SRI International

KLAUS: A SYSTEM FOR MANAGING INFORMATION AND COMPUTATIONAL RESOURCES

Technical Note 230

October 1980

By: Gary G. Hendrix, Program Director

Artificial Intelligence Center
Computer Science and Technology Division

SRI Project 1894

This research was supported by the Defense Advanced Research Projects Agency with the Naval Electronic Systems Command under contract N00039-80-C-0575. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency of the United States Government.

SRI International

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

ABSTRACT

This report presents a broad-brush description of the basic goals and philosophy of a research program at SRI International (SRI) aimed at developing the technology needed to support systems that can be tutored in English about new subject areas, and that can thereafter aid the initial or subsequent user in filing and retrieving information, and in conveniently applying to the new subject area other computer software, such as data-base management systems (DBMS), planners, schedulers, report generators, simulators and the like. These systems, which we call Knowledge Learning and Using Systems (KLAUS), are intended to act as brokers between the user's needs, as expressed in the user's terms, and the resources available in a rich computational environment.

I INTRODUCTION

This report presents a broad-brush description of the basic goals and philosophy of a research program at SRI International (SRI) aimed at developing the technology needed to support systems that can be tutored in English about new subject areas, and that can thereafter aid the initial or subsequent user in filing and retrieving information, and in conveniently applying to the new subject area other computer software, such as data-base management systems (DBMS), planners, schedulers, report generators, simulators and the like. These systems, which we call Knowledge Learning and Using Systems (KLAUS), are intended to act as brokers between the user's needs, as expressed in the user's terms, and the resources available in a rich computational environment. All KLAUS systems either have been or will be built on a central core that integrates knowledge acquisition, knowledge representation, deduction, and natural-language processing abilities. To meet the needs of particular classes of applications, the KLAUS core can be augmented with additional capabilities. Our first KLAUS implementation was completed in early 1980 and is described in [25].

In a nutshell, the core concept of a KLAUS is that of an interactive system preprogrammed with essential skills for readily learning the concepts and vocabulary of new subject domains, and with expertise for applying acquired knowledge in problem-solving situations. A KLAUS is tutored about new domains in English (perhaps also using tables, menus, and domain-specific formalisms). While being taught, a KLAUS does not play a passive role, but actively looks for gaps and inconsistencies in its knowledge, asking its tutor pointed clarification questions. In this manner, the KLAUS aids the user in formalizing, organizing, and clarifying his ideas. After tutoring, a KLAUS can aid its tutor and other users in performing tasks that require combining knowledge of the new domain with knowledge of how to use sophisticated computer systems. In particular, it can

* Construct knowledge bases for other AI systems.
* Perform the user's bookkeeping.
* Help recall facts forgotten by the user.
* Spot inconsistencies in what the system has been told.
* Generate high-level reports or detailed descriptions of parts of the system's knowledge.
* Answer questions and make deductions.
* Help the user interface with other computer software.
* Explain the system's operations.

The most immediate benefits from this type of information and resource management system will most likely occur in the development of personal assistants, automated command and control systems, project management systems, computer-based consulting systems, data-base query systems, and advanced computer-based instruction systems.

The ideas for KLAUS have been developed in response to problems that have arisen in our work at SRI on the DARPA/IPTO-sponsored LADDER project [54]. In essence, these problems come down to the need to be able to transfer advanced AI technology easily from one domain to another and the need to provide users with a uniform view of a body of computational resources.

One of the key requirements in building any intelligent system is the ready availability of a large body of machine-usable information about the domain of application. Currently, the construction of large knowledge-bases is a tedious and time-consuming operation that typically must be performed by a highly skilled computer specialist. Thus, research in the machine-based acquisition of knowledge about new domains and the use of acquired knowledge in the performance of a variety of tasks is central to the project.

Much of the knowledge needed by many intelligent systems is obtainable through discussions in the English language with humans. For this reason, systems that communicate with domain experts in a natural language seem to have the most plausible design for automating part of the knowledge acquisition process.

An important aspect of the KLAUS concept is that while learning about new domains, a KLAUS also learns about the language (primarily the vocabulary and jargon) that users employ in discussing the domain. This simultaneous acquisition of concepts and language gives full recognition to the fact that language and reasoning are integrally connected.

Most knowledge acquisition research to date has been rather preliminary and has concentrated on the development of "knowledge structure editors" (Davis [7], Stefik [60]), which may be viewed as special-purpose, interactive programming languages that ease the tedious burden of creating and modifying data structures encoding large, specialized knowledge bases. For some applications, such editors employed by trained technicians may be the best way to build knowledge bases. In contrast with such structure editors, an English-based acquisition system is intended to free the user from any need to ever notice or mention the data structures used by the system in its internal operations. That is, the system should automatically create and manipulate the internal data structures without any explicit instructions from users to do so.

Solving the knowledge acquisition problem and creating systems that can manage diverse computational resources will undoubtedly require many years of effort; however, there should be many significant milestones along the way that will provide highly useful systems. SRI's current research effort is exploratory in nature and focused on delineating the technical problems inherent in the KLAUS concept.

II THE KLAUS CONCEPT

A. The KLAUS Core System

The basic function of a KLAUS core system is to manage information. In this regard, it is related to conventional DBMSs--but there are very significant differences.

In a conventional DBMS, a data-base administrator defines a number of files with their various fields and interconnections. That is, he explicitly defines a special-purpose data structure for encoding information about some domain. Subsequently, this data structure is filled out by technicians (or programs) through interactions that deal explicitly with the form of the data and only incidentally with its meaning. Finally, once a conventional data base has been defined and filled out, questions may be posed against the data using query languages that typically compel users to know and be able to manipulate the structure of the data base explicitly. (Recent work on LADDER-like systems has eased part of the access problem [33].)

In contrast, a KLAUS user should never need to mention data structures. He is free to devote his attention, for example, to ships, ports, and weapons systems, rather than to files, fields, links, and records. He volunteers descriptions of his new domain in English and answers easily understood questions posed by the system. For example, suppose the user wishes to tell the system about the home ports of various ships. In a conventional DBMS, a SHIP-HOME file might be declared with the fields SHIP-NAME and PORT. Records might be entered in this file, such as

KENNEDY	MAYPORT
FOX	CHARLESTON

This would be done by creating special-purpose data structures and running special-purpose programs--activities that could be performed easily only by someone with substantial experience with the DBMS. In contrast, a KLAUS would be told about home ports through a dialogue something like the following:*

```
User: > EVERY SHIP HAS A HOME PORT

KLAUS:  Is a HOME PORT a PORT?

User: > YES

KLAUS:  Do only ships have home ports?

User: > (Types carriage return.  This indicates that KLAUS
         should stop asking questions.)

User: > THE HOME PORT OF FOX IS CHARLESTON

KLAUS:  Is FOX a new ship?

      (KLAUS asks because something other than a ship might
       have a home port, yet ships are the only objects that
       are known to have them.)

User: > YES

User: > KENNEDY'S IS MAYPORT

KLAUS:  The home port of KENNEDY is Mayport.

        (KLAUS already knows KENNEDY is a ship.)

User: > WHAT IS FOX'S HOME PORT

KLAUS:  Charleston

User: > WHAT ARE THE HOME PORTS OF SHIPS YOU KNOW ABOUT
```

* We assume KLAUS has already been told about ships and ports, and, in particular, about the ship named KENNEDY, but not the ship named FOX

KLAUS: I know about 127 ships, but I only know the home ports of 2. Here they are:

SHIP	HOME PORT
FOX	CHARLESTON
KENNEDY	MAYPORT

User: > (Types carriage return. Gives control to KLAUS.)

KLAUS: Do only ships have home ports?

User: > NO

KLAUS: What other things have them?

User: > (Types carriage return. Does not answer question. Gets control back from KLAUS.)

User: > (Types carriage return. Gives control back to KLAUS, which is expected to go on to some new questions.)

KLAUS: I am missing several pieces of information about the ship FOX.

Do you want to tell me more about it now?

User: > QUIT (User ends session.)

In short, the purpose of both conventional DBMSs and the KLAUS core is to file, sort, selectively retrieve, and display data in various formats. A primary distinction between DBMSs and KLAUS is that DBMSs are systems for dealing with formal data structures through a formal command/query language, whereas a KLAUS is a system for learning and manipulating concepts through interactions stated primarily in English.

A second and equally significant distinction is that conventional DBMSs store only propositions of the form that logicians call "ground literals." Ground literals are single-fact elements such as

THE KENNEDY IS OWNED BY THE U.S.

Facts involving logical connectives, such as "OR" in

KENNEDY IS EITHER IN PORT OR AT SEA,

or involving quantification, such as

ALL CARRIERS CARRY DOCTORS,

are not ground literals and cannot be explicitly represented in a conventional DBMS. They could, however, be represented, discussed and reasoned about by a KLAUS.

For example, a KLAUS should be able to engage in conversations similar to the following. Conventional DBMSs cannot handle this conversation even if it is converted into a formal language because it involves both the representation of quantification and the use of deductive logic.

```
User: > ALL CARRIERS CARRY A DOCTOR.

KLAUS:  OK.

User: > WHAT SHIPS HAVE DOCTORS?

KLAUS:  May KLAUS assume that if a ship CARRIES a doctor
        then the ship HAS a doctor?

User: > YES

KLAUS:  All carriers have doctors.  Do you want an explicit
        list of the carriers?

User: > NO

User: > CARRIERS ARE AT LEAST 500 FEET LONG.

KLAUS:  OK.

User: > HOW LONG IS THE KENNEDY

KLAUS:  I do not have the explicit length of the Kennedy.
        The length of the Kennedy is at least 500 feet.

User: > HOW DO YOU KNOW
```

```
|  KLAUS:  You told me:                                  |
|             F1)  Kennedy is a carrier.                 |
|             F2)  All carriers have a length of at least |
|                  500 feet.                             |
|          From F1 and F2 I deduce that:                 |
|             D3)  Kennedy has a length of at least 500 feet. |
|                                                        |
----------------------------------------------------------
```

In summary, a KLAUS core system will differ from a DBMS in that users

* Talk about domain concepts rather than data structures
* Use English for most interactions (including data input)
* Are not restricted to ground literals.

Moreover, as shown below, a KLAUS has the opportunity to grow in a number of directions beyond the potential of a DBMS.

B. The Larger Picture

The KLAUS core system is itself a major extension of conventional DBMSs because of its abilities to deal not only with ground literals, but also with other types of logical statements, and, most importantly, because of its ability to learn the concepts and vocabulary of new domains through dialogues in English that do not mention KLAUS's internal data structures explicitly. But the full power of the KLAUS concept only begins to emerge when the KLAUS core is enhanced with the ability to describe and use auxiliary software systems. The ultimate goal of the KLAUS project is not merely to provide a super-DBMS, but rather to provide a sophisticated software executive that combines a fluent knowledge of the English language with a command of facts about new domains and about how to coordinate the use of various complex software systems in the solution of user problems.

1. A Sample Dialogue

The need for and basic characteristics of an advanced KLAUS are illustrated by the following deceptively simple dialogue.

```
1.  User: > DO I HAVE NEW MAIL?

2.  KLAUS:  You have one message
              FROM: J. HARMS
              SUBJECT:  Possible bug in CG radar systems
            Do you want to read it now?

3.  User: > YES

4.  KLAUS:  (Presents message about possible defect in the
             radar used by CGs -- guided missile cruisers.)

5.  User: > WHO ARE THE COMMANDERS OF OUR GUIDED MISSILE
            CRUISERS IN THE MED

6.  KLAUS:  The following table presents the officers of
            guided missile cruisers in the Med.
```

officer	ship
CAPT. J. HARMS	DANIELS J
CAPT. O. EVANS	WAINWRIGHT
CAPT. T. FRENZINGER	JOUETT
CAPT. J. BRANIN	HORNE
CAPT. W. HOHMANN	STERETT
CAPT. C. MICHAELS	STANDLEY WH
CAPT. J. EVERETT	FOX
CAPT. J. TOWNES	BIDDLE
CAPT. H. GRAHAM	LEAHY
CAPT. P. PHILHOWER	YARNELL HE
CAPT. J. YOUNG	WORDEN

```
7.  User: > FORWARD THE MESSAGE TO ALL OF THEM EXCEPT
            GRAHAM AND PHILHOWER.
```

In this dialogue,* the user first asks about his mail (Sentence 1). When there are only a few messages (in this case, one), KLAUS displays the message SUBJECTs and the names of the senders (Sentence 2). Then the user says he wants to read his one message and

---

* This dialogue does not require KLAUS to understand the title or body of the message; however, KLAUS must understand that there was a message and that it was from J. Harms.

KLAUS causes it to be displayed (Sentences 3 and 4). The message is about a possible bug in the radar systems of guided missile cruisers. Let us assume that part of the user's mission is to convey information to U.S. ships in the Mediterranean. The user decides to find out who the commanding officers of the relevant ships are and so asks Question 5. When he sees the answer in 6, he is reminded that some of the CGs are of the Leahy class (these include Leahy, Yarnell and Worden), which, for reasons implied by the message and the user's personal knowledge, should not be affected by the radar problem. He therefore asks that the message be forwarded to all the non-Leahy officers, but he does not exclude YOUNG, because YOUNG has asked to be sent copies of all messages regarding any class of CG.*

It is important to note that the major work required for this dialogue can be performed by existing systems. Sentences 1-4 and 7 use the ARPANET mail system in elementary ways; Sentences 5 and 6 can be supported by existing DBMS technology. Indeed, the S-LADDER system can accept 5 in exactly the form stated.

But a key component needed to support such a dialogue does not currently exist. The missing component is a mechanism that

* Interacts with the user in natural language.
* Decides which of several subsystems should be invoked to meet the user's needs.
* Automatically creates and executes the calling sequences needed to invoke the appropriate subsystems.
* Maintains a dialogue context with the user that reflects and integrates the various local contexts established by and for the use of subsystems.

In other words, we lack a level of awareness of the relationships between the various software packages, the user, the domain of application, and their possible interactions. A long-term goal of the KLAUS project is to fulfill that need.

----

* The plausibility of this dialogue from the perspective of Navy operations is not important here. The point is that real problems involve interactions of the same type, even though the subject matter may be quite different.

This appears to be a major unsolved problem in the DoD's Advanced Command and Control Architectural Testbed (ACCAT).* While an ACCAT user has a number of sophisticated individual software packages available to help him, he must coordinate these packages himself. This requires knowing a large number of different user interfaces, and reestablishing the context of his current interest each time he moves from one package to another.

Sentence 7 from the dialogue illustrates the missing level of awareness very well. Having (perhaps unknowingly) invoked the mail package and the DBMS, the user asks that the message be forwarded to all of them except Graham and Philhower. But who knows to what the expressions the message, them, Graham and Philhower refer? The mail system and the DBMS certainly do not know. Therefore, it must be some executive system (a KLAUS!) that supplies the missing level of awareness, interacting in English with the user and in appropriate formal languages with the mail package and the DBMS. This system must remember that a message has recently been read and that it should be salient in the user's mind. It should know that messages are forwarded to people (actually, to directories) and that therefore the them must be people. It must realize that commanding officers of CGs in the Med are people** who have recently been mentioned, and must infer that they are collectively the referent of the user's them. It must also realize that Graham and Philhower are officers.***

In short, processing the sample dialogue requires a knowledge of the domain of application, a knowledge of computer programs (including the meanings of their inputs and outputs), and a knowledge of how these pieces fit together in a larger framework.

* A meeting of many of the researchers developing software for the ACCAT was held in October 1978 to address this problem. It was concluded that a solution was clearly beyond the state of the art, and that basic research in this area was warranted.

** The DBMS does not even know this simple fact.

*** Note that KLAUS's knowledge of Graham and Philhower may have come only from its interactions with a DBMS. That is, it may be that no user ever before mentioned these particular officers to KLAUS.

2. Knowing About and Using Computer Resources

The KLAUS core will be able, by engaging in a dialogue with a user, to learn and reason about a DBMS or a simulation model in much the same way as it does about a ship or a port. But because software systems are resident in the same computer environments as KLAUS itself, rather than just "reason" and interact with a user about them as it does about other objects, the KLAUS core could be augmented with special abilities actually to use the software systems it has been told about. For example, an advanced KLAUS could not only know that the Blue File is on the Datacomputer, it could be given the ability to use the DBMS to retrieve data. In this sense, a KLAUS would have a "deeper" knowledge of computer-manipulable entities, such as the Blue File, than it has of other types of objects, such as the ship Kennedy.

Giving a KLAUS the ability not only to discuss but also to use other software is a key objective of our long-term research program. In general, we hope that KLAUS will ultimately be able to coordinate as many types of computer resources as our imagination can conceive. For example, KLAUS could be told about various special systems for sending mail, for sensing the environment, or for performing physical manipulations. Other special-purpose systems KLAUS could know about might include routines to calculate routes that avoid land masses, to retarget weapons systems, or to create and maintain situation displays in special formats.

3. Knowing About a DBMS

As a concrete representative of the whole set of possible software systems that a KLAUS might both have knowledge of and be capable of using, let us consider just one in some detail--a DBMS. Our aim in considering this example is to show how a KLAUS might be utilized to help mediate between a user's view of a domain and the view implicit in the data base's structure.

The relationship of a data base to the part of the world it is intended to describe is generally unclear to all but those who work closely with it on a technical level. For example, DARPA's Blue File data base (which is relatively clean and well-organized) concerns the characteristics of Navy and merchant ships, including information about ship positions, weapons systems, physical dimensions, the chain of command, operations, and the like. These are concepts familiar to everyone in the Navy. But the DBMS does not really know about these concepts; it knows how to manipulate tables and the codes listed in various fields. The ways in which the tables, field names, and field entries are associated with relationships and objects in the Navy world are not described anywhere in the system. Such correlations are in the minds of competent, specially trained data-base users. In particular, only these specialists know how all the following Blue File attribute names relate to real-world phenomena:

ANAME BEAM CADAT CARAT CASREP CONAM DEP DFT DISPL DOCTR
DPC DSC DST DWT EBEG EEND EIC EICNOM EMBRK ENDUR ETA
ETD ETERM FTP GWT HIT HOGEO HTP HUL IRCS LGH LINEAL
MAXRANGE MCM MCS MED NAM NAT NCARCITY NCARRCC
NCCGTYPP NCCOMM NCCQTYP NCDEPCC NCDPCITY NCDRPOS
NCDRTIME NCESDES1 NCETA NCETD NCIRCS NCM NCNAMTIT NCS
NCSOA OPCON OWN PCFUEL PTC PTD PTP PTS RANK READY REASN
SHIPCLAS TYPE UIC VCN WEPSCAP WEPSDES WEPSNOM

The specialists know that a "D" in the MED field indicates that a ship has a doctor on board, and, being humans, these specialists understand what that implies. All the DBMS knows is that there is a field called MED that takes the letter "D" as one of its values. To the DBMS, the terms MED and D are only strings of characters with no further significance. The DBMS has no knowledge about ships sailing in water, about doctors being people who sometimes travel on ships, about people being ill or injured, or about the ability of doctors to remove an appendix.

But an advanced KLAUS is intended to "comprehend" the larger enterprise in which a data base is to be used. It is supposed to be capable of learning about people and doctors and data-base management systems. It is our intention that a KLAUS be able to represent how the data base is interconnected logically with KLAUS's knowledge about ships, doctors, and other aspects of the real world that have been communicated to KLAUS by its tutors. By establishing this connection, KLAUS can relate the utility of the data base to users who know only about ships and doctors (the real application), but who do not understand the particular DBMS technology or the encoding schemes used, and who do not know all the specific facts stored in the data base.*

C. Summary

With an untutored KLAUS system, a user will be able to interactively create a knowledge base for any area of interest that is amenable to formulation in terms of first-order logic. Once created, the knowledge base could be used for retrieving facts, answering questions, making deductions, and browsing. The system becomes, in effect, a super-DBMS that interacts with its users in English.

KLAUS should ultimately not only learn about new domains and the natural-language constructions users wish to employ in discussing those domains, but it should also be able to learn about new software packages and the formal languages used to interact with them. Having learned about a domain of application and the use of various software packages, a KLAUS could act as a coordinator that organizes resources to meet the user's needs and that communicates with the user and the various software systems, in each one's own language. In such a situation, a KLAUS's knowledge base would serve as a uniform medium through which external resources could be coordinated with one another and with the user. In short, KLAUS acts like a competent technician who understands his client's problem as stated in the client's own terms, and who knows how to solve that problem by using a variety of software aids.

To provide such capabilities, a KLAUS must be fluent in both natural and formal languages, it must be able to represent both application domain and computer software concepts, and it must have

----

* Note that KLAUS's use of a knowledge base in this context fills the role of the so-called "conceptual schema" introduced in the ANSI-SPARC report [59], which described the possible new standards for data bases and data base management systems.

powerful commonsense reasoning skills. The ultimate objective of our research is the creation of this technology.

III THE RESEARCH PROGRAM

A. Program Goals

The long-range goals of the SRI KLAUS project are both technological and scientific in nature. The technological goal is to produce sophisticated computer systems that can be taught about new areas of human endeavor, learn about other computer systems that can support such endeavors, and aid humans both in organizing and communicating bodies of knowledge and in using computer resources. The project's scientific goals are manifold and are related to a number of fields, including computer science, philosophy, mathematics, linguistics, and psychology. More specifically, we seek to identify the mechanisms (both logical and computational) needed to acquire new knowledge about the concepts and language of previously unfamiliar domains. We seek to learn how new knowledge can be imparted through tutorial dialogues conducted in natural language. We seek to identify the types and structure of prerequisite knowledge needed to participate in learning dialogues and needed to participate in problem-solving dialogues in which acquired knowledge must be used in the solution of domain-related problems. We seek to explore efficient and flexible computer-based representations for such knowledge, and to develop procedural mechanisms for applying it.

Obviously, the long-range research goals are years from realization and are attainable only by a major research effort. But this effort is being staged as a sequence of modest efforts that build upon one another in an orderly fashion, providing demonstrable milestone systems at each step.

Our approach to creating a KLAUS capability involves developing and extending two fundamental technologies: knowledge representation and deduction, and natural-language processing. The representation and deduction component is at the heart of the system, serving as the depository for acquired knowledge and as the semantic base for natural-language understanding. Our approach to knowledge acquisition involves the translation of natural-language statements into the structured, formal, logical representation used by the representation/deduction component. Our approach to knowledge use involves the application of deductive processes to the knowledge encoded in formal structures.

A KLAUS must possess several "innate" abilities before a user or tutor can begin to interact with it. In particular, we have preprogrammed our pilot KLAUS, called NANOKLAUS, with a basic knowledge of English syntax, with abilities to store, manipulate, and retrieve logical expressions, and with essential skills for learning the vocabulary and concepts of new subject domains. The untutored NANOKLAUS is also endowed with a seed vocabulary and knowledge of key concepts needed for acquiring more knowledge. For example, NANOKLAUS is preprogrammed with such concepts as WORD and PHYSICAL OBJECT. Ultimately, an untutored KLAUS should include at least the following subsystems:

* A sophisticated natural-language understanding system including
    * A general, linguistically motivated syntax of the English language.
    * A vocabulary of the function words of English (including articles, quantifiers, prepositions, pronouns, numbers, conjunctions, copulas, and the like) and a skeletal set of words needed for the definition of new words and concepts (including words such as NOUN, VERB, OBJECT, RELATION, FUNCTION, SET, PROGRAM, TIME, SPACE, and so forth).
    * A general system for testing the semantic feasibility of inputs and for constructing internal descriptions of the meanings of those inputs that pass the feasibility tests.
    * Routines for determining the referents of phrases in context.

* A sophisticated deduction system including

    * Algorithms and data structures that are not less than first-order complete.

    * Indexing mechanisms on the data structures for efficient retrieval.

    * Mechanisms that allow the deductive process to appeal to "specialist" procedures for the efficient answering of certain types of questions or for access to computer software external to KLAUS.

    * Special mechanisms for reasoning about processes, including a planning system.

    * Special mechanisms for reasoning about knowledge and belief.

    * Facilities for storing and managing large external files.

* A fuzzy matcher for browsing through data in search of items loosely related to a given description.

* A learning module that aids in the acquisition of new vocabulary, concepts, and procedures. This module should include

    * Interactive procedures for the linguistically motivated classification of new terms.

    * Procedures for building new vocabulary entries.

    * Procedures for adding new classes of objects, particular individuals, classes of relationships, and particular assertions to the knowledge base.

    * Procedures for discovering new terms or new uses of old terms in the input from users.

    * Generalization procedures.

    * Gap detection procedures.

    * Procedures for detection and resolution of inconsistencies.

    * Analogy/metaphor-understanding procedures.

* A natural-language generation system, including

    * A well-formed sentence generator.

    * A context-sensitive noun phrase generator.

    * A system for planning presentation strategies for the output of data requiring multiple sentences.

* An information formatter for displaying data in an easy-to-read form, including

    * Table generators.

    * Graphics.

* A self-explanation system for explaining how KLAUS arrives at its answers to questions.

IV RELATED WORK IN KNOWLEDGE ACQUISITION

The terms "knowledge acquisition" and "learning" have been used to describe a wide range of research problems in AI that differ both in the kinds of knowledge being acquired and in the means of acquisition. KLAUS is distinct from all of these in addressing the problem of how a computer system can learn about a totally _new_ area by carrying on a _dialogue in natural language_ with a user. Research in the following six subareas of "knowledge acquisition" is related to KLAUS:

(1) Representation of knowledge

(2) Expansion of the knowledge base of rule-based expert systems

(3) Discovery programs

(4) Concept formation

(5) Analogical reasoning

(6) Menu/frame traversal systems.

The following sections briefly characterize work in each of these areas, distinguish such work from the research reported here, and discuss those aspects of such work that are likely to contribute to our current effort or future efforts at increasing the power of KLAUS systems.

1. _Representation of Knowledge_

Research in this area (e.g., Brachman [5]; Fox [12]; Hendrix [16]; Kowalski [30]; Mark [35]; Moore [39], [40]; Nevins [43]; Sacerdoti [53]; and Smith [58]) is concerned with fundamental issues about the basic structures needed to represent knowledge. Because the question of what structures are needed depends to a large extent on what those structures will be used for, research in this area is closely tied to research on deductive reasoning. Although the questions of inference and deduction lie behind many of the design choices in work on representation, much of the implementation effort to date has been concentrated on systems for building different kinds of representational structures, with less effort devoted to developing inference engines that reason with such structures. Kowalski [30], Nevins [43], and Fikes and Hendrix [11] are notable exceptions. More importantly from the perspective of KLAUS research, although the designers of different representation schemes have often developed formal languages for building knowledge bases in their particular schemes (e.g., Bobrow and Winograd [3] and Hendrix [16] are good examples), none has considered the problem of adding to such knowledge bases by describing new information in English.

In building KLAUS we are building on a variant of an already well-developed representation scheme, the partitioned semantic networks of Hendrix [16], in which the basic symbols of logic are organized in graphical form. The representation has the full expressive power of first-order logic and, in addition, explicitly displays much of the important indexing and type-hierarchy information that is needed for efficient computer implementations. We make use of techniques that have been developed in work on partitioned networks for aggregating sentences, indexing predicates and terms, and representing type hierarchies. Moreover, this basic structure has been augmented by adding to it the knowledge and mechanisms for controlling inference discussed in Moore [39].

However, there is still much to be learned about basic structures of knowledge (see Smith [58]) and we expect to follow developments in research on representation of knowledge closely. We will augment and modify the internal structures that KLAUS builds (as well as the kinds of questions it asks the user) when such research indicates new kinds of structures are needed. Conversely, we expect the research on KLAUS to benefit research on basic representational issues because the acquisition of new information should be a key use of these structures.

2. Rule Learning and Modification

Much recent work on knowledge acquisition has been done in the context of rule-based expert systems (e.g., MYCIN [55], PROSPECTOR [9], HEARTS [42]), systems that encode the judgmental knowledge of experts and apply it to classification problems in a specific domain (infectious diseases for MYCIN, geological ore deposits for PROSPECTOR, card games for HEARTS). The central knowledge acquisition problem faced by builders of such systems is how to transfer the judgmental knowledge of a human expert to a computer system. Research in this area (e.g., Davis [7]; Mitchell [37]) has been concerned with how to encode and reason with meta-level knowledge, the knowledge a system has about what it knows and how it can use that knowledge.

Research on knowledge acquisition in rule-based systems and the research program described herein differ primarily in the kind of system to which knowledge is being added. KLAUS is intended to aid in the construction of knowledge bases for information management. The research emphasizes learning the logical structure of new subject areas rather than learning new rules, heuristics, and strategies for performing a particular classification task. Basically, the job of a judgmental reasoning system is to characterize a given situation as being in one of an explicitly specified set of situation classes. Information about the class indicates what properties are common to or what actions are appropriate in situations in the class. For example, given a geological area (a situation), PROSPECTOR seeks to determine the class of the area, which in turn indicates what formations and minerals are likely to be present. Acquisition systems for rule-based "experts" need to learn the names of the situation classes and a set of rules that indicate how various bits of information about a situation may support or refute the possibility of the situation being in a particular class. This interesting and important area of research is orthogonal to KLAUS.

3. Concept Formation

Another major area of research on knowledge acquisition is that concerned with learning from examples and learning by analogy. Much of this research has evolved from early work on learning systems and pattern recognition, but the techniques have progressed substantially from such work. In essence, the problem is to construct a characteristic description of some set of entities (e.g., the set of all arches) that distinguishes that set from all others the systems knows about. Some of the work on extending rule-based systems can be seen as a specialization of concept formation research (e.g., Mitchell [37]) in which pattern-action rules, rather than descriptions, are being learned. Work in this area (e.g., Winston [65], Colman [6], Dietterich and Michalski [8], Tanimoto [61], Fox and Reddy [13], Hayes-Roth and McDermott [27]) has been concerned with such problems as determining relevant descriptor sets, combining and modifying particular descriptions as new examples are presented, and determining what features are critical in distinguishing different sets.

A major difference between KLAUS and research on concept formation is that KLAUS is particularly concerned with how to talk about the new concepts it learns. In particular, KLAUS seeks to learn the language associated with a concept at the very time it learns the concept.

KLAUS is, of course, concerned with concept formulation even though it learns by being told, rather than by being presented with examples. Specifically, KLAUS must know whether it has enough information to determine how a new concept is like or unlike other concepts it knows about. Some of the strategies developed for concept formation programs will be useful in constructing those parts of KLAUS systems that integrate new concepts into the existing knowledge base.

4. Analogy

One of the most powerful techniques for learning new information is the use of analogy. This can be considered a special kind of learning by example in which a critical problem is deciding which aspects of the old concept are important to the new concept. Analogical reasoning has been used in systems that learn about processes as well as systems that learn about objects by analogy (e.g., McDermott [36], Moll and Ullrich [38], and Goldstein and Grimson [20]). Learning by analogy is a difficult problem and one our initial KLAUS efforts will not address. However, because analogy is such a powerful tool for learning (whether the learning is by being told in language or by being shown some other way), we anticipate profiting from analogy research in the development of more advanced KLAUS systems.

5. Discovery Programs

Some recently developed programs (e.g., Lenat [32]; Langley [31]) have been able to "discover" new rules about their domain of expertise. In essence, they start with a body of rules and data and "learn" new rules through internal manipulation of these rules and data. This kind of learning is quite sophisticated and of a very different sort from that of the KLAUS effort. In particular, the domain of expertise remains constant for discovery systems and, because the learning is all internal, no communication with another agent is involved.

6. Menu/Frame Traversal Systems

For some applications, menu selection on terminals with display capabilities provides an alternative to natural language as a means of communicating naturally and effectively with a user. ZOG [47] is a system that (among other things) allows a user to create a menu/frame selection system for a new domain. ZOG differs from KLAUS not only in the different means of communication, but also in how much it knows about the information in the domain. In many ways ZOG is like a DBMS: when it is used in a new domain, it doesn't really learn anything about the concepts of that domain, but rather it acquires new complex, interconnected structures through which the competent user can navigate in search of information. All structuring is done by the user building the system and all interpreting of the structures is done by the user doing the navigation. In other words, ZOG allows a user to create information structures so that other people can access that information easily and naturally. Although ZOG can manipulate the structures it is given, and even invoke procedures, it doesn't know about what the structures mean in the sense of being able to reason with them. In contrast, when KLAUS learns about a new domain, it itself integrates the basic concepts in this domain into its knowledge structures and hence is able not only to retrieve information, but also to reason about it. Furthermore, KLAUS automatically integrates this new knowledge with previous knowledge.

V SUMMARY

The goals of the KLAUS project are to study knowledge acquisition and to develop the scientific base needed to provide a sophisticated software executive that combines a knowledge of the English language with a command of facts about new domains and about how to coordinate the use of various complex software packages in the solution of user problems.

Basic KLAUS systems are intended to allow a user to interactively create a knowledge base of any area of interest that is amenable to formulation in terms of first-order logic. Once created, the knowledge base can be used for fact retrieval, question answering, deduction, and browsing. The system becomes a type of extended DBMS that interacts with its users in English.

We have implemented and tested a pilot KLAUS system called NANOKLAUS [25] that incorporates the features of the basic KLAUS concept.

Advanced KLAUS systems should not only learn about new domains and the natural-language constructions users wish to employ in discussing those domains, but it should also be able to learn about new software packages and the formal languages used to interact with them. Having learned about a domain of application and the use of various software packages, a KLAUS could act as a coordinator that organizes resources to meet the user's needs and that communicates with the user and the various software systems, each in the appropriate language. In such a situation, a KLAUS's knowledge base would serve as a uniform medium through which external resources could be coordinated with one another and with the user. To provide these abilities, a KLAUS must be fluent in both natural and formal languages; it must be able to represent both application-domain and computer-software concepts; and it must have powerful commonsense reasoning skills.

REFERENCES

1. "Knowledge Base Management System," Proposal for Research submitted to DARPA, Computer Science Department, Stanford University, in cooperation with SRI International (January 1978), also SRI International Proposal for Research No. ISD 78-20 (March 1978).

2. D. Bobrow, and B. Raphael, "New Programming Languages for Artificial Intelligence Research," ACM Computing Surveys, Vol. 6, pp. 153-174 (1974).

3. D. Bobrow, and T. Winograd, "An Overview of KRL, A Knowledge Representation Language," Cognitive Science, Vol. 1, No. 1 (January 1977).

4. R. S. Boyer and J. S. Moore, "Proving Theorems about LISP Functions," Journal of the Association for Computing Mach., 22, pp. 129-144 (January 1975).

5. R. J. Brachman, "A Structural Paradigm for Representing Knowledge," Technical Report No. 3605, Bolt Beranek and Newman, Inc., Cambridge, Massachusetts (1978).

6. R. W. Colman, "Manipulation Extrapolation, a System for Controlling Trainable Robots," Proc. 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

7. R. Davis, "Interactive Transfer of Expertise: Acquisition of New Inference Rules," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts (August 1977).

8. T. G. Dietterich and R. S. Michalski, "Learning and Generalization of Characteristic Descriptions: Evaluation Criteria and Comparative Review of Selected Methods," Proc. 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

9. R. O. Duda, P. E. Hart, P. Barrett, J. G. Gaschnig, K. Konolige, R. Reboh, and J. Slocum, "Development of the Prospector Consultation System for Mineral Exploration," Final Report, SRI Project 5821 and 6415, SRI International, Menlo Park, California (October 1978).

10. B. Elspas, K. N. Levitt, and R. J. Waldinger, "An Interactive System for the Verification of Computer Programs," Technical Report, Stanford Research Institute, Menlo Park, California (September 1973).

11. R. E. Fikes, and G. G. Hendrix, "A Network-Based Knowledge Representation and Its Natural Deduction System," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, pp 235-246 (1977).

12. M. S. Fox, "On Inheritance in Knowledge Representation," Proc. 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

13. M. S. Fox and D. R. Reddy, "Knowledge-Guided Learning of Structural Descriptions," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts (August 1977).

14. G. G. Hendrix, "The LIFER Manual: A Guide to Building Practical Natural Language Interfaces," SRI Artificial Intelligence Center Technical Note 138, SRI International, Menlo Park, California (February 1977).

15. G. G. Hendrix, "Human Engineering for Applied Natural Language Processing," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachussetts (August 1977).

16. G. G. Hendrix, "Encoding Knowledge in Partitioned Networks," in Associative Networks - The Representation and Use of Knowledge in Computers, N. V. Findler, ed. (Academic Press, New York, New York 1979).

17. G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum, "Developing a Natural Language Interface to Complex Data," ACM Transactions on Database Systems, Vol. 3, No. 2 (June 1978).

18. G. G. Hendrix, D. Sagalowicz, E. D. Sacerdoti, "Mechanical Intelligence: Research and Application," SRI International Proposal for Research No. ECU 78-9 (June 1978).

19. G. G. Hendrix, D. Sagalowicz, and E. D. Sacerdoti, "Research on Transportable English Access Mediums to Distributed and Local Data Bases," SRI International Proposal for Research No. ECU 79-103 (September 1979).

20. I. P. Goldstein and E. Grimson, Annotated Production System, a Model for Skilled Acquisition," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts (August 1977).

21. C.Green, The Application of Theorem-Proving to Question-Answering Systems, Stanford Artificial Intelligence Project Memo AI-96, Stanford, California (June 1969).

22. B. J. Grosz, "The Representation and Use of Focus in a System for Understanding Dialogs," Proc. 5th International Joint Conference on

Artificial Intelligence, Cambridge, Massachusetts, pp. 67-76 (August 1977).

23. B. J. Grosz, "Focusing and Description in Natural Language Dialogues," To Appear in: A. K. Joshi, I. Sag, and B. Webber, eds., *Proc. of Workshop on Computational Aspects of Linguistic Structure and Discourse Setting*, Cambridge University Press, Cambridge, Great Britain (1981).

24. B. J. Grosz, "Utterance and Objective: Issues in Natural Language Communication," *Proc. 6th International Joint Conference on Artificial Intelligence*, Tokyo, Japan (August 1979).

25. N. Haas and G. Hendrix, "An Approach to Acquiring and Applying Knowledge," *Proc. 1st Annual National Conference on Artificial Intelligence*, Stanford, California (August 1980).

26. Hart, P. E., "Progress on a Computer-Based Consultant," *Proc. 3rd International Joint Conference on Artificial Intelligence*, Tbilisi, USSR (August 1975).

27. F. Hayes-Roth and J. McDermott, "Knowledge Acquisition from Structural Descriptions," *Proc. 6th International Joint Conference on Artificial Intelligence*, Tokyo, Japan (August 1979).

28. J. Hobbs, "Coherence and Coreference," *Cognitive Science*, Vol. 3, No. 1, pp. 67-90 (1979).

29. K. G. Konolige, "Toward a General Natural-Language Interface," *Proc. of the ACCAT Principal Investigators Meeting*, Monterey, California (October 1979).

30. R. Kowalski, "Predicate Logic as a Programming Language," *Information Processing*, Amsterdam, North-Holland (1974).

31. P. Langley, "Rediscovering Physics with BACON.3," *Proc. 6th International Joint Conference on Artificial Intelligence*, Tokyo, Japan (August 1979).

32. D. B. Lenat, "AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search," Stanford University Computer Science Dept. Report STAN-CS-76-570 (July 1976).

33. W. H. Lewis, "TED: A Transportable English Datamanager," *Proc. of the Principal Investigators' Meeting of the ACCAT Program*, Monterey, California (October 1979).

34. Z. Manna, and R. Waldinger, *A Deductive Approach to Program Synthesis*, SRI Artificial Intelligence Center Technical Note 177, Menlo Park, California (December 1978).

35. W. S. Mark, "The Reformulation Approach to Building Expert Systems," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts (August 1977).

36. J. McDermott, "Learning to Use Analogies," Proc. 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

37. T. M. Mitchell, "Version Spaces: a Candidate Elimination Approach to Rule Learning," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts (August 1977).

38. R. Moll and J. W. Ulrich, "The Synthesis of Programs by Analogy," Proc. 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

39. R. C. Moore, "Reasoning from Incomplete Knowledge in a Procedural Deduction System," MIT Artificial Intelligence Laboratory, Cambridge, Massachusetts AI-TR-347 (1975).

40. R. C. Moore, "Reasoning about Knowledge and Actions," PhD dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts (1979).

41. R. C. Moore, "Handling Complex Queries in a Distributed Data Base," Artificial Intelligence Center Technical Note 170, SRI International, Menlo Park, California (September 1979).

42. J. Mostow and F. Hayes-Roth, "Operationalizing Heuristics: Some AI Methods for Assisting AI Programming," Proc. 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

43. A. Nevins, "A Human-Oriented Logic for Automatic Theorem-Proving," Journal of the Association for Computing Mach., 21 pp. 606-621 (1974).

44. Nilsson, N. J., ed., "Artificial Intelligence - Research and Applications," Progress Report to DARPA, Stanford Research Institute, Menlo Park, California (May 1975).

45. N. J. Nilsson, A Production System for Automatic Deduction, Artificial Intelligence Center Technical Note 148, SRI International, Menlo Park, California (July 1977).

46. W. H. Paxton, "A Framework for Speech Understanding," PhD dissertation, Stanford University, Stanford, California (1977).

47. Robertson, G. et al., "ZOG: A Man-machine Communication Philosophy," Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania (August 1977).

48. A. Robinson, "Investigating the Process of Natural-Language Communication: A Status Report," AI Center Technical Note 165, SRI International, Menlo Park, California (July 1978).

49. A. E. Robinson, "Natural-Language Communication," SISTM Quarterly, Vol. 2, No. 4, pp. 20-28 (1979).

50. J. J. Robinson, "DIAGRAM: an Extendable Grammar for Natural Language Dialogue," To be published.

51. J. F.Rulifson, "QA4: A Procedural Calculus for Intuitive Reasoning," PhD dissertation, Stanford University, Stanford, California. Technical Note 73. Stanford Research Institute, Menlo Park, California (1972).

52. E. D. Sacerdoti, R. E. Fikes, R. Reboh, D. Sagalowicz, R. J. Waldinger, B. M. Wilber, "QLISP - A Language for the Interactive Development of Complex Systems," Proceedings of AFIPS National Computer Conference, pp. 349-356 (1976).

53. E. D. Sacerdoti, A Structure for Plans and Behavior (Elsevier-North Holland Publishing Company, New York, New York 1977).

54. E. D. Sacerdoti, "Language Access to Distributed Data with Error Recovery," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts (August 1977).

55. E. H. Shortliffe, Computer-Based Medical Consultations: MYCIN (Elsevier-North Holland Publishing Company, New York, New York 1976).

56. R. F. Simmons, and J. Slocum "Generating English Discourse from Semantic Nets," Communications of the ACM, Vol. 15, No. 10 (1972).

57. J. Slocum, "Generating Verbal Responses," in Understanding Spoken Language, D. E. Walker, ed. (Elsevier-North Holland Publishing Company, New York, New York 1978).

58. B. C. Smith, "Levels, Layers, and Planes: The Framework of a System of Knowledge Representation Semantics," MS thesis, Massuchesetts Institute of Technology, Cambridge, Massachusetts (February 1978).

59. T. B. Steel Jr., "ANSI/X3/SPARC Study Group on Data Base Management Systems, Interim Report 75-02-08," FDT (Pub. ACM-SIGMOD), Vol. 17, No. 2 (1975).

60. Stefik, M. "An Examination of a Frame-Structured Representation System," Stanford Heuristic Programming Project Memo HPP-78-13, Stanford, California (September 1978).

61. S. L. Tanimoto, "Inductive Learning of Categories from Examples Using Minimum Cost Representation," Proc. 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

62. W. Teitelman, "INTERLISP Reference Manual," Xerox Palo Alto Research Center, Palo Alto, California (December 1975).

63. D. E. Walker, Understanding Spoken Language (Elsevier-North Holland Publishing Company, New York, New York, 1978).

64. D. E. Walker, B. J. Grosz, G. G. Hendrix, W. H. Paxton, A. E. Robinson, and J. Slocum, "An Overview of Speech Understanding Research at SRI," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts (August 1977).

65. P. H. Winston, "Learning Structural Descriptions from Examples," Chapter 5 in P. H. Winston (Ed.), The Psychology of Computer Vision (McGraw-Hill Book Company, New York, New York, 1975).