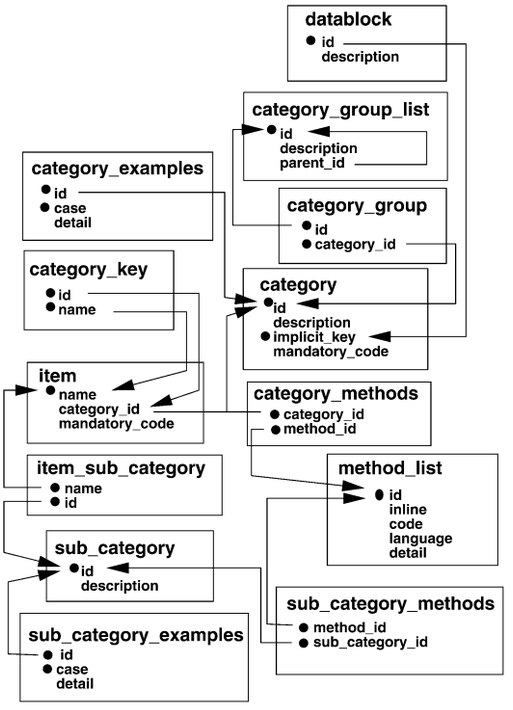

Fig. 2.6.4.2. DDL2 attributes used to specify category, subcategory and category group information. Category identifiers are given in a large typesize and item names are given in a smaller typesize. Parent–child relationships are specified by lines connecting data items with the arrow pointing at the parent item. Key items within a category are marked with a bullet.

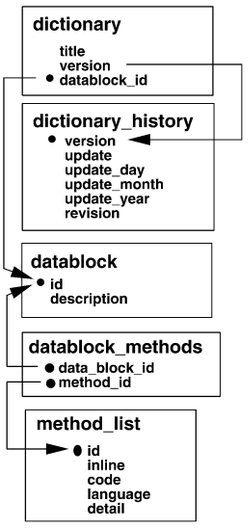

Fig. 2.6.4.3. DDL2 attributes used to specify dictionary and data-block information. Category identifiers are given in a large typesize and item names are given in a smaller typesize. Parent–child relationships are specified by lines connecting data items with the arrow pointing at the parent item. Key items within a category are marked with a bullet.

Fig. 2.6.4.2 illustrates the organization of attributes describing categories, subcategories and category groups. Similarly, Fig. 2.6.4.3 shows the organization of DDL2 attributes at the data-block and dictionary level. All of these attributes are discussed in terms of their application in building data dictionaries in the next section.

2.6.4. DDL2 organization

Figs 2.6.4.1–2.6.4.3 provide schematic illustrations of the definitional features provided by DDL2. These figures represent the elements of the DDL in terms of its own language constructs (*i.e.* categories and the relationships between attributes within those categories). This self-defining presentation has the important consequence of validating the internal consistency of the DDL data model.

Fig. 2.6.4.1 shows the organization of the attributes available to define each data item. These include: a description, examples, data type, allowed values and ranges, default values, internal structural features (*e.g.* vector and matrix properties), units, and other dependency relationships. These DDL attributes are shown as a collection of DDL categories enclosed in boxes in the figure. For instance, the description or textual definition for a data item is specified in a category named ITEM_DESCRIPTION. This DDL category contains the attributes 'name' and 'description'. The attribute 'name' corresponds to the DDL data item `_item_description.name`. This item is the key item in the category named ITEM_DESCRIPTION. In Fig. 2.6.4.1 this is denoted by a bullet. The name attribute in the ITEM_DESCRIPTION category is related to the parent definition of this data item in the category named ITEM. This is reflected in Fig. 2.6.4.1 by the line pointing to the parent data item.

The data-block level ties the contents of a dictionary to the `data_` section in which it is contained. The identifier for the data block and hence the dictionary is added implicitly to the key of each category. This builds into the data model a convenient means for distinguishing similar information recorded in separate data blocks. This feature is important in organizing the results from different crystallographic experiments, each being reported as a separate block of data.

2.6.5. DDL2 dictionary applications

In this section, several examples are presented which illustrate how the elements of the DDL are used to build dictionary definitions. Example 2.6.5.1 shows the definition of the `_citation.journal_abbrev` data item from the mmCIF dictionary.

The category ITEM_DESCRIPTION holds a text description of each data item. The category ITEM holds the item name, category name and a code indicating whether this item is mandatory in any row of this category. The value of the mandatory code is either `yes`, `no` or `implicit`. The `implicit` value is used to indicate that a value is required for the item but it can be derived from the context of the definition and need not be specified. This feature is most often used in DDL2 dictionaries to avoid re-specifying data-item names in each category since these values can be derived from the name of the save frame enclosing the definition. The value of the `_item.name` in the above example is enclosed in quotation marks. This is a requirement of the STAR syntax so that a value containing a data name is not mistaken for a dictionary attribute.

The mmCIF dictionary contains a superset of the definitions that were originally defined in the core CIF dictionary. In order to maintain backward compatibility with original definitions, the ITEM_ALIASES category was introduced to hold the item name, dictionary name and version in which the original definition of an item was published. In this example, the data name used in the core dictionary differs from the example definition only in the period that distinguishes the category and attribute portions of the item name.

The category ITEM_TYPE holds a reference to a data type specified in the ITEM_TYPE_LIST category. A reference to the data type is used here rather than a detailed data-type description in order to avoid repeating the description for other data items. A single list

Example 2.6.5.1. *A rather simple data definition.*

```
save_citation.journal_abbrev
   _item_description.description
; Abbreviated name of the cited journal given in
   the Chemical Abstracts Service Source Index.
;
   _item.name                '_citation.journal_abbrev'
   _item.category_id           citation
   _item.mandatory_code        no
   _item_aliases.alias_name
                             '_citation_journal_abbrev'
   _item_aliases.dictionary  cif_core.dic
   _item_aliases.version     2.0.1
   _item_type.code           line
   _item_examples.case       'J. Mol. Biol.'
save_
```

of data types and associated regular expressions is stored in the ITEM_TYPE_LIST category and this may be referenced by all of the definitions in the dictionary. In the mmCIF dictionary, the codes that are used to describe the data types are generally easy to interpret. In this example, the type code 'line' indicates that a single line of text will be accepted for this data item.

Descriptive examples of data items can be included in the ITEM_EXAMPLES category. In Example 2.6.5.1, one value, `'J. Mol. Biol.'`, is specified, but multiple examples can be provided using a **loop_** directive.

Other DDL item attributes are illustrated in the mmCIF definitions for the items **_cell.length_a** and **_cell.length_a_esd** in Example 2.6.5.2. Some data items are only meaningful as part of a complete set. The ITEM_DEPENDENT category is used to store this type of information. Those additional data items within the irreducible set are listed in this category. In Example 2.6.5.2, the cell lengths in the *b* and *c* directions are defined as dependent items of the cell length in the *a* direction.

The permissible ranges of values for a numerical data item are stored in the ITEM_RANGE category. Each boundary condition is defined as the non-inclusive range between a pair of minimum and maximum values. If multiple boundary conditions are specified using the **loop_** directive, then each condition must be satisfied. A discrete boundary value may be set by assigning the desired boundary value as both the maximum and minimum value. In the above example, the permissible cell-length range is defined as greater than or equal to zero, where the latter boundary condition is specified by setting both extrema as zero.

A number of special relationships may be defined between data items. For some relationships which occur frequently, the source or function of the relationship has been standardized. In the example above, this feature is used to identify that the **_cell.length_a_esd** is the standard uncertainty (estimated standard deviation) of **_cell.length_a**. The recognized relationships are fully described in the DDL definition of the data item **_item_related.function_code** in category ITEM_RELATED. The current list includes the kinds of relationships in Table 2.6.5.1.

Sets of data items within a category may be collected into named subcategories. ITEM_SUB_CATEGORY is used to store the subcategory membership of a data item. In the above example, item **_cell.length_a** is added to the subcategory CELL_LENGTH. The items **_cell.length_b** and **_cell.length_c** are similarly added to this subcategory in their definitions.

The ITEM_UNITS category holds the name of the system of units in which an item is expressed. The name assigned to **_item_units.code** refers to a single list of all of the unit types used in the dictionary. This list is stored in the category ITEM_UNITS_LIST. Conversion factors between different systems of units are provided in the data table stored in the ITEM_UNITS_CONVERSION category.

Example 2.6.5.2. *Definition of a data item that has dependencies and associated items.*

```
save_cell.length_a
   _item_description.description
; Unit-cell length a corresponding to the structure
   reported in angstroms.
;
   _item.name                    '_cell.length_a'
   _item.category_id               cell
   _item.mandatory_code            no
   _item_aliases.alias_name      '_cell_length_a'
   _item_aliases.dictionary        cif_core.dic
   _item_aliases.version           2.0.1
    loop_
   _item_dependent.dependent_name
                                 '_cell.length_b'
                                 '_cell.length_c'
    loop_
   _item_range.maximum
   _item_range.minimum               .    0.0
                                   0.0    0.0
   _item_related.related_name    '_cell.length_a_esd'
   _item_related.function_code     associated_esd
   _item_sub_category.id           cell_length
   _item_type.code                 float
   _item_type_conditions.code      esd
   _item_units.code                angstroms
save_

save_cell.length_a_esd
   _item_description.description
; The standard uncertainty (estimated standard
   deviation) of _cell.length_a.
;
   _item.name                    '_cell.length_a_esd'
   _item.category_id               cell
   _item.mandatory_code            no
   _item_default.value             0.0
    loop_
   _item_dependent.dependent_name
                                 '_cell.length_b_esd'
                                 '_cell.length_c_esd'
   _item_related.related_name    '_cell.length_a'
   _item_related.function_code     associated_value
   _item_sub_category.id           cell_length_esd
   _item_type.code                 float
   _item_units.code                angstroms
save_
```

Example 2.6.5.3 shows the definition of the CELL category from the mmCIF dictionary. The name and textual description of a category are stored in the category named CATEGORY. The item named **_category.mandatory_code** indicates whether the category must appear in any data block based on this dictionary.

The list of data items that uniquely identify each row of a category are stored in the CATEGORY_KEY category. In the example above, the item **_cell.entry_id** is defined as the category key. This item is a reference to the top-level identifier in the mmCIF dictionary, **_entry.id**. Because only a single entry may exist within an mmCIF data block, this choice of category key specifies that only a single row may exist in the CELL category.

Membership in category groups is stored in the category named CATEGORY_GROUP. Each category group must have a corresponding definition in the category CATEGORY_GROUP_LIST. In the above example, the CELL category is assigned to category groups **cell_group** and **inclusive_group**. The former contains categories that describe properties of the crystallographic cell and the latter includes all the categories in the mmCIF dictionary. Organizing categories in category groups is a convenient means of providing a high-level organizational structure for a complex dictionary.

Complete and annotated examples of a category are stored in the CATEGORY_EXAMPLES category. The text of the category example is stored in the item **_category_examples.case** and any associated annotation is stored in the item **_category_examples.detail**.

Table 2.6.5.1. *Relationships defined by* **_item_related.function_code**

Code	Meaning
alternate	The item identified in **_item_related.related_name** is an alternative expression in terms of its application and attributes to the item in this definition
alternate_exclusive	The item identified in **_item_related.related_name** is an alternative expression in terms of its application and attributes to the item in this definition; only one of the alternative forms may be specified
convention	The item identified in **_item_related.related_name** differs from the defined item only in terms of a convention in its expression
conversion_constant	The item identified in **_item_related.related_name** differs from the defined item only by a known constant
conversion_arbitrary	The item identified in **_item_related.related_name** differs from the defined item only by an arbitrary constant
replaces	The defined item replaces the item identified in **_item_related.related_name**
replacedby	The defined item is replaced by the item identified in **_item_related.related_name**
associated_value	The item identified in **_item_related.related_name** is meaningful when associated with the defined item
associated_esd	The item identified in **_item_related.related_name** is the standard uncertainty (estimated standard deviation) of the defined item

Example 2.6.5.3. *Definition of an mmCIF category.*

```
save_CELL
   _category.description
; Data items in the CELL category record details
  about the crystallographic cell parameters.
;
  _category.id                    cell
  _category.mandatory_code        no
  _category_key.name            '_cell.entry_id'
   loop_
  _category_group.id             'inclusive_group'
                                 'cell_group'
  _category_examples.detail
# - - - - - - - - - - - - - - - - - - - - - - - - -
;
Example 1 - based on PDB entry 5HVP and laboratory
            records for the structure corresponding
            to PDB entry 5HVP
;
  _category_examples.case
;
_cell.entry_id                            '5HVP'
_cell.length_a                            58.39
_cell.length_a_esd                         0.05
_cell.length_b                            86.70
_cell.length_b_esd                         0.12
_cell.length_c                            46.27
_cell.length_c_esd                         0.06
_cell.angle_alpha                         90.00
_cell.angle_beta                          90.00
_cell.angle_gamma                         90.00
_cell.volume                             234237
_cell.details
    ; The cell parameters were refined every twenty
      frames during data integration. The cell
      lengths given are the mean of 55 such
      refinements;  the esds given are the root mean
      square deviations of these 55 observations
      from that mean.
    ;
;
save_
```

Example 2.6.5.4 illustrates the definition of a pair of mmCIF categories, CITATION and CITATION_AUTHOR, which share a common data item, **_citation.id**. This example illustrates how an item that occurs in multiple categories may be defined. In the case of the citation identifier, **_citation.id**, the ITEM category is preceded by a **loop_** directive and within this loop all of the definitions of the citation identifier are listed. For instance, the citation identifier is also an item in category CITATION_AUTHOR, where it has the item name **_citation_author.citation_id**. For conformity with the manner in which the core CIF dictionary has been organized, a skeleton definition of the child data item **_citation_author.citation_id** has been included in the dictionary. In fact, this skeleton definition is formally unnecessary.

As a matter of style, the mmCIF dictionary generally defines all of the instances of a data item within the parent definition. Items that are related to the parent definition are also listed in the ITEM_LINKED category. The repetition of a data item in multiple categories gives rise to parent–child relationships between such definitions. These relationships are stored in the ITEM_LINKED category. In Example 2.6.5.4, this category stores the list of data items that are children of the citation identifier **_citation.id**. These include **_citation_author.citation_id**, **_citation_editor.citation_id** and **_software.citation_id**.

2.6.6. Detailed DDL2 specifications

DDL2 is presented here (Chapter 4.10) in the form of a dictionary that is defined in terms of its own definitional elements. This self-consistent description not only provides a prototype for other application dictionaries, but also provides a mechanism by which the consistency and relational integrity of the DDL data model can be independently verified. DDL2 defines a relatively simple set of organizational elements including data blocks, categories, category groups, subcategories and items. Data dictionaries (*e.g.* mmCIF) apply these elements provided by the DDL to describe the knowledge base of an application domain. The following sections provide detailed specifications of each definitional element of DDL2.

2.6.6.1. DDL2 definitions describing data items

In this section, the DDL2 categories that describe the properties of data items are presented. Figs 2.6.4.1 and 2.6.4.2 illustrate the organization of definitional elements in these categories.

2.6.6.1.1. *ITEM*

The category named ITEM is used to assign membership of data items to categories. This category forms the bridge between the category and data-item levels of abstraction. The key data item in this category is the full data-item name, **_item.name**. This name contains both the category and data-item identifiers, and is thus a unique identifier for the data item. The category identifier, **_item.category_id**, is included in this category as a separate mandatory data item. This has been done to provide an explicit reference to those categories that use the category identifier as a unique identifier.

One could alternatively use the category and item identifiers as the basis for this category rather than the concatenated form of the item name, and thus eliminate the redundant specification of the

Example 2.6.5.4. *Related categories linked by parent–child relationships.*

```
save_CITATION
_category.description
; Data items in the CITATION category record details
  about the literature cited as being relevant to
  the contents of the data block.
;
_category.id                  citation
_category.mandatory_code      no
_category_key.name          '_citation.id'
 loop_ _category_group.id    'inclusive_group'
                             'citation_group'
#      ---------  abbreviated definition  ----------
save_

save__citation.id
  _item_description.description
; The value of _citation.id must uniquely identify a
  record in the CITATION list. The _citation.id
  'primary' should be used to indicate the citation
  that the author(s) consider to be the most
  pertinent to the contents of the data block.
;
 loop_       _item.name
             _item.category_id
             _item.mandatory_code
'_citation.id'                   citation         yes
'_citation_author.citation_id'   citation_author  yes
'_citation_editor.citation_id'   citation_editor  yes
'_software.citation_id'          software         yes
_item_aliases.alias_name    '_citation_id'
_item_aliases.dictionary     cif_core.dic
_item_aliases.version        2.0.1
 loop_       _item_linked.child_name
             _item_linked.parent_name
'_citation_author.citation_id'   '_citation.id'
'_citation_editor.citation_id'   '_citation.id'
'_software.citation_id'          '_citation.id'
_item_type.code              code
 loop_ _item_examples.case       'primary'   '1'   '2'
save_

save_CITATION_AUTHOR
_category.description
; Data items in the CITATION_AUTHOR category record
  details about the authors associated with the
  citations in the CITATION list.
;
_category.id                  citation_author
_category.mandatory_code      no
 loop_
_category_key.name    '_citation_author.citation_id'
                      '_citation_author.name'
 loop_ _category_group.id     'inclusive_group'
                              'citation_group'
#      --------- abbreviated definition  ----------
save_

save__citation_author.citation_id
_item_description.description
; This data item is a pointer to _citation.id in the
  CITATION category.
;
_item.name              '_citation_author.citation_id'
_item.mandatory_code         yes
_item_aliases.alias_name
                       '_citation_author_citation_id'
_item_aliases.dictionary     cif_core.dic
_item_aliases.version        2.0.1
save_
```

category identifier. The full name has been used here in order to provide compatibility with existing applications.

The item category also includes a code to indicate whether a data item is mandatory in a category and therefore must be included in any tuple of items in the category. This code, **_item.mandatory_code**, may have three values: `yes`, `no` and `implicit`. This last named value indicates that the item is mandatory, but that the value of this item may be derived from the context. In the case of an item name or a category identifier, these values can be obtained from the current save-frame name. Implicit specification dramatically simplifies the appearance of each dictionary definition because it avoids the repeated declaration of item names and category identifiers that are basis components or the unique identifiers for most categories.

Although the data item **_item.name** is the basis for all of the item-level categories, its definition and properties need only be specified at a single point. Here, the data items that occur in multiple categories are defined only in the parent category. In certain situations, a child data item may be used in a manner which requires a description distinct from the parent data item. For instance, **_item_linked.parent_name** and **_item_linked.child_name** are both data-item names as well as children of **_item.name**, but clearly the manner in which these items are used in the ITEM_LINKED category requires additional description. It is important to note that although the design of this DDL supports the definition of data items in multiple categories within the parent category, it is also possible to provide separate complete definitions within each category.

2.6.6.1.2. *ITEM_ALIASES*

The DDL category ITEM_ALIASES defines the alias names that can be substituted for a data-item name. The alias mechanism also provides a means of identifying items by names other than those that follow the naming conventions used in this DDL. This feature should be used primarily to guarantee the stability of names defined in previously published dictionaries. The items **_item_aliases.name**, **_item_aliases.dictionary** and **_item_aliases.version** form the key for this category. The items **_item_aliases.dictionary** and **_item_aliases.version** are provided to distinguish between dictionaries and different versions of the same dictionary. Any number of unique alias names can be defined for a data item.

2.6.6.1.3. *ITEM_DEFAULT*

The DDL category ITEM_DEFAULT holds default values assigned to data items. Default data values are specified in item **_item_default.value**. Default values are assigned to data items that are not declared within a category. The key item for this category, **_item_default.name**, is a child of **_item.name**. A single default value may be specified for a data item.

2.6.6.1.4. *ITEM_DEPENDENT*

The ITEM_DEPENDENT category defines dependency relationships among data items within a category. Each data item on which a particular data item depends is specified as an item **_item_dependent.dependent_name**. For a data item to be considered completely defined, each of its dependent data items must also be specified.

2.6.6.1.5. *ITEM_DESCRIPTION*

The DDL category ITEM_DESCRIPTION holds a description for each data item. The key item for this category is **_item_description.name**, which is defined in the parent category ITEM. The text of the item description is held by data item **_item_description.description**. A single description may be provided for each data item.