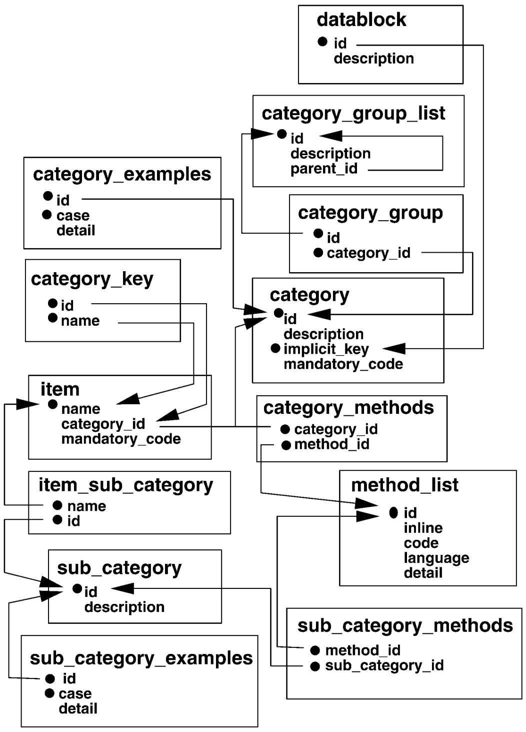

Fig. 2.6.4.2. DDL2 attributes used to specify category, subcategory and category group information. Category identifiers are given in a large typesize and item names are given in a smaller typesize. Parent–child relationships are specified by lines connecting data items with the arrow pointing at the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

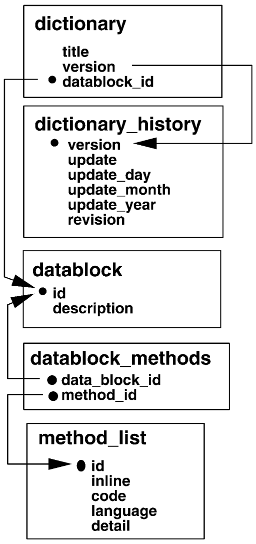

Fig. 2.6.4.3. DDL2 attributes used to specify dictionary and data-block information. Category identifiers are given in a large typesize and item names are given in a smaller typesize. Parent–child relationships are specified by lines connecting data items with the arrow pointing at the parent item. Key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

## 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.

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.

## 2.6.5. 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