<DORNBUSH>CML.NLS;1, 15-JUN-73 14:44 CFD ;

COMMAND META LANGUAGE -- CML 02

INTRODUCTION 03

The command meta-language (CML) is a vehicle for describing the syntax and semantics of the user interface to the NLS system. The syntax is described through the tree-meta alternation and succession concepts. The semantis are introduced via built-in functions and semantic conventions. 011

No attempt is made to describe the full semantics of any command via CML, but it is hoped that the front-end interface (parsing and feedback operations) may be explicitly accomodated with these facilities. It will still be necessary, and desirable, to use execution functions to perform the low-level semantics of the command. The CML describes how the command "looks" to the user, rather than what it does in the system. 012

ELEMENTS OF CML 09

RECOGNIZERS 05

Keyword Recognition 013

The process of keyword recognition is independent of the description of the keywords for CML. In the CML description, each keyword is represented by the full text of the keyword. The algorithm used to match a user's typed input against any list of alternative keywords is known as keyword recognition, and is a function of the command interpreter and is independent of the CML description of the command. 014

SELECTION SPECIFICATION 08

Three types of selections are built into CML. They are DSEL, SSEL, and LSEL (see -- the writeup on the command language for the explicit definition of the selections). Basically, they are recognizers which require some entity type as an argument and they return a pointer to a pair of text pointers. The entity type is obtained either by some previous invocation of the recognition function for some list of keyword entities, or use of the VALUEOF built in function. 089

The DSEL, SSEL, and LSEL functions perform all evaluation and feedback operations associated with the selection operations. 090

FEEDBACK CONTROL 06

The feedback control elements of CML are used to provide feedback in addition to the normal feedback generated by the recognizers. This is used to implement additional "noise words" and help feedback. 092

1) adding feedback to the command feedback link. 093

A string may be added to the current command feedback line by enclosing the quoted string in angle brackets. 094

```
extra feedback = '< .SR '>                           095
```

2) replacing the last word in the feedback line. 096

It is possible to replace the last string in the command feedback line by using the string replace facility. This is similar to (1) above except the previous word in the feedback line is deleted before adding the new string. 097

```
replace extra feedback = "<..." .SR '>               098
```

FUNCTION EXECUTION 07

Functions may be invoked at any point in the parse by writing a name of some routine and enclosing a parameter list in parentheses. 099

```
     control = .ID % routine name % '( $<',> var ')                 0100
FORMAL SYNTAX OF CML                                                 010
   system          = "SYSTEM" .ID %system name% sysdef $subsys       065
                           "FINISH";                                 066
   subsys          = "SUBSYSTEM" .ID % subsystem name -- must have
   been previously included in sysdef % #command "END.";             067
   sysdef          = listdef;                                        068
   listdef         = .ID '= #<'/>( .SR / .ID % name of list %) '; ;
                                                                     069
   command         = keyword $( exp ) [confirm] '; ;                 070
   exp             = #<'/>alternative;                               071
   alternative     = #factor;                                        072
   factor          = term/ '( exp ')/ '[ exp ']];                    073
   term            = recognition/ feedback/ control/ help/ assign/
                                                                     074
                           var;                                      075
   assign          = var '← term;                                    076
   var             = .UID;                                           077
   confirm         =(+'<CA>).CHR; % call routine to terminate cmd %
                                                                     078
   recognition     = keyword/ builtinrec/ keyelm;                    079
   keyword         = .SR [ '[ #qualifier '] ];                       080
   keyelm          = .ID % name of  list %;                          081
   qualifier       = "NOTT"/ "NOTD"/ "L1";                           082
   builtinrec      = (("SSEL"/ "DSEL"/ "LSEL") '( var/ ("VALUEOF("
                                                                     083
                           .SR ')) '))/ "VIEWSPECS"/ "LEVADJ"/
   "LIT";                                                            091
   feedback                = "CLEAR";                                086
   help            = '< .SR '>;                                      087
   control         = .ID '( $<',>var ');                             088
```

"delete" (textspec

textspec = "TEXT" TEXT~~spec~~select / "WORD" wordselect

~~TEXTselect = bugselect bugselect~~

TextSelect = A← [illegible] B← Bugselect / MakeText P(A,B)