Vol. 16 no. 2 2000
Pages 159–168

STAR/mmCIF: An ontology for macromolecular structure

John D. Westbrook[1] and Philip E. Bourne[2,*]

[1]Rutgers, The State University of New Jersey, Department of Chemistry, 610 Taylor Road, Piscataway, NJ 08854-8087, USA, and [2]San Diego Supercomputer Center and Department of Pharmacology, University of California, San Diego, 9500 Gilman Drive, La Jolla CA 92093-0505, USA and The Burnham Institute, 10901 N. Torrey Pines Road, La Jolla CA 92037, USA

Received on July 7, 1999; revised on August 16, 1999; accepted on September 14, 1999

Abstract

Motivation: *Crystallographers were motivated 10 years ago to develop a simple and consistent data representation for the exchange and archiving of data associated with the crystallographic experiment and the final structure. As this process evolved (and the data grew at near exponential rates) came the recognition that this representation should also facilitate the automated management of the data and, with the aid of additional software for verification and validation, provide improved consistency and accuracy and hence improved scientific inquiry. This realization led to a new Dictionary Definition Language (DDL) and an extensive dictionary based on this DDL for describing macromolecular structure. In broad terms this could be considered an ontology. An important feature in the development of the ontology was the endorsement and ongoing maintenance and support of the International Union of Crystallography (IUCr). While the description of macromolecular structure and the x-ray crystallographic experiment used to derive it represent explicit data, the ontology is extensible and applicable to other less well-characterized data domains.*

Results: *Details of the DDL, the dictionaries that have been developed, and software for reading and using this ontology are presented.*

Availability: *Extensive documentation, software tools and the DDL and dictionaries are available from http://ndbserver.rutgers.edu/mmcif and associated mirror sites.*

Contact: *Bourne: bourne@sdsc.edu and Westbrook: jwest@rcsb.rutgers.edu*

Introduction

Ontologies have previously been described by Guarino (1996) and implemented in systems such as Ontolingua (http://ontolingua.stanford.edu). However, within the bioinformatics community the term ‘ontology’ has been used to mean different things. This was apparent during a session devoted to ontologies at the 1998 Intelligent Systems for Molecular Biology (ISMB) conference in Montreal (http://www-lbit.iro.umontreal.ca/ISMB98/). In part this reflected the diverse backgrounds of the audience who ranged from computer scientists to experimental molecular biologists. We did, however, share one common albeit broad desire for what an ontology should be—the unambiguous definition of biological data within a given scope. As pragmatists these authors believe that the encoding rules embodied in Self-defining Text Archival and Retrieval (STAR) and applied to define a dictionary definition language (DDL) and the macromolecular Crystallographic Information File (mmCIF) dictionary do meet this definition and we describe it here. Certainly STAR/DDL/mmCIF complies with Schulze-Kremer’s description of an ontology as ‘a concise and unambiguous description of what principle entities are relevant to an application domain and the relationship between them.’ Although many users would argue that this ontology is not concise, and computer scientists would likely argue that the expression of the relationships between entities is neither formal nor well described. Nevertheless, STAR/DDL/{mm}CIF has proven useful to crystallographers and informaticists, working with small molecules and working with biological macromolecules. Moreover, since it forms the foundation for the new Protein Data Bank (PDB) as maintained by the Research Collaboratory for Structural Bioinformatics (RCSB; http://www.rcsb.org), of which we are members, it is worth some discussion as to how it arose, how it is evolving, and what it means to structural biology.

History

In the late 1980s the International Union of Crystallography (IUCr) established a committee to develop a general purpose data exchange format for the exchange of data

*To whom correspondence should be addressed.

associated with a small molecule single crystal x-ray diffraction experiment. This exchange format became known as the Crystallographic Information File (CIF), which consisted of a set of encoding rules defined by STAR, DDL (Hall and Cook, 1995) based on these encoding rules and a data dictionary conforming to the DDL. The IUCr at their triennial congress approved version 1.0 (v1.0) of this dictionary in 1990. Details of the 1200 data definitions in this dictionary were later published (Hall *et al.*, 1991). The dictionary was quickly adopted for two reasons. The lesser reason being that the dictionary was endorsed by a strong scientific society. The major reason being a reduced time to publication when submitting data to the journal *Acta Crystallographica* C in the form of a CIF. The data, notably the atomic Cartesian coordinates, are required for validation and archiving prior to a paper being accepted. Submitting this information in a form conforming to CIF assured more timely processing and hence publication of accepted papers. In later years the IUCr established two servers which accept CIF files, one for geometric validation of atomic coordinates and one for producing a final version of the paper in a form identical to the final journal article. The validation report returned to the submitter is identical to that given to the referee when the paper is reviewed so the author has at least some indication of whether the paper will be accepted prior to publication. Once the paper is accepted the CIFs are passed to the Cambridge Crystallographic Data Center (CDCC) for loading into their databases.

While it is true that small molecule crystallography is a particularly quantitative science which permits a large part of the experiment to be defined in exact terms, the same basic idea could be applied to less quantitative data. Extending this idea of an interchange format, an IUCr sponsored committee was formed in 1990 to develop an extension to the CIF dictionary specific to biological macromolecules. This became known as mmCIF. What initially appeared to be a small task became 7 years of dedicated work by a small group in defining the dictionary terms (Fitzgerald *et al.*, 1992). The end result was v1.0 of the mmCIF dictionary that comprised 1700 terms and was ratified by an IUCr committee (COMCIFS) established to approve these dictionaries. (By the time the mmCIF dictionary was complete a number of other dictionary efforts of interest to the IUCr had begun, see below.)

Perhaps the greatest contribution of this work is the dictionary itself. As discussed later, the emergence of new technologies in the future, and/or inherent limitations in CIF when applied to the more general STAR encoding rules, may limit the use of the DDL, but the detailed description of a biological macromolecule and the experiment used to resolve it is likely to last much longer. In short, the dictionary terms may be cast into a different form, but the generalized way of describing any biological macromolecule is likely to persist.

While developing v1.0 of the mmCIF dictionary there was a fundamental realization that the DDL used to describe the small molecule structure and experiment (Hall and Cook, 1995) was too informal. In short, it left too much of the decision about how to interpret an item of data in the hands of the programmer, not to a rigorous machine-readable set of definitions. Different programmers could interpret the item of data differently leading to different software producing different results when using what, in fact, were the same items of data. The problem was compounded when trying to load the data into a database, since relationships between items of data were specified in a non-formal way. An interchange format is only of limited use when human intervention is required to make the exact interpretation between data being written and data being read. This is less of a problem in a single discipline where those involved 'speak the same language,' but for a multi-disciplinary audience it was insufficient.

At this juncture the emerging macromolecular structure dictionary could have been cast into a different form based, for example, on Abstract Syntax Notation (ASN.1), a Unified Modelling Language (UML) or an Object Modelling Technique (OMT), all of which were well established. However, it was felt by the primary developers at the time that the best approach was to stay with a standard defined by the crystallographic community. Hence, DDL version 2.0 (v2.0) (Westbrook and Hall, 1995) was developed. Version 2.0 of DDL addressed these shortcomings using the same STAR encoding rules as DDL v1.*x* (there were subsequent releases of the dictionary by this time) and included definitions to provide mapping between terms described in DDL v1.*x* and v2.0. Version 2.0 of DDL defines data dictionaries and associated data files, both of which are easily mapped into a relational data model. There is the notion of categories (tables) with primary and foreign keys. Additionally there is the notion of data hierarchies (category groups, categories, and sub-categories). Categories are self- defining—all categories are members of the ddl group and the ddl group is a member of itself.

In brief the STAR encoding rules and the DDL provide a notion of scope that is used to segment data and create associations between segments of data. There is the notion of sets and allowed ranges that can be used to enumerate data. Thus, the dictionary can maintain allowable ranges of values for a particular data item useful in validating data during reading. Since this is defined in the dictionary which is external to any software program different software can easily use the same validation criteria. There is the notion of units and units conversion to be applied to items of data. The latter implies the application of methods to items of data that can be expressed in a general form. While this has not been used extensively in the current dictionaries, it has been used externally and is described subsequently.

It is not the purpose of this paper to detail the form and content of the mmCIF dictionary. The purpose is to provide an overview of what STAR/DDL/mmCIF provides and to direct the reader to Web-accessible resources for further detail. The mmCIF Web resource (http://ndbserver.rutgers.edu/mmcif) provides an entry point to a full specification of the DDL, the mmCIF dictionary, software that uses it, and various tutorials including one for writing other dictionaries using DDL v2.0.

Relationship to the PDB and the PDB format

The RCSB, under a contract from the US government, has assumed the operation of the PDB which was previously operated by Brookhaven National Laboratory (BNL). Internally the RCSB uses mmCIF as a way to represent structural data. Users, on the other hand, are most familiar with the PDB format since the majority of programs use that format. Several PDB formats have appeared over the years. Each incorporating new features of an emerging discipline and each intended to be upwardly compatible. These format changes, while welcome, suffered from not being specified in a formal way and hence are not defined explicitly for use by a computer program. Interpretation is left to the programmer having consulted a PDB Guide to Authors.

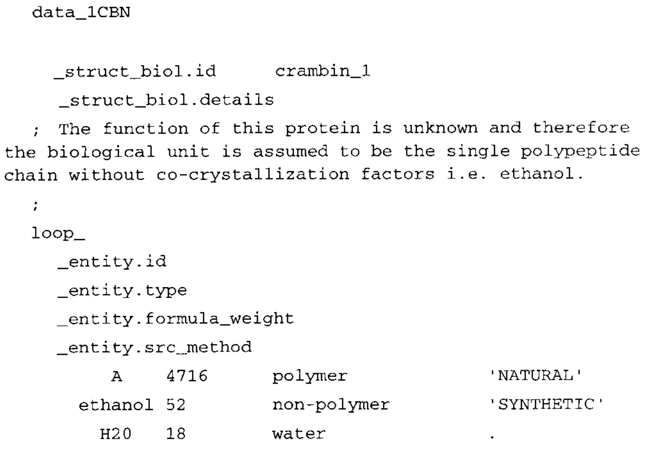
For reasons of backward compatibility, the RCSB continues to write and distribute data using PDB v2.2 format, but at the same time is about to provide data conforming to the mmCIF dictionary. That PDB files are produced consistently from an mmCIF representation. Although it should be noted the converse is not possible—the informality of the PDB format prevents consistent automatic conversion of structure data represented in a PDB format to mmCIF. Internal use of mmCIF permits a more detailed and consistent means of representing macromolecular structure data. While mmCIF provides a richer and more consistent data definition, it is recognized that good software tools will be needed if mmCIF is to be adopted more widely outside of the PDB. This issue is discussed further below.

Results

A summary of the features provided by STAR/DDL/mmCIF are now given.

STAR

Self-defining Text Archival and Retrieval (STAR) was first described by Hall (1991) as a simple, general, upward compatibly and flexible means of representing electronic data which could be read by human or machine. This was later expanded upon by Hall and Spadaccini (1994) and expressed in a Backus–Naur form Aho *et al.* (1985). Figure 1 illustrates a simple example of the encoding rules embodied by STAR when used to represent a data file containing information on a protein.

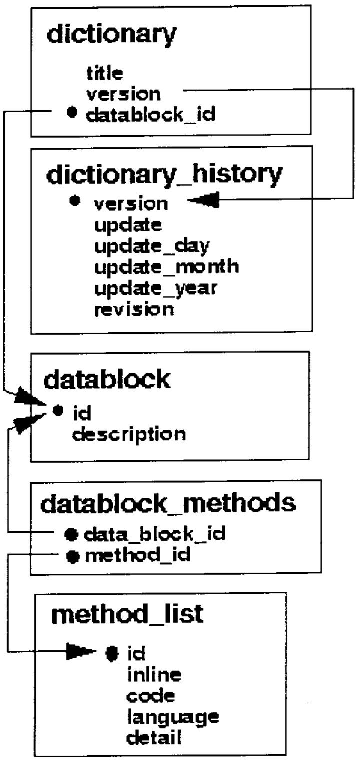


```
data_1CBN


   _struct_biol.id     crambin_1
    _struct_biol.details
 ;  The function of this protein is unknown and therefore
the biological unit is assumed to be the single polypeptide
chain without co-crystallization factors i.e. ethanol.

 ;
 loop_
    _entity.id
    _entity.type
    _entity.formula_weight
    _entity.src_method
        A     4716      polymer             'NATURAL'
      ethanol 52        non-polymer         'SYNTHETIC'
        H2O   18        water               .
```

Fig. 1. Fragment of an mmCIF containing data on the protein crambin (Teeter *et al.*, 1993) illustrating STAR encoding rules.

A set of *name-value* pairs is defined by STAR. Each name-value pair is referred to as a *data item*. Thus, a data item is identified by its value and the unique name that the value has associated with it. Names are distinguished from values by the use of a leading underscore (_). Syntax and semantics are clearly separated since any semantics associated with the data item are defined in specific dictionaries, as defined above and discussed below. Name-value pairs are enclosed in a data block which defines the scope of information being conveyed. A data block starts with a *data_blockcode* tag, where *blockcode* is a unique identifier for the data block (in this instance a PDB id code) and is followed by the associated name-value pairs. A data block ends when another data block starts or at an end-of-file. A STAR file consists of one or more data blocks and an optional leading global data block (not shown), which contains information that is applicable across multiple data blocks in a STAR file. A global data block is identified by the *global_blockcode* statement. A *save frame* is an optional referenced subcomponent nested inside a data block. A save frame starts with a *save_savecode* where *savecode* is an identifier used to reference a save frame within a data block. A save frame ends with the reserved word *save_*. Repetitive data items can be packaged into a loop structure contained within a single data block. A data loop structure consists of a *loop_* statement followed by a list of data names and then

(a)



(b)
```
loop_
        _dictionary_history.version
        _dictionary_history.update
        _dictionary_history.revision

    1.0.00 1997-10-14
    ;
      Changes (PMDF):
      + Editorial changes in light of proof-reading by B.
  McMahon and I.D. Brown
      + No data names or enumeration values have changed, with the
  exception of the addition of the enumeration value 'other' to
  exptl.method
      + The bulk of the changes, which are too numerous to list there,
  were fixing spelling and grammar errors, and providing missing
  definitions
      + In a few cases, data definitions were reworded for clarity
      + Replaced erroneous occurances of Y~calc~ with Y~obs~ in the
  defintions of weighted R factors (Ian Tickle)
    ;
```

(c)
```
save_DICTIONARY_HISTORY

    _category.description
;
   Attributes for specifying the revision history of the dictionary
;

    _category.id                       dictionary_history
    _category.mandatory_code           no
    _category_key.name                '_dictionary_history.version
     loop_
    _category_group.id
                                      'ddl_group'
                                      'dictionary_group'
     save_

save__dictionary_history.version

    _item_description.description
;
    A unique version identifier for the dictionary revision.
;
     loop_
    _item.name
    _item.category_id
    _item.mandatory_code
         '_dictionary_history.version'  dictionary_history     yes
         '_dictionary.version'          dictionary             yes
    _item_type.code                     char
    _item_linked.child_name           '_dictionary.version'
    _item_linked.parent_name          '_dictionary_history.version'
     save_

save__dictionary_history.update

    _item_description.description
;
    The date that the last dictionary revision took place.
;
    _item.name                        '_dictionary_history.update'
    _item.category_id                   dictionary_history
    _item.mandatory_code                yes
    _item_type.code                     yyyy-mm-dd
     save_

save__dictionary_history.revision

    _item_description.description
;
    Text description of the dictionary revision.
;
    _item.name
                                      '_dictionary_history.revision'
    _item.category_id                   dictionary_history
    _item.mandatory_code                yes
    _item_type.code                     text
     save_
```

Fig. 2. The DDL specification for the dictionary and the data block containing the dictionary. (a) Boxes surround data items that belong to the same category. Category identifiers are given in large font and item names are given in small font. Parent–child relationships are specified by lines connecting data items with the arrow pointing at the parent item. Key items within a category are identified by a preceding black dot. (b) An example taken from the mmCIF dictionary showing the category dictionary_history. (c) Specification of the category dictionary_history as it appears in the DDL dictionary.

a repeated list of data values that can be decomposed and matched to a corresponding data name. To maintain the correct correspondence between names and values, values cannot be missing from the loop. If a data value is not known it must be represented as either a period (.) to signify that it is missing, or a question mark (?) to signify that it is not relevant in the current context. A loop structure can be nested inside of another data loop structure to construct arbitrarily complex data loop structures. Each level of loop must be terminated by a *stop_* statement, except the outermost loop, which is terminated by the occurrence of a new data item, a save frame, a data block, or an end-of-file. Nested loops are not currently used for representing macromolecular structure data to be compatible with the small molecule use of a subset of the STAR encoding rules. The exception is the

NMR dictionary that does use nested loops. Figure 1 also includes the use of comments—a hash (#) terminated by an end-of-line, and the use of semi-colons (;) as the first character of a line delimiting a body of text as a single data value.

Dictionary Definition Language

Given this basic set of encoding rules it is possible to define a DDL from which a variety of dictionaries can be written to define specific subject domains. Only DDL v2.0, used to represent macromolecular structure data, is described here, since it is the most pertinent to this paper. Readers interested in DDL v1.x, used to describe small molecular crystallographic data, can refer to Hall (1991).

The DDL contains definitions for:

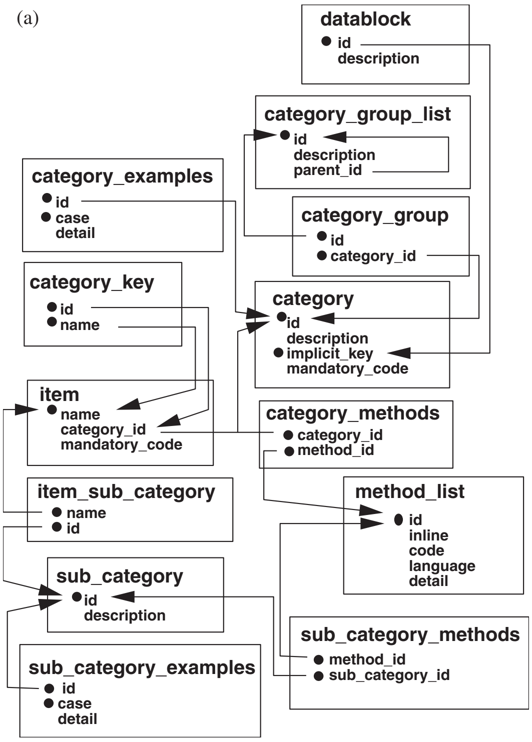
- dictionaries and data blocks
- category groups, categories, and sub-categories
- data items
- methods

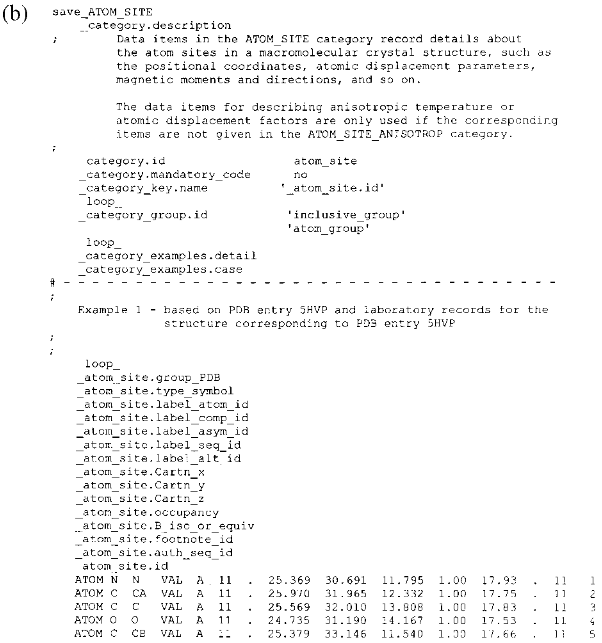
each is discussed.

Dictionaries and Data Blocks

Figure 2 illustrates how a dictionary is described by the DDL. A dictionary is contained within a single data block where each dictionary definition is contained in a save frame within that data block. The dictionary has a name, version history, and method identifiers that define methods to be applied within the context of the data block [Figure 2(a)]. Figure 2(b) illustrates how the specification of dictionary history appears in the mmCIF dictionary.

Since STAR is self-defining the DDL is itself defined in a STAR/DDL compliant dictionary (Westbrook and Hall, 1995). The part of the DDL dictionary that deals with dictionary history is shown in Figure 2(c). The DDL dictionary contains name value pairs where the name is the DDL component being defined and the value is the definition. To simplify the conceptualization of the DDL in the following discussion each feature is not compared with its definition, but with its application in the mmCIF dictionary and hence an instance of its use.



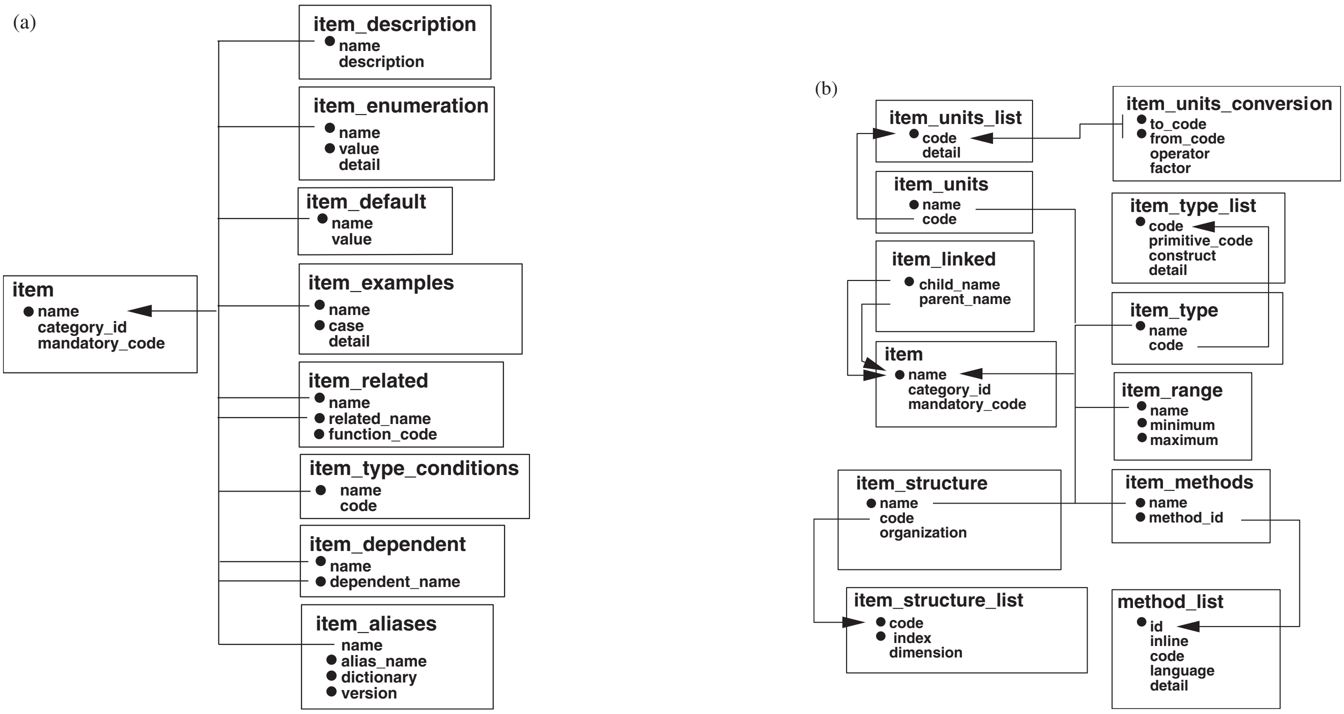

(b)

```
save_ATOM_SITE
    _category.description
;        Data items in the ATOM_SITE category record details about
         the atom sites in a macromolecular crystal structure, such as
         the positional coordinates, atomic displacement parameters,
         magnetic moments and directions, and so on.

         The data items for describing anisotropic temperature or
         atomic displacement factors are only used if the corresponding
         items are not given in the ATOM_SITE_ANISOTROP category.
;
     category.id                  atom_site
    _category.mandatory_code      no
    _category_key.name           '_atom_site.id'
     loop_
    _category_group.id            'inclusive_group'
                                  'atom_group'
     loop_
    _category_examples.detail
    _category_examples.case
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
;
    Example 1 - based on PDB entry 5HVP and laboratory records for the
                structure corresponding to PDB entry 5HVP
;
;
     loop_
    _atom_site.group_PDB
    _atom_site.type_symbol
    _atom_site.label_atom_id
    _atom_site.label_comp_id
    _atom_site.label_asym_id
    _atom_site.label_seq_id
    _atom_site.label_alt_id
    _atom_site.Cartn_x
    _atom_site.Cartn_y
    _atom_site.Cartn_z
    _atom_site.occupancy
    _atom_site.B_iso_or_equiv
    _atom_site.footnote_id
    _atom_site.auth_seq_id
     atom_site.id
    ATOM N  N   VAL  A  11  .  25.369  30.691  11.795  1.00  17.93  .  11  1
    ATOM C  CA  VAL  A  11  .  25.970  31.965  12.332  1.00  17.75  .  11  2
    ATOM C  C   VAL  A  11  .  25.569  32.010  13.808  1.00  17.83  .  11  3
    ATOM O  O   VAL  A  11  .  24.735  31.190  14.167  1.00  17.53  .  11  4
    ATOM C  CB  VAL  A  11  .  25.379  33.146  11.540  1.00  17.66  .  11  5
```

Fig. 3. The DDL specification for category groups, categories, and subcategories. (a) Boxes surround data items that belong to the same category. Category identifiers are given in large font and item names are given in small font. Parent–child relationships are specified by lines connecting data items with the arrow pointing at the parent item. Key items within a category are identified by a preceding black dot. (b) Specification of the atom_site category taken from the mmCIF dictionary.

Category groups, categories, and sub-categories

Figure 3(a) illustrates how data items can be grouped. Central to this is the notion of a category (_category) which groups data items and is easily mapped into a relational table or data object. Each category is identified, described, including examples, has a primary key, and has associated methods that can be applied to all data items in that category. Whether that category need be defined



(c)

```
save__atom_site.Cartn_x
     _item_description.description
;       The x atom site coordinate in Angstroms specified according to
        a set of orthogonal Cartesian axes related to the cell axes as
        specified by the description given in
        _atom_sites.Cartn_transform_axes.
;
     _item.name                      '_atom_site.Cartn_x'
     _item.category_id                 atom_site
     _item.mandatory_code              no
     _item_aliases.alias_name        '_atom_site_Cartn_x'
     _item_aliases.dictionary          cif_core.dic
     _item_aliases.version             2.0.1
      loop_
     _item_dependent.dependent_name
                                     '_atom_site.Cartn_y'
                                     '_atom_site.Cartn_z'
     _item_related.related_name      '_atom_site.Cartn_x_esd'
     _item_related.function_code       associated_esd
     _item_sub_category.id             cartesian_coordinate
     _item_type.code                   float
     _item_type_conditions.code        esd
     _item_units.code                  angstroms
      save_
```

Fig. 4. The DDL specification for data items: (a) Part A; (b) Part B. Boxes surround data items that belong to the same category. Category identifiers are given in large font and item names are given in small font. Parent–child relationships are specified by lines connecting data items with the arrow pointing at the parent item. Key items within a category are identified by a preceding black dot. (c) The specific data item describing a Cartesian *x* coordinate are taken from the mmCIF dictionary.

within the data block to insure the integrity of that data block is also defined. Categories can be grouped and each category subgrouped as needed. Figure 3(b) illustrates the atom_site group (_category.id) from the mmCIF dictionary that includes the data items that characterize the location of an atom. This group is not mandatory for a valid data block (_category.mandatory_code)—a macromolecule can be described without atomic coordinates. The name of the category group is inherent in each data item within that category. The data item atom_site.id, which uniquely identifies the atom, must be present for the category to be valid (_category_key.name). The category is a member of a category group called atom (_category_group.id), which jointly characterize all features of an atom. For example, _atom_site_anisotrop defines another category within the atom site group for specifying the anisotropic thermal motion present in the atomic position and expressed as six thermal displacement vectors.

Data Items

Figure 4(a) and (b) illustrate how individual data items are represented by the DDL. Figure 4(c) illustrates many of the DDL data item features when they are applied to a single *x* coordinate representing the position of that atom in Cartesian space as found in the mmCIF dictionary. The _item.description.description describes the item of data that is named explicitly by _item.name. The item belongs to the category of data items atom_site as described above. It is not mandatory that this data item be present for the category to be valid (_item.mandatory_code) - an atom site can be defined by the known primary protein sequence, but the position of the atom may be unresolvable from the experimental data. The item_aliases are used to alias this data item to equivalent data items in specific versions of other dictionaries. Thus if two dictionary authors insist on providing different data names for the same data values the DDL supports the definition of this correspondence. The _item_dependent describes dependent data items. In the example given in Figure 4(c) the validity of the *x* Cartesian coordinate requires that *y* and *z* be present also. This provides an opportunity for a consistent means of validation since the validation criteria is not defined in each individual program, but defined in a common dictionary external to any program. The _item_related category specifies related data items, in this instance an estimated standard deviation associated with the atomic position. The *x* Cartesian coordinate is a member of a subcategory called cartesian_coordinate (_item_sub_category.id). Data typing is provided (_item_type.code) and conditions are associated with that data type (_item_type_conditions.code). In this instance the data value is a floating point with an associated esd, for example 1.321(3). This value is specified in Angstroms (_item_units.code).

Methods

Methods can be applied to data blocks, category groups, categories, subcategories and individual data items. While the DDL is in place, no dictionary has yet to include methods as part of the dictionary, although novel ways to include methods externally have been devised (Biggs *et al.*, 1997) and will be discussed subsequently.

Dictionaries

Given the STAR encoding rules and the DDL a variety of dictionaries can be written. A dictionary defines all the terms in a given domain. The limited expression of relationships offered by the DDL permits relationships between terms in the dictionary. The DDL also facilitates the validation and formal description of items of data as well as defining specific methods to operate on the data. A number of dictionaries mostly associated with molecular structure have been developed and are summarized in Table 1. Here we focus on the mmCIF dictionary. This dictionary is maintained by COMCIFS, a committee appointed by the International Union of Crystallography that is responsible for overseeing that a standard data representation is maintained.

Table 1. Known Dictionaries based on STAR

Dictionary	Description
Core CIF	The atomic details of small molecules derived from an x-ray crystallography experiment http://www.iucr.ac.uk/iucr-top/cif/cif_core
imgCIF/CBF	Storage of two-dimensional area detector data and other large datasets http://ndbserver.rutgers.edu/mmcif/cbf
Macromolecular CIF	The atomic details of biological macromolecules at the level of detail found in detailed scientific publications http://ndbserver.rutgers.edu/NDB/mmcif/dictionary
Powder CIF	Adds to the core CIF dictionary details of the powder diffraction experiment http://www.iucr.ac.uk/iucr-top/cif/pd
Modulated structures CIF	Adds to the core CIF dictionary details of incommensurately modulated crystal structures http://www.iucr.ac.uk/iucr-top/cif/ms
EPIF	Primary sequence and enzymatics http://www.sdsc.edu/Kinases/development/PIF/SFBrowser.html
MDB	Theoretical models determined by Glaxo Wellcome
NMR	Details of the NMR experiment http://www.bmrb.wisc.edu/elec_dep/Forms/complete_form_v21.txt

The mmCIF dictionary (Bourne *et al.*, 1997) contains over 1700 terms (data items) and took 7 years to complete. The predefined scope for the dictionary was to provide a full description of any biological macromolecule and the x-ray crystallographic experiment used to determine that structure at the level of detail provided in a good scientific publication. Further, the dictionary should include descriptions of all information contained in a PDB file. Since PDB files also describe structures determined by NMR and by theoretical calculations there are data items that pertain to the specifics of these experiments, but they are not fully developed at this time.

The 1700 data items are of the form shown in Figure 3(b) for _atom_site.Cartn_x. Details of the contents of the mmCIF dictionary and their relationship to the contents of a PDB file are fully described by Bourne *et al.* (1997) and only a summary is given here for how structures are represented. Central to the description of a macromolecular structure are entities, which are of three types, polymer, non-polymer and water. Entities are subcomposed into chemical components. For polymers these components are canonical or non-canonical amino acids or nucleotides. For non-polymers they are typically full ligand descriptions. Connectivity within and between polymers, non-polymers and water is fully described. Entities can be used to build the contents of the asymmetric unit as found in the crystal structure and also the biologically active molecule. Data items exist to provide a full description of the secondary structure, but at present tertiary structures, quaternary structures, and assemblies are not fully defined.

Data files

Data files contain one or more data blocks where each data block contains data items that are defined in the appropriate dictionary. While data files based on STAR are common for describing small molecules they are yet to be found routinely for describing macromolecular structures. Those that are available have been generated from PDB files using programs like PDB2CIF (Bernstein *et al.*, 1998). These derived files provide insights into working with mmCIF, but in terms of content contain no information beyond that which can be parsed from a PDB file. This restriction will change in the near future based on developments within the PDB. These developments are discussed below.

The STAR concept is extensible and can include information described in external reference files (ERFs). These include data items that are not part of the generic dictionary but are added for specific projects. Examples might be enumeration of standard values of basic amino acid geometry to be used for checking purposes or a sequence feature table.

Software

A variety of software has been developed for dealing with mmCIF, with the emphasis on proof-of-concept software or foundation libraries (Table 2). In hindsight the lack of end-user application programs has hampered the adoption of mmCIF. This experience leads us to say that defining an ontology is the beginning, not the end of the story. Without good software to use the ontology its adoption as a community standard is in doubt, regardless of how comprehensive and insightful.

Table 2. Available software for use with STAR/mmCIF

Name	Description
cif2pdb	Program to convert mmCIF to pseudo-PDB format (H.J.Bernstein & F.C.Bernstein, 1998, unpublished; Perl)
CIFLIB	Application Program Interface (Westbrook *et al.*, 1997, C)
CIFOBJ	A class library of mmCIF dictionary access tools (S. Schirripa and J. D. Westbrook, 1996, unpublished; C++)
CIFPARSE	A library of access tools for mmCIF (S-H. Hsieh and J. D. Westbrook, 1996, unpublished; C)
CIFTABLE	A class library of table access tools (S. Schirripa and J. D. Westbrook, 1997, unpublished; C++)
CIFtbx2	Routines for basic file manipulation (Hall and Bernstein, 1996, Fortran)
OOSTAR	Data structures and associated applications to manipulate STAR files (Chang and Bourne, 1998, Objective-C)
pdb2cif	Filter a PDB entry and produce mmCIF (Bernstein *et al.*, 1998, Perl)
PDBTool	Graphical review of a PDB or mmCIF structure entry (Biggs *et al.*, 1996, C++, X/Motif)

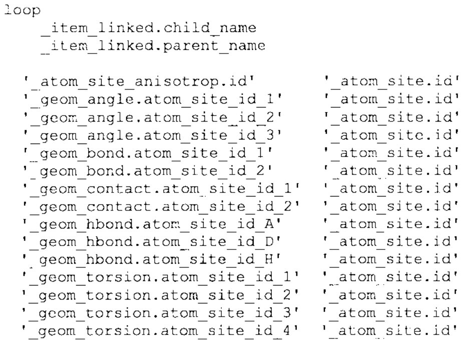
Access to this software is available at http://ndbserver.rutgers.edu/mmcif/software

Tutorials

A variety of tutorials on using and writing mmCIF dictionaries and software tools have been written by John Westbrook, Herbert Bernstein and Phil Bourne and are available from http://ndbserver.rutgers.edu/mmcif/workshop/mmCIF-tutorials/.

Discussion

STAR and the corresponding CIF dictionary were developed to facilitate data exchange and archiving for small molecule crystallography data. The IUCr facilitated the extension of this concept by fostering developments in other areas of crystallography, including the determination of biological macromolecules. Early in the process of developing a dictionary for biological macromolecules (mmCIF) those involved realized the potential of this effort to become an ontology for structural biology, even if it was not recognized by that name at the time. This realization came about, in part, by introducing those trained in computer science and informatics to a concept that had its roots with those experimentalists working in the subject domain. In retrospect the process would have been facilitated if these two distinct groups had been brought together in the beginning. It is now apparent that the makings of useful ontology requires both groups.



```
loop
    _item_linked.child_name
    _item_linked.parent_name

  '_atom_site_anisotrop.id'       '_atom_site.id'
  '_geom_angle.atom_site_id_1'    '_atom_site.id'
  '_geom_angle.atom_site_id_2'    '_atom_site.id'
  '_geom_angle.atom_site_id_3'    '_atom_sitc.id'
  '_geom_bond.atom_site_id_1'     '_atom_site.id'
  '_geom_bond.atom_site_id_2'     '_atom_site.id'
  '_geom_contact.atom_site_id_1'  '_atom_site.id'
  '_geom_contact.atom_site_id_2'  '_atom_site.id'
  '_geom_hbond.atom_site_id_A'    '_atom_site.id'
  '_geom_hbond.atom_site_id_D'    '_atom_site.id'
  '_geom_hbond.atom_site_id_H'    '_atom_site.id'
  '_geom_torsion.atom_site_id_1'  '_atom_site.id'
  '_geom_torsion.atom_site_id_2'  '_atom_site.id'
  '_geom_torsion.atom_site_id_3'  '_atom_site.id'
  '_geom_torsion.atom_site_id_4'  '_atom_site.id'
```

Fig. 5. Example of parent-child relationships from DDL v2.*x* showing the mapping of _atom_site_id.

It could be argued that software such as Ontolingua is independent of the subject domain. While true, the counter argument is that it got that way by applying it to several specific domains simultaneously.

The result for structural biology is a hybrid that tries to merge concepts developed by domain scientists to those developed by computer scientists. For example, while STAR permits nested loops, arbitarily long records, and data names, the subset of the encoding rules in which the small molecule dictionary is developed does not. The macromolecular case follows the small molecule case making for a cumbersome representation. This combined with the STAR encoding rule that a data name may appear only once in a data block makes the issue of data representation even more complex. Version 2.*x* of DDL deals with this through the use of parent–child relationships (Figure 5). In this example the unique identifier for an atomic site is used in the definition of basic connectivity, geometry, hydrogen bonding, and non-bonded contacts. The end result is a complex data representation. Other problems that have been noted in using STAR/mmCIF but are not described in detail here are: inability to recognize data files as mmCIF files or to know what dictionary they can be validated against (metadata); category groups, categories and sub-categories are conceptually different leading to complex data structures; interpreting certain data items requires a good knowledge of the overall dictionary content; and some data are defined too coarsely requiring second-level parsing.

While the data dictionary is comprehensive, this of itself makes its difficult for a novice to use. Tools exist for browsing the dictionary or abstracting subsets of the dictionary for particular uses. It turns out abstraction is also necessary for efficient programming when using the dictionary. An ideal program (Chang and Bourne, 1998) would simply code the STAR encoding rules, read a DDL dictionary and from that build a data structure to contain specific dictionaries conforming to that DDL, and subsequently data files conforming to that dictionary. In practical terms this does not happen. First, the size and the complexity of the dictionary make this impracticable, and second, the majority of programmers are scientists who have not been trained to work in this way. Rather, they write programs that read specific items of data from data files. Nevertheless, if tools are already available the benefits of using the ontology become apparent.

The mmCIF dictionary provides the conceptual schema for the new Protein Data Bank (PDB) as developed by the RCSB (http://www.rcsb.org). The conceptual schema is used to define physical schema for both relational and object oriented database implementations. Reading different subsets of the mmCIF dictionary by an interface builder application enables different subsets of data to be entered conforming to different views of the data. This is used by the PDB to create both depositor and annotator views of the data used in data entry. The dictionary defines what data items are mandatory, enumeration values and ranges where appropriate, definitions and so on, which are used by the depositor in making the deposition to the PDB and by the annotator in further expanding the entry. The obvious advantage of this approach is that changes to a rapidly changing scientific field are independent of the underlying software to support those changes. The changes are added to the dictionary and new views for data input generated immediately. Moreover, the data representation is governed by a body appointed by the scientific society that represents the field and not by an individual group of developers. This has the potential of leading to a more consistent and globally acceptable representation of the data. A challenge for the PDB is recasting data in at least three or more distinct PDB data formats into an mmCIF form. This is not a problem for new depositions that capture this information from the user and can produce consistent PDB files while internally maintaining mmCIF compliant data, but it is a problem for the approximately 10 000 structures (June 1999) already deposited.

Once a more stringent data representation is imposed new possibilities emerge. For example a prototype for the use of code generation (Biggs *et al.*, 1997) in conjunction with the mmCIF dictionary is already it place. We defined a domain specific language (DSL) and added this to categories within the mmCIF dictionary. The DSL calls specific mapping modules that map the PDB to the mmCIF data representation. A code generator reads the mapping modules, generates an executable, and maps structures represented in a PDB format to an mmCIF format and vice-versa. Thus as the mmCIF description evolves a new convertor is generated without recoding anything, but by simply adding a dictionary pointer to an associated mapping module which is easily maintained by non-programmers.

There are competing and compelling advances in technology that have taken place since STAR/DDL/mmCIF was introduced. Nevertheless, the groundwork that has been done will be useful even if the ontology is expressed using different technology. We are working with the Object Management Group (OMG) to define a standard for macromolecular structure data. The work done in developing mmCIF will be of great value here. Further, we and others are in the process of using eXtensible Markup Language (XML) and the associated Document Type Definition (DTD) for representing protein structure and function. It is a straightforward process to map the components of a macromolecular structure as defined in the mmCIF dictionary into a DTD. Thus while the form of the representation may change in the future the content will only grow. It is our hope that the many years of hard work that went into that content will soon begin to prove valuable to the biology community at large.

In conclusion, STAR/mmCIF provides a means of characterizing that content which, taken together, we consider an ontology. The characterization includes, data typing, relationships between items of data, which data are mandatory, and so on. Data cast into STAR/mmCIF can be reliably exchanged and interpreted by human or machine. While DDL v2.x addresses some significant shortcomings in DDL v1.x, problems remain. Whether these problems and/or competing technologies preclude the widespread use of STAR/mmCIF remains to be seen. Either way, the early realization of the importance of good data representation and the willingness of an international scientific society to support its development should be commended.

Acknowledgements

The development of the STAR/mmCIF standard has truly been a community effort. Most prominent and providing the greatest contributions to the work described in this paper are Paula Fitzgerald who was responsible for mmCIF dictionary development with assistance from Helen Berman who is now lead editor for new dictionary developments. Sid Hall carried out much of the pioneering work on STAR and Herbert Bernstein has been a key software developer in support of this effort. Finally, Brian McMahon has provided expertise on developing the standard on behalf of the IUCr. This work has benefited from the previous support of the National Science Foundation and the US Department of Energy.

References

Aho,A.V., Sethi,R. and Ullman,J.D. (1985) *Compilers Principles, Techniques, and Tools*. Addison-Wesley, MA.

Bernstein,H.J., Bernstein,F.C. and Bourne,P.E. (1998) CIF applications. VIII. pdb2cif: translating PDB entries into mmCIF format. *J. Appl. Cryst.*, **31**, 282–295.

Biggs,J., Pu,C., Groeininger,A. and Bourne,P.E. (1996) PDBtool: An interactive browser and geometry checker for protein structures. *J. Appl. Cryst.*, **29**, 484–490. http://www.cse.ogi.edu/DISC/PDBTool/.

Biggs,J., Pu,C. and Bourne,P.E. (1997) Code generation through annotation of macromolecular structure data. In Gaasterland,T.T. *et al.* (eds), *Fifth International Conference on Intelligent Systems for Molecular Biololgy.* AAAI Press, MenlowPark, CA, pp. 52–55.

Bourne,P.E., Berman,H.M., McMahon,B., Watenpaugh,K.D., Westbrook,J.D. and Fitzgerald,P.M.D. (1997) The macromolecular CIF dictionary (mmCIF). *Meth. Enzymol.*, **277**, 571–590.

Chang,W. and Bourne,P.E. (1998) CIF applications. IX: A new approach for representing and manipulating STAR files. *J. Appl. Cryst.*, **31**, 505–509.

Fitzgerald,P.M.D., Berman,H.M., Bourne,P.E. and Watenpaugh,K. (1992) Macromolecular CIF Working Group, International Union of Crystallography.

Guarino,N. (1996) Understanding, building, and using ontologies. *Proceedings of the Tenth Knowledge Acquisition for Knowledge-based Systems Workshop.* http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96Proc.html

Hall,S.R. (1991) The STAR file: A new format for electronic data transfer and archiving. *J. Chem. Inf. Comput. Sci.*, **31**, 326–333.

Hall,S.R., Allen,F.H. and Brown,I.D. (1991) The crystallographic information file (CIF): a new standard archive file for crystallography. *Acta Cryst.*, **A47**, 655–685.

Hall,S.R. and Spadaccini,N. (1994) The STAR file: Detailed specifications. *J. Chem. Inf. Comput. Sci.*, **34**, 505–508.

Hall,S.R. and Cook,A.P.F. (1995) Data definition language for STAR file dictionaries. *J. Chem. Inf. Comput. Sci.*, **35**, 819–825.

Hall,S.R. and Bernstein,H.J. (1996) CIF applications. V. CIFtbx2: Extended tool box for manipulating CIFs. *J. Appl. Cryst.*, **29**, 598–603.

Teeter,M.M., Roe,S.M. and Heo,N.H. (1993) Atomic resolution (0.83 A) crystal structure of the hydrophobic protein crambin at 130 K. *J. Mol. Biol.*, **230**, 292–311.

Westbrook,J.D. and Hall,S.R. (1995) A dictionary description language for macromolecular structure, Rutgers University, New Brunswick, NJ, Report NDB-110.

Westbrook,J.D., Hsieh,S.-H. and Fitzgerald,P.M.D. (1997) CIF applications. VI. CIFLIB: an application program interface to CIF dictionaries and data files. *J. Appl. Cryst.*, **30**, 79–83.