A discussion paper by David Brown (COMCIFS Chair) concerning decisions that COMCIFS needs to make on questions raised during the formal review of the *methods* Dictionary Definition Language (DDLm).

****

In the course of reviewing the proposed *methods Dictionary Definition Language*, DDLm, and preparing CIF dictionaries in this language, three issues have arisen that require a decision by COMCIFS. These relate to the use of features in DDLm that are not available in the existing DDLs. An open discussion of these issues is needed before the voting members of COMCIFS are asked to make their final decisions.

In this discussion paper I first provide background by describing the relevant additions to DDLm and how they will change the way CIF is used. I then describe the three issues that must be resolved and list some possible solutions as the basis for discussion.

Background

Two new features of DDLm are relevant to this discussion.

1. CIF dictionaries may now include *'methods*', i.e., machine readable algebraic relationships that can instruct a computer how to generate the value of a missing item from other items present in the CIF.

2. The CIF dictionary that is used to read a CIF may be a virtual dictionary created at run time by merging a number of smaller CIF dictionaries according to instructions included in the CIF itself. This works in the following way: The CIF is first read in as a STAR file in which no meaning as attached to the datanames or their values except for those items in the audit_conform category. The program then uses the information in the _audit_conform items to direct the assembly of the virtual dictionary. Finally the program uses the virtual dictionary to interpret the other items in the CIF. The reading program must be hardcoded for reading STAR files and interpreting items in the audit_conform category.

There are four traditional uses for CIF

1. Archiving the results of crystal structure determinations, e.g., Protein Databank, Acta Cryst. archive.

2. Transferring crystallographic information between institutions, e.g., submission of structure reports to Acta Cryst., uploading and downloading structure reports to and from databases, transferring data from a synchrotron to a researcher's home institution.

3. Transferring crystallographic data between programs within the same institution.

4. Input into web programs, e.g., checkcif.

DDLm offers the following additional possibilities that are not available under DDL1 or DDL2.

5. Calculating the values of items not present in the CIF.

6. Including local dictionaries in the virtual dictionary used to read a CIF.

7. Combining 5 and 6 to allow a user to calculate items not defined in the standard dictionaries, e.g., the dipole moment or refractive index of a crystal according to the user's definition.

There are two important CIF principles that are relevant.

1. CIF (normally) defines only one data item for a given piece of information and the format of that item is the one closest to the natural representation of the quantity (rather than a format in common use or one designed for ease in computation). For example, the CIF specifies which units that are to be used and alternative units are not permitted.

2. A given piece of information may appear only once in a given datablock. This is necessary to ensure uniqueness of the reported value. If the same information is given twice there could be an ambiguity if the two values do not agree. This restriction follows necessarily from 1, but if 1 is relaxed, this principle could still be preserved.

Issues and their possible resolutions

Three issues are raised. In each case I suggest some possible solutions and their implications.

1. Virtual dictionaries.

DDLm is designed to allow dictionaries to be merged. This is achieved through *import* items that allow all or parts of dictionaries stored at the specified URIs, to be imported into a higher level dictionary, with a calling 'head dictionary' as the top level. Because merging will be simple, it is likely that the current dictionaries, such as the Core CIF dictionary, will be broken into several smaller, more specialized, dictionaries during their conversion to DDLm. One consequence of this approach is that each CIF will specify the contents of its own run-time virtual dictionary. This may range between a small head dictionary containing only *import* statements or a complete existing dictionary in which no further merging is needed. The CIF items needed to identify the head dictionary are already present in the audit_conform category. Currently these items are rarely used, but in the future CIFs will be expected to contain the _audit_conform items needed to identify the (head) dictionary to be used for reading the CIF as well as the URI where it can be found. Any *import* items in the head dictionary will in turn have pointers to where the required subdictionaries can be found. For most applications this dictionary will be one of a small number stored on the IUCr web site where the COMCIFS-approved subdictionaries will also reside.

However, if CIF is to exploit its potential for flexibility, there are other possibilities that some people may occasionally wish to use. For some specialist application the originator of a CIF may wish to create a custom head dictionary, particularly if a local dictionary is to be incorporated in the virtual dictionary since the standard head dictionaries on the IUCr web site clearly will not contain calls to local dictionaries. This is not a problem if the CIF is only to be used locally since the head dictionary can be stored on a local server. However, if the CIF is to be passed to another institution the question arises as to where the relevant head dictionary (and any required local dictionaries) are to be stored. There are a number of possibilities:

A, ***These dictionaries are stored on a local web server.***
Problems: The dictionaries would have to remain in the server with a stable URI as long as the CIF was in existence, but if the CIF were circulated to other institutions the originator would no longer have control over when or whether the last copy of the CIF was deleted. Therefore the head and local dictionaries would need a permanent and stable URI or the CIF would need an expiration date. Are such conditions easily met?

B. ***The dictionaries could be stored on the IUCr web server.***
Advantages: This would be stable,
Problems: Would the IUCr be willing to provide a home for such personal files with no indication of how long they would need to remain active?

C. ***The head dictionary could be included in the same file as the CIF that calls it.***
Advantages: The dictionary would travel with the CIF and no permanent URI would be needed.
Problems: There is a risk that the CIF and dictionary could become separated over time, making reading the CIF problematic.

D. ***The head dictionary could be included as a text field within the CIF itself as an item called _audit_conform_included_dictionary.***
Advantages: The head dictionary is an integral part of the CIF and would remain with it. The head dictionary could incorporate any required items from a local dictionary or the _audit_conform_included_dicrtionary could be looped.
Problems: There may be some technical issues in this solution but these should not be insurmountable. Mixing different file types within a datablock may be considered inappropriate and possibly dangerous. However, there is a precedent in imgCIF which includes a representation of a binary file within the ASCII text field of a CIF

E. ***CIF rules would disallow the export of a CIF that calls a head dictionary without a stable URI.***
Advantages: It solves the problem
Problems: It would be difficult to enforce. It would effectively limit uses 6 and 7 above to in-house use or to institutions that can provide stable URIs

The audit_conform items will in any case be privileged as their use must be hardcoded into the software Thus an AUDIT_CONFORM.DIC would be a small CIF subdictionary that would be stable for the foreseeable future.

2. Hierarchy of methods

Methods are used to calculate the value of an item from other items in the CIF. Items for which *methods* are defined are necessarily derivative since by definition experimental measurements cannot be calculated in this way. There is therefore a natural hierarchy with measured quantities forming the foundation.

There are, however, some quantities that can be calculated in more than one way. Fobs^2 for example can be calculated either from Fobs or from I and the calculations are different (though they should give the same result if the CIF is self consistent). There are two approaches that can be taken and these represent two different views of how CIF should be developed.

A. ***The methods are seen as hierarchical. For example, Fobs^2 is always calculated from Fobs, but if Fobs is not present, it is first calculated from the observation, I. This approach implies an hierarchy of items in which a given item is always calculated from more fundamental items.***

Advantages: It creates a unique definition for each derivative item. As this is only a problem for dictionary writers it can be monitored by COMCIFS.

Problems: However, local dictionaries that include methods would not be monitored by COMCIFS. The hierarchy would have to be explicit so that the writers of dictionaries would know that Fobs^2 should be calculated from Fobs rather than I. Most calculations involve several items and it is easy to imagine that there may be more convoluted cases where it might not be possible to maintain the hierarchy, or in which one inadvertently ends up with circular definitions (see for example the problem described in item 3 below).

B. ***Several looped methods are allowed.***

Advantages: A derived item could be calculated in different ways. The loop key would be a number that defines the order in which the *methods* would be attempted. Method 1 would be the default (and primary definition), method 2 would be used if method 1 could not be used because the required items were not present and could not be calculated. If required the loop key could be specified by the user at run time to indicate which *method* should be used.

Problems: The presence of multiple definitions could result in confusion about the primary definition even though the default would be regarded as primary.

The DDLm standard allows for different algebraic languages to be used, though the language currently used is dREL which has been developed for this purpose. Presumably only dREL will be used in COMCIFS approved dictionaries, but other languages may be used in private dictionaries. The *method* in a private dictionary can always be programmed in the head dictionary to take precedence over the *method* in the COMCIFS approved dictionary.

3. Intermediate items used in computation

Since *methods* break the calculations into small steps, any given computation may involve the generation of values for several intermediate items. Some of these will be items already defined in the current dictionaries, for example the generation of Fobs when calculating Fobs^2 from I, but others may be items that are not currently defined (often for good reasons). This is well illustrated by an example taken from the current draft DDLm CIF dictionary: the use of the beta matrix format to simplify calculations involving the atomic displacement parameters.

The two CIF principles described above are at stake here. For example the atomic displacement parameters are given in the physically natural U form (though a concession was reluctantly made of allowing protein structure reports to use the B form because of its ubiquitous use in that important field). The generation of a beta matrix from the U matrix during a routine calculation violates both principles: the availability of two different formats for the same information (atomic displacement parameters), and the presence of the same information twice in the same CIF (as U and as beta). Further, we cannot assume that everyone will use the U format in generating a CIF. If the beta format is defined in the dictionary eventually someone will create a CIF with atomic displacement parameters given only in the beta format.

This example raises a number of red flags. In the crystallographic literature, beta is defined in two different ways, either with or without an explicit 2 in the cross terms. Rarely do the original papers that report betas state which convention was used. The definition of beta in the draft dictionary is, of course, unambiguous, but how many people would make sure that the beta they report conforms to that definition? Further, there is no easy way to check whether the correct convention was used. This is no hypothetical danger; we have already received a request for a CIF definition of the beta format so that a database could use it to output entries containing beta parameters copied from the original paper. This request was turned down on two grounds, firstly as being contrary to the CIF principle requiring all items to be reported in the same format, and secondly on the grounds that most published betas are ambiguous and CIF should avoid any ambiguity. Possible solutions to this problem are

A. ***Prohibit the use of intermediate values that would not be welcome in an archival CIF.***

Advantages: This ensures that we adhere to current policies that are important principles in defining our dictionaries.

Problems: It limits the ways in which calculations could be made and may prevent certain shortcuts from being used.

B. ***Flag items that should not be included in an archive.***

Advantage: Even though the intermediates are included internally in the CIF, they can be removed when a CIF is saved. A program can easily check if temporary intermediates are present, i.e., atomic displacement parameters reported as betas would be flagged as non-conforming and removed or a warning issued.

Problems: It creates two classes of items in the dictionary, archival and computational (but this may be the price of the exploiting the multifunctionality of CIF)

C. ***Allow the intermediate values to remain in archived CIFs and abandon the two CIF principles described above, namely that an item of information be given only in one format and that the same information appear only once in the same CIF.***
Advantages: Avoids having two different types of item.
Problems: Two important CIF principles would be lost with serious implications for the future of CIF. Either it places an additional burden on software producers who must be prepared to read items in a variety of formats, or the CIF dictionary would need to contain a *method* for generating U from beta. Since beta is currently defined in terms of U, this would create a circular definition and play havoc with any hierarchical scheme in the computation of missing items (see the discussion in point 2 above). If two forms of the same information do not agree, we need a flag to indicate which form is the default. This solution raises many specters and would put the CIF project in peril.

David Brown