International Tables for Crystallography (2006). Vol. G, Section 2.4.4, pp. 45–47.

During this period, the IUCr Working Party on Crystallographic Information had commissioned one of us (SRH) to coordinate the development of a universal file to replace the existing fixed-format Standard Crystallographic File Structure (SCFS: Brown, 1988). As documented in Chapter 1.1, the CIF approach was adopted as the international standard in 1990 and published by Hall *et al.* (1991). Although the small-molecule CIF is able to store a representation of 2D chemical topology, its data definitions do not meet all the needs of the chemical community. In 1991, the IUCr became interested in further extending CIF into the chemical arena and discussions took place between representatives of the CIF project and of the SMD Technical Working Group. These meetings decided that an integration of the SMD format and the STAR syntax was desirable because it provided a number of advantages over the existing SMD specifications (Barnard & Cook, 1992). In particular, SMD/STAR provides for a clearer separation of the data structure and the data content roles, together with more flexible data extensibility in future versions. In addition, automated data validation of STAR/SMD files is possible using electronic data dictionaries. In a wider context, there were obvious opportunities for integrating with other applications of the STAR File.

2.4.3. MIF objectives

Molecular information embraces a broad spectrum of data related to chemical and molecular structure. It includes both individual and linked data items, *inter alia* spectroscopic measurements, thermochemical data, electrochemical properties, crystal structure information and so on. These items represent the data descriptors of molecular chemistry and it is intended that all of these will eventually be accommodated in the MIF approach. However, the initial MIF implementation (Allen *et al.*, 1995), summarized in this chapter, treated only the important core information: the data items needed to specify the connectivity and stereochemistry of molecules and their 2D and 3D spatial representations. The MIF data items needed for more extensive applications must, in the future, involve the collaborative efforts of informatics and database experts from chemical industry and academia.

A dictionary of the initial MIF core data items described in this paper is given in Chapter 4.8. This is the abbreviated text version of the definition attributes contained in the electronic dictionary file. The core MIF data items provide descriptors for representing the 2D connectivity of a molecule or substructure, the conventions for relative or absolute stereochemical relationships, and the coordinates and conventions used for the generation of 2D and 3D graphical depictions. These data items apply to complete molecules, or to substructures with incomplete or variable attributes. As a consequence they are well suited for query definitions in substructure search systems, a feature that will be discussed later in this chapter.

2.4.4. MIF concepts and syntax

The syntax of the Molecular Information File is based on that of the STAR File (Hall, 1991; Hall & Spadaccini, 1994). A MIF is an ASCII text file that can be read or amended using a standard text editor, and that can be processed computationally without conversion to another format. The organization and expression of MIF data is summarized in Table 2.4.4.1. Each file consists of a series of data blocks and each block consists of a series of individual data items. There may be any number of items within a block and any number of blocks within a file. A data block represents a logical grouping of data items and, in most MIF applications, a data block will usually specify a complete chemical entity, *i.e.* a fully defined molecule or a query substructure.

Table 2.4.4.1. *Brief overview of the MIF syntax*

A text string is a string of characters bounded by white space, single or double quotes, or semicolons in column 1.
A data name is a text string bounded by white space starting with an underline.
A data value is a text string not starting with underline, preceded by an identifying data name.
A list is a sequence of data names, preceded by '`loop_`' and followed by a list of data values.
A save frame is a collection of data within a data block, preceded by '`save_framecode`' and closed with '`save_`'.
A data block is a collection of data, preceded by '`data_blockcode`'.
A global block is a collection of data, preceded by `global_`, that is common to all subsequent data blocks.
A file may contain any number of data blocks or global blocks.
A data name must be unique within a data block.

The MIF syntax, unlike that of a CIF, places no restrictions on line lengths or nested loop levels. For a detailed understanding of the differences between a MIF and a CIF, the reader should compare this chapter with Chapter 2.2 or refer to the published details of the STAR syntax (Hall & Spadaccini, 1994), the specification of the CIF core data items (Hall *et al.*, 1991) and the Dictionary Definition Language (Hall & Cook, 1995) used to define data items in the electronic version of a STAR dictionary.

CIF data, described by over a thousand items in the current dictionaries (see Part 4), encompass the fields of crystallographic structure and diffraction techniques, and these data items could readily be incorporated into a MIF. It should be noted, however, that currently the reverse is not possible because the current CIF syntax does not support nested loops or save frames.

2.4.4.1. Data identification

The fundamental principle that underpins MIF is exactly as for CIF: every data item is represented by a unique data tag followed by its associated data value. These combinations are referred to as tag–value pairs or tuples. Data names must start with an underscore (*i.e.* underline) character and data values may be any type of string, ranging from a single character to many lines of text. Here are some simple examples of MIF data items:

```
_atom_mass_number       79
_atom_type              Se
_display_colour         blue_medium
```

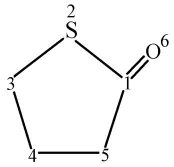
The complete list of MIF core data items is given in Chapter 4.8.

2.4.4.2. Looped lists

Repetitive data are stored in a MIF as lists of values, as they are in a CIF. Each list is prefaced by a `loop_` statement and a sequence of data names that identify the data values that follow in 'packets' of equal length. The values in each packet match the order and number of the data names. Any number of packets may appear in a looped list.

Atom and bond properties are typical of the information to appear in a looped list. The atoms and bonds of thiabutyrolactone in MIF format are shown in Fig. 2.4.4.1. The description of each data item in this example is given in Chapter 4.8, although the meanings are clear from the self-descriptive data names. The number of data values in each list is an exact multiple of the number of data names at the start of each loop structure. Looped lists are terminated by the next list or by any other data name, data block or end of file. Comments may be included in a MIF and are preceded by a `#` character, as illustrated in Fig. 2.4.4.1.

Hierarchical data may require the use of nested loop structures (see the `_display_*` loop in Fig. 2.4.4.2). Note that the packet for `_display_id` of 7 has two sets of `_display_conn_` values giving



```
# atom attributes list
     loop_
         _atom_id
         _atom_type
         _atom_attach_h
                  1  C  0     2  S  0     3  C  2
                  4  C  2     5  C  2     6  O  0

# bond attributes list
     loop_
         _bond_id_1
         _bond_id_2
         _bond_type_mif
                  1  2  S     2  3  S     3  4  S
                  4  5  S     5  1  S     1  6  D
```

Fig. 2.4.4.1. MIF coding of atom and bond properties for thiabutyrolactone.

connections to atom sites 1 and 4 (the other connections to site 7 appear in the next two packets). Data items that appear in looped lists are identified in the MIF dictionary (see Chapter 4.8) as having the attribute `_list` set to either 'yes' or 'both'. Other relationships between looped data items are also specified in the dictionary.

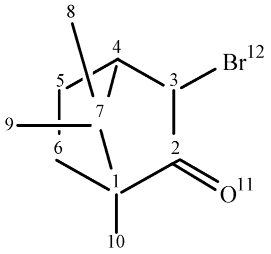
2.4.4.3. Save frames

Save frames are employed in a MIF to encapsulate grouped data for efficient cross-referencing. If a set of data needs to appear repeatedly in a data application, it is efficient to place this data into an addressable save frame. Molecular fragments, such as amino-acid units, are a case in point. A save frame is bounded by the statement **save_*framecode*** and terminated by a **save_** statement. It can be referenced within the parent data block using the value **$*framecode*** where the *framecode* matches the string in the **save_*framecode***. Note that all data names must be unique within a save frame, but the same data names may appear in other save frames or in the parent data block. Save frames may not contain other save frames but save-frame references (**$*framecode***) may appear in other save frames.

Save frames can be used in a MIF for many purposes. A simple application, the storage of alternative 3D conformational representations describing cyclohexane, is illustrated in Fig. 2.4.4.3. Within the STAR syntax, save-frame references (**$*framecode***) may occur before or after the save-frame definition within any data block. MIF preserves this basic STAR syntax. Save frames are particularly useful for defining commonly referenced structural templates and examples of this facility are discussed and illustrated (Figs. 2.4.7.1 and 2.4.7.2) in Section 2.4.7.

2.4.4.4. Data blocks

A data block is a sequence of unique data items or save frames. It is opened with a **data_*blockcode*** statement and closed by another data-block statement or a **global_** statement (see below). The *blockcode* string identifies the block within the file. Examples of data blocks are shown in Figs. 2.4.4.2, 2.4.4.3 and 2.4.6.1. Each data block in a file must have a unique *blockcode*.



```
data_bromocamphor

loop_
    _atom_id
    _atom_label
    _atom_type
    _atom_attach_h
    _atom_coord_x
    _atom_coord_y
    _atom_coord_z
          1  C1  C  0  4.69027   2.57756   2.10705
          2  C2  C  0  3.61112   2.44777   3.16754
          3  C3  C  1  4.16317   2.26258   0.69475
          4  C4  C  1  5.39943   1.87018  -0.13167
          5  C5  C  2  6.48959   2.15784   0.79703
          6  C6  C  2  6.57842   3.69178   1.11990
          7  C7  C  0  5.27609   3.93542   1.94650
          8  C8  C  3  5.93386   1.67891   2.14924
          9  C9  C  3  5.57194   0.17837   2.14326
          10 C10 C  3  6.85298   2.00528   3.35651
          11 O2  O  0  3.03484   2.36201   0.29318
          12 BR3 Br 0  5.41337   2.68686  -1.87368

loop_
    _bond_id_1
    _bond_id_2
    _bond_type_mif
          1  2  S   2  3  S   3  4  S   4  5  S
          5  6  S   6  1  S   1  7  S   7  8  S
          7  9  S   1 10  S   2 11  D   3 12  S
          4  7  S

_display_define_scale      50
_display_define_span_x    500
_display_define_span_y    500

loop_
    _display_id
    _display_object
    _display_symbol
    _display_colour
    _display_size
    _display_coord_x
    _display_coord_y
    loop_
        _display_conn_id
        _display_conn_symbol
        _display_conn_colour
  1     .  .      .  .  251 195 2 1b black stop_
  2     .  .      .  .  334 244 3 1b black stop_
  3     .  .      .  .  334 339 4 1b black stop_
  4     .  .      .  .  251 387 5 1b black stop_
  5     .  .      .  .  168 339 6 1b black stop_
  6     .  .      .  .  168 244 1 1b black stop_
  7     .  .      .  .  217 292 1 1b black
                                4 1b black stop_
  8     .  .      .  .  191 426 7 1b black stop_
  9     .  .      .  .  100 292 7 1b black stop_
  10    .  .      .  .  251 100 1 1b black stop_
  11 text  O   blue 10  401 206 2 2b black stop_
  12 text Br yellow 10  417 387 1 1b black stop_
```

Fig. 2.4.4.2. MIF coding of atom properties (including 3D coordinates), bond properties and display information for (+)-3-bromocamphor.

```
data_cyclohexane

_molecule_name_common           cyclohexane

    loop_
        _atom_id
        _atom_type
        _atom_attach_h         1  C  2   2  C  2   3
                   C  2   4  C  2   5  C  2   6  C  2
loop_
    _bond_id_1
    _bond_id_2
    _bond_type_mif
           1 2 S  2 3 S  3 4 S  4 5 S  5 6 S  6 1 S

loop_
    _reference_conformation
                       $chair   $boat   $twisted_boat

save_chair
    loop_
        _atom_id
        _atom_coord_x
        _atom_coord_y
        _atom_coord_z    1   1.579  0.159  0.263
                         2   0.756  0.507 -0.986
                         3   0.825  0.493  1.541
                         4  -0.549 -0.131  1.590
                         5  -1.377  0.222  0.347
                         6  -0.626 -0.158 -0.937
save_

save_boat
    loop_
        _atom_id
        _atom_coord_x
        _atom_coord_y
        _atom_coord_z    1   1.657 -0.426  0.356
                         2   1.031  0.133 -0.927
                         3   0.960  0.133  1.602
                         4  -0.568 -0.040  1.558
                         5  -1.051 -0.738  0.279
                         6  -0.499 -0.028 -0.964
save_

save_twisted_boat
    loop_
        _atom_id
        _atom_coord_x
        _atom_coord_y
        _atom_coord_z    1   0.933  0.922  0.971
                         2   1.186  0.220 -0.368
                         3  -0.119  0.161  1.796
                         4  -1.135 -0.581  0.911
                         5  -1.371  0.181 -0.397
                         6  -0.083  0.236 -1.238
save_
```

Fig. 2.4.4.3. Atom and bond properties for cyclohexane, together with 3D coordinate representations of three alternative conformations: chair, boat and twisted boat.

2.4.4.5. Global blocks

A global block is similar to a data block except that it is opened with a **global_** statement and contains data that are common or 'default' to all subsequent data blocks in a file. Global data items remain active until re-specified in a subsequent data block or global block.

In some applications it may be efficient to place data that are common to all data blocks within a global block. In particular, save frames may be defined within global blocks and then referenced in subsequent data blocks [this statement corrects an error in Hall & Spadaccini (1994)]. Examples of global data are shown in Figs. 2.4.7.1 and 2.4.7.2, in which a variety of frequently referenced structural units are encapsulated within save frames specified in global blocks.

2.4.5. Atoms, bonds and molecular representations

The MIF dictionary (see Chapter 4.8) contains definitions of the principal data items needed to specify molecular connectivity and spatial representations. These definitions are grouped according to purpose or, as referred to in the DDL dictionary language (Hall & Cook, 1995), by category. Categories are formally specified in the MIF dictionary using the data attribute **_category** but they may also be identified from the data-name construction '***_<category>_<subcategory>_<descriptor>***'. Note that data items appearing in the same looped list must belong to the same category.

The values of some data items are restricted, by definition in the MIF dictionary, to standard codes or states. For example, the item **_bond_type_mif** can only have values S, D, T or O as in its dictionary definitions:

S: single (two-electron) bond;
D: double (four-electron) bond;
T: triple (six-electron) bond;
O: other (*e.g.* coordination) bond.

The MIF dictionary plays the important additional role of validating and standardizing data values. This is illustrated with the data item **_display_colour**, which identifies the colours of 'atom' and 'bond' graphical objects. The colour codes or states for this item are specified in its dictionary definitions as a set of permitted red/green/blue (RGB) ratios, and no other colours may be used in a MIF. This has the technical advantage of making colour states searchable for chemical applications.

Fig. 2.4.4.2 shows MIF data for the molecule (+)-3-bromocamphor. The 'atom' list contains the items **_atom_id**, **_atom_type** and **_atom_attach_h**, which identify the chemical properties of the atoms, plus the items **_atom_coord_x**, ***_y** and ***_z**, which specify the 3D molecular structure in Cartesian coordinates [these are taken from diffraction results (Allen & Rogers, 1970)]. The item **_atom_label** is also used with any graphical depiction of the 3D model. The 'bond' loop in this example uses the simple **_bond_type_mif** conventions described above. The data names needed to depict stereochemistry are discussed with examples (Figs. 2.4.8.1, 2.4.8.2 and 2.4.8.3) in Section 2.4.8.

The MIF approach to representing 2D chemical structure separates the specification of chemical atom and bond properties. This provides additional flexibility in the description of the graphical objects, such as atomic nodes and bonded connections. The MIF data required to generate a 2D chemical diagram are shown in Fig. 2.4.4.2. The diagram generated from this data will be in a display area of 500 × 500 coordinate units at a scale of 50 units per cm (the 2D chemical diagram shown in Fig. 2.4.4.2 is not to this scale). The default origin (the bottom left corner of the display area) can be specified with the item **_display_define_origin**. The data used to depict a 2D structure form a two-level loop with the 'atomic' graphical objects at level 1 and the 'bond' graphical objects at level 2. The item **_display_object** has the values '.' (null or no object), 'text' (an element or number string) or 'icon'. The size and colour of the atom site are specified with **_display_size** and **_display_colour**. The bonds connected to each atom site are specified as a sequence of **_display_conn_id** numbers (in loop level 2). These numbers must match one of the