguson
_publ_section_title
    '3-Benzo[b]thien-2-yl-5,6-dihydro-1,4,2-oxathiazine-4-oxide'
_publ_section_abstract
; The title compound, C~11~H~9~NO~2~S~2~ contains the 1,4,2-oxathiazine
   ring system in a half-chair conformation with the methylene C atoms
   0.510(3) and -0.367(3)\%A from the plane of the other four ring atoms.
;
data_99107abs
_chemical_name_systematic
   '3-Benzo[b]thien-2-yl-5,6-dihydro-1,4,2-oxathiazine 4-oxide'
_chemical_formula_moiety          'C11 H9 N O2 S2'
_chemical_formula_weight          251.31
_symmetry_cell_setting            orthorhombic
_symmetry_space_group_name_H-M    'P 21 21 21'
loop_
 _symmetry_equiv_pos_site_id
 _symmetry_equiv_pos_as_xyz
 1  'x, y, z'
 2  'x+1/2, -y+1/2, -z'
 3  '-x, y+1/2, -z+1/2'
 4  '-x+1/2, -y, z+1/2'
_cell_length_a                    7.4730(11)
_cell_length_b                    8.2860(11)
_cell_length_c                    17.527(2)
_cell_angle_alpha                 90.00
_cell_angle_beta                  90.00
_cell_angle_gamma                 90.00
_cell_volume                      1085.3(3)
_cell_formula_units_Z             4
_exptl_crystal_description        plate
_refine_ls_wR_factor_ref          0.073
loop_
 _atom_site_label
 _atom_site_type_symbol
 _atom_site_fract_x
 _atom_site_fract_y
 _atom_site_fract_z
 _atom_site_U_iso_or_equiv
S4  S  0.32163(7)  0.45232(6)  0.52011(3)  0.04532(13) Uani
S11 S  0.39642(7)  0.67998(6)  0.29598(2)  0.04215(12) Uani
O1  O -0.00302(17) 0.67538(16) 0.47124(8)  0.0470(3)   Uani
O4  O  0.2601(2)   0.28588(16) 0.50279(10) 0.0700(5)   Uani
N2  N  0.14371(19) 0.66863(19) 0.42309(9)  0.0402(3)   Uani
C3  C -0.2776(2)   0.07587(19) 0.07691(9)  0.0332(3)   Uani
C5  C  0.1497(3)   0.5457(3)   0.57608(11) 0.0498(5)   Uani
C6  C -0.0171(3)   0.5529(2)   0.52899(12) 0.0460(4)   Uani
C12 C  0.4215(2)   0.57488(19) 0.38139(9)  0.0344(3)   Uani
loop_
 _geom_bond_atom_site_label_1
 _geom_bond_atom_site_label_2
 _geom_bond_distance
 _geom_bond_site_symmetry_2
 _geom_bond_publ_flag
S4  O4   1.485(2) . yes
S4  C3   1.813(2) 3 yes
S4  C5   1.792(2) . yes
S11 C12  1.742(2) . no
```

Figure 1
Extract from a CIF journal submission (not all required items are shown and some atoms have been omitted for brevity).

2. The structure of CIF

The CIF is based on the STAR (Self-Defining Text Archive and Retrieval) file structure described by Hall (1991). STAR has a conceptually simple syntax, or grammar, in which each item of information is encoded into a pair of character strings, the first being the tag (the data name), the second its value, *e.g.* `_cell_volume 1763.8`.

The leading underline of '`_cell_volume`' indicates that this string is a data name. Strings are separated by arbitrary amounts of 'white space' (space, tab or end-of-line characters) and, to ensure long-term stability, only characters found in the ASCII character set may be used.

A STAR file is divided into data blocks, each of which starts with the data name `data_`*xxxxx*, where *xxxxx* represents a name chosen by the user. The data block is broken into lines that are normally kept to less than 80 characters to ensure that each line can be viewed on a screen or printed on standard sized paper without wrap around. A data block ends with either the beginning of the next data block or the end-of-file. Fig. 1 shows an example of a CIF, illustrating the variety of information it may contain and the way this information is stored. It is a slightly edited and very abbreviated extract from a file presented to authors as an exemplar submission for *Acta Crystallographica Section C*.[1]

Before a STAR file can be written, a dictionary of data names and their meanings is required. The dictionaries that contain the names of crystallographic items are known as CIF dictionaries and the files written with them are called CIFs (Hall *et al.*, 1991).

Since each data name may only appear once in a data block, tables, such as the table of atomic coordinates, appear in loops introduced by the string '`loop_`' followed first by an ordered list of data names and then by an ordered list of values.

Items are divided into those that represent numbers and those that represent text. Number strings represent real numbers or integers (and may include an indication of a standard uncertainty in physically measurable quantities); text strings may contain any character in the printable ASCII character set. Any text string that contains spaces must be delimited with single or double quotes, otherwise the spaces will be interpreted as delimiters. If the string extends over more than one line so that the text string contains an end-of-line character or set of characters, it must be delimited by a semicolon appearing as the first character on the line, as shown in Fig. 2.

3. CIF dictionaries

In earlier years, file structures tended to be defined in terms of fixed formats in which each item was identified by its position in a particular record. More recently, designers of file formats have realised that there needs to be a clear separation between the form (structure or syntax) and the content (semantics or metadata) of a file. Keeping the description of its contents separate from the structure makes it easier to extend the list of items that can be included and allows the writing of generic software that can search for and extract specific items using their names (tags) without the need to know the nature of its value. A common example of this is the eXtensible Markup Language (XML) used increasingly in publishing and database activities (W3C, 2001). The syntax of an XML file conforms to the Standard Generalized Markup Language (SGML) specification (ISO, 1986), but different XML applications typically have different Document Type Definitions (DTDs), usually as separate files, that define the ordering and nesting of the permitted content. At the time CIF was first developed, the restricted availability and high cost of high-quality SGML tools were among the factors that discouraged exploration of this class of markup. However, CIF adopted the same paradigm of separating form from content. Descriptions and definitions of the tags or data names in CIF were held in external dictionaries, with roles analogous to the DTDs of XML. One consequence of the relatively flat file structure of CIF is that it contains a larger number of distinct tags (each labelling a specific crystallographic property) than is usually the case in XML, where distinct tags tend to describe different larger-scale structures in the data model. In XML, the finer details of the structure are contained in the qualifiers that form part of the tag, a feature that is not part of the STAR file structure. Qualifiers are useful when a given item may be defined in different ways, since they can indicate which convention has been adopted. In a field such as crystallography, where the concepts used are well defined and their

```
loop_
  _text_examples

'This is a character string delimited by single quotes'

"A string may include delimiting characters (") if not followed by space"

; This is a much longer piece of text that is so long that it must extend
  over more than one line.  Therefore it is delimited by using a semicolon,
  ';', as the first character of the first and last lines.  Note that the
  semicolon may be used in the string providing it does not immediately
  follow the end-of-line.
;
```

Figure 2
Examples of text strings in a CIF.

data name	a character string beginning with an underline. Any unique character string can serve as a data name, but usually the names are chosen to suggest their meaning.
description	a description or definition of the information contained in the item.
value	the numeric or text value assigned to the item in the CIF.
type	defines if the value is a text string or a number.
units	the units in which a numeric value is expressed.
category	all items that appear in the same loop must belong to the same category.
category key	a pointer used to identify different groups of items in a list.
mandatory flag	indicates if the item is required for data management purposes.
parent–child links	relationships between pointers that link items in different categories

Figure 3
A selected list of the properties of data items.

[1] The complete example file is available from http://journals.iucr.org/services/cif/examples.html.

meaning is stable, a more detailed dictionary with precise definitions is more appropriate.

Thus, since each item in a CIF is identified by a separate data name, it is necessary to have access to the dictionary that describes the properties associated with the name. A CIF dictionary, in addition to giving the name and a description of the item, also lists a number of other properties, such as whether the value is numeric or text. Fig. 3 lists some of these properties, discussed in more detail below.

CIF dictionaries define the principal items required in crystallography and so far six dictionaries (Fig. 4) have been approved by COMCIFS, the committee established by the IUCr to oversee the CIF project. Further dictionaries covering such fields as small-angle scattering, magnetic structures and electron densities are in preparation. Copies of these dictionaries can be found on the IUCr Web site in the official ASCII version suitable as input to an application, as well as in formatted versions that are designed to be easy to read.

Not all dictionaries need COMCIFS approval. Anyone can create a private dictionary provided that it does not contain data names that duplicate those already approved. Such a private dictionary may define items that are used locally, but if it defines items that are of more general interest it may be necessary to make the dictionary more widely available. The IUCr keeps a register of some of these private dictionaries, such as those that are used by structure-solving applications. Name clashes are prevented by the incorporation of unique character strings into the data names defined in the private dictionaries. A CIF may contain data names taken from more than one dictionary, either private or official. However, a program that reads a CIF needs to know only the names of the items it requires, since it can ignore any items whose data names it does not recognize. The presence of an unknown local data name does not therefore invalidate the CIF, but the information contained in that item will not be read unless the local dictionary is available.

It is in the structure of the dictionaries that the true potential of CIF as a crystallographic language lies. In order to make the extraction of information from a CIF as automatic as possible, CIF dictionaries are themselves stored as STAR files so that they can be consulted by the programs used to read, write and manipulate CIFs. To write a dictionary as a STAR file, a further dictionary defining the data names that appear in the dictionaries is needed. The material given in this dictionary, which assigns names to the properties listed in Fig. 3, is called the Dictionary Definition Language (DDL; Hall & Cook, 1995). There are currently two versions of this language, DDL1 and DDL2. The original version, DDL1, was used to define the dictionary of basic crystallographic items, known as the cif_core.dic (Fig. 4). DDL2 was developed in order to provide the much tighter definitions needed for macromolecular crystallography where automatic computer handling of information is required in large-scale operations such as the genome project.

Name	DDL	Remarks
cif_core.dic	1, 2	Defines basic crystallographic terms. Originally written in DDL1, a DDL2 version is incorporated in cif_mm.dic (Hall et al., 1991).
cif_mm.dic	2	Defines terms used in macromolecular crystallography. It includes most of the items in the cif_core.dic (Fitzgerald et al., 1998).
cif_pd.dic	1	Defines terms needed for describing powder patterns (Toby, 1998).
cif_img.dic	2	Contains the terms needed to define an image, particularly ones produced by 2-dimensional detectors. Because of the large size of images, a file format that permits a binary coding of an image CIF has also been defined (Westbrook, 2000).
cif_ms.dic	1	Defines terms needed for describing modulated structures (Madariaga, 2002).
cif_sym.dic	2, 1	Defines the basic symmetry terms. Written in DDL2 but a DDL1 version will also be available (Brown, 2001).

Figure 4
Currently approved CIF dictionaries.

Since the dictionaries themselves are written as STAR files, they can be read and manipulated by computer applications in the same way as CIFs. By reading both the CIF and its dictionary, it is no longer necessary to include the CIF data names in the code of an application, since both the names and their associated properties can be found in the dictionary. Future CIFs will identify the dictionaries used in their construction so that an application will be able to download any dictionary it needs from the Web. There is a protocol which allows several dictionaries to be merged into a single virtual dictionary in order to read or write a particular CIF. This means that the basic crystallographic information, such as the space group and lattice parameters, that is defined in the core dictionary can be used in conjunction with items defined in one of the more specialized dictionaries.

To give an example of how this might work, a program designed to create or edit a CIF would first find, load and merge all the dictionaries it needs. The data names stored in the resulting virtual dictionary are then displayed as needed on the screen, directing the user how to enter the values in the correct format. As the values are entered, the editor checks them against the dictionary and, when satisfied that they conform to the dictionary definition, it writes them to a syntactically correct STAR file. The resulting file could contain spectroscopic or thermodynamic as well as crystallographic information, providing suitable dictionaries had been loaded. To be properly described as a CIF, however, a file should contain at least one data name taken from an approved CIF dictionary.

In order to ensure that future generations will be able to access the current archive of CIFs, it is important that data names and their definitions be stable. They cannot be removed from the dictionaries nor can their definitions be changed. However, dictionaries have to evolve and this is achieved by adding new items or assigning new names to items that have undergone a change in meaning. Where a given data name has become obsolete through the natural evolution of the discipline, it is not removed from the dictionary though its use may be discouraged.[2] Because new data names are continually being added, it is wise to use the dictionary to supply the data

[2] More correctly, it is not removed from the stable of dictionaries that an application may consult in attempting to resolve a data name. In practice, deprecated items may be moved to ancillary dictionaries to trim the current active dictionaries.

names to an application, rather than relying on coding the data names directly into the program. In this way the application will not become obsolete as the CIF dictionaries evolve. At the same time, archived CIFs can still be read by loading the version of the dictionary used to prepare the CIF. For this reason, all versions of the dictionaries, both current and obsolete, are stored on the Web with stable URLs at the IUCr Web site. Several of the approved dictionaries already exist in more than one version, though every attempt is made to keep these upwardly compatible.

The crystallographers who prepared the official CIF dictionaries usually found the job to be a bigger challenge than they had anticipated. What at first looked like a simple exercise in writing definitions for the commonly used crystallographic terms, turned out in practice to be an in-depth exploration of the underlying concepts of crystallography and their interrelationships, ideas that we often take for granted and never fully explore. The definitions in a CIF dictionary have to be precise and unambiguous since related items must be compatible, meaning that the information that each item gives, and the form in which it appears, must be well defined. It is not sufficient, for example, to define an item such as the hypothetical `_atom_site_adp`, since atomic displacement parameters (adps) can be given in many different forms, such as *B*, *U* and two different and incompatible forms of beta. One could define a series of items, *e.g.* `_atom_site_B`, `_atom_site_U`, `_atom_site_beta1` and `_atom_site_beta2`, to allow the author to specify in which form the information is given, but this adds complexity to the dictionary and requires an unnecessary overhead since every application has to be able to read this information in all four formats. The philosophy adopted by CIF is, so far as possible, to define only one form for each item of information. The reader only needs to recognize this one form and the onus is then on the writer of the CIF to make the necessary conversion. There are a few exceptions where CIF allows the same information to be given in more than one form, but these are reluctant concessions to human weakness.

4. Relational structure of CIF

Each of the items stored in a CIF can be treated as an object whose properties, such as those listed in Fig. 3, are defined in the dictionaries. Among these properties are cross references to other data items that give CIF its relational character. These pointers link the information found in one loop, such as the table of bond lengths (`_geom_bond`, Fig. 1) with related information that appears in other loops, such as the table of atomic coordinates (`_atom_site`) and the table of symmetry operations (`_symmetry_equiv`). In the table of bond lengths, the labels identifying the bonded atoms (`_geom_bond_atom_site_label_1` and `_2`) must correspond to the labels of the individual atoms in the table of atomic coordinates (`_atom_site_label`), and the labels of the symmetry operations (`_geom_bond_site_symmetry_1` and `_2`) must correspond to the identifiers in the table of symmetry operations (`_symmetry_equiv_pos_site_id`). Although anyone with a crystallographic background will find this relationship obvious, computers cannot acquire background information by experience in the way that people can. Unless this knowledge is specifically loaded into the computer, the computer treats all objects as meaningless character strings. By reading a CIF dictionary, the computer has access to the data names of the pointers that link the bonds in the `geom_bond` loop with the corresponding items in the loops of atomic coordinates and symmetry operations, allowing it to retrieve the coordinates of the two atoms that form any given bond. This link between items in different parts of the data block gives the file the same structure as is found in a relational database. The explicit information about the pointers found in the dictionary is the first step towards supplying an application with the crystallographic knowledge that it needs to generate items of information that are not explicitly stored in the CIF.

In order to implement pointers in the dictionary, items that appear in the same loop in a CIF are defined as belonging to the same category. This allows the dictionary to identify the special role of the pointers in file management within the category. As an aid to people using the dictionaries, the category name usually (though not necessarily) forms the first element of the data name of an item within the category. Thus, `_atom_site_label` belongs to the `atom_site` category, `_geom_bond_atom_site_label_1` and `_geom_bond_site_symmetry_2` belong to the `geom_bond` category and `_symmetry_equiv_pos_site_id` belongs to the `symmetry_equiv` category.[3] The dictionary defines the properties of the categories and among these is the list of pointers (the category keys).

The pointers are hierarchically arranged, with the pointer in one category being nominated as the parent and pointers having the same values in other categories being nominated as the children. In the case of `_atom_site_label` the parent category is `atom_site` since the list of coordinates appearing in this category serves to define all the atoms in the structure. These atoms may or may not be mentioned in the child categories, since it is not necessary to give the lengths of bonds formed by all, or even any, of the atoms, but the presence of a list of bond lengths does imply that there exists a list of the atoms that form the bonds. Although such a list of atom sites must appear, it is not necessary to include their coordinates if these are not required or are not known.

5. CIF in crystallographic database and publishing activities

There can be no doubt that CIF has been successful in its initial goals of providing a standard and portable data interchange format between crystallographic applications, databases and publications. This is particularly the case for small-cell crystal structure determinations. From data collection at the diffractometer, through solution and refinement, almost all

[3] In future editions of the core dictionary, the `symmetry_equiv` category will be replaced with the category `space_group_symop` (Brown, 2001).

contemporary software is capable of writing CIF output, and most software can also import well formed CIFs. Details of the structures of small molecules may be deposited with the Cambridge Crystallographic Data Centre with CIF being the preferred format. IUCr journals require CIF format for supplementary information accompanying small-molecule or inorganic structure reports, while *Sections C* and *E* of *Acta Crystallographica* require the whole article to be submitted as a CIF. Many other journals welcome supplementary information in CIF format, and a number of journals encourage their authors or editors to use automated checking services that not only validate the syntax of such deposited files, but also test for their internal consistency and crystallographic plausibility. Details of structures may be output in CIF format from the Cambridge Structural Database (CSD; Allen, 2002; Bruno *et al.*, 2002) and the Inorganic Crystal Structure Database (ICSD; Belsky *et al.*, 2002). The Metals Data File (White *et al.*, 2002) and Powder Diffraction File (Faber & Fawcett, 2002) are also capable of processing material in this format.

Powder diffraction studies are well covered by the relevant CIF dictionary (Toby, 1998), which is used for submitting powder diffraction papers to *Acta Crystallographica*. The recently approved modulated-structures dictionary (Madariaga, 2002) will provide a standard method for reporting such structures in the journals, and is intended to be used for databases of modulated structures. This dictionary matches closely the requirements for describing modulated structures constructed by the IUCr Commission on Aperiodic Crystals (Chapuis *et al.*, 1997).

In the domain of biological macromolecules, CIF as a relative newcomer has had to compete with the existing Protein Data Bank (PDB) file format, originating in the early days of the Brookhaven Protein Data Bank (Abola *et al.*, 1987; Berman *et al.*, 2002). Much existing software still has to be modified to support the macromolecular CIF format (mmCIF; Fitzgerald *et al.*, 1998; Bourne *et al.*, 1997). Nevertheless, the current version of the Protein Data Bank managed by the Research Collaboratory for Structural Biology uses data processing engines based on mmCIF data structures, and macromolecular data sets are available to the public in mmCIF format. The mmCIF dictionary, designed as a superset of all the specific data fields present in files stored in PDB format, is used both as the basis for database schemata within the Protein Data Bank and as the basis for a common object resource dictionary for data exchange among a wider range of life-sciences disciplines (Greer, 2000).

Macromolecular structure work is aided by a CIF dictionary of terms describing measurements made using image plates (Westbrook, 2000) and an accompanying software library (Ellis & Bernstein, 2001) handling such data files in ASCII or compressed binary formats.

6. Data models and information interchange

As mentioned above, the CIF dictionaries and the Dictionary Definition Languages continue to evolve in response to the introduction of new concepts into crystallography and information handling. The very flexibility built into the structure of CIF invites this evolution and allows CIF to exploit new ideas in information technology. The standards will change but upward compatibility will be maintained.

The initial requirements for a standard crystallographic file structure addressed the need for uniform tagging of items of information to allow them to be exchanged between computer programs. Little thought was given at first to how these data items related to other components in the complete description of a crystal structure. However, the choice of the STAR file structure with its extensible tagging format was felicitous, because it rapidly became apparent that the STAR file lent itself to a variety of hierarchical, relational and object-oriented data models.

Early pilot applications of the STAR file structure in quantum electrochemistry and chemical structure description showed how the basic mechanism could be used to describe complete electron basis sets, or to pose chemical database queries based on Markush structures of generic formula. The same STAR structure has also been used to create NMR structural data files at the BioMagResBank of the University of Wisconsin (Ulrich *et al.*, 1996), a directory of personal contact information and professional interests within the crystallographic community (Epelboin, 1995, 1997), and, latterly, taxonomic descriptions in botany.

For crystallographic applications, CIF imposes some restrictions on the full functionality of the STAR file (Hall *et al.*, 1991), and the resulting file structure and the organization of data names or identifying tags are well suited to object-oriented or relational data models. In practice, the most fully developed data model for CIF has been the relational schema based on DDL2 devised for the mmCIF dictionary which, as already mentioned, forms the basis for the current Protein Data Bank operations. However, other studies and experiments have demonstrated the flexibility and extensibility of CIF data models.

An early set of CIF C++ object classes was subsequently reworked in Java and implemented as a component of a generic structured document browser (Murray-Rust, 1998). The object model of CIF has been developed in a production environment into a full class hierarchy and attendant application programming interface (Schirripa & Westbrook, 1996; Westbrook *et al.*, 1997). Murray-Rust (1998) has further explored the development of a document object model (DOM) for CIF that could form the basis of interoperability with applications based on the extensible markup language, XML, of the World Wide Web Consortium (W3C, 2001).

There is a complementarity between CIF and XML. XML, derived from the Standard Generalized Markup Language, SGML (ISO, 1986), is particularly well suited to the markup and classification of the contents of a document. While there is nothing to prevent numeric or mixed numeric/text data sets from being marked up as XML instances – and numerous individual examples of such data sets do exist – there are rather few fully developed XML document type definitions (DTDs) that provide tags for as rich a set of well defined data items as the CIF dictionaries provide in the field of crystal-

lography. Chemical Markup Language (CML; Murray-Rust & Rzepa, 1999, 2001) is an ambitious attempt to exploit XML in a broader chemical context and has won acceptance as a basis for chemical information exchange by the International Union of Pure and Applied Chemistry (Becker, 2000).

Of course, technical considerations do not of themselves guarantee the adoption of innovative solutions. The exploratory molecular STAR applications in chemistry described above built on the Standard Molecular Data (SMD) format (Bebak *et al.*, 1989; Barnard, 1990) to produce a chemical information exchange mechanism compatible with CIF. Many novel developments were tested during its early research stage. These included the creation of a Dictionary Definition Language which was subsequently used to build the CIF dictionaries, the use of the complete set of syntactic constructs permitted by STAR for encapsulating and nesting data, and the possible use of STAR as a query language. A subset of this material was published as the Molecular Information File (MIF; Allen *et al.*, 1995), but there has been no subsequent development of MIF tools or applications, despite its potential as an open and non-proprietary standard. The reasons for this must remain speculative, but the crystallography community, being both smaller and more homogeneous than the larger community of chemists, has been better placed to adopt new solutions. It will be interesting to see whether XML is adopted within the chemistry community as a collection of well integrated consensus DTDs covering separate topic areas, or whether individual application builders will tailor their own DTDs to afford different views of the same topic. The problems of integrating conflicting DTDs in the latter case should not be underestimated.

7. Future CIF developments

Including pointers in a CIF informs the computer which items are related, but it does not say what this relationship is. In current versions of the dictionaries, this information is found in the verbal definitions which are not designed to be computer interpretable. However, future versions of the dictionaries will include methods, that is algorithms, that will instruct the computer how to derive one quantity from others that are in the CIF (Spadaccini *et al.*, 2000). For example, if a user requests the density of a crystal and the density is not already present in the CIF, the application will look in the dictionary where it will find an algorithm that defines the density as `_chemical_formula_weight * _cell_formula_units_Z / _cell_volume`.

If any of these values is not present, the method property would contain the algorithm for calculating them from other items that are present.

With a dictionary that includes methods, an application would be able to use a CIF containing only the essential experimental measurements to obtain information about any property defined in the dictionary. The user would not need to know whether the item was already stored in the CIF. For example, someone interested in the refractive index could interrogate CIFs that have been retrieved from a database, using a private dictionary that included an algorithm for calculating the refractive index from the crystal structure. By merging the official CIF dictionary with a local dictionary containing the algorithm, the refractive indices of all the structures could be rapidly retrieved. Different models for calculating the refractive index could be compared by creating a number of different local data names, each containing a different algorithm. The application would read the input request, search the dictionary for the algorithm, calculate the required numbers and display the results.

We can look forward to the day when we will all have a generic STAR program that will understand the syntax of STAR and will therefore be able to read the dictionaries and use them to direct the retrieval of information from the CIFs. The result will be a revolution in the way we access the accumulated information available on crystal chemistry. The CIF dictionaries will constitute a knowledge base of crystallography that computers can access to retrieve any desired piece of crystallographic information. Though hidden from the user by the computer applications, CIF will be the underlying common language of crystallography.

8. Further details

Further details of the CIF standard, including both ASCII and formatted versions of the approved CIF dictionaries, can be obtained from the International Union of Crystallography Web site: http://www.iucr.org.

We wish to acknowledge the efforts of all the people, too numerous to name, who have contributed to the work described in this paper.

References

Abola, E. E., Bernstein, F. C., Bryant, S. H., Koetzle, T. F. & Weng, J. (1987). *Crystallographic Databases*, edited by F. H. Allen, G. Bergerhoff & R. Sievers, pp. 107–132. Chester: IUCr.

Allen, F. H. (2002). *Acta Cryst.* B**58**, 380–388.

Allen, F. H., Barnard, J. B., Cook, A. P. F. & Hall, S. R. (1995). *J. Chem. Inf. Comput. Sci.* **35**, 412–427.

Allen, F. H., Bellard, S., Brice, M. D., Cartwright, B. A., Doubleday, A., Higgs, H., Hummelink, T., Hummelink-Peters, B. G., Kennard, O., Motherwell, W. D. S., Rodgers, J. R. & Watson, D. G. (1979). *Acta Cryst.* B**35**, 2331–2339.

Barnard, J. (1990). *J. Chem. Inf. Comput. Sci.* **30**, 81–96.

Bebak, H., Buse, C., Donner, W. T., Hoever, P., Jacob, H., Klaus, H., Pesch, J., Roemelt, J., Schilling, P., Woost, B. & Zirz, C. (1989). *J. Chem. Inf. Comput. Sci.* **29**, 1–5.

Becker, E. D. (2000). *Chem. Int.* **23**, 135.

Belsky, A., Hellenbrandt, M., Karen, V. L. & Luksch, P. (2002). *Acta Cryst.* B**58**, 364–369.

Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., Jain, S., Fagan, P., Marvin., J., Ravichanran, V., Schneider, B., Thanki, N., Padilla, D., Weissig, H., Westbrook, J. D. & Zardecki, C. (2002). *Acta Cryst.* B**58**, 899–907.

Bourne, P. E., Berman, H. M., McMahon, B., Watenpaugh, K. D., Westbrook, J. D. & Fitzgerald, P. D. M. (1997). *Methods Enzymol.* **277**, 571–590.

Brown, I. D. (1983). *Acta Cryst.* A**39**, 216–224.

Brown, I. D. (2001). *The Symmetry CIF Dictionary*, ftp://ftp.iucr.org/pub/cif_sym.dic.
Bruno, I. J., Cole, J. C., Edgington, P. R., Kessler, M., Macrae, C. F., McCabe, P., Pearson, J. & Taylor, R. (2002). *Acta Cryst.* B**58**, 389–397.
Chapuis, G., Farkas Jahnke, M., Pérez-Mato, J. M., Senechal, M., Steurer, W., Janot, C., Pandey, D. & Yamamoto, A. (1997). *Acta Cryst.* A**53**, 95–100.
Ellis, P. J. & Bernstein, H. J. (2001). *CBFlib: An API for CBF/imgCIF Crystallographic Binary Files with ASCII Support*, http://www.bernstein-plus-sons.com/software/CBF/Doc/DBFlib.html.
Epelboin, Y. (1995). *World Directory of Crystallographers and of Other Scientists Employing Crystallographic Methods*, 9th ed. Dordrecht: Kluwer Academic Publishers.
Epelboin, Y. (1997). *World Directory of Crystallographers and of Other Scientists Employing Crystallographic Methods*, 10th ed. Dordrecht: Kluwer Academic Publishers.
Faber, J. & Fawcett, T. (2002). *Acta Cryst.* B**58**, 325–332.
Fitzgerald, P., Berman, H. M., Bourne, P. E., Watenpaugh, K. D. & Westbrook, J. D. (1998). *The Macromolecular CIF Dictionary*, ftp://ftp.iucr.org/pub/cif_mm.dic.
Greer, E. S. (2000). *Macromolecular Structure RFP Response*, revised submission to OMG document lifesci/99-08-15, http://pdb.rutgers.edu/mmcif/dictionaries/ascii/lifesci_00_08_01.pdf.
Hall, S. R. (1991). *J. Chem. Inf. Comput. Sci.* **31**, 326–333.
Hall, S. R., Allen, F. H. & Brown, I. D. (1991). *Acta Cryst.* A**47**, 655–685.
Hall, S. R. & Cook, A. P. F. (1995). *J. Chem. Inf. Comput. Sci.* **35**, 819–825.
ISO (1986). *Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML)*. ISO Standard 8879.
Madariaga, G. (2002). *CIF Dictionary for Modulated Structures*, ftp://ftp.iucr.org/pub/cif_ms.dic.
Murray-Rust, P. (1998). *Acta Cryst.* D**54**, 1065–1070.
Murray-Rust, P. & Rzepa, H. S. (1999). *J. Chem. Inf. Comput. Sci.* **39**, 928–942.
Murray-Rust, P. & Rzepa, H. S. (2001). *J. Chem. Inf. Comput. Sci.* **41**, 1113–1123.
Schirripa, S. & Westbrook, J. D. (1996). *CIFOBJ. A Class Library of mmCIF Access Tools. Reference Guide, CIFOBJ v. 1.01*. Technical Report NDB-269. Rutgers University, New Brunswick, NJ, USA.
Spadaccini, N., Hall, S. R. & Castleden, I. R. (2000). *J. Chem. Inf. Comput. Sci.* **40**, 1289–1301.
Toby, B. H. (1998). *Powder CIF Dictionary*, ftp://ftp.iucr.org/pub/cif_pd.dic.
Ulrich, E. L., Agentar, D., Klimowicz, A., Westler, W. M. & Markley, J. L. (1996). *Acta Cryst.* A**52**, C-577.
W3C (2001). *Extensible Markup Language (XML)*, http://www.w3c.org/xml/.
Westbrook, J. D. (2000). *Image CIF Dictionary (imgCIF) and Crystallographic Binary File Dictionary (CBF) Extending the Macromolecular CIF Dictionary (mmCIF)*, ftp://ftp.iucr.org/pub/cif_img.dic.
Westbrook, J. D., Hsieh, S.-H. & Fitzgerald, P. D. M. (1997). *J. Appl. Cryst.* **30**, 79–83.
White, P. S., Rodgers, J. R. & Le Page, Y. (2002). *Acta Cryst.* B**58**, 343–348