

# THE VEX-93 ENVIRONMENT AS A HYBRID TOOL FOR DEVELOPING KNOWLEDGE SYSTEMS WITH DIFFERENT PROBLEM SOLVING TECHNIQUES <sup>1</sup>

Julio J.Valdés<sup>2</sup> Ramón Hita<sup>3</sup> Katia Peón<sup>3</sup>  
Dania Hernández<sup>3</sup> Ada García<sup>3</sup> Raul Paredes<sup>3</sup>  
Yazna García<sup>3</sup> Alfredo Rodríguez<sup>3</sup>

<sup>2</sup>Departamento de Lenguajes y Sistemas Informáticos. Sección de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Catalunya, Barcelona, Spain.  
e-mail:valdes@goliat.upc.es Tel: + 34 3 401-7334, Fax: +34 3 401-7014

<sup>3</sup>Laboratorio de Inteligencia Artificial y Análisis de Datos. Academia de Ciencias de Cuba-COPEXTEL, Ave 7ma, calle 192, Playa, La Habana,Cuba. e-mail: laisoft@tinored.cu

## Abstract

The paper describes VEX-93 as a hybrid environment for developing knowledge-based and problem solver systems. It integrates methods and techniques from artificial intelligence, image and signal processing and data analysis, which can be mixed. Two hierarchical levels of reasoning contains an intelligent toolbox with one upper strategic inference engine and four lower ones containing specific reasoning models: truth-functional (rule-based), probabilistic (causal networks), fuzzy (rule-based) and case-based (frames). There are image/signal processing-analysis capabilities in the form of programming languages with more than one hundred primitive functions. User-made programs are embeddable within knowledge basis, allowing the combination of perception and reasoning. The data analyzer toolbox contains a collection of numerical classification, pattern recognition and ordination methods, with neural network tools and a data base query language at inference engines's disposal.

VEX-93 is an open system able to communicate with external computer programs relevant to a particular application. Metaknowledge can be used for elaborate conclusions, and man-machine interaction includes, besides windows and graphical interfaces, acceptance of voice commands and production of speech output.

The system was conceived for real-world applications in general domains, but an example of a concrete medical diagnostic support system at present under completion as a cuban-spanish project is mentioned.

Present version of VEX-93 is a huge system composed by about one and half millions of lines of C code and runs in microcomputers under Windows 3.1.

**Keywords:** problem solving, knowledge base systems; mixing several reasoning models; image and signal processing; neural networks; data analysis; hybrid systems

---

<sup>1</sup>Project partially supported by a grant (1993) from the Alexander von Humboldt Foundation (Bonn, Germany).

## 1 Introduction

Knowledge-based systems are complex computational constructs oriented to real world problem solving and are starting to have their own history in computer science, informatics and technology. With a variable degree of success they have addressed tasks like diagnostics, planning, scheduling, design, manufacturing and many others.

Almost all of these research and development efforts have been conducted in the framework of artificial intelligence, triggering both theoretical as well as practical activities from a variety of different approaches, and as a consequence a broad spectrum of paradigms and reasoning models has emerged. Also some morals have been derived. In particular, that problem solving in real world domains is much more than using artificial intelligence but a complex activity demanding also methods, models and tools from other fields in an integrated way.

In which concerns to the AI component, the availability of different reasoning models naturally led to two attitudes or streams. One try to define a single representation language able to model "the way " humans reason (production systems, probabilistic models, connectionistic models, etc.) and is often called the *reductionistic* approach [Hájek, Zvárová, Esteva 1994]. Another called the *hybrid* approach does not consider possible to model complex problems with only few concepts, nor to describe uniquely the way humans reason or solve problems, and try to integrate or mix several such models or techniques in a covering architecture.

Real world problems are usually very complex and this is expressed by several features: the different parts or stages of a problem, the different kind of tasks related with each of them, the variety of information sources and types of knowledge and data which must be used, the different kind of operations which must be done with them, the way in which all these leads to conclusions and so on. Moreover, at present no single representation model can cover all these. If human reasoning is considered as such, a similar conclusion can be reached. To have such a unique general theory would be nice but it seems unlikely. Moreover, the practice of knowledge engineering (very often neglected) shows that whenever a non-toy or non-simple problem is addressed, huge complications arises. One of the reasons for the failure of many expert systems application projects, or for the reluctancy/rejection to use them, is because usually different parts of the same problem are forced to be solved by the same reasoning model or technique. Often this is something imposed by the software tool used (usually an expert system shell with a narrow scope of possibilities).

The sense of multi-conceptual modelling arises in almost any real-world domain, since in them the knowledge used covers the entire range from purely heuristic (empirical rules or principles) to formal (e.g: processes described by differential equations). The types of evidences can be purely judgmental (e.g: "the patient is depressed"), perceptual (e.g: a tumor recognized from an X-ray image from the thorax), numerical (results from laboratory tests), and so on. Moreover, all of them have inherent an uncertainty nature in a general sense. Each piece of information has a different importance and the specific consequences which can be derived from them are very much contextually dependent.

Important points in this respect are knowledge processing (of different kinds and at different abstraction levels), data analysis (of different nature, from different sources and with space/time dependencies), managing uncertainty, and the interaction with other informatic systems. The VEX-93 project represents an experimental environment inspired in this philosophy, conceived a a tool for problem solving in general domains by integrating different knowledge processing and other techniques. Therefore it belongs to the branch of *hybrid systems*. Although conceived originally as a research project in computer science, the VEX-93 environment even in its present state (version 1.0) has shown to be a very useful tool in handling problems from real world applications.

This paper outlines VEX-93's v1.0 general context and content, and its possibilities in the areas of problem solving, decision support, study of complex systems and other topics.

## 2 The VEX-93 approach

First of all, VEX-93 is a hybrid tool and this concept is present in many of its features:

- By integrating artificial intelligence with data analysis in a broad sense (including data with space-time dependencies).
- By combining perception with abstraction (VEX-93 may have direct access to the real world).
- By allowing evidences of different nature (judgements, images, pictures, data basis, digital signals, etc.).
- By allowing a mixture of heuristic with formal models.
- By supporting different AI reasoning models (rule-based, probabilistic, connectionistic models, fuzzy models, case-based reasoning), which can be used to describe parts of a unique problem.
- By allowing different knowledge representation forms, different uncertainty notions and several hierarchical levels.
- By allowing problem decomposition and a variable degree of dependence in the solution of the individual parts of the problem. This means structuring the knowledge.
- By being an *open system*, offering not only a set of internal tools but also by giving the user the possibility to incorporate external tools in the form of computer programs with which VEX-93 can exchange information.
- By offering four internal *programming languages* in which the developers may describe their problem solving views and conceptions about the concrete application engineered.

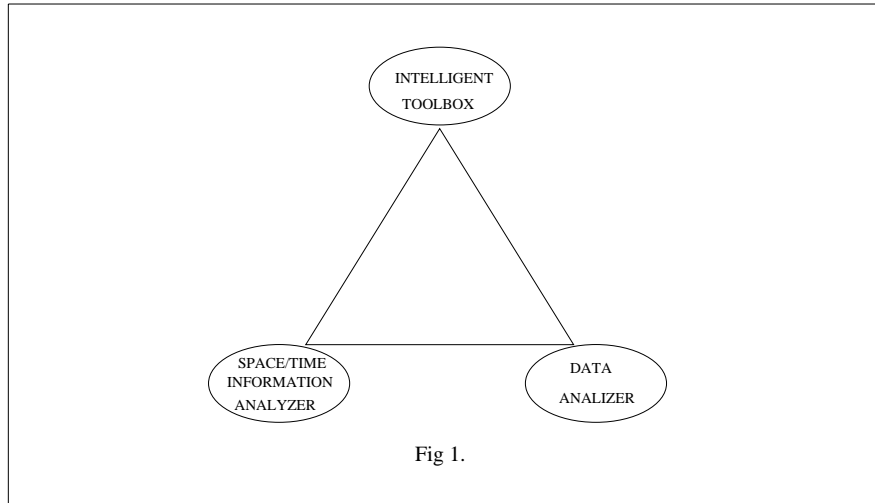
A key concept in VEX-93 is that of complex processing of information of very different nature at different hierarchical and abstraction levels, going from raw data to strategies. The idea is that of information flow and processing according to: its nature, the task or subtask to be solved and the general problem solving strategy the system should use (specified by the developers). VEX-93 v1.0 is composed by three integrated set of tools shown in fig.1.

## 3 Components and architecture

VEX's architecture is shown in fig.2.

Each set of tools can be used with a variable degree of interconnection conditioned by both the nature of the application and the idea of the developers about the problem. The Intelligent Toolbox contains a collection of inference engines working at two different hierarchical levels: the Strategic level (using metaknowledge) and the Specific level (containing the different reasoning models). The Space/Time Information Analyzer contains two interpreters for languages dedicated to image and signal processing/analysis. These languages contain more than one hundred functions and operators which can be used as primitives for programs by means of which digital images and/or signals can be processed, and analyzed. They can be used also as a way how to give to the knowledge system *perception* capabilities for image/signal *processing and understanding*. This is done by transforming what is relevant for the problem in terms of objects or features, into numerical or *conceptual* descriptions of interesting events and *logical statements*. The Data Analyzer is a collection of tools for data querying and analysis containing methods for numerical classification, pattern recognition, ordination, etc. and also for data mining and discovery. The data base querying tool allows the inference engines to formulate queries of different kinds to data basis, to extract information from them or to compute logical or numerical quantities required by the reasoning process.. VEX-93 v1.0 can also *generate* internal data basis for its work.

VEX-93 has two non-exclusive input sources: a user, and real world. The information combining



from the user may consist of evidences of different kinds, facts about the problem (e.g: data basis), external programs (with mathematical models, or other information systems which he knows are relevant for his problem), etc. Input from the real world may consist of digital pictures, signals, data directly gathered by sensors monitoring a process (eg: a patient, a natural environment), etc. With these inputs, a problem solving model or strategy (usually in the form of a *knowledge processing program* interpreted by the strategic inference engine) and with VEX's internal machinery, an output is constructed as a set of conclusions, recommendations and/or actions (shown or executed). This is a rough description of the usual way in which VEX-93 could be used. Now the main parts will be outlined.

### 3.1 The Perception Level

There are two main hierarchical levels in VEX-93: the lower models *perception* and the upper models *abstract reasoning*. The former accepts direct information from the real world which might consist of images (eg: from a TV camera or a microscope), single or multichannel digital signals (from instrument recordings, sounds, etc.), or punctual data (e.g. a temperature taken from a sensor). They are taken as evidences in the same way as user's judgments are. VEX-93 contains interpreters for two *dedicated languages* for image and signal processing and analysis. They provide a flexible way to incorporate knowledge in the form of complex processing and computations over images and signals, as well as constructs for the recognition, description and extraction of objects or events of interest and for their representation as *abstract concepts* which can be used within the reasoning process at a higher level by the inference engines. The latter allow the system to *understand* in some sense the relevant information from these sources. In this case system's developers might let the system work either with what the user is perceiving or interpreting from the world, or with what it finds by itself without user's intervention. Another possibility is to define a cooperative behaviour between the system and the user in a variable degree.

Thus, *perception* is modelled in VEX as a preprocessing of raw information of the above mentioned kind with its transformation into abstract concepts, objects or assertions (often of logical or qualitative nature). These results can be used either for direct decision/action making or as inputs to more abstract processes like the reasoning done by the inference engines.

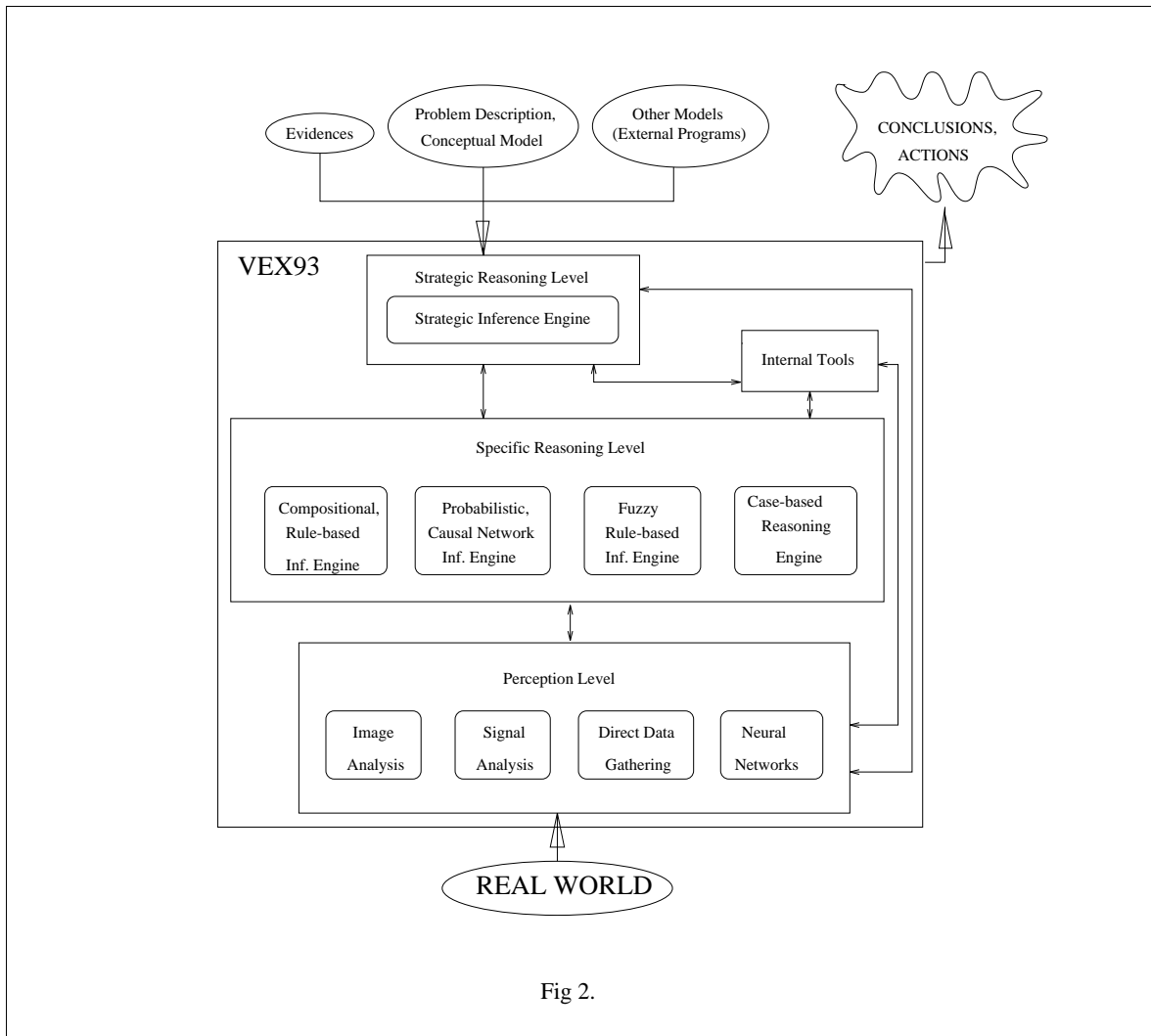


Fig 2.

### 3.2 The reasoning level

The reasoning level is composed by two hierarchical layers. The top one contains an inference engine which handle problem solving strategy. This engine has access to all sets of information and evidences concerning the problem and can operate or activate all VEX's machinery. More precisely, is the interpreter of a *language* in which the developers may write a *problem solving or knowledge processing program* using metaknowledge describing the main problem solving steps, the subtasks to be solved in them, the way in which the partial solutions obtained must be linked, how partial or final conclusions should be elaborated, etc. In other words, this engine is programmable so as to give knowledge engineers the possibility to express their problem solving views or conceptual models about the application to be developed, in a flexible way in order to address different application domains. Although developers may use VEX as a whole like a single tool at this level by supplying a program to this upper engine, this is not obligatory as there is a default program.

The *specific* or lower reasoning level is composed by four inference engines according to different reasoning models or paradigms. They can be mixed in some extent and jointly used when solving parts of a given problem. It is in this place where much of VEX's hybrid philosophy takes place. When addressing knowledge engineering applications it is usual to face big problems in which different

hierarchically related parts or stages can be recognized. Very seldomly a single kind of uncertainty reasoning, knowledge representation, etc. fits all parts or stages of the problem, or expert's behaviour. What is often found is that some part of the problem is better described by a given model, another by another model and so on. For this reason in VEX there is a "*repertoire*" of specific reasoning tools (in no way exhaustive), which allows the knowledge engineer describe each part of his problem by the model he think is most appropriate for it according to his views and experience. He is not forced to use a single reasoning model as there is a representative *offer* of models at his disposal. Four such models are implemented as individual inference engines: A rule-based, belief factor, compositional inference engine, a probabilistic engine working with multiply connected causal (bayesian) networks, a fuzzy logics rule-based engine, and a case-based reasoning engine. There a fifth model, a connectionistic one, in the form of feed-forward neural networks but it has been considered more an internal tool than an inference engine in the logical sense. It is included in the set of Internal Tools described below.

The rule-based compositional model is a variant of the classical MYCIN-PROSPECTOR's, consisting in rules of the form  $A \rightarrow B (\omega)$  ( $A$  = antecedent,  $B$  = conclusion,  $\omega$  = weight). The antecedent is an open logical formula made from propositional variables and logical connectives, and the succedent is a proposition. The weight express the *degree of belief* about  $B$  if  $A$  is known to be present, so it is in fact a conditional belief. A set of combining functions [Hájek 1985] plus a backward, a forward, or a combined inference engine, propagate the evidences from a consulted case through the set of rules. Extensive theoretical studies have been made in the AI community, making clear the kind of uncertainty notion that such weights represents, their defects, and their relation with other uncertainty notions [Hájek, Havránek, Jiroušek 1992], [Hájek-Valdés 1990, 1991, 1994], [Valdés 1987].

VEX-93 include some tools for handle the problem of dependencies among premisses like an optional Möbius transform of this kind of knowledge basis (not present in most systems). VEX-93 actually works with a more general variant of the original engine called *functional*. In it, rule components are not only logical propositions but *functions* in general (numerical, logical etc.) and whether a particular function value should be taking into account when evaluating a rule is definable, so that some function calls may produce only side effects without affecting the application of combining functions on uncertainties. Besides, a variety of mechanisms like contextual links attached to propositions or rules, and definable prior and posterior actions, allows the knowledge engineer to exert considerable control deep inside the inference engine.

The probabilistic inference engine present in VEX-93 assume expert's beliefs in the form of conditional probabilities in their strict mathematical sense, as coming from a joint distribution representable by a graphical model derived from a causal network (single or multiply connected), which express the dependence structure. The original directed graph expressing expert's views about independencies is finally transformed into a hypertree and the given conditional probabilities used for computing the apriori/aposteriori distributions of hypertree components using the method of local computations [Lauritzen and Spiegelhalter 1988]. VEX-93's probabilistic engine is inherited from the GRAMUR project [Valdés 1992].

The fuzzy inference engine deals with rules of the form  $X \text{ is } A \rightarrow Y \text{ is } B (f)$  where  $A$  and  $B$  are fuzzy properties (old, tall, etc.)  $X$ ,  $Y$  are variables and  $f$  a  $[0,1]$  real value assigned to the relation (used compositionally). This number express the degree in which the variable  $Y$  is the property  $B$ , given that variable  $X$  is known to be  $A$ . More generally, the antecedent is composed by several such fuzzy propositions linked by logical connectives. In VEX-93 fuzzy values are interpreted as the *truth value* of a *vague statement*. There is a repertoire of 20 T-norms and T-conorms for adscribing semantics to logical connectives and rule weights, as well as for parametrize each part of the compositional scheme for processing and aggregating rules. The output from a fuzzy knowledge base may be either a collection of fuzzy sets or a set of numeric variables after defuzzification [Zimmermann 1992], [Klir, Folger 1993].

In case-based reasoning (CBR) problem solving is based on the principle of using previously acquired or learned experience. In many situations humans do not go back to first principles (laws, axioms, or causal relations) to construct correct solutions to particular problems or to make correct decisions. Instead, similar situations are remembered and confronted with others known from the past, by exploiting similarities between the actual problem and user's past experiences. CBR models have been considered in VEX-93 modestly in this first stage, as only a dynamic memory was included [Riesbeck Shank 1989],[Kolodner 1993]. Therefore adaptation and automatic indexing have not been considered for this version. Cases are frame-like represented and the engine acts as a *reminder*. As with the other tools, this case memory can be used either independently or in combination with other inference engines.

There are in AI other models for uncertainty and vagueness, as well as other reasoning models (Dempster-Shafer theory, possibilistic logics, (non monotonic, truth maintenance, etc.). They were not included in this first version of VEX-93.

Finally observe that each of the considered engines could be used independently as the kernel of an expert system shell (as is usually the case), but in VEX-93 they are single functions used as building blocks as part of a *knowledge program* used by the strategic inference engine.

### 3.3 Internal/External Tools

This module contains resources which the inference engines may use during their operation and cover tools, mainly for data analysis. There are procedures for data base managing and querying (at present only for relational data basis in the form of DBASE III and ASCII files), procedures for numeric classification, pattern recognition, ordination methods (principal components, non-linear mapping), factor analysis, etc. There are also neural network tools (for feed-forward models with extra tools for separate backpropagation training), formula evaluators, a module for performing fuzzy arithmetics and routines for different operations on graphs. These tools can be used from different places within VEX93 and they can be mixed in some extent. Also what they compute is available at different parts of the reasoning process, and there is an information exchange among the inference engines and the peripheral tools (internals and also user-supplied).

In VEX-93 v1.0, the set of internal tools is rather numerically oriented and accounts for most of the two lower vertices of VEX-93's conceptual triangle from fig.1. Again, only a *representative* set of tools was chosen. In this respect the role of the "Other Models" input source (c.f. fig 2) achieves its importance. As mentioned, the knowledge engineer may introduce *other models* in the form of *external programs* with which VEX-93 can *communicate*. These external models may describe specific aspects of the problem (e.g: a set of differential equations describing a part of process) and these models, shaped in the form of external computer programs, can be "plugged" into VEX-93 and used by the inference engines exactly in the same way as VEX-93's internal resources. Since there is a simple communication protocol format for exchanging information, almost any computer system may be used as an external tool for VEX-93. For example, deterministic or stochastic models for simulation, other statistical or data analytic methods, and even other AI models or inference engines.

## 4 Knowledge representation

Several knowledge representations can be used as there might be different reasoning models, and abstraction levels. Since the amount of knowledge is often huge, questions concerning its structure are important. Two main representation forms are defined: General and Specific. The former describes

problem's structure as a collection of *domains* (parts, subproblems, units, etc) composed by a set of *partitions* (subproblems). There might be links between domains and in general they are nodes of a directed graph (similarly with partitions within a domain). Knowledge corresponding to a partition is what is considered in VEX-93 as a *knowledge base*, and a *problem* for VEX-93 is composed by a *set of domains* whose relations are given by a graph and which communicate among them through a *blackboard* (fig.3). For example, cardiovascular diseases can be considered as a domain which in turn is composed by families of diseases like coronary, ventricular, etc. The simplest problem for VEX-93 is one containing only one domain composed by one partition.

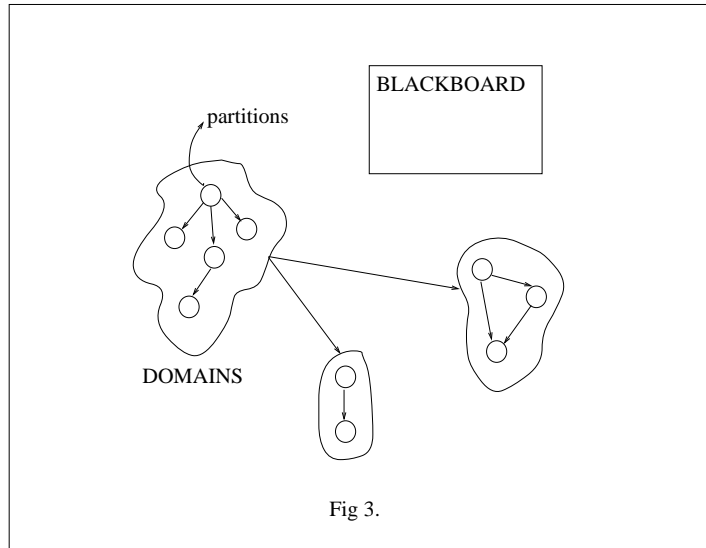


Fig 3.

In this way the general knowledge base or *problem* is given by a *domain graph*, representing problem's structure of tasks and/or meaningful parts and their relations. Since the same basic knowledge can be described at a higher level in different ways, different expert's behaviors can be simulated.

At the specific level of representation, knowledge can have the form of logical functions (propositions are particular cases), production rules (with certainty factors, or fuzzy rules), metarules, context links, causal networks, graphs, fuzzy sets, linguistic variables, neural networks, taxonomies, equivalencies, formal models in the form of external processes and metaknowledge, etc. The dynamic memory for cases uses a frame-like representation using logical propositions and variables from problem's knowledge base (the set of all basis composing the partitions of all domains).

Another important issue is that of *elaborating conclusions*. In VEX-93 it is possible to introduce expertise about *what* is a conclusion of a consultation in the particular field worked out [Valdés J.J, et.al. 1989]. For example, in medical diagnosis usually a conclusion is understood as a *complex statement* composed by some parts: one stating the proposed diagnosis, another prescribing therapies, and a final with observations, or remarks to the patient. In each field, experts conceive conclusions differently and in VEX-93 this specific kind of knowledge is representable. There are tools for elaborate what the inferences engines have found in order to construct outputs in the style of human experts.

## 5 Auxiliary tools

Knowledge basis in VEX-93 may be very big involving many files of different kinds (knowledge basis themselves, data basis, external programs, images, data files, etc.). Thus there are several off-



line tools for compile and handle them. Moreover there are on-line explanation and tracing facilities for debugging and tuning the knowledge basis, plus other facilities.

## 6 Control

There are many ways in which knowledge processing can be controlled in VEX-93. The most important one is that done at the strategic level (where the knowledge processing program is used). Also at the lower reasoning level, specific knowledge representation forms exerts implicit or explicit control too. This is a too broad subject and can't be detailed here.

## 7 Man-Machine interaction

VEX-93 may be used in both batch and interactive mode. Besides usual windows and pop-down menus, there are also capabilities for *voice* interaction. There are tools for elaborate statements and paragraphs which can be given as output in either text or voice forms either during the inference process and/or when elaborating conclusions. The use of external programs allows developers to incorporate not only other reasoning or mathematical models, but also externally designed front-end user interfaces, complementing or even replacing VEX-93's resources.

## 8 Software - Hardware comments

VEX-93 was programmed in C, and conceived originally in dual versions for both Unix and Dos (Windows) operating systems. Non scientific reasons forced a concentration on a quickly developed version 1.0 only for PC's based on 386 or higher processor, running Windows 3.1. This version has been in operation since 1993.

VEX-93's v1.0 overall size is over one million and half lines of C code sorted in different programs, modules, and static and dynamic libraries. It requires about 10 MB hard disk space but due to its modular architecture it is by far not necessary to have the whole system installed.

## 9 An example in the field of medicine

SEEA is a decision support system for the domain of general medicine at practitioner's level and is integrated as a module into the SIAS system for health care management. SEEA is been developed as a cuban-spanish project and uses a part of VEX93's machinery. Its general knowledge base is still under development but about 85 % of the whole has been engineered already and is under testing. At its present state SEEA is composed by about 13 domains corresponding to the main families of diseases SEEA is mainly a consultation aid for diagnosis covering about several hundred diseases considered under the norms of the European Community. The size of problem's base is big and only speaking about production rules, they sum up to several thousand. SEEA will be operative at the end of 1996 and will be embedded into the SIAS system, a large integrated system for health care and sanitary management.

## 10 Final considerations

VEX-93 is an experimental system and a research project in knowledge engineering trying to integrate into a unique platform some representative reasoning models with other problem solving and analytical tools from other fields of mathematics and information processing. This makes VEX93 potentially useful in many application domains by combining AI with other information processing techniques. The flexible architecture of VEX93 can be used for designing a wide variety of applications ranging from pure numerical data analyzers to sophisticated processors and integrators of heterogeneous information. Some examples are smart monitoring systems, with a programmable degree of autonomy, analyzers of dynamic data basis, decision support systems integrating different reasoning schemes (mixing mathematical with heuristic models), etc.

### References

1. Hájek P., Zvárová J., Esteva F.: MUM - state of art. Internal report of the MUM Project. Third Framework Programme of European Community Activities in the Field of research and Technological Development. RTD Programme "Telematic Systems of General Interest". Sub-Programme: AIM, September 1994.
2. Hájek P.: Combining Functions for Certainty Factors in Mycin-like expert systems. *Int. Journal of Man-Machine Studies*, Vol.22, pp.59-67, 1985.
3. Hájek P., Havránek T., Jiroušek R.: *Uncertain Information Processing in Expert Systems*. CRC Press, Boca Ratón, Florida, 1992.
4. Hájek P., Valdés J.J.: Algebraic foundations of uncertainty processing in rule-based expert systems (group theoretical approach), *Computers and Artificial Intelligence* 9, 1990, pp.325-334.
5. Hájek P., Valdés J.J.: Generalized algebraic foundations of uncertainty processing in rule-based expert systems (dempsteroids). *Computers and Artificial Intelligence* 10, 1991, pp.29-42.
6. Hájek P., Valdés J.J.: An analysis of MYCIN-like expert systems. *Mathware & Soft Computing* 2, 1, 1994, pp.45-68.
7. Klir G., Folger T.: *Fuzzy sets, uncertainty and information*. Prentice Hall Int. Editions 1988, 355 pp.
8. Kolodner J.: *Case-based reasoning*. Morgan Kaufmann Pub. Co. 1993, 688 pp.
9. Lauritzen S.L., Spiegelhalter D.J.: Local computation with probabilities in graphical structures and their applications to expert systems. *Jour. Royal Stat. Soc. B*, Vol 50, No. 2, 1988.
10. Valdés J.J.: *Logical and algebraic foundations of uncertainty processing in rule-based expert systems of artificial intelligence*. PhD Thesis Inst. of Mathematics, Academy of Sciences of Czechoslovakia, Prague, 1987.
11. Valdés J.J., de la Cruz A., Pérez A., Jocić E., Balsa J., Rodríguez A. : *Ingeniería de conocimiento en el medio ambiente del ARIES*. (Knowledge engineering in the ARIES environment.) Higher Polytechnical Institute José Antonio Echevarría press. 1989, 200 pp, La Habana (in spanish).
12. Valdés J.J.: An integrated environment for uncertainty processing using graphical models. *Proceedings IPMU'92 Int.Conference on Information Processing and Managing Uncertainty*, Palma de Mallorca, 1992, pp.537-540.
13. Valdés J.J.: *Development tools for intelligent systems with hybrid problem solving techniques*. Project submitted to the Alexander von Humboldt Foundation (Bonn, Germany), 1992.
14. Zimmermann H.J.: *Fuzzy set theory and its applications*. Kluwer Academic Publishers 1992, 399 pp.