

# Deontic Logic, Computational Linguistics and Legal Information Systems

EDITED VERSIONS OF SELECTED PAPERS FROM THE  
INTERNATIONAL CONFERENCE ON "LOGIC, INFORMATICS, LAW"  
FLORENCE, ITALY, APRIL 1981

*Editor:*

ANTONIO A. MARTINO

*Professor of Political Science,  
University of Pisa, Italy*

VOLUME II



1982

NORTH-HOLLAND PUBLISHING COMPANY  
AMSTERDAM • NEW YORK • OXFORD

## IMPLEMENTATION OF COMMON COMMAND LANGUAGE ON STAIRS/V5-TLS

Rigoletto Bartoli  
Giuseppe Antonio Romano  
Oreste Signore

CNUCE  
Via S. Maria, 36  
56100 PISA (Italy)

Summary: 1. Introduction: CCL-background - 2. CCL primitives  
- 3. CCL Implementation on STAIRS/V5-TLS - 3.1 Main problems  
- 3.2 Solutions adopted - 4. Conclusions.

### 1. Introduction: CCL-background

The purpose of the EURONET-DIANE network is to make a large number of databases accessible to users throughout the European Community at a cost which is based purely on connection times and number of bytes transmitted and is completely independent of the distances involved.

The network is composed of a number of national nodes and a number of "Hosts" which offer their services, i.e. principally their databases, to the user community. At the present, most of the available databases are of a bibliographic or documentary character.

In a network of this type, it is quite possible that more than one host is offering the same database or that different hosts offer databases which, at least partly, cover the same field of interest and may indeed contain similar information. In the first case, in order to obtain the same information from different dbs it may be necessary to use two or more different query languages. In the second example, a EURONET-DIANE network user may be obliged to access more than one host in order to cover his particular fields of interest as completely as possible. The user may thus be faced with the problem of using (and learning) several query languages as hosts frequently use different information retrieval systems.

Although EURONET-DIANE should not be viewed as a distributed database system, such situations may cause considerable difficulty and inconvenience, particularly for the less expert user. In fact, these problems may well cause the user either to reject the network services totally or access always the same host, thus requesting increasingly complex and heavy services, e.g. duplication of databases already existing on other hosts.

In order to overcome these difficulties and generally to encourage the use of the EURONET-DIANE network of documentary information, the CEC has

produced Guidelines to establish a Standard Command Set for Information Retrieval Systems (NEGUS77).

## 2. CCL primitives

The CCL primitives have been derived from an analysis of the characteristics of a number of different information retrieval systems which, in fact, generally offer similar services in response to commands which have both different names and syntax.

The present paper gives a brief description of the commands and their general characteristics, for a more detailed explanation the reader should refer to (NEGUS77).

### Initialization:

BASE identification of the database to be queried;  
STOP end of a session or part of a session.

### Query formulating:

FIND to formulate a query;  
DISPLAY to have a list of possible search parameters displayed in alphabetic order or logically connected;  
SAVE to save a query for later use.

### Output:

SHOW for on-line printing or display of documents;  
PRINT for off-line printing of documents.

### General:

DEFINE to alter system parameters or create user "macros";  
DELETE to delete queries, printing requests, etc.;  
MORE to display additional data;  
BLACK to display "previous" data;  
HELP to obtain on-line guidance on how to use the system;  
NEWS to obtain the most recent news on the system;  
INFO to receive general information.

The following INFO sub-commands are to be implemented:

EURONET to receive the most recent news on EURONET;  
COST to obtain information on the cost of a query, etc.;  
SCHEDULE to learn hours in which the service is available;  
USERS to see how many users are currently using the system;  
STATUS to receive information on the query currently being processed.

"Transparent" mode usage

OWN to allow use of the original database commands.

In addition, there are a number of general rules governing the user/system dialog.

- All user input is interpreted as a command unless it is a reply to a specific system query.
- All commands must end with a semi colon or a return key.
- Each command can be followed by one or more arguments, which must be separated by a suitable delimiter.
- The system can only be accessed by terminals acceptable to EURONET, i.e. TTY-type terminals.
- Two types of dialog must be available: for expert users (SHORT) and for inexpert users (LONG). The type of dialog can be chosen by the user for the entire session (using the DEFINE command) or just for the current query (in this case a full stop appears before the command, e.g. .BASE).
- Commands can be abbreviated (except for STOP). To avoid ambiguity, a greater number of characters will be necessary for the commands which are most effective.
- A ? is used to request information. In many cases the use of ? is synonymous with the use of the HELP command. In particular, ? followed by a command, signifies that an explanation of that command is requested. A command followed by ? will give information on the use of that command during the current session.
- The words used to formulate a query can be truncated not only on the right hand side but also on the left or in the middle.
- The fields of a document (or record) must be identified by a two-character "field label". A list of the suggested values is given in (NEGUS77) and (NEGUS80).

Certain other recommendations are also made and the reader is referred to the bibliography. Some of these suggestions will be partially discussed in the implementation description.

### 3. CCL implementation on STAIRS/V5-TLS

This chapter examines the main problems which occurred during the implementation of CCL on STAIRS/V5-TLS, and presents the solutions which were adopted.

The choice of STAIRS/VS was determined by the fact that CNUCE had already acquired specific knowhow with this product as an information retrieval system. Furthermore, STAIRS/VS is considered by the manufacturer as a strategic package and should, therefore, have a reasonably long life-term.

TLS was chosen because CCL permits access to a thesaurus and thus the acquisition of TLS implied a considerable reduction in the implementation time and the advantage of a commercial package, i.e. maintenance by the manufacturer. These advantages can still be considered as important even though in actual fact CCL only employs certain TLS functions.

The main problem encountered during implementation was the basic problem which arises whenever an attempt is made to improve a standard package: whether to create an interface or to modify the package itself.

Clearly, the development of an interface offers a number of advantages: it is the most simple solution; the times and human effort involved are more easily estimated; the interface remains mainly stable even if the basic package should be changed. In particular, the introduction of an interface means that the implementation will have an easier, longer life and should be substantially independent of successive package releases unless these involve substantial modifications to the facilities offered by the system. For this reason, other authors have preferred to implement ad hoc interfaces (SCHREI80).

Alterations to the package itself have the big disadvantage of involving a considerable work load which it is difficult to evaluate a priori. Such alterations are often difficult to achieve, and may well be invalidated by subsequent manufacturer's modifications.

These considerations would seem to strongly favour the development of interfaces rather than altering the system itself. However, it should be remembered that CCL is rather a special case. In fact, in view of the reasons behind the original decision to implement a Common Command Language, the development of an interface which merely translates the CCL commands into commands which can be accepted by STAIRS/VS-TLS does not appear to be adequate. In effect, this solution would lead to the implementation of only a subset of the CCL syntax. However, it is very important that the possibilities offered by CCL are not neglected. If CCL offers certain features which though not directly available on the Host system are not in contrast with its philosophy or with its internal data structure, then an attempt should be made also to implement these features. Clearly, once a decision has been made concerning the system modifications to be made, then the advantages and disadvantages of this choice should be weighed carefully, item by item, together with the possible impact with future developments in the system product.

In view of these considerations, we chose to adopt a "mixed" strategy: a certain number of ad hoc modules interpret the CCL commands, transform them into one or more commands in STAIRS format, adding the necessary flags, and finally pass the command to STAIRS. The STAIRS and TLS modules

have been altered in order to satisfy the requirements of the CCL syntax which could not be directly translated as it is not supported by STAIRS.

### 3.1 Main problems

The main problems encountered during the implementation have obviously been those regarding the differences in philosophy between CCL and STAIRS/VS-TLS.

In general, CCL does not offer a user/system dialog. STAIR/VS-TLS, however, privileges this type of dialog. In EURONET, where the user costs are dependent on the number of bytes transferred, this difference is important.

In CCL, each entry is interpreted as a command unless it is in reply to a specific system query.

STAIRS/VS-TLS adopts a very different philosophy: the user enters a certain sub-environment and remains in that environment until a command is entered. An entry is recognized as a command when the first two characters are two full stops (e.g. ..SEARCH or ..BROWSE, etc.).

CCL treats all the fields in a record in the same way. STAIRS/VS distinguishes between "paragraphs" and "formatted fields", and a search is made on the two types of field using two different commands (..SEARCH and ..SELECT).

### 3.2. Solutions adopted

One of the first necessities was to extend the Transaction Work Area, in order to be able to reserve all the specific CCL flags and parameters. This choice means that in practice it is possible to have a product which is mainly based on the functional characteristics of STAIRS/VS-TLS, as the parameters are managed in an area completely external to the area in which the host system parameters and variables are managed.

The modifications made can be grouped into three types relative to:

- general support;
- individual commands;
- logical flow.

The general support modifications are substantially the same as those made to CICS (the Data Communication support maintained at CNUCE) and to the two Macro Services, i.e. DLN020 and VIE900. These include:

TTY Support: elimination of time-out, carriage return after writing the results, support of terminals with odd or even parity, sending of the characters "/\*" to signal that the system is ready to receive new input.

Input Conversion: Conversion of all entries by preceding them with two

full stops (..) so that they can be interpreted by STAIRS as commands.

Treatment of Conversational Input: the input conversion described above does not take place when the system has posed a specific query to the user, e.g. system requests for parameters omitted or incorrect. Nevertheless, STOP is always interpreted as a command, in accordance with the CCL syntax.

Dialog Selection: when a command is read, it is controlled to see whether it is headed by just one full stop; this signifies that the user desires the abbreviated form of dialog.

Messages Selection: as a result of the type of dialog chosen, the system selects the message file to be used. The system will search for the message on the short message file (if the user has specified SHORT), then on the modified message file, and finally on the standard message file.

Map Selection: a philosophy has been adopted which is analogous to that used for message selection.

We have already mentioned that it was felt that the implementation of a simple interface for command transformation would be too limiting. The solution which has been adopted has implied the writing of an ad hoc code practically external to the system, which would be substantially unaffected by any future changes to STAIRS/VS-TLS, and also the modification of certain modules of the host. The transformation of the commands has not been a particularly simple task considering that STAIRS/VS-TLS has no real command driver; the commands are intercepted and accepted at different points in the system. This difficulty probably originated because the system adopts the "environment" and "subcommand" philosophy (e.g. SEARCH environment, SELECT environment, BROWSE subcommands). Consequently, the text of the CCL commands are affected in various points. In addition, certain commands are controlled and accepted even by "low" level modules, i.e. modules which are not at the command driver level. It has thus been necessary to introduce a number of modifications, which are conceptually very simple but quite complex from the maintenance viewpoint, to these modules. These are the "specific modifications to modify the logical flow". However, in general, the modifications necessary for the implementation of the single commands have been made using specific modules written in PL/I, which transform CCL commands into STAIRS commands (or if necessary a series of commands) and introduce a number of parameters in order to extend the CCL's own TWA.

The modifications to the modules mainly consist of blocks of instructions, inserted at points with specific functional characteristics, which mask the display of certain results and simulate certain input.

Consequently, in practice, the basic logic of STAIRS/VS-TLS has not been modified.

#### 4. Conclusions

The Common Command Language has been found to be a fairly powerful and concise language and on the whole quite easily learnt by the inexperienced user.

STAIRS/VS-TLS should be considered as a substantially "complete" package in the sense that it offers practically all the functions included in CCL. However, this product seems to have been designed mainly for video terminals of the "full screen" type. This leads to a very rich dialog which in EURONET, where the user costs are calculated on the basis of the number of characters transferred and connection times and where only TTY type terminals are accepted, become particularly "heavy". Furthermore, the STAIRS/VS-TLS logic which causes "paragraphs" and "formatted fields" to be recognized within a document (see [STAIRSgi], [STIRSog], [STAIRSprm]) produces an asymmetry which is not to be found in CCL.

From our experience we feel that, although the CCL can be implemented on any basic information retrieval system, a complete implementation would probably always necessitate modifications to the internal system software.

#### Acknowledgements

We should like to express our gratitude to S. Lippi of IBM, Italy, for all his help and collaboration during the analysis and development of this project.

In addition, our thanks go to N. Aloia for coding the program for the transaction CONN.

#### Appendix A. Command Description

All the commands which have been implemented are described in this Appendix. For each command is given a description of the acceptable parameters and their significance.

Certain factors which are common to the entire implementation are listed here below:

- the implementation supports terminals of the 3270 type in addition to TTY type terminals;
- on TTY type terminals, the system advises when it is ready to receive input by sending '/?';
- except when the system makes specific queries, each input is interpreted as a command. STOP is always interpreted as a command;
- all the commands can be preceded by a stop (.), which indicates



that in the execution of that command the SHORT dialog is preferred (experienced user).

### A1 - Sign on

The sign-on procedure has been implemented by coding an ad hoc CICS transaction called CONN.

The input format is as follows:

```

-----
| CONN [ECT] CNUCE CCL TLS [user-password] user name |
| STAIRS |
-----

```

Abbreviated forms of CONNECT (CONN, CONNE, CONNEC) are accepted as valid input. Node indication is mandatory (i.e. CNUCE).

If erroneous parameters are entered, the user is given a list of the available transactions:

```

SCST for information retrieval using CCL
AQTL for information retrieval using STAIRS/VS-TLS
AQUA for information retrieval using STAIRS/VS

```

and is requested to enter the code for the transaction he requires.

Valid parameters are CCL, TLS or STAIRS. For CCL or TLS these parameters are followed optionally by the user password and name. If STAIRS is entered, name and password are mandatory.

If no parameter is entered, CCL is assumed by default.

When the sign on procedure is complete, the user can either issue a command, or press the "enter" key, thus obtaining a map which displays the permitted commands.

### A2 - HELP Command

The HELP command gives the user general advice on the use of the system.

H and ? are also accepted as valid inputs.

The command format is:

Command	Parameter	Default value
H[ELP] or ?	[command]	HELP

Where:

command = any valid CCL command

If the command is issued without parameters, HELP or ? is assumed, and the system displays a list of allowed inputs.

If incorrect input is given:

HELP HELP

is assumed.

The user can move backward or forward using the paging commands BACK, MORE and PAGE.

### A3 - BACK Command

During a DISPLAY of HELP, the BACK command allow the user to move backward to review a preceding page.

B is also accepted as valid input.

The command format is:

Command	Parameter	Default value
B [ACK]	[number]	1

Where:

number indicates how many pages the user want to move backward

The input is converted into:

p-n        for 3270

and

doc-n     for TTY

A4 - MORE Command

During a DISPLAY, SHOW or HELP, the MORE command allows the user to move forward to retrieve a successive page.

M is also accepted as valid input.

The command format is:

Command	Parameter	Default value
M[ORE]	[ number ]	1

Where:

number indicates how many pages the user want to move forward

A "carriage return" is understood as:

MORE 1

The input is converted into:

p-n for 3270

and

doc-n for TTY

A5 - PAGE Command

During a DISPLAY, SHOW or HELP, the PAGE command allows the user to retrieve a specific page.

PA is also accepted as valid input.

The command format is:

Command	Parameter	Default value
PA [GE]	[ number ]	current page + 1

Where:

number indicates the page the user want to retrieve.

The input is converted into:

p=n

### A6 - DEFINE Command

The DEFINE command is used to assign values to specific parameters of the system.

The abbreviated form DE is also valid input.

The command format is as follows:

Command	Parameter	Default value
DE[FINE]	[ DL = char ]	X'00'
	[ ;BS = char ]	X'00'
	[ ;M = S(hort)/L(ong) ]	LONG
	[ ;PAGE (pl,ln,mg) ]	24,24,0
	[ ;Default ]	

Where:

DL defines the character used as "delete line" for TTY's

BS defines the character used as "backspace" for TTY's

M defines whether LONG or SHORT messages are desired

PAGE defines the "page size" as:

pl = page length

ln = number of lines per page

mg = margin at the top of the page

The parameters are positional, the absence of a parameter is denoted by a comma. Missing parameters are calculated from the given values. The parentheses are mandatory. If only one parameter is given, the parentheses are not mandatory and the values are taken as pl, setting ln=pl and mg=0.

DEFAULT resets all parameters to their default values.

The order of the parameters is unimportant. Should a parameter be specified more than once, the last specification is taken as that one valid.

### A7 - STOP Command

The STOP command is used to log out of the system.

The command format is:

```

-----
| STOP                                     |
|-----|

```

Whenever the STOP command is given, the user logs out of the systems.

#### A8 - OWN Command

The OWN command allows STAIRS/VS-TLS own commands to be used.

The command format is:

```

-----
| OWN                                     |
|-----|

```

In order to return to the CCL environment, the user should enter the command ..CCL.

#### A9 - BASE Command

BASE selects the database that is to be searched.

BAS is also accepted as valid input.

The command format is:

```

-----
| BAS[E]  [ dbname ]                    |
|          [ ;P      = password ]      |
|          [ ;TL     = thesaurus language ] |
|          [ ;THES   = thesaurus name or LIST] |
|-----|
|          or                             |
|-----|
| BAS[E]  ?                               |
|-----|

```

Where:

dbname	name of database to be accessed (four characters)
P =	indicates the database password, if any (max 8 characters)
TL =	asks for a thesaurus in a specific language (by default E = English)
THES =	specifies the thesaurus name (four characters)

or

THES=LIST	asks for a list of available thesauri
BASE ?	asks for the name of the currently active database and its

structure (i.e. field labels)

Notes that no parameter can be entered if the data base name is missing. If no parameter is entered, a list of available databases is shown, and the user is asked to select one of them (by name or number).

If THES=<thname> is missing and a thesaurus is associated to the database, this thesaurus is automatically selected.

If T=NONE is specified in the DBCB, no thesaurus is selected.

If essential parameters should be missing (e.g. the database name, the password or the thesaurus identification) a normal TLS dialog occurs.

The maps and the read routines have, however, been modified; the databases and the thesauri available are numbered so that they can be chosen either by their name or their order number in the output map.

When the user enters the command BASE ?, the name of the database in which the user is operating and a list of field labels are displayed.

#### A10 - DISPLAY Command

The DISPLAY command is used to obtain listings of logically related thesaurus terms or dictionary words in alphabetical order.

The abbreviated form D is also accepted as valid input.

In the FIND command references to the displayed terms can be made via the "T=" parameter.

The command format is:

D [ISPLAY]		[CT [=] [thr1] descriptor]
		or
		[FT] [=] term
		or
		<number>
		or
		?

Where:

CT asks for a display of logically related terms;  
 FT asks for a display of dictionary words in alphabetical order;  
 thr1 = thesaurus relator, i.e. relator of a semantic field in the selected thesaurus to be displayed (as default ALL is as-

sumed);

descriptor = main descriptor of the semantic field to be displayed (may be masked with \$ sign);

term = masked search term in the dictionary

<number> = a number indicating any display already made. If later on during the session a reference with T= is made in a FIND command, it will be relative to the display which is indicated by <number>;

? asks for a display of the last successful display. If later on during the session a reference with T= is made in a FIND command, it will be relative to the last successful display.

As a result of this command, a map is displayed with sequentially numbered terms, containing relators and descriptors if there has been a CT display, or number of occurrences and words if there has been a FT display.

The user can move backward or forward using the paging commands BACK, MORE and PAGE.

#### All - FIND Command

The FIND command is used to enter search terms and search statements.

The abbreviated form F is also valid input.

The command format is:

F[IND]	identifier	[operator identifier]	...
or			
F[IND]	?		

Where:

identifier =

a) a search term or code which may be truncated or restricted to a particular type (see later identifier modification);

b) a literal search phrase, enclosed in double quotes (e.g. "black and white");

c) a label identifying one or more search terms displayed at the terminal (e.g. T= n [ TO m ] ). If the operator TO is used; the terms are logically OR'ed.

d) a label identifying one or more previous search statements (e.g. S = n [TO m]).

Once again, if the operator TO is used, the terms are logically OR'ed.

and

operator =

a) any boolean logical connector (AND, OR, NOT).

Examples:

    FIND smith and wesson

    FIND S=1 OR bond

    FIND T=3 TO 5 S=2 TO 4 NOT butterfly

If brackets are used to ensure that a sequence of operators is executed in the intended sequence, the logic within the brackets is executed first.

When the same logical operator is to be used to connect a number of terms, a shortened form of list notation can be used.

Example:

    FIND linus AND lucy AND schroeder AND snoopy is equivalent to:

    FIND (AND linus;lucy;schroeder;snoopy)

and

FIND ? asks for a list of the previous searches.

### A11.1 - Identifier modification

An identifier may be modified by using a prefix or suffix to restrict searches to individual fields or to indicate special types of search terms.

#### A11.1.1 - Prefix

- A prefix is divided into a field (or data element).label and a relational connector.
- The accepted relational connectors are:  
= , < , > , <= , > =  
For field labels which do not have linear ranges (e.g. free text) only the connector = is valid.
- With field labels with tree structured ranges, in particular for structured thesauri, standard connectors are:  
DOWN UP NT BT  
Other connectors may be defined by the user in the table VIET <thname>.

#### A11.1.2 - Suffix

- The search statement can be further qualified by using the suffix



facility.

Example:

FIND <f11> = computer/<f12>,<f13>

where <f11>, <f12>, <f13> are field labels.

Note that search statements referring to field labels with linear ranges may not be qualified by using the suffix facility. Therefore, a search statement may be qualified by using prefixes and suffixes only if the relational '=' or a thesaurus relator are used.

#### A11.1.3 - Prefix and parentheses

To avoid repeated use of a given prefix, a shortened form may be used.

Examples:

FIND <f11> = butterfly AND <f11> = daisy

is equivalent to:

FIND <f11> = (butterfly AND daisy)

or:

FIND (butterfly AND daisy)/<f11>

and:

FIND AU = (schultz OR parker OR hart)

is equivalent to:

FIND AU = (OR schultz;parker;hart)

or:

FIND (OR schultz;parker;hart)/AU

#### A11.2 - Adjacency connector

When searching free text, it is possible to specify that two words should occur in the same paragraph.

This is achieved by entering:

<word1> ... <word2>

#### A11.3 - Truncation

The character masking symbols (\$) is used to search masked terms.

Example:

FIND compu\$

will retrieve computational, computer, computing, etc.

#### A11.4 - Syntax limitations

- Field labels with a linear range cannot be mixed with others in one query.
- Backreference in queries with field labels having linear ranges is allowed. However, the backreferenced query(ies) must appear at the beginning. A logical AND between backreferenced queries and entered identifiers is assumed, irrespectively of the actually entered operator.

e.g. FIND S=1 TO 3 OR na=10 AND py >=79

is converted into:

FIND S=1 TO 3 AND na=10 AND py >=79

#### A12 - SHOW Command

SHOW causes the retrieved documents to be displayed at the terminal.

S is also assumed as valid input. The input is not positional. If a parameter should be entered more than once, the last value entered is held to be valid.

The command format is as follows:

Command	Parameter	Default value
S[HOW]	[ S = qn ]	last query
	[ ;R = n [ TO m ] ]	1 TO 5
	[ ;l = k ]	1
	[ ;F = p1;p2;... ]	
	or [ ;Fn ]	ALL

Where:

qn = query number

n = first document number in the list to be displayed

m = last document number in the list to be displayed

k = increment document number (for skimming list)

pn = field label (=STAIRS paragraph or formatted field)

Fn = predefined format

where "n" is a number ranging from 1 to 23. These numbers are in correspondence with the formats "D" to "Z" that can be defined by using the DLNPCDEF macro of STAIRS/V5-TLS.

### A13 - PRINT Command

The PRINT command is used to have retrieved documents printed on the offline printer or on a private print queue.

The abbreviation form P is also valid input.

The input is not positional. If a parameter should be entered more than once, the last value entered is held to be valid.

The command format is as follows:

Command	Parameter	Default value
P[PRINT]	[ S = qn ]	last query
	[ ;R = n [ TO m ] ]	1 TO 50
	[ ;l = k ]	1
	[ ;F = p1;p2;... ]	
	or	ALL
	[ ;Fn ]	
pn	[ ;D = OFFLINE ]	
	or	D = OFFLINE
	[ DISK = prtq ]	

Where:

- qn = query number
- n = first document number in the list to be displayed
- m = last document number in the list to be displayed
- k = increment document number (for skimming list)
- pn = field label (= STAIRS paragraph or formatted field)
- Fn = predefined format
  - where "n" is a number ranging from 1 to 23. These numbers are in correspondence with the formats "D" to "Z" that can be defined using the DLNPCDEF macro of STAIRS/V5-TLS.
- D = OFFLINE asks for offline printing of documents
- DISK = prtq asks for offline printing of documents on private data set, identified as "prtq" in DCT

### Appendix B. Field labels

(quoted from A.E. Negus and A.E.Snowden (NEGUS80)).

AU	author
TI	title
SO	source
CC	classification code (other specialized codes are generally database dependent)
CT	controlled term (i.e. thesaurus term)
UT	uncontrolled term (i.e. free indexing)
FT	free text term
LA	language
NR	report number
CS	corporate source
AF	author affiliation
NP	patent number
CP	patent country
CY	publication country
JT	journal title
JA	journal abbreviation
CO	CODEN
SS	ISSN
SB	ISBN
AB	abstract
NA	abstract number
ND	document number
NC	contract number
ED	computer entry date
PD	publication entry date
PY	publication year
DT	document type
RT	reference
RA	referenced author
RI	referenced inventor
RJ	referenced journal
RP	referenced patent
CR	Chemical Abstracts Registry Number
WL	Wiswesser line notation

### References

- (CCL-001-1) Euronet/CCL Project team: EURONET/CCL implementation with STAIRS/VS - Document number CCL-0001-1 (April 23rd, 1980).
- (CNUCE79) Common Command Language - Implementation on STAIRS/VS (December 1979).
- (CNUCE80a) Bartoli, R., Lippi, S., Romano, G.A., Signore, O., Common Command Language - Implementation on STAIRS/VS - Release 0, in: CNUCE Report No. 174 (July 1980).
- (CNUCE80b) Bartoli, R., Lippi, S., Romano, G.A., Signore, O., Common Command Language - Implementation on STAIRS/VS-TLS - Rel-

ease 1 (July, 1980).

- (CNUCE80c) Bartoli, R., Lippi, S., Romano, G.A., Signore, O., Common Command Language - Implementation on STAIRS/VS-TLS - Implementation Description - Release 1.1 (October 1980).
- (CNUCE80d) Bartoli, R., Lippi, S., Romano, G.A., Signore, O., Common Command Language - Implementation on STAIRS/VS-TLS - User Guide, Release 1.1 (October 1980).
- (Modder80) Modderkolk, A.A., Standardized Command Set - An Overview of the Modification to the IBM Products STAIRS/VS and STAIRS/VS-TLS (January 28th, 1980).
- (NEGUS77) Negus, A.E., Euronet Guideline: Standard Commands for Retrieval Systems (The institution of Electrical Engineers, London 1977) 66.
- (NEGUS80) Negus, A.E., Snowden, A.E., EURONET-DIANE User's Guide (Common Command Language) (1980, Scicon Consultancy International Ltd).
- (SCHREI80) Schreiber, F.A., Difilippo, C., Zagolin, M., Un preprocessor per l'interfacciamento di sistemi eterogenei di information retrieval su una rete di calcolatori. Proceedings of the AICA Annual Congress '80 (29/31 October 1980) 1245-1263.
- (STAIRSgi) Storage and Information Retrieval System/Virtual Storage (STAIRS/VS) - General Information, IBM GH12-5114.
- (STAIRSog) Storage and Information Retrieval System/Virtual Storage (STAIRS/VS) - Operations Guide, IBM SH12-5500.
- (STAIRSprm) Storage and Information Retrieval System/Virtual Storage (STAIRS/VS) - Program Reference Manual, IBM SH12-5400.
- (TLStgi) Storage and Information Retrieval System/Virtual Storage - Thesaurus and Linguistic Integrated System - General Information Manual, IBM GH12-5123.
- (TLStor) Storage and Information Retrieval System/Virtual Storage - Thesaurus and Linguistic Integrated System: Online Retrieval - Program Description and Operations Manual, IBM SH12-5318.
- (TLStg) Storage and Information Retrieval System/ Virtual Storage - Thesaurus and Linguistic Integrated Systems: Thesaurus Generator - Program Description, IBM SH12-5319.