

### 3.2 Genericity

A second, non-trivial function of OMBI resides in its genericity. As stated, OMBI is able to create databases on the basis of different database grammars, as long as they respect the fundamental architectural OMBI-principles (the UDS), such as the distinction between FU and LU. This makes OMBI a highly flexible and multifunctional tool, capable of being used in many different environments.

The implementation of this function is such that OMBI is delivered with a standard database grammar. This standard grammar can be varied in a simple way by the user, by adding or deleting data categories, changing finite lists, changing feature co-occurrences etc. More fundamental changes to the grammar have to be programmed on a lower level, in the underlying system Paradox. This is still a relatively simple operation, and does not require highly sophisticated programming skills.

The OMBI User Manual (see Wijne & van Elswijk 1995) contains a complete description of the OMBI-SGML Grammar. The notation used is BNF (Backus-Naur Form) a well-known standard for describing grammars. The top-most rules for e.g. form units, respectively lexical units, read:

```
form-unit      ::= <FORM> spelling [ # wordcat]
                form-field*
                lexical-unit*
                </FORM>

lexical-unit   ::= <LU> resume
                [ # syntactic-subcategory
                [ # semantic-type]]
                lu-field*
                {translation/description}*
                example-unit*
                </LU>
```

### 3.3 Importing and exporting

OMBI's third function is that it can import into its database structure existing MRD's and databases, as long as they have certain minimal structural indications for the recognition of the basic units that OMBI works with. In addition, OMBI can export from its database different SGML-databases and/or dictionaries for specific purposes, by using transformation components. Along the lines mentioned above, a standard

