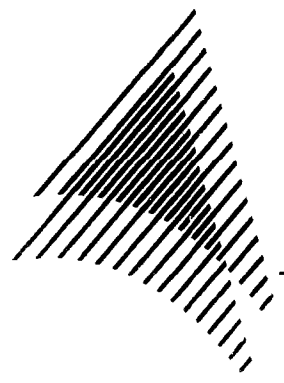


DTIC/TR-88/12

AD-A199 215

①



AD-A199 215

**DOD Gateway Information System (DGIS)
Common Command Language:
The Decision for Artificial Intelligence**

DAITC/TR-88/003

Allan D. Kuhn
Randy L. Bixby
(Defense Technical Information Center)

Duc T. Tran
(Control Data Corporation)

March 1988

DTIC
ELECTE
OCT 27 1988
S D
E

**Defense Applied Information Technology Center
Hypermedia Laboratory**

1800 North Beauregard Street
Alexandria, Virginia 22311
(703) 998-4787
Fax No. (703) 931-3968

88 1027 002

This document has been approved
for public release and sale; its
distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE			5. MONITORING ORGANIZATION REPORT NUMBER(S)	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) DAITC/TR-88/003			7a. NAME OF MONITORING ORGANIZATION Defense Technical Information Center	
6a. NAME OF PERFORMING ORGANIZATION Defense Applied Information Technology Center		6b. OFFICE SYMBOL (If applicable)	7b. ADDRESS (City, State, and ZIP Code) Cameron Station Alexandria, VA 22304	
6c. ADDRESS (City, State, and ZIP Code) 1800 N. Beauregard Street Alexandria, VA 22311			9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8a. NAME OF FUNDING / SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	10. SOURCE OF FUNDING NUMBERS	
8c. ADDRESS (City, State, and ZIP Code)			PROGRAM ELEMENT NO.	PROJECT NO.
			TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) DoD Gateway Information System (DGIS) Common Command Language: The Decision for Artificial Intelligence				
12. PERSONAL AUTHOR(S) Allan D. Kuhn, Randy L. Bixby, Due T. Tran				
13a. TYPE OF REPORT		13b. TIME COVERED FROM TO		14. DATE OF REPORT (Year, Month, Day) 8803
				15. PAGE COUNT 21
16. SUPPLEMENTARY NOTATION Presented at RIA088 Conference, Massachusetts Institute of Technology, Cambridge 21-24 Mar 88				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	Common Command Language, CCL, C language, PROLOG, Artificial Intelligence, Knowledge bases, Blackboard architecture, DoD Gateway Information System, DGIS	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) The first stage of the development of the DGIS Common Command Language (CCL) is described. The first CCL prototyping of DIALOG, BRS, NASA/RECCN, and DRCLS in C-language, and the issues raised by this prototyping, is summarized. The issues concern dealing with the distinctions of information systems. The decision to convert to Artificial Intelligence is explained, including the use of PROLOG, and knowledge base and blackboard architectures. At the time this paper was presented, basic PROLOG-based prototypes had been established for the same four information systems.				
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Allan D. Kuhn			22b. TELEPHONE (Include Area Code) (703) 998-4600	22c. OFFICE SYMBOL DAITC HypLab

DOD GATEWAY INFORMATION SYSTEM (DGIS) COMMON COMMAND LANGUAGE:
THE DECISION FOR ARTIFICIAL INTELLIGENCE

Allan D. Kuhn
Randy L. Bixby
Duc T. Tran

*Presented at
RIA088 Conference on
User-Oriented Content-Based Text and Image Handling
Massachusetts Institute of Technology, Cambridge
March 21-24 1988*

DRITC Hypermedia Laboratory Report Number 1

The Hypermedia Laboratory

Defense Applied Information Technology Center
Alexandria, VA 22312

RIAO 88 PROGRAM

CONFERENCE

**with presentation of prototypes
and operational demonstrations**

USER-ORIENTED CONTENT-BASED TEXT AND IMAGE HANDLING

**Massachusetts Institute of Technology
Cambridge MA.**

March 21-24 1988



Multimedia representation

organized by

**Centre de Hautes Etudes Internationales d'Informatique
Documentaire**

with the assistance of

Centre National de la Recherche Scientifique (CNRS)

**Institut National de Recherche en Informatique et
Automatique (INRIA)**

Ecole Nationale Supérieure des Mines de Paris

Centre National d'Etudes des Télécommunications (CNET)

US Participating Organizations

**American Federation of Information Processing Societies
(AFIPS)**

American Society for Information Science (ASIS)

Information Industry Association (IIA)

**under the direction of Professor Lichnerowicz
de l'Académie des Sciences de Paris**

VOL 2

DOD GATEWAY INFORMATION SYSTEM (DGIS) COMMON COMMAND LANGUAGE:
THE DECISION FOR ARTIFICIAL INTELLIGENCE

Allan D. Kuhn

Defense Technical Information Center, Alexandria, VA.

The results of our first prototyping experiences with the DGIS Common Command Language (CCL) are related. DGIS began its initial prototyping in C language with DIALOG, BRS, NASA/RECON, and DROLS. These prototypes in a third generation algorithmic language brought to surface a number of problems and questions in dealing with the distinctions of information approach. Experiences, results, and conclusions in working with these systems are brought out. The decision to convert to and continue our CCL development with Artificial Intelligence tools is explained. Our effort is a merging of PROLOG and C capabilities, to provide the DGIS user an AI-based searcher assistant interface that makes the human-machine interaction more human-like on DGIS.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

DOD GATEWAY INFORMATION SYSTEM (DGIS) COMMON COMMAND LANGUAGE:
THE DECISION FOR ARTIFICIAL INTELLIGENCE

Allan D. Kuhn

Defense Technical Information Center, Alexandria, VA.

January 1988

I. INTRODUCTION

The Defense Technical Information Center (DTIC) of the U.S. Department of Defense (DoD) has sponsored development of a DoD Gateway Information System (DGIS) since 1982. The purpose of DGIS is to provide online, streamlined methods for identifying, accessing, searching and analyzing data from heterogeneous databases of interest to the DoD community [CGA85]. The following figure is the top menu of DGIS, and shows its core operations to achieve this purpose [KAD86]:

```
0      WELCOME TO THE DoD GATEWAY INFORMATION SYSTEM

*****INFORMATION TRANSFER MODULES
1      directory      DGIS Directory of Online Resources.
2      communicate    Connect to information resources and people
3      process        Information product listing.

*****INFORMATION UTILITIES
4      em             Electronic Mail.
5      files          File operations.

*****SUPPORT INFORMATION
6      help           Description of features.
7      users          DGIS registered users.
8      info           DGIS news and information.
9      cddlog         DGIS full text retrieval.

DGIS HOTLINE NUMBER: (703) 276-6182
or send questions via DGIS EM to dghelp.

Enter a menu number, a command, '?' to backup, 'T' for top, or 'r' to end.
```

DTIC	DOD Gateway Information System	DGIS
------	--------------------------------	------

MISSION

- Provide access to remote systems
- Aggregate information from those systems
- Process and tailor that information

GOAL

- Streamlined system for intermediaries and end-users

DTIC	DOD Gateway Information System	DGIS
------	--------------------------------	------

COMMON COMMAND LANGUAGE

Barrier:

- Diversity of information systems
- Learning requirement/curve
- Constrained information universe

Need:

- Comprehensive information
- Standard language

One of the barriers to searching diverse databases is the lack of a common command language set for retrieving in heterogeneous databases. To overcome this barrier, the Program Manager for the DoD Gateway Information System (DGIS) requested that a Common Command Language (CCL) design activity begin, and in April 1986 a team of three people was assigned the effort.

Just prior to this time a literature search was made to check progress in CCL development. The review showed that CCL literature was sparse, which portended that CCL applications were not readily available for DGIS needs. A search of the Library and Information Science Abstracts (LISA) database on DIALOG in February 1986, for example, resulted in 9 finds, compared to 22 in January 1988. This lack was confirmed by conversations with people working in the field, the conclusion being that extensive CCL literature was only on the verge of surfacing. A review of the citations obtained showed us CCL had two basic thrusts at that time: one thrust was conceptual dissertation, the other the experimenting with specific purposes in mind, rather than general. This situation was further highlighted at the DTIC-MIT jointly sponsored Conference on Computer Interfaces, Boston, May 1986 [DTIC86]; of the three papers submitted specifically addressing CCL, one was about gateway processor software for retrieving with how CCL concepts could be incorporated, one on issues raised by CCL standards, and one a review of the CCL standard drafted by the National Information Standards Organization (NISO). The absence of extensive and helpful

literature at that time on real applications amenable to DGIS requirements showed us that we had the burden of giving definition to the organization of our DGIS CCL development activity.

II. OUR FIRST PROTOTYPING DEVELOPMENTS

Our initial effort to implement CCL was motivated by beginning with a simple approach, and trying to get several prototypes up quickly. We adopted the draft standard for common commands prepared by the National Information Standards Organization (NISO) [NISO86/87]. Programming was done in C, merged with two UNIX utilities that were immediately adaptable to CCL needs. Those two utilities were LEX (generator of lexical analysis programs), and YACC (Yet Another Compiler-Compiler) [UNIXol]. LEX was used for lexical analysis of the CCL prototype C programs, YACC for the syntactical analysis. C was used to implement all remaining semantic processing and miscellaneous tasks [TDTpip].

Communications was highly critical. DGIS had NAM (Network Access Machine) software agents available in the DGIS software for connecting users to databases for native language searching. NAM provided programming for:

- a. Establishing the connection.
- b. Validating user access.
- c. Logging on to the target database, including entry of the logon codes.

The NAM agent was reviewed and found adaptable to CCL for communicating the command and response in searching the remote database system [TDTpip].

Our first prototype was CCL for DIALOG. DIALOG was chosen because it was a system with which many users in the DoD community are familiar, and find easy to use.

The DIALOG prototype was followed by BRS, NASA-RECON, and DROLS in fairly rapid succession. BRS was chosen because it was a another major vendor system with many databases, NASA-RECON because it was a Federal government database system, and finally DROLS, DTIC's database system.

III. FIRST PROTOTYPE RESULTS

We terminated C-programming with completion of the four prototypes. The experience we gained was immeasurably useful. The following issues and features resulted from this first prototyping:

1. The Adaptation of the NAM Connection Agent: As mentioned NAM software for connecting with remote databases was already available. Once the sign-on is completed, the user is connected directly with the database. The user then invokes the CCL translator.

2. CCL Invocation: Currently, once one has accessed a database system through the NAM connection agent, one may invoke the CCL translator with a special key.

3. CCL Translators: The creation of prototype CCL translators taught us that each information system is individualistic and must be treated as such. The translator programming is totally dependent on command mapping requirements for each system. The programmer must also detect anything "hidden" in the target database system that is needed for a response. The CCL translator is a filter that, once activated,

COMMON COMMAND LANGUAGE

First Prototyping

- Feasibility
- C-Language/UNIX System
- Functions
- NISO standard
- Native to standard
- Connection - DIALOG, BRS, DROLS, NASA-RECON
- CCL concepts

CCL Protoype Results

1. Connection Agent (NAM)
2. CCL Invocation
3. CCL Translators/Filters
4. Native Command Option
5. CCL Prompt
6. Command/Entry Verification/Echo
7. Online Documentation
8. Shell Utilities Spawning
9. Non-NISO Commands
10. NISO Commands
11. In-CCL Menus

Proto-conclusion:

Aids to help user navigate the unfamiliar system

intercepts all CCL commands from the user, translates the command, sends the translation (i.e., the target database native command) for execution, and brings the results back to the user [TDtpip]. The translator is deactivated by the conventional <CNTRL>d (exit from a process).

4. Native Command Language Option: The option to use the native command language was necessary when we were prototyping only a selected set of commonly used commands. The entry of a native command was made very simple: at the CCL prompt, one precedes the native command with a backward slash (\) to tell the translator that the native command is coming, e.g.:

```
CCL > \s      (for DIALOG 'select')
```

5. The CCL Prompt: The prompt 'CCL >' was incorporated as a reminder to the user that one has invoked the CCL utility.

6. CCL Command Verification: When the user invokes a common command, the translation of the invocation is echoed in the database system structure, e.g., for DROLS:

```
CCL > find artificial intelligence and psychology
```

```
(echo)      @str@
             artificial intelligence
             and
             psychology
             end
```

```
CCL: Searching...
```

The echo may also be turned off, currently with the command:

CCL> noecho .

7. Online Documentation: The HELP feature to show the user how to use the CCL. The documentation, in very abbreviated form, covers the CCL commands available. For example (DIALOG):

CCL > help find

CCL format

find <term> ...

DIALOG2 format

select <term> ...

DESCRIPTION

Initiate a search.

8. Shell Spawning while in CCL: We incorporated the capability to exploit a UNIX shell, file, or utility while in the CCL. Use of the capability is at the user's discretion; for example, the user may want to list one's files as a review measure while searching a database. The signal to the CCL translator is an initial bang (!), e.g.,

CCL > !ls (for listing files)

CCL > !w (for seeing who is on the system)

etc.

9. New Commands: In developing the DGIS CCL we found that the NISO standard did not cover several items that we deemed useful. Usefulness was based on the following:

- a. Functions, prevalent in systems, that aided the user; an example is successive session cost display.
- b. Functions, not prevalent, seen as highly useful; e.g., listing the accession numbers of finds.
- c. Functions that we found were needed for an operative CCL; an example is cancelling the translation echo display at one's discretion.

The non-NISO commands that we incorporated under the first prototyping are:

COMBINE Do Boolean operations (and, or, not) on previously created sets.

COST Display session cost thus far.

EXECUTE Execute a previously saved search strategy (in target database).

LIST List accession numbers of search results.

NOECHO Cancel native command function echo to CCL command function.

10. NISO Standard Common Commands Incorporated in the First Prototyping: As we developed the four prototypes, we included the following commands to enhance the prototype capabilities: CHOOSE, HELP, FIND, DISPLAY, RELATE, FORWARD, and BACK. START and STOP are taken care of by the DGIS automatic connect and disconnect.

11. CCL System Menu Development: As we progressed through the four prototypes, the more unfamiliar the database systems became. The programmer in particular was totally unfamiliar with DROLS. This is a normal situation because DROLS, in addition to being a terse, no-assist system, is a closed system with a relatively small, registered community. He was, therefore, as a highly skilled technical expert, an ideal person to look at DROLS and divine its appropriate functional CCL requirements.

The very terseness of DROLS (including that lack of a prompt) generated the need to experiment with menu sets to step the unfamiliar user through the database. These menus, basically, provide functional information the expert DROLS searcher knows, but is not on the system. CCL menu examples are:

When invoking CCL CHOOSE in DROLS without designating which database -

DTIC	DOD Gateway Information System	DGIS
------	--------------------------------	------

In-CCL Menu

CCL > choose

Select one of the following files:

- 1. Current Reports**
- 2. Technical Reports**
- 3. New Accessions**
- 4. Work Units**

Please enter your choice (1-4) - - > 2

CCL > find ... (etc.)

- Functional Information**
- Not On System**

When invoking CCL DISPLAY for search results in DROLS without designating a display format -

CCL > display

Select data type to be displayed:

1. Search Results.
2. Qualified Results.
3. User File.
4. Single Technical Report Number.
5. Single Current File Number.
6. Single Work Unit Number.
7. Available Files.
8. Information Log.
9. Order Log.
10. Inverted File.

Please enter your choice (1-10) --> 1
Please enter a field no (0 for end of field list) --> 3
Please enter a field no (0 for end of field list) --> 21
Please enter a field no (0 for end of field list) --> 23
Please enter a field no (0 for end of field list) --> 0

CCL > (Sub command for display mode)

Select a display mode :

1. Item by item display.
2. Continuous display.

Please enter your choice (1-2) --> 1

-- 1 OF 20
-- 1 - AD NUMBER: P003929
-- 3 - [etc.]

The inclusion of the menu sets aids the CCL user to navigate the unfamiliar system, and hopefully helps eliminate the need to totally rely on user manuals.

IV. MAJOR PROBLEM

Each prototype raised issues and problems which we used to refine the successive prototype. As the prototypes progressed, various problems in working with them led to solutions such as HELP features and menus as mentioned above.

The major problem, however, surfaced as a result of our cumulative experience. We learned that creating "Common Command Language" was NOT a panacea. Programming a "standard" command

CCL PROBLEM

- Information system idiosyncracies
- Language substitution
- System formats
 - Choose
 - Display

Proto-conclusion:

The problem is the individual system operating characteristics, not commands.

CCL-ISSUE ELEMENTS

- Standard Language
- CCL

System

- Purpose of CCL

S

Proto-conclusion:

CCLS must be built to serve the purpose of the parent system.

language was in actuality only substituting one command language for another.

This was most apparent when the DISPLAY function is employed. Quite factually, if the user does not know the DISPLAY formats of an unfamiliar system, one cannot see results. A command with less serious consequences is the FIND function. Using FIND, the user is very likely to be able to enter the query and foment results. But any function involving a display is likely to be dead-ended in no display. This situation simply does not obviate the need for referral to a system's user documentation, which gives instruction in terms of its native command language.

Another example is the CHOOSE function. Some systems identify databases by number, others by acronym. For BRS, one must enter CHOOSE NTIS; in DIALOG, CHOOSE 6. The hydra of options and formats keeps cropping up. Each system must be addressed individually, with the goal of having some central pattern program to draw upon. The crutch we used for the C language-based CCL prototype is the menu.

The creation of a CCL is only one component of the "CCL-need" issue. A second component is creation of a CCL System that allows a user to search in unfamiliar database systems without needing to know that system's operating characteristics. A third component is identifying the critical purposes that a CCL system is to serve.

In the case of DGIS, the criteria for CCL purposes are the DGIS information processing operations, particularly in postprocessing downloaded files. A DGIS postprocessing requirement is to have a tagged citation for translation.

Downloaded citations must be translated into the DGIS standard citation format before the automated processing routines can be applied.

This requirement is an example of a criterion for a DGIS CCL system. The CCL system must include function default results for those users unfamiliar with a database, particularly for DISPLAY. The default, on simple invocation of DISPLAY, will provide the fully tagged citation. Additional elements, such as menus and question prompts, e.g., "DISPLAY on last set? y/n," must also be incorporated.

The case of CHOOSE represents another problem environment. In DGIS the solution is the eventual integration of CCL with a Directory of Online Resources. When this is accomplished, the query will be forwarded automatically to the relevant databases through CCL.

The real demon for CCL has turned out to be the idiosyncratic operating characteristics of each database system.

V. THE DECISION FOR ARTIFICIAL INTELLIGENCE

1. THE CAUSE

The rigidity and constraints of a straight algorithmic program-based CCL discovered during the prototypings led us to exploring the potential of artificial intelligence. The natural language and expert system possibilities of AI were very appealing. The project's programming technical expert reviewed the main AI programming languages. His recommendation was to explore AI applications with PROLOG, a simple but powerful relational programming language based on the idea of programming in logic [BA-86].

The initial technical reasons for selecting PROLOG were [TDTpip]:

a. The Reversibility of PROLOG: In determining object relationships, a program can be written establishing those relationships, with the inverse of the relationship inherent in the program.

b. Its Database Capability: In that PROLOG has its own internal databases, this feature allows a PROLOG program to manipulate codes as relations that can be asserted or deleted. PROLOG incorporation in CCL includes extending to external databases, e.g., INGRES DBMS in the DGIS software, to achieve the flexibility of storing knowledge in both PROLOG internal databases and traditional external databases. This allows including more powerful database technology in the program system for greater performance and easier use of DGIS by the enduser.

c. The Separation of Logic and Control: A PROLOG program amalgamates rules and facts, basically making one also the other. Although they are governed by a default execution control, the control can be easily supplemented or replaced by more powerful meta-rules also coded in PROLOG.

d. Object Inheritance and Message Passing: These are two powerful features of object language methodology. Both are easily implemented and embedded in PROLOG. Both features are elemental for the more graceful functioning of CCL.

2. THE CONCEPTUAL RESTRUCTURING OF CCL

Using C language programming, the basic CCL elements consisted of the user, the CCL, the database language processor, and the database information accessed. The jump to PROLOG opened

COMMON COMMAND LANGUAGE SYSTEM - CCLS

Decision for AI

- Heterogeneous Universe
- Third Generation Rigidity
- Potential for Universality
- Human-machine Interface

CCLS - AI

PROLOG

Requirements Merger

- Ease of Learning
- Knowledge Base Building
- NL to CCL Requirements
- Requirements Control

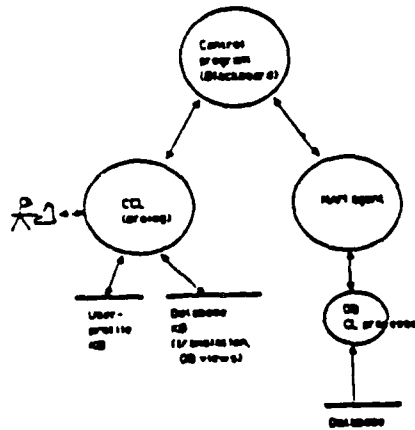
Technical Properties

- Reversible Relationships
- Database Capability
- Separation of Logic and Control
- Object Inheritance - Message Passing

*A simple but powerful programming language
based on programming logic.*

in

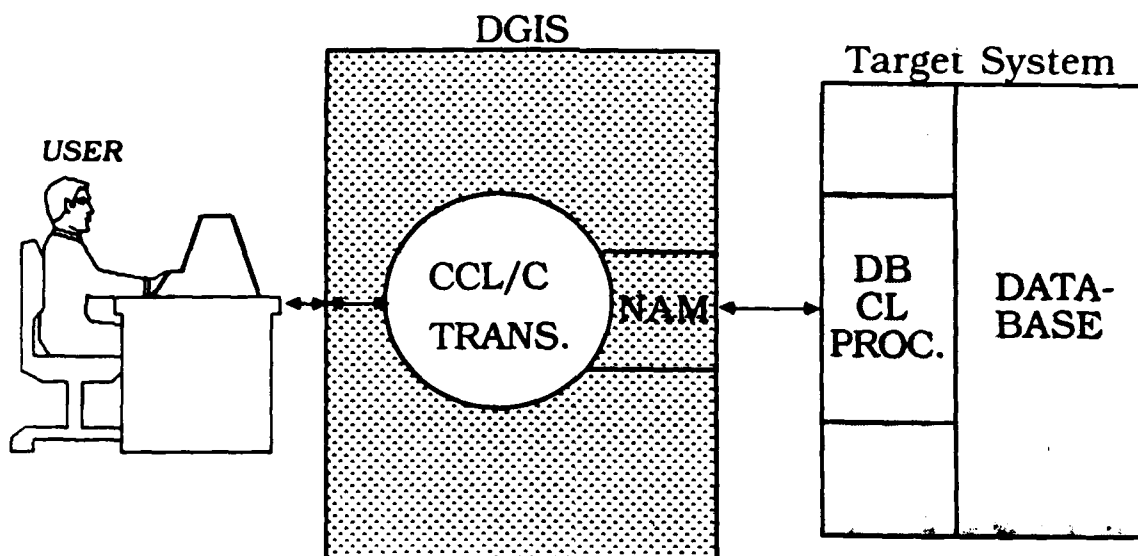
new possibilities in which CCL now could be handled as a knowledge-based system. The CCL conceptual structure now became [TDTpip]:



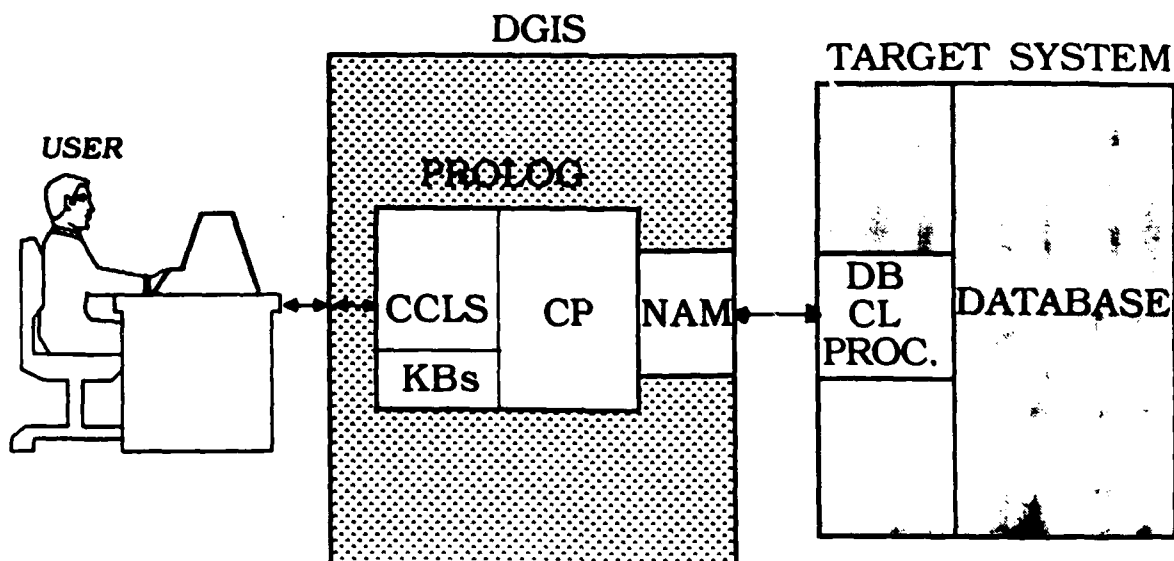
CCL as a knowledge-based system

To the DGIS user making use of CCL, the fact that CCL will be PROLOG-driven is transparent. The PROLOG CCL, however, in serving the user, will draw on the command language knowledge base, and also a CCL-user profile knowledge base (still to be developed). The user's query and profile data will be controlled through the control program blackboard, which will coordinate translation and communications in a continuous real-time system mode [BA-86] through the NAM connection agent. The NAM agent passes the communications to and from the target database system's command language processor for searching on the database information.

CCL - C



CCLS - AI



With the transition to an AI-based CCL System, the goals of DGIS CCL have been re-constituted to incorporate AI-supported capabilities as follows:

- a. One command language to communicate with all bibliographic databases.
- b. Creation a CCL System that assists the user in searching unfamiliar database systems.
- c. Provision of a user-friendly search session.
- d. Provision of an intelligent, user-useful search session.
- e. Flexibility to adapt easily to changes and enhancements.

3. OTHER CHANGES IN THE ACTIVITY

Because of the relative ease of learning PROLOG programming, another effect of making the transition to PROLOG was to transfer much of that programming from the technical expert to the requirements expert. This change allowed fuller control of the command requirements, from command language researching to command language knowledge base building. This also allowed the technical person to concentrate on the knowledge base - database system connector programs, in itself a programming-intensive activity.

4. FUTURE DIRECTIONS

Our next phase in CCL is a melding of PROLOG implementation, expert system building, and C supplementary programming. The PROLOG-based CCL has two parts [TDTpip]. One part is fixed, in compiled PROLOG code; the second is variable, in interpretive PROLOG code. The variable part loads and processes information from the two knowledge bases (KB), the command language KB and the user profile KB. Appropriate tools will be incorporated to

maintain the KBs (adding, deleting, modifying information). We are currently (December 1987) procuring an artificial intelligence processor system and an expert system building software tool. The processor will be networked to the DGIS computer system, and will serve to both develop and maintain AI applications in CCL and other AI applications on DGIS [KAD87b].

We are investigating several schemes for KB organization. In general, we plan to couple PROLOG with a Relational DBMS (RDBMS) where large KBs (most of which are facts) will reside. The technical issue here is the interface between PROLOG and the RDBMS (likely INGRES). We intend to make this interface through SQL (standard query language) so that it can work with any RDBMS, rather than only with INGRES [TDTpip].

Other CCL system application factors are:

a. CCL Integration with Other DGIS Functions: Other DGIS operations are potentials for AI applications, with which to link with CCL. One is the DGIS Directory of Online Resources, wherein a user's query resources are identified and communicated with automatically and simultaneously. Another is the DGIS postprocessing routines, with which the multiple resource responses are automatically downloaded, translated, merged, and processed (or tailored), based on a one-pass instruction entry with which the user invokes the whole process.

b. Planning Capability: Includes the preliminary structuring of multiple queries and the combining of target databases' result sets.

c. Learning Capability for CCL: Employing learning solution paths [BA-86] for optimizing the information added to the command language KB and the user profile KB.

d. Migrate to Natural Language: The NISO and appended CCL will be the backbone of the DGIS CCL, but in migrating to Natural Language dialogue, will become transparent in a command language translation supporting role.

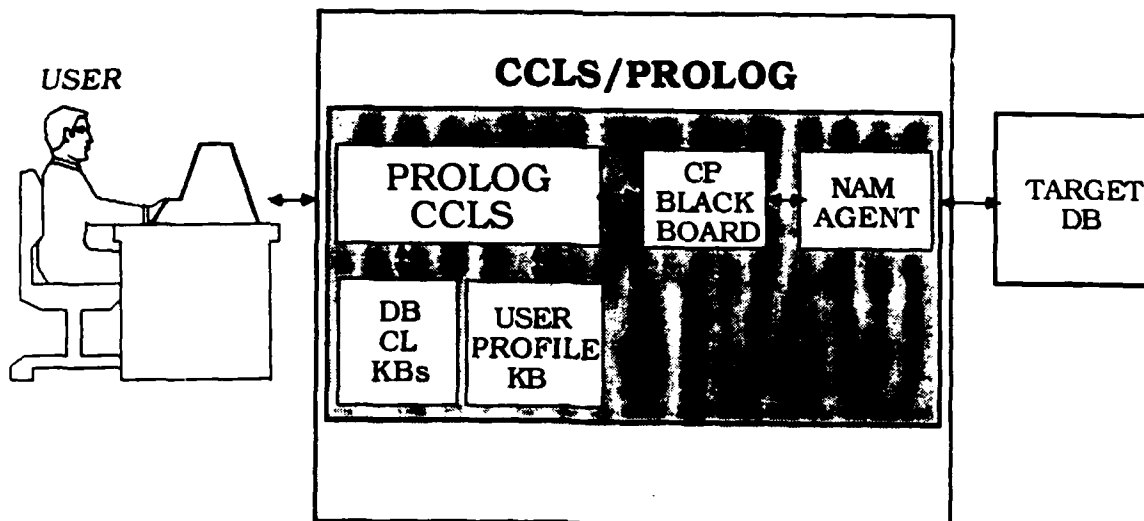
e. Provide Simultaneous Database Access Capability: That is, true concurrent connecting with, searching in, and downloading of results from multiple systems.

Considering these CCL application goals, the following technical aspects that explain our programming approach to the DGIS CCL interface are (and verbatim from [TDT87]):

The DGIS CCL is structured as a knowledge-based system. CCL can be thought of as a black box between the user and the host database. The control program (CP) of CCL is a blackboard-based architecture PROLOG program that controls the interaction between CCL agents and the communication agents (we use the term 'knowledge source' rather than agents to be consistent with the literature on blackboard systems). There will be two types of CCL knowledge bases. One is pertinent to the user information, and the other is the knowledge base about databases. The user knowledge base (UKB) system will store information relevant to a particular user, or a group of users. Examples of information stored in the UKB are areas of interest (database names), shorthand (CCL scripts, aliases, et al.), user privileges, etc. This information is needed by the CCL to intelligently converse with and interpret commands from the user. The database knowledge

CCLS - AI

DGIS



CCLS - AI

APPLICATION

- CCLS Integration With Other DGIS Functions
- CCLS Planning Capability
- Simultaneous DB Access
- CCLS Learning Capability
- Migration To Natural Language

To Assist The User To Navigate All Systems

base (DKB) contains information needed to translate CCL commands into host database commands and to understand the returning results and errors from the database.

The control program (CP) is a typical blackboard-based program. It is a PROLOG implementation of an object-oriented system where the blackboard is nothing more than a general object that registers and monitors progress of the related knowledge sources. Each knowledge source is an object that is activated and deactivated by messages. A knowledge source's progress and results are also composed in terms of messages whenever possible for the blackboard of the CP. This includes the packaged messages received through the NAM connection agent, already mentioned.

The construction of the CCL knowledge bases is being done with the aid of domain experts at DTIC. These experts help in stating the requirements for interpreting the native command languages, and capture expert searcher usage of command languages. Additionally, a number of PROLOG tools that help maintain and validate the knowledge bases will be implemented.

All this to resolve the problem of, as the requirements expert states [BRL87]: "Ever since the creation of the second online database search system, the problem of multiple command languages has plagued online searchers."

ACKNOWLEDGEMENTS

Ms. Randy L. Bixby, Defense Technical Information Center, is the requirements expert on the project. Mr. Duc Tien Tran, Consultant, Control Data Corp., is the programming technical expert. Thanks is given to Ms. Sharon Karalus for reviewing and properly styling this paper.

REFERENCES

- [BA-86] Bundy, Alan. Catalogue of Artificial Intelligence Tools. Springer-Verlag Berlin Heidelberg. 1986.
- [BRL87] Bixby, Randy L. The DoD Gateway Information System (DGIS): Common Command Language Mapping. Defense Technical Information Center, Alexandria, VA. October, 1987, AD-A185 951.
- [CGA85] Cotter, Gladys A. The DoD Gateway Information System. Defense Technical Information Center, Alexandria, VA. October 1985, AD-A161 701.
- [DTIC86] Jacobson, Carol E., and Shirley A. Witges. Proceedings of the Second Conference on Computer Interfaces and Intermediaries for Information Retrieval, May 28-31, 1986, Boston, Massachusetts. Defense Technical Information Center, Alexandria, VA, and Massachusetts Institute of Technology, Boston. AD-A174 000.
- [KAD86] Kuhn, Allan D., and Gladys A. Cotter. The DoD Gateway Information System (DGIS): User Interface Design. Defense Technical Information Center, Alexandria, VA. August 1986, AD-A174 150.

[KAD87b] Kuhn, Allan D. Toward an Artificial Intelligence Environment for DTIC: Proposed Tasks, Recommended Configurations, Projected Start-Up Costs. Defense Technical Information Center, Alexandria, VA. May 1987, AD-A181 103.

[NISO86/87] National Information Standards Organization (NISO), Washington, DC. American National Standard -- For Information Sciences: Common Command Language for Online Interactive Information Retrieval (DRAFT). [March 1987] ANSI Z39.58-198X.

[TDT87] Tran, Duc T. DoD Gateway Information System (DGIS) Common Command Language: PROLOG Knowledge Base Profile. Consultant; Defense Technical Information Center, Alexandria, VA. October 1987, AD-A185 150.

[TDTpip] Tran, Duc T. DoD Gateway Information System (DGIS): Prototype Programming the DGIS Common Command Language. Consultant; Defense Technical Information Center, Alexandria, VA. Paper-in-progress.

[UNIXol] University of California-Berkeley. UNIX Programmer's Manual. LEX(1); YACC(1); NAM(1T). Online manual, UNIX Operating System BSD 4.2.