REPRESENTATIONS OF KNOWLEDGE IN A PROGRAM FOR SOLVING PHYSICS PROBLEMS

Gordon S. Novak Jr. *
Computer Science Department
The University of Texas at Austin
Austin, Texas 78712

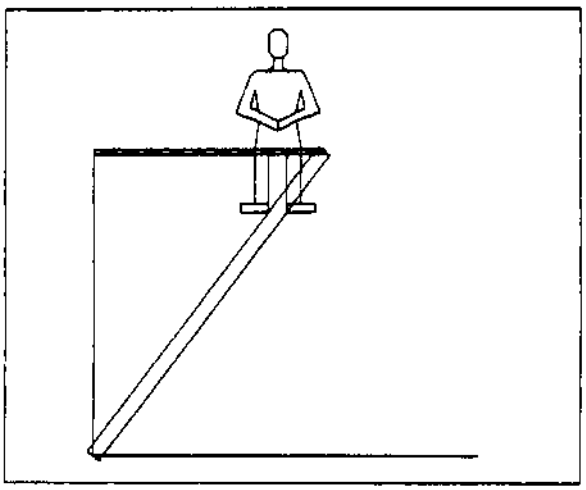
ABSTRACT

A computer program which solves physics problems stated in English is described in terms of the knowledge which is used to transform one type of representation into another. The English sentences of the problem statement are progressively transformed into a semantic network form, a language-free internal model of the objects in the problem and their attributes and relationships, a set of canonical object frames which interpret actual objects as canonical objects (such as a point mass), a geometric model, a set of equations, and a picture model. The general notion of a canonical object frame, which abstracts a subset of the properties of an object to form a representation of a canonical object whose interactions with related canonical objects can be formally modelled, is discussed as a method of organizing problem-solving programs.

1. INTRODUCTION

This paper describes the representations of knowledge used by a program which solves physics problems stated in English. Rather than using a single uniform representation, the program uses a number of different representations, each of which is specialized for a particular task, e.g., language syntax, language semantics, representing objects and their attributes and relationships, representing objects as canonical objects used in physics, modelling geometry, and solving equations. Many of these representations are based on the notion of frames [Minsky 75]. The use of specialized representations simplifies many of the processes which must be performed by the program; however, it requires that the program be able to translate between the various representations when necessary. Procedural knowledge is required to convert one representation into another, since it frequently happens that information which is essential in the target representation is unspecified or is specified only implicitly in the source representation; inferences are required to fill in such information. Specialized representations allow procedures to be attached to particular types of representations, both to convert them to other types and to solve problems which are associated with the specialized area. In this paper, we discuss the ways in which these techniques are used to coordinate the many kinds of knowledge which are necessary for solving physics problems.

* Present Address: Artificial Intelligence Lab
Stanford Research Institute
Menlo Park, CA 94025

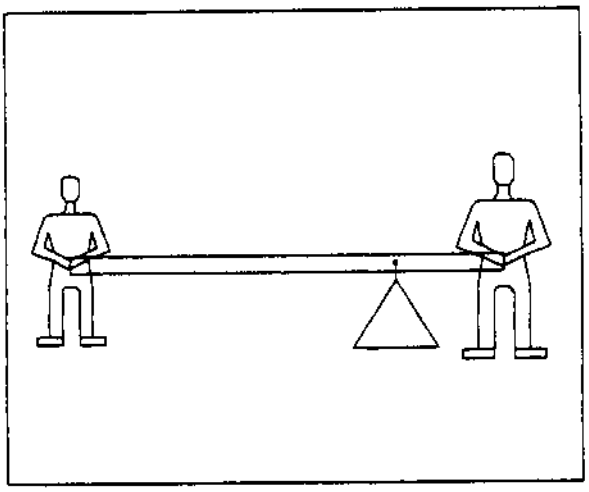


P8 SCHAUM PAGE 25 NUMBER 19

(THE FOOT OF A LADDER RESTS AGAINST A VERTICAL WALL AND ON A HORIZONTAL FLOOR)(THE TOP OF THE LADDER IS SUPPORTED FROM THE WALL BY A HORIZONTAL ROPE 30 FT LONG)(THE LADDER IS 50 FT LONG , WEIGHS 100 LB WITH ITS CENTER OF GRAVITY 20 FT FROM THE FOOT , AND A 150 LB MAN IS 10 FT FROM THE TOP)(DETERMINE THE TENSION IN THE ROPE)

ANSWER: 120.00000 LB

Figure 1.: An Example Problem



P2 SCHAUM PAGE 12 NUMBER 4

(WHERE MUST A WEIGHT BE HUNG ON A POLE , OF NEGLIGIBLE WEIGHT , SO THAT THE BOY AT ONE END SUPPORTS 1/3 AS MUCH AS THE MAN AT THE OTHER END)

ANSWER: (TIMES LENGTH76 7.50000E-1) FROM THE BOY , WHERE LENGTH76 IS THE LENGTH OF THE POLE

Figure 2.

2. OVERVIEW OF THE PROGRAM

The program, which we call ISAAC for ease of reference, is able to read, understand, solve and

draw pictures of physics problems stated in English. Representative examples of the class of problems solved by the program are shown in Figures 1 and 2. The program has solved a total of twenty problems in the area of rigid body statics, most of which were taken verbatim from high school and college physics texts.

The understanding and solution of such a problem proceeds in several distinct steps. The first step is, of course, the understanding of the English sentences of the problem statement. The sentences are parsed into a case-structured semantic network form by an augmented transition net grammar, which is aided by a large number of semantic programs. Semantic processing Is interwoven with the parsing process. Structures produced by the parser, which initially correspond closely to syntactic forms, are made progressively more semantic as the parsing proceeds. One major semantic process is referent identification, in which a phrase is identified with the object or relationship to which it corresponds in the program's developing model of the problem. A second major semantic process is the identification of a type of conceptual entity (such as a location or attribute) which is referenced by a phrase. Procedures associated with each type of semantic frame allow inferences to be made to fill "slots" in the semantic frame whose values are unspecified; for example, the phrase "at one end" impll-rcitly references an unspecified physical object, whose Identity must be inferred to complete the meaning of the semantic frame. The final step in the semantic processing of a sentence is the execution of the verb semantics. At the time when the semantic routine for a verb is executed, the case arguments of the verb will typically be represented as semantic frames (as opposed to syntactic-phrases); this representation greatly simplifies the processing of verbs. Execution of verb semantics typically causes the transfer of new information to existing objects in the internal model of the problem, or causes the creation in the model of new objects and relationships.

After all of the sentences of the problem statement have been read and processed semantically, the information conveyed by the sentences will have been transferred to a language-free internal model, and the structures produced in parsing the sentences will have been discarded. The internal model contains representations of physical objects (e.g., ladders, ropes, and persons), features or aspects of objects (e.g., locations), and relationships among objects (e.g., attachment relations),

In order to solve the physics problem, it is necessary to associate a Canonical Object Frame with each physical object in the problem, A Canonical Object Frame is an idealization or abstraction of certain features of an actual object, such as a rigid body or a point mass. Physical laws are defined in terms of these idealized canonical objects, which only approximate the behavior of real objects. The canonical object frame to be used for an object depends on the object's context; thus, a person might be modelled as a pivot when carrying a plank, or as a point mass when sitting on a plank. There are procedures associated with each type of canon*-ical object which test for the presence of necessary features and make appropriate inferences for features which are missing. For example, if the weight of a rigid body is unknown, it is assumed to be zero; if the length is unknown, the assumption is made that the object must have a finite length, and a symbolic constant is created to represent the length.

A geometric model is created to relate the positions of the objects in the problem to a common coordinate system. Knowledge of the prototypical geometry of each object is used to create a composite geometric model by combining the geometric models of individual objects at points of attachment. In some cases, this requires mapping features of objects (such as the distance between two points) into a formal system (such as a triangle), solving a problem in the formal system, and mapping the results back to the original object (for example, defining the size of an object based on the computed distance between two points on the object).

After canonical object frames have been selected for each object and the geometric model has been constructed, equations are written to describe the interactions of the objects according to physical laws. These equations are solved by a small symbolic manipulation package, and an answer to the question asked in the problem statement is generated from the equation solutions.

The last step performed by the program is the generation of a picture model and a diagram of the problem. The picture model is similar to the geometric model; however, it is necessary to determine reasonable sizes in the drawing for objects which have no geometric size (for example, a person who is modelled as a point mass), and to infer reasonable points of attachment when they are unspecified (for example, the point of attachment for a person whose canonical object frame is PIVOT is assumed to be HANDS). Finally, the composite drawing must be scaled to fit the available picture area.

3. UNDERSTANDING ENGLISH SENTENCES

Parsing of the English sentences of the problem statement is controlled by an Augmented Transition Network Grammar [Woods 70]. Rather than using a grammar interpreter, the grammar is implemented directly as a set of LISP functions, using a set of small system functions to control movement of the scanner and to provide automatic backup when an attempted parsing fails. The grammar functions in the parser build structures whiqh represent the content of the phrase being parsed, or make changes to already existing structures. In addition, the parsing functions frequently make calls to semantic routines; a grammar function may fail on semantic grounds even though it succeeded in parsing tne desired syntactic construct.

The structures produced by the parser initially bear a strong resemblance to Semantic Networks as used by Simmons [Simmons 73], These structures are not produced for all syntactic constructs, but only for those which represent larger units of meaning (primarily noun phrases and verb phrases). As the parsing continues, the parse structures are made

progressively more semantic and less syntactic, until finally the meaning of a phrase is extracted and incorporated into the internal model or into the semantic structure of another phrase; thereafter, the parse structure is no longer used (with the exception of possible later use for finding pronoun referents). Rather than being representations of the meanings of phrases, these parse structures may be considered to be temporary structures where the information provided by the syntactic parsing may be stored in a readily accessible form until sufficient context has been collected to allow the semantics of the phrase to be determined The form and use of the parse structures are shown later in this section, where the complete processing for an example sentence is outlined.

As the processing of a noun phrase proceeds, the phrase will generally be identified as an instance of a Semantic Frame. The semantic frame designation specifies the type of conceptual entity denoted by the phrase, e.g., PHYSENT (physical entity) ATTROF (attribute-of), or LOCPART (location/part). In addition, the semantic frame contains "slots" for the "arguments" of the frame; for example, the Attribute-Of semantic frame contains slots for the name of the attribute (e.g., LENGTH) and the referent object (that is, the object in the internal model of the problem) with which the attribute is associated. If the values of essential slots in the semantic frame are not specified explicitly in a sentence, procedures associated with the semantic frame can examine the internal model of the problem and make the inferences necessary to fill in the missing values. Such inferences are frequently required when, for example, a location is named without the object with which it is associated, as in "the center" or "one end". Thus, the procedures associated with a semantic frame perform the dual functions of determining when a conceptual entity is incompletely specified and providing specialist programs to complete the specifications by making appropriate inferences.

The use of semantic frames greatly simplifies the processing required for "higher-level" semantic routines which reference multiple phrases, such as the semantic routines for verbs and prepositions. A single type of conceptual entity, such as a location, may .be denoted by a variety of syntactic forms; by identifying the type of conceptual entity and collecting the arguments of the semantic frame into a standard form (making inferences as required), all of these syntactic forms are converted to an identical semantic frame form. Thus, the higher-level semantic routines may directly access the attributes of the conceptual entity without being concerned with the syntactic form which was used to specify that entity.

Phrases in a sentence frequently refer, either explicitly or implicitly, to objects in the program's developing model of the problem. Referent Identification is the process of determining the object(s) in the internal model to which a phrase refers. The referenced "object" may be a physical object, an attribute of a physical object (such as a location), or a relationship (such as an attachment between two physical objects); there is a set of specialist routines for identifying referents of each type. Referent identification is in general a very difficult problem [Chamiak 72]; even in the microworld of physics problems, the processes for identifying referents are fairly complex. First, it is necessary to determine whether a phrase refers to an existing object in the model, or whether a new object must be created and added to the model. If the model contains more than one possible referent, it is necessary to select the correct one(s); this selection may be specified in the sentence by means of an attribute value ("the left end" ... "the other end"), or an intenslonal description ("the load", i.e., the object which is being carried). When the referent identification process is complete, the "referent" which is returned is a list of the objects in the internal model to which the phrase refers. The various semantic frame forms have slots for the referent of the semantic frame and/or the object the semantic frame is about, e.g., the object whose attribute is referenced. In many cases, the referent constitutes the entire meaning of the phrase; thus, a phrase like "one end of a pole" has a meaning in isolation (even though it is not an acceptable sentence), namely, the pointer to the location object in the internal model to which the phrase refers. Once again, the use of semantic frames and the referent identification process relieve higjier-level semantic, routines of the burden of knowing how a referent was specified; these routines need only deal with the pointer to the referent object in the internal model.

The structures produced by the parser and the structures created in the internal model in response to an example sentence are shown in Figures 3 and A, respectively. Each word which is underlined represents a LISP GENSYM atom; the columns below the atom names are the indicator/value pairs from the atom's property list. The numbered statements below describe the sequence of events in processing the example sentence; the numbers shown in the Figures correspond to this sequence, and indicate the position of the scanner as the sentence is parsed, or the time at which atoms are created or new information is added to their property lists.

1. The initial noun phrase "one end" is parsed, producing the structure T0K1 with its first four properties.

2,3 The prepositional phrase parser causes the noun phrase "a pole" to be parsed, producing the structure T0K2 with its first four properties.

4. The phrase "10 ft long" causes a modifier to be added to T0K2.

5. The preposition semantics routine for OF is called with T0K1 and T0K2 as arguments. By means of a discrimination net using tests on these arguments, this particular instance of the use of OF is classified as being of the form "<location> OF <object>".

6. The function IDRFNT is called to identify the referent of "a pole". This causes T0K2 to be assigned the Semantic Frame PHYSENT (Physical Entity), and causes the object P0LE3 to be added to the internal model. P0LE3 is added to T0K2 as its referent.

```
"One end of a pole 10 ft long is supported by a man."
  +     +  +     +          +          +      +
  1     2  3     4          8          9      11

 1  TOK1                        3  TOK2
    TOK     END                    TOK     POLE
    LFRAME  NP                     LFRAME  NP
    NBR     (NS)                   DET     INDEF
    DET2    ONE                    NBR     (NS)
 7  SFRAME  LOCPART             4  MODS    ((LENGTH . (10 FT)))
    SEMOBJ  (POLE3)             6  SFRAME  PHYSENT
12  RFNT    (LOC7)                 RFNT    (POLE3)

 8  TOK4                        9  TOK5
    TOK     SUPPORT                TOK     PERSON
    LFRAME  VP                     LFRAME  NP
    MAINVB  SUPPORTED              WORD    MAN
    AUX     (IS)                   MODS    ((RESTRICT (SEX MALE))
    TRANS   *T*                             (RESTRICT (AGE ADULT)))
    PASV    *T*                    DET     INDEF
    OBJ     TOK1                   NBR     (NS)
10  SUBJ    TOK5               10  SFRAME  PHYSENT
                                   RFNT    (PERSON6)
```

Figure 3.: Structures Produced in Parsing a Sentence

```
 6  POLE3                       10  PERSON6
    TOK         POLE                TOK         PERSON
    ENTITY      PHYSENT             WORD        MAN
    LENGTH      (10 FT)             ENTITY      PHYSENT
12  LOCS        (LOC7)              RESTRICT    ((SEX MALE)
13  ATTACH      (ATTACH8)                        (AGE ADULT))
    SUPPORTBY   (PERSON6)           ATTACH      (ATTACH8)
                                    SUPPORT     (POLE3)

12  LOC7                        13  ATTACH8
    FRAME       LOCATION            FRAME       ATTACH
    ENTITY      LOCATION            TYPEATT     PINJOINT
    OBJECT      POLE3               LOCS        ((PERSON6 NIL)
    LOCNAME     END                              (POLE3 LOC7))
    SELECT      (ONE)
```

Figure 4.: Internal Model Produced in Response to Sentence

7. The semantic routine for OF next assigns the Semantic Frame LOCPART (Location/Part) to T0K1 and fills the Semantic Object slot of this frame with the referent of the phrase which was the object of the preposition, i.e., the list (P0LE3). Note that since the meaning of T0K2 has been extracted and used, T0K2 is no longer a part of the parse structure.

8. The verb phrase parser is called (with T0K1 as the syntactic subject argument). Since the verb phrase is passive, T0K1 is attached to the verb token structure T0K4 as the OBJ case argument.

9. The prepositional phrase parser causes the noun phrase "a man" to be parsed, generating the structure T0K5. "Man" is defined in terms of an underlying concept ("person") with modifiers.

10. The preposition semantics routine for BY calls IDRFNT to identify the referent of a "man"; this causes T0K5 to be assigned the Semantic Frame PHYSENT, and causes a new object PERS0N6 to be added to the internal model as its referent.

11. The verb semantics for the verb SUPPORT are called; this routine sees a structure of the form <physent> SUPPORT <locpart>.

12. The location object L0C7 is added to the internal model and specified as the referent of T0K1.

13. The semantic routine for the verb SUPPORT calls the function IDATT to identify an attachment relation between P0LE3 at location L0C7 and PERS0N6 (location unknown); this causes the creation of the attachment relation ATTACH8.

4. CANONICAL OBJECT FRAMES

A Canonical Object Frame represents an idealization of an actual object as a canonical object whose behavior approximates the behavior of the idealized aspects of the actual object. For example, in a physics problem, a man standing on a plank might be modelled as a Point Mass canonical object. A canonical object frame does not represent a superset of the actual object (typically represented by an ISA link in semantic networks), but rather represents a "view" of the object in the sense of [Bobrow and Winograd 77], In a large A.I. system, a complex object such as a person might be modelled by a variety of canonical object frames, each of which would represent some aspect of the person; in our microworld of physics problems, each object is modelled (within a single problem) as a single canonical object. The types of canonical object frames which may be used to represent an object depend upon its context; a person would be modelled as a Pivot object when carrying a plank rather than standing on It,

In order to model an object using a canonical object frame, it is necessary to select the appropriate frame, and then to abstract the characteristics of the object which are required for the frame representation. Selection of the proper frame is done by specialist programs for the type of problem area (in this case, physics problems). The specialist programs know what types of canonical objects might be appropriate representations for each type of actual object; selection of the proper one is done by examining the context of the object (e.g., how many objects it is attached to, whether it supports something or is supported or is attached to something that supports something, etc.).

In general, a mapping function is required to abstract the features of an object for use in its corresponding canonical object frame. The necessary features may be specified implicitly, or in a different form from that which is needed, or they may not be specified at all, For example, in the problem shown in Figure 2, the length of the pole and the weight of the weight are unspecified; the values of these necessary attributes must be inferred (in this case, by creating symbolic constants for them) in order to complete the canonical object frame representations. A second set of

mapping functions may be needed to translate the results of reasoning processes performed on the canonical objects into the form required for the "actual" object representations, and to propagate the consequences of the new results.

The advantage of viewing an actual object as a canonical object is that canonical objects interact with each other in precisely specified ways. The information about the actual object which may be used in an attempt to solve a problem is restricted to those aspects of the object which were abstracted to form the canonical object frame, which greatly reduces the size of the problem space compared to a uniform representation in which all information is equally accessible. In addition, specialist procedures may be attached to a canonical object frame type to specify the interactions with other canonical objects and to solve special problems which occur frequently. In the physics problem solver, the interactions of canonical objects obey physical laws which are expressed as equations; these laws include those taught in physics texts (e.g., the laws of rigid body statics) as well as other physical laws which are "obvious" (e.g., the tension forces exerted at the ends of a rope are equal in magnitude, nonnegative, and directed toward the center of the rope). The specialist procedures associated with each type of canonical object examine the objectrs relationships with other objects and write equations which describe the interactions of the objects; solution of the set of generated equations yields the answer to the problem.

As an example of the use of Canonical Object Frames, consider the man standing on the ladder in the problem of Figure 1, The problem statement says only that "a 150 lb. man is 10 ft. from the top"; in order to solve the problem correctly, the program must determine what role the man plays in this particular problem. It is inadequate to assume that a man will always play a single role, since in the problem of Figure 2 the man is carrying the pole rather than standing' on it. In order to choose the correct Canonical Object Frame, the context of the man is examined: he is attached (in some way) to the ladder, which is known to be supported at two points; therefore, the ladder is assumed to support the man, and a "point mass" Canonical Object Frame is chosen. Choosing the correct Canonical Object Frame is essential to solving a problem correctly: if the wrong one is chosen, it is likely to cause unsolvable equations to be written, or to cause essential information to be left out of the equations, so that the answer will be wrong.

The use of Canonical Object Frames is an important respect in which this program differs from the earlier programs STUDENT [Bobrow 68] and CARPS [Chamiak 68]. In STUDENT, which solved algebra word problems, the words or phrases between keywords were directly reduced to algebraic variables, without any analysis of the meanings of the words (in fact, nonsense words could be used). In CARPS, which solved calculus word problems, equations were associated directly with object names (for example, $x^2 + y^2 = 1$ was associated with LADDER). These direct associations caused these earlier programs to fail on problems which were otherwise within their abilities because an object played a "role1' different from the one which the program assumed. By the use of Canonical Object Frames, ISAAC separates the role played by an object from the representation of the object itself, so that a single object can play different roles in different contexts. Parts or features of an object may be abstracted. For example, the imaginary line between two points on a ladder may form one side of a triangle; it is not necessary that the entire ladder be a side.

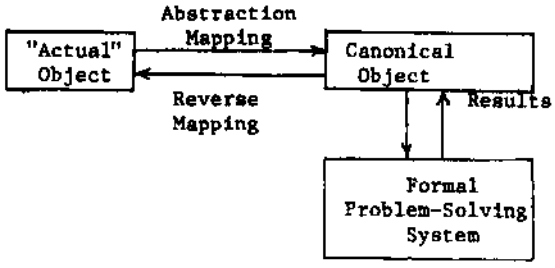
5. GEOMETRIC MODEL

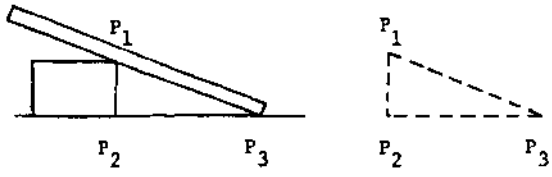
In order to write equations describing the interactions of the objects in a physics problem, it is necessary to construct a geometric model which relates positions on objects to a common coordinate system. Geometric models for individual objects are constructed by using prototype geometric models for each type of object; a prototype geometry is parameterized by the rotation of the object, the offset of the object relative to a larger geometric model, and scale factors for the x and y coordinates. Given the relative positions of locations on an object in the prototype geometry, it is easy to calculate the coordinates of a location on an "actual" object using simple vector geometry.

The composite geometric model is constructed from the individual geometric models by requiring that the coordinates of two points attached to each other be equal; thus, if an object is to be added to the composite model, it is only necessary to recalculate its offset so that its point of attachment will have the coordinates of the. corresponding point in the composite model. However, if several objects are connected so that they form a triangle, and the sizes or rotations of some of the objects are unknown, this simple method is inadequate. In this case, it is necessary to map the known line segment lengths and angles into a canonical triangle form, solve the triangle, and map the results back to give the desired rotations and object sizes.

6. CONCLUSION

The program for solving physics problems which is described in this paper makes frequent use of the notion of frames [Minsky 75], In our implementation, frames may be considered to be interpretations or "views" of other objects, which then allow problem solving by specialist programs which know how to deal with the canonical objects represented by the frames; a diagram and example of this process are shown below.





In this hypothetical example, the "abstraction mapping" creates representations of line segments (none of which, presumably, existed in the original representation) to form a canonical triangle model; the triangle can then be solved by a specialist program for triangles, and the result can be mapped back into the orignial representation (e.g., by defining the x coordinate of P, if the coordinates of P2 are known).

The notion of canonical object frames is a powerful technique for constructing problem-solving systems. By separating the recognition and abstraction processes from the formal problem solving system, it allows the latter to be written cleanly and efficiently, without regard for the specifics to which it may be applied. An existing formal problem-solving system may be used with a new type of object simply by adding a routine to interpret the new object as the appropriate canonical object. By restricting access to both procedures and data, the canonical object frame reduces the size of the problem space with which the problem solver must contend, and keeps irrelevant knowledge from being accessed.

This notion of the abstraction of a small number of properties of an actual object to form a representation of a canonical object is basic to the sciences and to engineering. Indeed, major scientific advances (e.g., coordinate geometry, Newtonian mechanics, atomic theory, phrase-structure grammar) have generally been accompanied by the introduction of new types of canonical objects with which the theories deal. Laws such as those of Newtonian mechanics derive their great power from their ability to predict the behavior of any physical object on the basis of only a few attributes, e.g., the object's mass and the resultant of the forces on it, The practices of engineering are heavily based upon the use of canonical object models; for example, an integrated circuit is described in terms of at least seven canonical models: a logic model, a geometric model and topology model of the circuit package, a thermal model, a timing model, a power consumption model, and a circuit model (which in turn is expressed in terms of canonical objects such as transistors and lumped resistances).

We believe that the organization of knowledge in terms of canonical object frames with associated specialist procedures for solving problems in particular areas is basic to a variety of disciplines (not only physics and engineering, but also such disciplines as medicine and law), and that it has considerable potential for artificial intelligence programs as well. The nature of the procedures which recognize an actual object as an instance of a canonical object and construct the mappings between the model of the object and the canonical object frame is an important area for future research.

REFERENCES

1. Bobrow, D. G. "Natural Language Input for a Computer Problem-Solving System", in Minsky (ed.) Semantic Information Processing, Cambridge: M.I.T. Press, 1968.

2. Bobrow, D. G. and Winograd, T. "An Overview of KRL, a Knowledge Representation Language", Cognitive Science, Vol. 1, No.l (Jan. 1977).

3. Charniak, E. "CARPS, a Program which Solves Calculus Word Problems", M.I.T. report MAC-TR-51, 1968.

4. Charniak, E. "Toward a Model of Children's Story Comprehension", M.I.T A.I. Lab Report AI TR-266, Dec. 1972.

5. Minsky, M. "A Framework for Representing Knowledge", in Winston (ed.) The Psychology of Computer Vision, New York: McGraw-Hill, 1975.

6. Novak, G. S. "Computer Understanding of Physics Problems Stated in Natural Language", Technical Report NL-30, Computer Science Department, The University of Texas at Austin, 1976.

7. Novak, G. S. "Computer Understanding of Physics Problems Stated in Natural Language", American Journal of Computational Linguistics, Microfiche 53, 1976.

8. Schank, R. C. and Colby, K. M. (eds.) Computer Models of Thought and Language, San Francisco: W. H. Freeman, 1973.

9. Simmons, R. F. "Semantic Networks: Their Computation and Use for Understanding English Sentences", in Schank and Colby (eds.), 1973,

10. Woods, W. A. "Transition Network Grammars for Natural Language Analysis", Coram. ACM, Vol, 13, No. 10 (October 1970), pp. 591-606.