2.6.6.2.7. *SUB_CATEGORY*

The category SUB_CATEGORY provides data items to describe a subcategory and to associate a procedure with the subcategory (see Section 2.6.6.2.9). A subcategory is a set of data items within a category that have a particular association. A typical example would be a triad of positional coordinates *x*, *y*, *z* that are collectively assigned to a 'cartesian' subcategory.

2.6.6.2.8. *SUB_CATEGORY_EXAMPLES*

The DDL category SUB_CATEGORY_EXAMPLES holds examples of a subcategory. A subcategory example might illustrate valid instances of the items comprising the subcategory. An example specification contains the text of the example, `_sub_category_examples.case`, and an optional comment item, `_sub_category_examples.detail`, that can be used to qualify the example. The key for this category includes the items `_sub_category_examples.id` and `_sub_category_examples.case`. This compound basis permits multiple unique examples to be provided for each subcategory.

2.6.6.2.9. *SUB_CATEGORY_METHODS*

The SUB_CATEGORY_METHODS category is used to associate method identifiers with subcategories. Any number of unique method identifiers may be associated with a subcategory. The method identifiers reference the full method definitions in the parent METHOD_LIST category.

The procedure that is identified as `_sub_category_methods.method_id` may be used to validate the subcategory identified as `_sub_category_methods.sub_category_id`. Subcategory validation may be required in instances where conditions are placed on the values of data items within the subcategory that are more restrictive than those associated with each component data item. A simple example of such a restriction would be a normalization restriction on the components of a subcategory. Any procedure referenced in this category must also be defined in the category METHOD_LIST.

2.6.6.3. DDL2 definitions describing methods

In this section, the DDL categories that define the methods associated with data blocks, categories, subcategories and items are presented. Figs. 2.6.4.1, 2.6.4.2 and 2.6.4.3 illustrate the relationships between the method categories and other DDL categories.

2.6.6.3.1. *METHOD_LIST*

The METHOD_LIST category defines methods that can be associated with data blocks, categories, subcategories and items. This category attempts to capture only the essential information required to define these methods, without defining any implementation details. The implementation details are appropriately left to application-dictionary developers. It is assumed here that, within a domain of dictionaries, a consistent method interface will be adopted that is tailored to the requirements of that domain. This of course complicates the sharing of methods between domains; however, it would be impossible at this time to define an implementation strategy inside the DDL that would even begin to satisfy the diverse requirements of potential DDL users. Consequently, the definition of each method is limited to: its unique identifier, `_method_list.id`; a textual description, `_method_list.detail`; the source text of the method, `_method_list.inline`; the name of the language in which the method is expressed, `_method_list.language`; and a code to identify the purpose of the method, `_method_list.code`.

2.6.4. DDL2 definitions describing dictionaries and data blocks

In this section, the DDL categories that describe the characteristics of dictionaries and data blocks are presented. In this context, a dictionary is defined as a group of related definitions within a STAR data block. Fig. 2.6.4.3 illustrates the organization for these categories.

2.6.6.4.1. *DATABLOCK*

The DATABLOCK category holds the essential identifying information for a data block: the name of the data block, `_datablock.id`; and a description of the block, `_datablock.description`. The `_datablock.id` is the parent identifier for both `_category.implicit_key` and `_dictionary.datablock_id`. The former guarantees that the identifier for the data block, and hence the dictionary, is added implicitly to the key of each category.

2.6.6.4.2. *DATABLOCK_METHODS*

The DATABLOCK_METHODS category may be associated with a data block. The method identifiers reference the full method definitions in the parent METHOD_LIST category.

2.6.6.4.3. *DICTIONARY*

The DICTIONARY category holds the essential identifying information for a data dictionary. The items recorded in this category include the title for the dictionary, `_dictionary.title`, the current version identifier, `_dictionary.version`, and the data-block identifier in which the dictionary is defined, `_dictionary.datablock_id`. The version identifier references the parent identifier in the DICTIONARY_HISTORY category in which each dictionary revision is described.

2.6.6.4.4. *DICTIONARY_HISTORY*

The DICTIONARY_HISTORY category holds the revision history for a dictionary. Each revision is assigned a version identifier that acts as the key item for the category. Along with the version information, a text description of the revision and date of revision must be specified.

References

Allen, F. H., Barnard, J. M., Cook, A. F. P. & Hall, S. R. (1995). *The Molecular Information File (MIF): core specifications of a new standard format for chemical data. J. Chem. Inf. Comput. Sci.* **35**, 412–427.

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. & Brown, I. D. (1991). *The Crystallographic Information File (CIF): a new standard archive file for crystallography. Acta Cryst.* A**47**, 655–685.

Hall, S. R. & Cook, A. P. F. (1995). *STAR dictionary definition language: initial specification. J. Chem. Inf. Comput. Sci.* **35**, 819–825.

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

Westbrook, J. D. & Hall, S. R. (1995). *A dictionary description language for macromolecular structure.* http://ndbserver.rutgers.edu/mmcif/ddl/.

references