PANGEA: AN EXPERT SYSTEM SHELL FOR MACHINE DESIGN PROBLEMS

I. HORVÁTH

Institute of Machine Design
Technical University, H-1521, Budapest
Received December 7, 1989
Presented by Prof. Dr. L. Varga

Abstract

The greatest problem in automating conceptual design is to computerize heuristic problem solving. Intelligent design systems must be able to process various types of knowledge components if they are to become effective tools for mechanical design. The developed PANGEA system combines predicative, procedural, propositional and meta-level knowledge in a manner that enables the solution of several generative problems of conceptual design.

Introduction

The extreme complexity of the human thinking process is evident. The fundamental mechanism of this process is only partially understood. The human scheme of knowledge interpretation, analysis, storage, access and superposition as applied by the conceptual designer is difficult to unravel. In developing intelligent machine design systems, on the one hand, novel system organization, information processing and programming principles must be followed; on the other hand, a formalized description of the design process has to be realized [5, 10]. Japanese and American researchers have attempted to develop a mathematically exact general theory of design, but actual computer architectures are insufficient for the implementation of this theory at a level that meets practical needs [9]. Additionally, theoretical fundamentals have to be further developed in the area of automating conceptual design. Research must address three main areas:

- knowledge processing as required for mechanical engineering design,
- formalization and typification of the problems of mechanical engineering design,
- development of computer methodologies for executing the design process entirely within the computer environment either as a fully automated process or as a process under the guidance of a human expert.

Present attempts to develop problem solving modules for intelligent design systems have made use of expert system concepts and the latest research into knowledge based systems [1, 4, 7]. In the case of machine design oriented expert systems, many related sub-problems of knowledge processing remain to be effectively implemented. Among these sub-problems are:

- synergic processing of several kinds of knowledge;
- multiple deduction plus evaluation strategies;
- data base and method base (algorithms) definition and processing;
- development of a system shell suitable for solving different conceptual problems by only changing the knowledge of the system.

Processing of designer knowledge by expert systems

The knowledge of machine design is, of course, a subset of human knowledge as applied to one human skill. This human knowledge subset is extremely complex. If we hope to capture it on a practical level we must separate it into distinguishable parts. Without claim to completeness, the following components of human knowledge (in particular with regard to problem solving strategies) may be identified:

- cognitive knowledge,
- plausible knowledge,
- abstracting knowledge,
- deductive knowledge,
- analog knowledge,
- predicative knowledge,
- procedural knowledge,
- meta-level knowledge.

To be effective, conceptual design has to utilize these knowledge components as an integrated whole. Accordingly, to be successful, intelligent design systems have to provide for several computer-oriented knowledge processing mechanisms but not necessarily in a manner that emulates the human problem solving process.

Structurally, an expert design system requires an integration of knowledge representation and inferencing modules [2]. To be effective, machine design oriented expert systems not only have to include or simulate the complex knowledge processing demands outlined above, but must also do them in concert. Thus, an effective system must be able to make heuristic combinations of structural elements, to perform algorithmic evaluations, to perform logic inferencing, to access databases of components, to display numerical and graphical results, etc. [3].

The expert system described in Fig. 1 is an illustrative example of such a system. The RULE BASE contains a structured set of rules and the FACT BASE includes

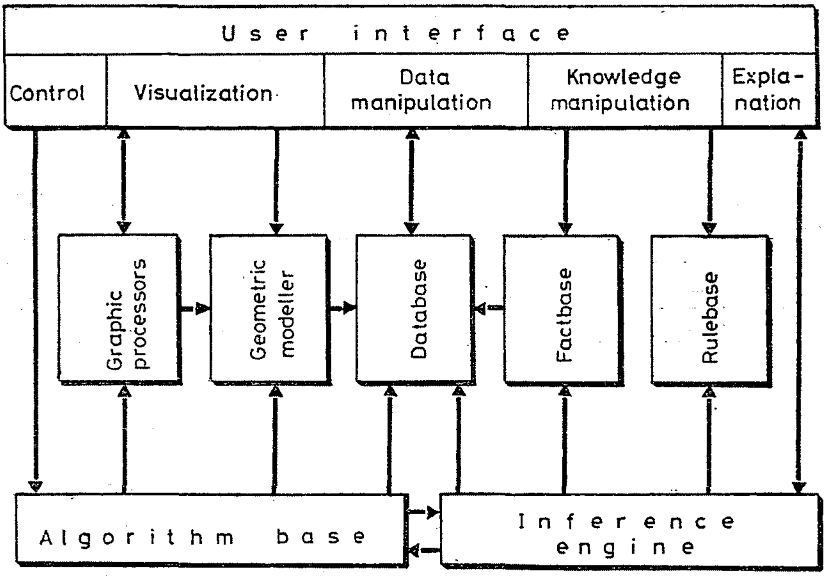

Fig. 1. Basic elements of a mechanical engineering design oriented expert system

facts describing the actual condition of the given problem. This Fact Base is constantly updated as the design process proceeds. The INFERENCE ENGINE is an algorithm based module that performs manipulations necessary for evaluating the RULE BASE and initiating actions with the ALGORITHM BASE. The USER INTERFACE provides a dialogue type communication with the user.

General features of the PANGEA system

As an adjunct to research on the applicability of expert systems in machine design, an experimental PC based expert system has been developed at the Institute of Machine Design, TUB [8], [6]. This system, denoted as PANGEA, is both a knowledge programming environment and a rule based processing shell and attempts to incorporate most of the features described in Fig. 1. From the point of view of system functioning, PANGEA enables the following activities:

— development of a knowledge processing environment;
— development and processing of rule bases;
— development and processing of fact bases;
— composition and compilation of knowledge components into symbolic forms;
— selection of different modes or methods of inferencing;
— development and construction of data (object) bases;

— development of specialized algorithmic methods;
— complex problem solving based on the integration of inferencing, database management and algorithmic methods;
— file management and housekeeping functions;
— capabilities for user guidance, help and system explanation.

PANGEA's user interface is designed as a series of pop-up menus accessible from the keyboard or by a mouse (Fig. 2). The UTILITY sub-menu enables the installation of the knowledge processing environment and the setting of system parameters. A HELP system provides an introduction to the various system activities. A TUTOR facility provides initial training in how to use the system.

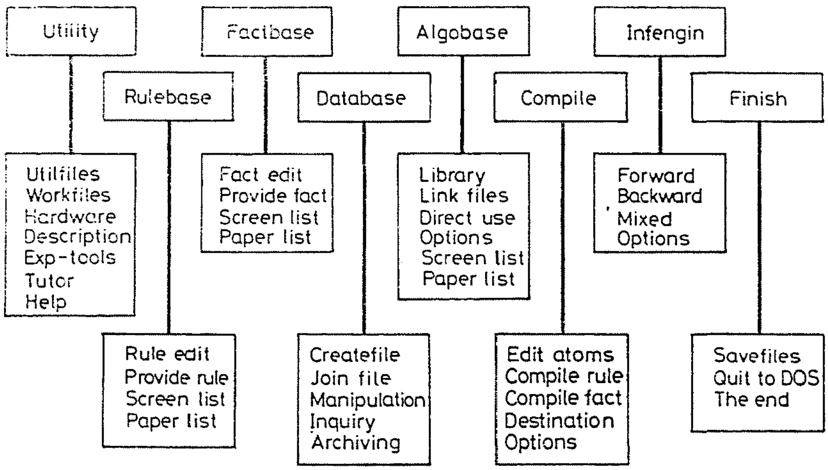

Fig. 2. Main menu options of PANGEA

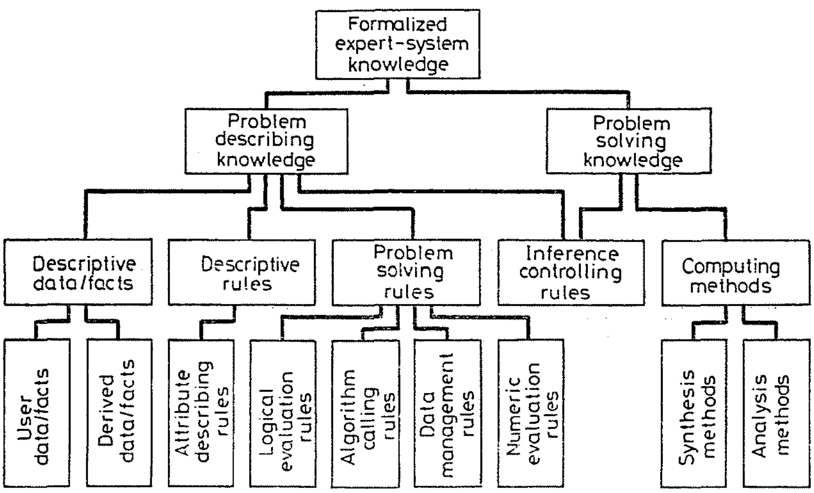

Fig. 3. Principle elements of the knowledge base of PANGEA

Information elements of the knowledge base of PANGEA are shown in Fig. 3. The rules of the rule base have an IF—THEN structure in which each rule has a single conclusion and up to five premises. The number of premises is limited only for the sake of screen display. Among rule premises, essentially all of the propositional logic operations are supported. Although all rules are of the same general structure, they may be classified by the type of activity they perform. There are three classes of rules: Problem Descriptive Rules, Problem Solving Rules and Meta-Rules. Problem Descriptive Rules are those that describe general conditions. Problem Solving Rules perform four types of activities. These are:

1. make logical evaluations;
2. obtain numeric evaluations by asking for user input or by triggering calculations;
3. call various problem solving methods (algorithms);
4. perform data base transactions.

Meta-Rules are essentially rules about the rules. They provide for a selection of sub-sets of rules that apply at different points in the problem solving process. Algorithmic routines can be user added and separately compiled as the method base is developed. The capability for rules to call routines in the method base integrates the abilities of the system.

The rule base as input by the user utilizes macro identifiers in the development for IF-THEN structures. The macros themselves are textual phrases that describe situations, form questions and describe conclusions. A rule base is stored as a text file that is subsequently deciphered and compiled by PANGEA. The data base is a structured data file complex describing objects and object attributes. The method base contains program files (.exe) that are called by specialized rules when the rule "fires".

File handling is enabled by several useful, built-in accessories. PANGEA includes a text editor for composing the macro phrases, the rule base, and the initial fact base conditions. However any text editor that produces standard ASCII files may be used. PANGEA performs a special inner decoding of the ASCII rules and fact base. The needed transformation of the rules and facts is performed in the COMPILER module. If a syntax error occurs, the translator places the user back into the text editor at the appropriate error position. The translator weights premises according to their frequency of occurrence. This weighting increases the efficiency of the interrogation process.

The RULE BASE module allows the user to select from a library of compiled rule base files. Thus, several bases can be stored in the system and the desired base selected through an option provided in this module. Previously defined rules, which in their definition phase use macro identifiers, can be displayed in their full text form. The rules may be listed at the screen or on paper. Individual rules or all of the rules may be selected for display.

The FACT BASE module allows the display and manipulation of the fact base files in essentially the same manner as in the RULE BASE module. After selection of a particular fact base file (consistent with the rule base file previously selected) certain premises (facts) can be established initially as true. As the system progresses in its analysis, inferred conclusions based upon the rule base and upon the truth or falseness of premises are added to the fact base. The user is queried about other premises and additional conclusions (facts) are added to the fact base. The FACT BASE module allows a full display of the facts at the screen or on paper so that the user can see the completed inferencing process.

Methods of inferencing

The INFENGIN module of PANGEA presently allows for two inferencing mechanisms: forward chaining and backward chaining. However, it is possible to incorporate inductive inferencing and a mixture of forward and backward chaining.

The INFENGIN module has some selectable modes of operation that are useful in developing and verifying a rule base. Rule base development is a time consuming and exacting process, so these special operational modes are essential. The system can be placed in a single step mode in which each rule that is tested is displayed on the screen. The fact base is also monitored and displayed in order to track its expansion and development by the inferencing process. Meta-rules that select the activity modules of the rule base can also be monitored. Finally, the inferencing process can be timed so that the knowledge engineer can determine the efficiency of the used rule bases.

The forward chaining unit uses pseudo-nonmonotonic logic such that rules are evaluated in three-state-logic (true, false, don't care). The inferencing engine uses the relative weighting of the various macros to determine the most appropriate query.

In the backward chaining mode a set of hypotheses is selected by evaluating the rule base to find possible end conclusions. Each hypothesis is evaluated for truth by using the relevant rules and checking the items in the fact base or by interrogating the user. Final solution is achieved only if one hypothesis can be proved. Backward chaining does not allow a meta rule capability since the manner of inferencing prohibits considering them. Stepwise inferencing and time measurement development modes can be selected as described above.

The METHOD BASE module of PANGEA contains the methods for arithmetic calculations and parameter determinations. The system DATA BASE contains catalogs of components and descriptive data sets of the various standard parts, assemblies and parameterized prototypes that must be evaluated by the method base. The data base is updated by rule calling procedures on the basis of data assigned to the design parameters.

Knowledge programming

Perhaps the most difficult phase in the development of an expert system is the knowledge input. In general rule based systems, this is equivalent to translating knowledge into a series of IF-THEN type structures. For a mechanical design system, the knowledge is also in the form of component data bases and algorithmic procedures. The closer the representation of knowledge to the human thinking process and to natural language, the less is the likelihood for errors or misinterpretations in the knowledge. With this in mind, PANGEA uses a natural descriptive mode for the macros. But the knowledge rules and procedures must be formalized. For this purpose, PANGEA uses a syntax similar to that of common highlevel programming languages.

PANGEA syntax covers the following language operations:

— definition of rule base;
— rule definition;
— definition of fact base;
— fact definition;
— definition of macros;

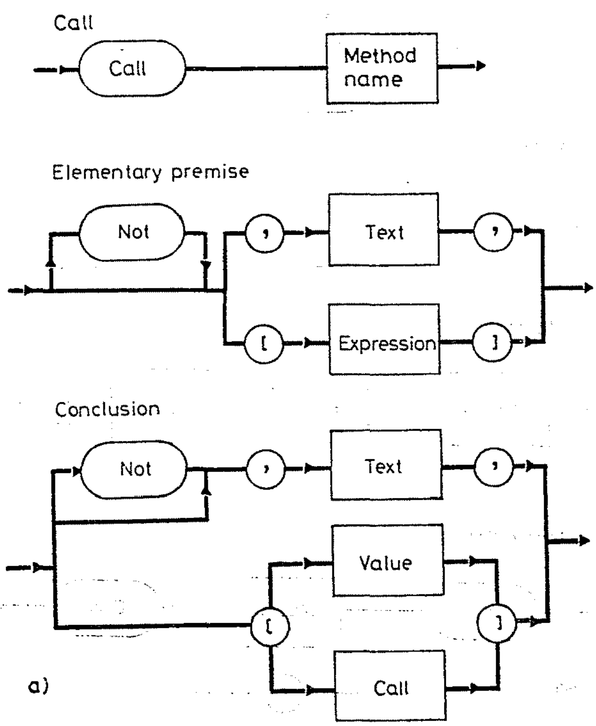

Fig. 4a.

— definition of the list of macros;
— definition of methods;
— condition definition;
— definition of the elementary conditions;
— definition of consequences;
— definition of the procedure calls;
— expression definitions;
— definition of valuation;
— factor definition;
— text definition;
— name definition;
— constant definition;
— unsigned real number definition;
— unsigned integer definition.

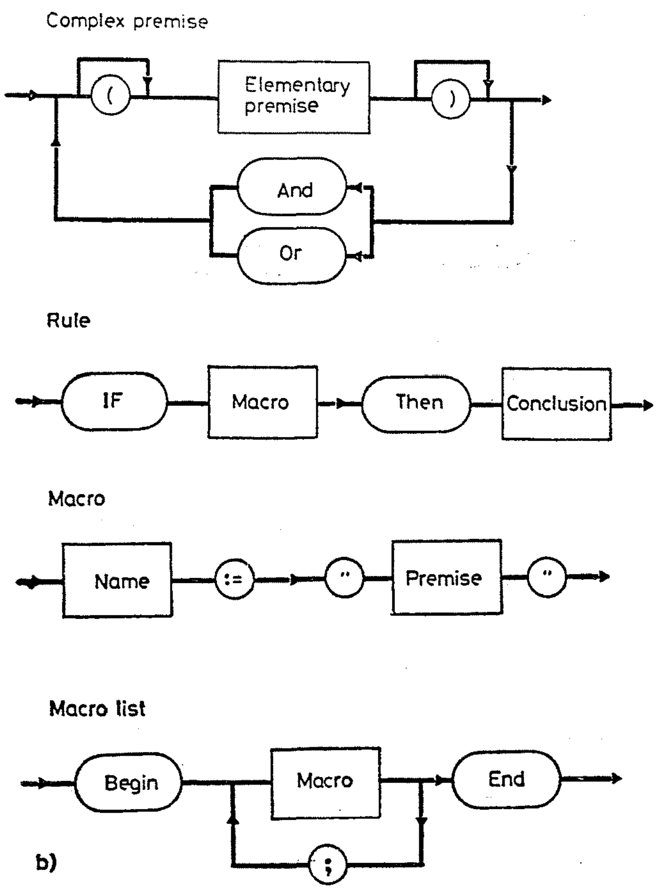

Fig. 4b.

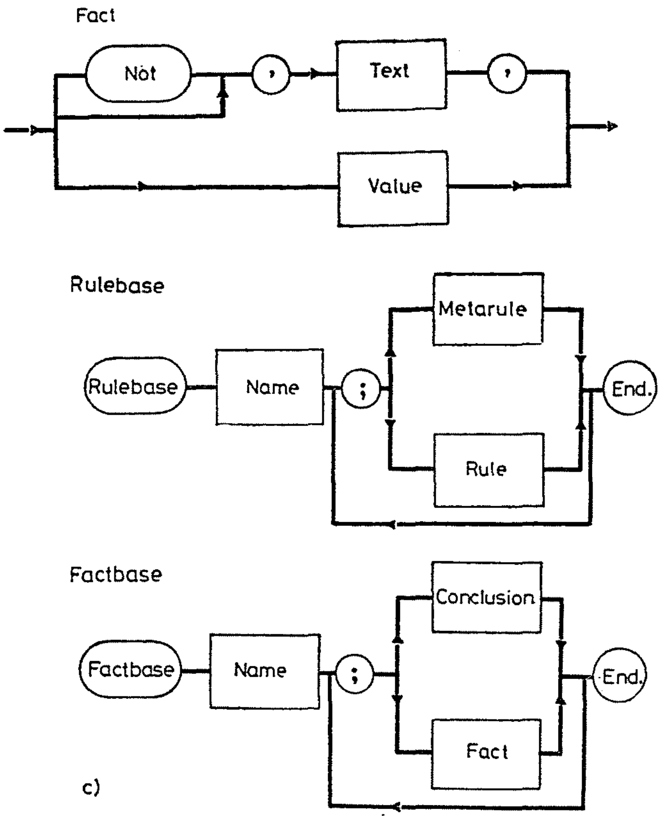

Fig. 4. Syntax diagrams for PANGEA language

Some of the syntax diagrams are shown in Fig. 4. The rule base as entered by the knowledge programmer is a text file that is later "compiled" by a syntactic analyser. During the compilation process, the system assigns inner symbols to the macros. During the inferencing, instead of the macro identifiers, an internal coding is utilized. Practically, the meaning attached to the macro by atom phrases is of use only for human communication.

An application example

Theoretically there are four general types of mechanical engineering design problems. These are as follows:

1. Qualitative Selection
Selecting a member from a set of known design objects that meets given functional conditions and demands

```
begin
t000 := "'feladat ket tengelyveg osszekapcsolasa'" ;
t001 := "'nagy tengelyiranyu tavolsag athidalas szukseges'" ;
t002 := "'vegleges tengelykotes alkalmazhato'" ;
t003 := "'tengelykapcsolo alkalmazhato'" ;
t004 := "'tengelyvegek kozott allando kapcsolat kell'" ;
t005 := "'tengelyek kozott mechanikus kapcsolat kell'" ;
t006 := "'tengelyek gyakorlatilag egytengelyuek'" ;
t007 := "'gepegysegek tengelyiranyban mozgathatok'" ;
t008 := "'jatekmentes kapcsolat szukseges'" ;
t009 := "'szogsebesseg ingadozas megengedett'" ;
t010 := "'konnyu szetbonthatosag szukseges'" ;
t011 := "'gyakori kapcsolasra van szukseg'" ;
t012 := "'kis teljesitmeny atvitele szukseges'" ;
t013 := "'kozepes teljesitmeny atvitele szukseges'" ;
t014 := "'tengelyvegek kicsuszasa megakadalyozando'" ;
t015 := "'magas uzemi fordulatszam (n<=6000 1/perc)'" ;
t016 := "'kapcsolofelek finom szogbeallitasa szukseges'" ;
t017 := "'dinamikus igenybevetel fellep '" ;
t018 := "'mechanikus tengelykapcsolo alkalmazhato'" ;
t019 := "'zsir/olajkod szennyezodes mersekelt'" ;
t020 := "'merev tengelykapcsolo alkalmazhato'" ;
t021 := "'nagy teljesitmeny atvitele szukseges'" ;
t022 := "'alakkal zaras szukseges'" ;
t023 := "'kozpontositas szukseges'" ;
t024 := "'fuggoleges beepites igenyelt'" ;
t025 := "'kis helyszukseglet kell'" ;
t026 := "'rovid tengelyek osszekapcsolasa szukseges'" ;
t027 := "'hosszu tengelyek osszekapcsolasa szukseges'" ;
t028 := "'hofokvaltozas szamottevo'" ;
t029 := "'kis uzemi fordulatszam'" ;
t030 := "'kozepes uzemi fordulatszam'" ;
t031 := "'zsir/olaj kenes lehetseges'" ;
t032 := "'eros zajhatas megengedheto'" ;
t033 := "'parhuzamos tengelyhiba megengedheto'" ;
t034 := "'altalanos mech nikus hajtas kornyezet'" ;
t035 := "'jarmuipari alk lmazasi kornyezet'" ;
t036 := "'szerszamgepip  alkalmazasi kornyezet'" ;
t037 := "'alacsony kolt  u gyartas szukseges'" ;
t038 := "'lehetseges t   yhiba axialis eltolodas'" ;
t039 := "'lehetseges t   yhiba parhuzamos eltolodas'" ;
t040 := "'lehetseges     yhiba szogelfordulas'" ;
t041 := "'nagy dinam     omatek terheles mukodik'" ;
t042 := "'nagy parhu     ngelytav athidalas szukseges'" ;
t043 := "'nagy szog      s uzemszeruen fellep'" ;
t044 := "'csak uzem      dasra van szukseg'" ;
t045 := "'uzemkozb       apcsolasra van szukseg'" ;
t046 := "'egyiran        megengedheto'" ;
```

Fig. 5. Series of macros for the clutch problem

```
***********RULEBASE***********

1. RULE :
      IF          feladat ket tengelyveg osszekapcsolasa  50
     AND          nagy tengelyiranyu tavolsag athidalas szukseges  50
     AND      NOT nagy parhuzamos tengelytav athidalas szukseges  50
     AND          tengelyek gyakorlatilag egytengelyuek  50
    THEN          transzmisszios tengely valaszthato

2. RULE :
      IF          feladat ket tengelyveg osszekapcsolasa  50
     AND      NOT nagy tengelyiranyu tavolsag athidalas szukseges  50
     AND      NOT nagy parhuzamos tengelytav athidalas szukseges  50
    THEN          tengelykapcsolo alkalmazhato

3. RULE :
      IF          feladat ket tengelyveg osszekapcsolasa  50
     AND      NOT nagy tengelyiranyu tavolsag athidalas szukseges  50
     AND          nagy parhuzamos tengelytav athidalas szukseges  50
    THEN          teljesitmeny atvivo hajtas szukseges

4. RULE :
      IF          tengelykapcsolo alkalmazhato  50
     AND          tengelyek kozott mechanikus kapcsolat kell  50
    THEN          mechanikus tengelykapcsolo alkalmazhato

5. RULE :
      IF          tengelykapcsolo alkalmazhato  50
     AND      NOT tengelyek kozott mechanikus kapcsolat kell  50
    THEN          kulonleges elvu tengelykapcsolo szukseges

6. RULE :
      IF          mechanikus tengelykapcsolo alkalmazhato  50
     AND          tengelyvegek kozott allando kapcsolat kell  50
    THEN          szerelessel bonthato kapcsolo alkalmazhato

7. RULE :
      IF          mechanikus tengelykapcsolo alkalmazhato  50
     AND      NOT tengelyvegek kozott allando kapcsolat kell  50
    THEN          oldhato tengelykapcsolo szukseges

8. RULE :
      IF          szerelessel bonthato kapcsolo alkalmazhato  50
     AND          tengelyek gyakorlatilag egytengelyuek  50
     AND          rovid tengelyek osszekapcsolasa szukseges  50
     AND      NOT lengescsillapitas szukseges  50
    THEN          merev tengelykapcsolo alkalmazhato

                                                    azhato  50
                                                      gathatok  50
```

Fig. 6. Coded rules for qualitative selection

```
***********FACTBASE***********

User defined ===>      feladat ket tengelyveg osszekapcsolasa
User defined ===>  NOT nagy tengelyiranyu tavolsag athidalas szukseges
From 1.rule ===> NOT transzmisszios tengely valaszthato
From 29.rule ===> NOT kardantengelyes hajtas valaszthato
From 30.rule ===> NOT parhuzamos kettoskardan valaszthato
From 31.rule ===> NOT bordashuvelyes kardantengely valaszthato
User defined ===>  NOT nagy parhuzamos tengelytav athidalas szukseges
From 2.rule ===>     tengelykapcsolo alkalmazhato
From 3.rule ===> NOT teljesitmeny atvivo hajtas szukseges
User defined ===>      tengelyek kozott mechanikus kapcsolat kell
From 4.rule ===>     mechanikus tengelykapcsolo alkalmazhato
From 5.rule ===> NOT kulonleges elvu tengelykapcsolo szukseges
From 49.rule ===> NOT acelporos tengelykapcsolo valaszthato
From 50.rule ===> NOT Triumph tengelykapcsolo valaszthato
User defined ===>      tengelyvegek kozott allando kapcsolat kell
From 6.rule ===>     szerelessel bonthato kapcsolo alkalmazhato
From 7.rule ===> NOT oldhato tengelykapcsolo szukseges
From 12.rule ===> NOT kormos tengelykapcsolo alkalmazhato
From 14.rule ===> NOT kapcsolhato tengelykapcsolo szukseges
From 36.rule ===> NOT surlodo tengelykapcsolo valaszthato
From 37.rule ===> NOT egymunkafeluletu surlodo tengelykapcsolo valaszthato
From 38.rule ===> NOT kupos surlodo tengelykapcsolo valaszthato
From 39.rule ===> NOT dobos surlodo tengelykapcsolo valaszthato
From 43.rule ===> NOT Gumidugos tengelykapcsolo valaszthato
From 44.rule ===> NOT Forst tengelykapcsolo valaszthato
From 45.rule ===> NOT Man-Renk tengelykapcsolo valaszthato
From 46.rule ===> NOT belso dobos tengelykapcsolo valaszthato
From 47.rule ===> NOT olajos lemezes tengelykapcsolo valaszthato
From 48.rule ===> NOT kulso dobos tengelykapcsolo valaszthato
User defined ===>      tengelyek gyakorlatilag egytengelyuek
From 21.rule ===> NOT kiegyenlito kapcsolo alkalmazando
From 24.rule ===> NOT Oldham tengelykapcsolo alkalmazhato
From 25.rule ===> NOT szogkiegyenlito tengelykapcsolo alkalmazando
From 26.rule ===> NOT egyszeru kardancsuklo valaszthato
From 27.rule ===> NOT szinkroncsuklo alkalmazhato
From 28.rule ===> NOT ives belsofogazatu tengelykapcsolo valaszthato
From 11.rule ===> NOT dilatacios tengelykapcsolo valaszthato
From 13.rule ===> NOT egyenes belsofogazasu tengelykapcsolo alkalmazhato
From 22.rule ===> NOT bordastengelykotes alkalmazhato
From 23.rule ===> NOT tarcsas dilatacios tengelykapcsolo alkalmazhato
User defined ===>      lengescsillapitas szukseges
From 8.rule ===> NOT merev tengelykapcsolo alkalmazhato
From 9.rule ===> NOT tokos tengelykapcsolo valaszthato
From 10.rule ===> NOT nyirt csapszeges tokos tengelykapcsolo
From 15.rule ===> NOT reteszes tokos tengelykapcsolo
From 16.rule ===> NOT zsugorkotesu tokos tengelykapcsolo
From 17.rule ===> NOT kupos kapcsolohuvely alkalmazhato
From 18.rule ===> NOT hejas tengelykapcsolo alkalmazhato
From 19.rule ===> NOT tarcsas tengelykapcsolo alkalmazhato
From 20.rule ===> NOT homlokfogazatu tengelykapcsolo valaszthato
User defined ===>      lokesszeru igenybevetel fellep
From 32.rule ===>     rugalmas tengelykapcsolo alkalmazhato
```

Fig. 7. Resulting factbase after an inferencing session

2. *Prototype Evaluation*
   Finding the most appropriate set of design values for a parameterized object prototype that meets functional requirements

3. *Configurative Synthesis*
   Selecting appropriate items from a given set of paradigm type design objects, coupling them into assemblies or groups and evaluating for functional requirements

4. *Generative Synthesis*
   Selecting the most elementary building entities to build up sub-groups or sub-assemblies and combining them with known groups or sub-assemblies to develop a generic prototype for functional evaluation. The activities for the design process in this case are not known.

At the present stage of development, PANGEA is able to cope with problems of the first three types. The fourth type is a generative, innovative design which is oriented towards the development of previously unknown objects. If this problem is to be solved by computers, a completely new design methodology is required.

Details of solving the problems belonging to the first category can be illustrated relatively easily. A test rule base has been developed with PANGEA for the selection of mechanical clutches appropriate to a specific application. The rules are of simple concluding format using macro statements associated with the clutch selection problem. A part of the set of macros is shown in Fig. 5. A partial set of the rules associated with these macros is shown in Fig. 6. A sample selection dialogue and the resulting fact base is shown in Fig. 7. Examples of application to the other design problem types will be discussed in other publications. There remains much to do in applying the PANGEA shell to the fourth problem type.

Conclusions

The knowledge elements necessary for problem solving in machine design have been identified — without claim to completeness. An expert system architecture appropriate for mechanical machine design has been identified and a first generation system applying this architecture has been developed. Initial results are promising.

References

1. Charniak, E.—McDermott, D.: Introduction to Artificial Intelligence, Addison-Wisley, Reading, 1985.
2. Edmunds, R. A.: The Prentice Hall Guide to Expert Systems; Prentice Hall, Englewood Cliffs, 1988.

3. HORVÁTH, I.—BERCSEY, T.: Entwicklungstendenzen der Rechnergestützten Technologien und neue Aufgaben der Methodischen Konstruierens von Maschinenelemente; 6th Workshop MeKoMe, Rigi-Kaltbad, 1989. March (in German).
4. HORVÁTH, I.: A műszaki szakértőrendszerek fejlesztésének szabályai; Computer World Számítástechnika, *3.* évf., 20. szám, 1988. október (in Hungarian).
5. HORVÁTH, I.: Szakértőrendszerek géptervezési alkalmazása; MicroCAD 89, Miskolc, 1989. február, pp. 56—64, (in Hungarian).
6. HORVÁTH, I.—TAKÁTS, I.: Szakértőrendszerváz-fejlesztés géptervezési feladatokra: Automatizálás, XXII. évfolyam, 9. szám, 1989. szeptember (in Hungarian).
7. SIEGEL, P.: Expert Systems; Tab Books, Inc., USA, 1986.
8. TAKÁCS, I.: Következtető mechanizmus és programozási környezet fejlesztése géptervezési feladatokhoz; Tudományos Diákköri Dolgozat, BME GSZI, Budapest, 1988.
9. TANIMOTO, S. L.: The Elements of the Artificial Intelligence; Computer Science Press, New York, 1987.
10. TOMIYAMA, T.—YOSHIKAWA, H.: Requirements and Principles for Intelligent CAD Systems; in Knowledge Engineering in Computer Aided Design, ed. by Gero, J. S., North Holland, Amsterdam, 1987. pp. 1—23.

Dr. Imre HORVÁTH, H-1521, Budapest