CHAPTER 6.

TRANSCRIPTION TO THE EXPERT SYSTEM

6.1. CHOICE OF A REPRESENTATIVE PROBLEM

6.1.1. Presentation of the problem

According to the standard methodology for developing an expert system (fig. No. 3.4), a representative problem subset has to be selected right from the start in order to support the development of an initial prototype and to limit the knowledge that should be collected. Knowledge acquisition and its analysis are supposed to be prerequisite steps of the final transcription to an expert system. Yet, the elicitation of the characteristics and the structure of knowledge is the source as well as the result of the transcription exercise and therefore, the knowledge acquired was analysed simultaneously with the trials of transcription to the expert system. The formal logic required by expert systems forced to submit the expert's diagnosis process to a rigorous dissection into logical inferences and non-logical statements. Only such analysis of knowledge will decide the technical functionalities required for the transcription.

In order to be representative, the problem subset should contain most of the characteristics of the main problem. There is no point in selecting a subproblem which is less complex than the main problem so that the feasibility of the subproblem would not bring any information on the general feasibility. The important features of our problem are the multi-expertise aspect connected with not necessarily consistent reasoning process, the modification of strategies according to objectives, the spatial perception, the time scenarios and the risk evaluation. The typical functionalities are the link with data base, the communication with the user and the execution of external procedures with transfers of parameters.

6.1.2. Choice of a representative problem subset for the initial prototype

A representative subset can be defined as the one which limits the general problem of diagnosis of a system to the particular diagnosis under one or two objectives in which the number of fields would be limited. Initially, the objectives of labour intensification and productivity of the agricultural system were retained among other possible objectives. Such a problem requires an analysis of different domains, data collection and interpretation, data base connection, procedural attachment and also compromises between objectives and performance criteria or decisions on the relative interest of alternatives in each field as well as system wise.
Since the access to experts and therefore to their expertise has been based on the opportunities rather than on a well thought-out selection, at first a simultaneous transcription of whatever acquired was led until complete knowledge would be available for our problem subset or until another representative problem with complete knowledge could be identified. Until then, the chunks of reasoning process that were identified in each field would be coded independently. But since logical rules used to make compromises or decisions could not be identified (§5.3.2), and because of the nature of the knowledge finally acquired, the initial prototype corresponding to this initial representative subset could not be fully developed. Therefore we had to limit our subproblems to hypothetical problem subsets and coherent limited prototypes were only developed within one field and up to the point where arbitrary decisions are required to issue the final diagnosis.

The PWD approach is one of the approaches, in which more experts were found. This approach is oriented toward the irrigation system structures, and was given more emphasis. This diagnosis is too simple to be characteristic of a real multiexpertise diagnostic in which combined objectives and decisions are required to carry out the diagnosis. Any subexpert system that can be developed has a fragmentary aspect, owing to the number of political decisions that are usually imbedded within the diagnostic process.

6.2. SELECTION OF A SHELL

6.2.1. Characteristic of the software to be used

A feasibility study is a first step to the elaboration of a final output. Therefore it is not so important to have the best inference engine and the right knowledge representation from the very beginning. Even if it would be optimal for our expert system development it might be too long and too time consuming to get this inference Engine (especially if it has to be written from scratch). Any good development shell having most of the basic specificities will give a major share of the answers about how relevant the expert system technique is to your particular field and how feasible the expert system development is.

In addition to experts using many logical rules, they also use categories extensively, typology of irrigation systems and problems. They associate the case under study to a particular relevant class that they have come across during their experience. On recognising that a system (an object) belongs to a particular group (class) because some of its characteristics are similar (pattern matching), the expert infers that this particular system will have similar proprieties to that of the identified class (inheritance mechanisms); therefore, the expert system should have both rules and object representation. The type of pattern matching required must enable to deal with variables (the same pattern matching methods are used by experts on different aspects of the system) which can be achieved only with first order logic.
As indicated in §2.7, experts usually have a two way thinking process: they make hypotheses that they try to verify (backward chaining) and identify facts or values for parameters which enable them to infer some intermediate conclusions (forward chaining). Therefore selected expert systems must have both forward chaining and backward chaining.

Several experts imply that different knowledge bases are used; since different diagnostic options can be chosen, all the knowledge modules will not always be required and it might be handy and faster if a modular structure can be given to the knowledge base.

There are things that experts systematically do before trying to infer any conclusions; this can only be performed by an expert system if a kind of control on the inference engine is provided (metarules, priorities, islands of knowledge).

At times experts make an assumption, then assess its possible consequences and accordingly modify the value initially assumed. Therefore value modifications and backtracking facilities are other characteristic required. They ask questions (convivial user interface) and collect information from existing sources (data base connection) which have to be fed during the reasoning process. Whenever an answer cannot be obtained, experts try to get to the information expected through indirect evaluation methods or provide a likely default value to carry on with the diagnosis; expert systems can do this if default value mechanisms are provided and if the answer "not known" does not halt the inference engine.

Economists, agronomists as well as engineers use computation methods for farm budgets, water requirements, hydraulic balance, losses and discharge distribution in the channels or rainfall analysis; These computations are carried out efficiently with traditional algorithms and to that end, procedural attachment should be added.

To avail of all these features and to deal with them satisfactorily the ideal development tool must be quite efficient. It may not be possible to acquire such a tool at the feasibility level because of the type of hardware configuration required or owing to budgetary considerations. This kind of softwares requires indeed a high working memory and a fast clock speed. They usually are quite expensive and run on even more expensive machines. For any research institution whose main activity is not expert system development, such a constraint is important. At the beginning, Macintosh, Apple Inc. computers were used until a shift to IBM compatible was requested. Necessarily the tool to be purchased had to run on the computers available.
6.2.2. SNOPS

The initial software used by us was a development tool called SNARK; it was developed on Vax station (Dec) in Pascal. This programme could not be implemented on the computers available for this study since the core of the software required more than a Megabyte working memory (which is more than what is available for the Apple Pascal compiler on Macintosh).

Therefore a new engine has been written in Lisp for the Macintosh based on both the main algorithms and concepts of SNARK and OPS5.

The main features of this engine, called SNOPS, are rules, metarules, forward chaining, lists, first order logic, different level of tails, sub-expert systems, procedural attachment in Lisp with any Lisp functions, Lisp user interface, etc... Some diagnostic modules have been developed with this shell, (water balance according to the type of year crop and soil, first overview of the system module,...).

Yet after some time it became clear, that the concept of tank diagnosis requires the object representation (structure and inheritance). De facto, the characterisation of a system by the expert is done from the choice of an initial pattern which is progressively clarified (§ 2.5). It would be easier to represent this thinking process if object and class were available. Besides this, a more user friendly interface would have been much welcome as well as data base connection.

Therefore based on the study of existing expert system shells (fig 3.2) one of them has been selected as the best considering the particular constraints assigned for this research work; consequently we have chosen and purchased a new expert system development shell called Nexpert which appeared more adequate.

6.2.3. NEXPERT

Nexpert from Neuron Data inc. is a shell that works on PC AT under a Microsoft 'Windows' and DOS environment. Nexpert is written in C and is claimed to have most of the features initially identified.

On ATs, Nexpert is a hybrid tool written in C with the libraries specific of any 'Windows' applications. Since 'Windows' programming style is quite particular (over 300 new functions have to be used instead of standard C functions), working with Nexpert requires that this 'Windows' language be learnt as well as standard C. Contrary to classical C programmes, 'Windows' applications are based on an event driven logic. The main programme is a loop...
catching messages on a queue, which translates them into actions. A message is the internal
sign that an event has occurred and that steps must be taken accordingly. Events can be evolved
either by user calls (keyboard or mouse clicking), or by the application (your specific orders) or
'Windows' main programme (dealing with the screen appearance and the operation on
messages). All this provides Nexpert with a ready made user friendly environment but at the
cost of dependence on 'Windows' software.

'Windows' software has been developed under the DOS operating system in the
perspective of the new operating system (OS2) designed for computers working with a 80386
chip which works with 'Windows' Presentation Manager software. Applications developed
under 'Windows' are (to some extent) compatible with those developed under 'Windows'
Presentation Manager environment. But, 'Windows' is highly limited under the MSDOS
operating system. Nexperts inherits most of these limits when running on ATs. No real
multitasking is possible and Nexpert is so slow that it often becomes tedious to wait upon the
expert system programme.

Anyway, with these accepted limitations, Nexpert is an expert system development
shell with both object and rule representations which utilises an incremental compiler. The rules
are triggered on pattern matching. Integrated forward and backward chaining can be performed
in a non monotonic reasoning. Object, classes, properties and different inheritance mechanisms
enrich the knowledge representation. The interface evolved from 'Windows' environment has
graphic knowledge editors as well as knowledge browsers. A custom end user interface can be
developed.

Nexpert's architecture is open; it enables integration with the external programme,
parameter passing, external monitoring, data base queries and multitasking (simulated under
'Windows'). There is a hardware cross compatibility with Macintosh and Vax; meaning that the
rules, the objects and the class can be transferred from one environment to the other, but all calls
based on 'Windows' environment must be adapted to changing devices since 'Windows'
environment specificities are not identical to other devices. Therefore all external access,
monitoring of the expert system from outside, and calls of external procedures cannot be
transferred.

In fact extensive using of Nexpert on PC AT tend to show that if all the interesting
features cited above are possible with Nexpert, most of them require complex and time
consuming programming to be implemented. This cannot be done on a large scale for a
feasibility study.
6.3. TRANSCRIPTION IN EACH DOMAIN

(i) The irrigation system

For example, the different rules that lead to the assessment of the irrigated system are developed in a separate knowledge base, IRRSYS.KB. (Annex4) They are based on a set of their conclusions on a set of parameters or facts describing the resource but do not show how this can be related to the general diagnostic process. As far as the logic in the master diagnosis process is not identified, these different chunks of knowledge cannot be chained up in a general diagnosis process. The external judgements which are the basis of the general diagnostic process have to be left out of the expert system. The organisation of this knowledge base is founded on three rules:

Rule 24
If
there is evidence of resource_diag
And there is evidence of system_diag
And if <modifications>.value is possible
Then irr_sys_diag is confirmed
And execute rep_irr_sys(<modifications>)
do modif_eval modif_eval

Rule 50
If
there is evidence of rainfall_diag
And there is evidence of run_off_diag
And there is evidence of storage_diag
And there is evidence of intakes_diag
And there is evidence of losses_diag
And there is evidence of losses sluice_diag
And there is evidence of ground_wat_diag
Then resource diag is confirmed

Rule 65
If
there is evidence of main_diag
And there is evidence of secondary_diag
And there is evidence of ofd_diag
And there is evidence of general_diag
Then system_diag is confirmed
The meaning of these three main rules is quite straightforward; rule 24 divides the diagnosis of the irrigation system into resource and layout. Rules 50 and 65 divide each of these subgoals into their principal components. Therefore, on launching the expert system, the hypothesis to be suggested is \texttt{irr\_sys\_diag} (meaning make the diagnosis of the irrigation system) and the inference engine starts backward chaining to try to achieve the diagnosis. This backward chaining process lasts until all necessary hypotheses are tested; then the hypotheses \texttt{irr\_sys\_diag} is set to \texttt{TRUE} and the right-hand actions of rule 24 are executed.

The actions on the right-hand side are:
(i) first execute a report on the main findings and
(ii) suggest \texttt{modif\_eval}.

\texttt{irr\_sys\_diag} is achieved through a set of 73 rules (Annex 4) that assess different aspects and identify which \texttt{modifications} could be "possible" (meaning: might be interesting).

\texttt{modifications} is a class of objects consisting of all the potential modifications. The class standard attributes of class are value (possible or not possible), cost, feasibility, interest, priorities, etc.,

When rainfall_diag hypothesis is under test, rainfall data are retrieved from a data file written in WKS (microsoft lotus) format and external procedures are launched to compute some average graphs and histograms to help the user with the statistical analysis. But the interpretation of the graphs is left to the user.

\texttt{modif\_eval} is an intermediate hypothesis of another decision tree (2 rules) that aims at evaluating the values of the attributes of each possible modifications (Annex 4; rule 45). The cost estimate often is the easiest part of it, then the contribution of the modifications to some predefined targets can be set to a numeric value to provide a relative idea of the extent of the contribution. But the real benefits are difficult to quantify in a more precise way. (it is a typical political decision in which multidisciplinary effects are crossed with objectives weighs to yield a subjective judgment) and the conclusions on priorities are not either logically based.

The outcome could be a table of possible modifications with their cost, their contributions to targets driven through rule 66 (Annex 4), but the final conclusions or even
fig No. 6.3: A sample screen of agrotool display
The agronomic diagnosis can be divided into three different types of actions besides interacting with the team:

- **data collection**

- mind simulation of the behaviour of irrigated system under various hypotheses in which, the behaviour is represented by adequate parameters.

- judgments bearing on the selection of preferable alternatives.

The judgment part is not rationally based and cannot be transcribed to expert system rules. The data collection consists in gathering information from an adequate data base and in holding a survey with a questionnaire to obtain those information is still missing. Of these two types of data collection, acquisition of preexisting relevant data from literature could be made automatic provided adequate database were built in. Some sample fields and files of the database are illustrated below. The trial data base that was used is a DBIII data base with informations relevant to crops. Some of the fields have a variable length (those depending on the crop duration) and others have a fixed length.

<table>
<thead>
<tr>
<th>FIELDS</th>
<th>CROP1</th>
<th>CROP2</th>
<th>CROP3</th>
<th>CROP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>crop name</td>
<td>PADDY</td>
<td>GROUNDNUTS</td>
<td>RAGI</td>
<td>SUGARCANE</td>
</tr>
<tr>
<td>crop varieity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total number of decades</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical practices calender/decade</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>labour req(men+women days)/decade</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inputs req(Rs)/decade</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>water requirement/decade</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yield/ha</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>price/t</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>loans</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>suitable soils</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>suitable follow up crop</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>by-product quantity/ha</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>by-product price/t</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

fig No.6.1: Sample empty crop data sheet

Like all data collection programme of similar nature, the data collection based on surveys depends on the communication with the informants as well as on the ability of the expert to interpret what he sees and to elucidate the meaning of what he hears. This cannot be performed automatically; (even Eliza, the expert system that gave the impression of understanding the psychiatric patients was merely based on syntax analysis and not on real
understanding §3.2). The computer working under an expert system is not like a robot; an intermediary is required to communicate between the irrigated system and the expert system since the user interface only communicates between the expert system and the expert system user. Therefore it is not interesting to expand on these aspects of field data collection since they remain as the entire responsibility of human beings.

The third phase, mind simulations of the irrigated system under various hypotheses is not easily amenable to expert reasoning process that could be automated with an expert system development. To serve the purpose of automation of the various phases of the diagnosis a search was held on the possible means for making this phase automatic. It leads us to a proposal for a decision aid tool.

The aim of this software would be to visualise the situation in the system. A set of initial data would enable the user to represent the characteristic of the system that one is interested in (these criteria can be selected from a list accessible under the item criteria of the menu bar). A set of functions that can be accessed by a menu would enable him to induce changes on one parameter in the system and to visualise the consequences of these changes on the selected parameters.

fig No.6.2: Function menus in an agronomic simulation tool
The expert's "mind representation" would be transformed into a computer visualisation of some aspects of the system that can be modified on demand (item criteria in the menu). The screen window would be divided into six smaller windows in which one of the evaluation criteria can be represented. The maps represent the tank and the different plots with colours representing the crops or three dimension histograms representing the water used or the net income obtained on each plot; the histogram illustrates the benefit distribution according to social groups and the graphs show the variation around the year of either labour used and labour opportunity or cash flow (expenses and revenues) or the amount of rainfall water, the water used from the tank, from the wells, by evaporation or transport losses.

This tool enables the user to visualise the evolution of the criteria selected when the crops are changed, the prices are affected, the climatic conditions change, or when the management rules of the irrigation system are changed. The way the changes affect the results are clear except for the management policies which must be clarified to be able to work out the programmes propagating the effects of these changes on the general system.

This simulation tool is a trick to bring into the open air the spatial understanding, the risk assessment and the type of mind representations that may be used by experts to convince themselves of the better quality of a particular solution. This provides a tool with a multicriteria analysis. On getting the reaction of experts to such a tool we might elicitate how their decision making uses the criteria displayed. This part of the diagnosis is therefore much better represented through a traditional simulation model than with an expert system.

(iii) The economic analysis

This case has some similarity with the preceding one. Four types of actions characterise economists' diagnoses: data collection (DC), interaction with team (TI), data analysis (DA) and judgments (J) with regard to the results obtained. Moreover, economists follow rather a fixed sequential approach:
- identify market requirements and price trends, commodity prices (DC),
- give conclusions on relative interest at the state level of various crops now and in future (J); this information is given to the agronomists to assist his decisions regarding cropping patterns (TI),
- compute crop budgets (DA) taking into account the information on agricultural practices received from agronomists (TI), compute both financial and economical values (DA),
- receive from the sociologist the characteristics of the different farm models (TI), compute farm budgets and family yearly income and their variations according to climate and price conditions (DA),
- study loan conditions (DC) and assess the impact on implementations (J),
- receive project proposals (TI) and compute the cost/benefit evaluations for the various options,
with comparisons of economic and financial profits in between the with and without situations. Sensitivity analysis of these results (DA) -This provides an economic picture from which the final conclusions have to be drawn (J).

End of analysis

Out of these different processes, whatever comes under J, DC or TI cannot be formalised in logical rules and what comes under DA (the computations) follow well defined sequential algorithms. Many programmes are available on these topics and therefore they need not be translated into an expert system that would be far less efficient.

(iv) The sociological module

The sociological module is organised around the rule number 18 of the base SOCSYS.KB (Annex 5)

Rule 18

If there is evidence of social_struct
And there is evidence of social_state
And there is evidence of adm_eval
Then Soc_diag is confirmed
And execute reportsoc(@EXE)

Therefore this module is meant to be launched on suggesting the hypothesis soc_diag, even though there is an infinite way to launch the expert system, it may not lead to a lot of information. The process transcribed here, is a three fold analysis of the social context describing the existing structures, the general social state and thirdly, the assessment of the administrations.

For each existing social group, the expert system SOCSYS.KB asks which part of the population belongs to that group, what share of the area do they own and what share of the area do they work on. This leads to a histogram with these three characteristics for each existing group. In the same rules some qualitative appreciation of the group’s political, and economical, and religious powers can be given and the results presented in a kind of Lotus Table.

The same type of question and information is gathered for social groups, economic groups and the religious groups. The typology of the economic group should be transferred to the economic module to enable the economist to work out his model farm budgets.
The second fold of the social diagnosis deals with the presence and type of conflicts, whether the commercial deals are socialised or not (market law is then predominant) and requires that an assessment of labour and traditions according to social group be made. This is supposed to be done by a separate knowledge base.

The state of the social context should be thus described but no conclusion are issued on the way this information can be used. The expert only provides a set of kaleidoscopic information. The knowledge base development had to be limited even though the set of informations that may be asked is very important. And indeed experts do not ask for the whole set of possible parameters that may be found in well known checklists (Bottrall, 1986). Usually they only ask for the information that they require to make out their convictions. They manage to evolve sense from a limited vision of some aspects of the social background and to issue conclusions based on this restricted vision. Since we did not know either how to limit the investigations on social aspects or how to draw conclusions out of them; this part has not been implemented. There is not much interest in transforming the social assessment of the situation in a systematic exhaustive but tiresome checklist of possible indicators that no one will be willing to provide.

The third part of it is the assessment of the existing administration. On trying to verify the goal adm_eval, the engine starts backward chaining until all existing administration within the set of seven alternatives presented has been reviewed. The seven types of administration are external administration, village institutions, command area organisation, farmers unions, farmers cooperatives, temple administration and water user associations.

It is interesting to outline an other drawback of Nexpert development tool; when you present seven boolean expressions in the premises part of the rules, all the rules based on the possible combinations of the values taken by the premises should be explicitly written: $2^7 = 128$; only rules 1, 2 and 3 have been written. This kind of development software has limitations and often makes the transcription of rules more than tedious!
For each administration, rule 4 (case of a rule that induce a loop on itself until all present administration have been reviewed) asks its name, its type of service, its quality of functioning, its system of conflict resolution, its costs and its ability to abide by its rules. Qualitative values can be given and the results presented in an external report.

Again the way that this information can be used to go deeper into the diagnosis could not be formalised in rules.

6.4. The main diagnostic module

The standard methodology to build an expert system specifies that once the knowledge has been collected and analysed, adequate technical solutions should be identified and selected in order to proceed to the transcription of the operating prototype. The attempts made tend to show that expert systems are not adapted to represent the full range of different kinds of knowledge and reasoning strategies used by experts. In fact the most important missing part of the overall diagnosis is the reasoning process that should be used to jump from parameters and objectives to decisions, from a set of multidisciplinary aspects to final conclusions or to identify what is the nature of a system or of a problem which are only known by some criteria. Since we have not been able to elicitate the mechanisms of these processes (either because the decision processes are not clear or because limits between the different categories are too fuzzy), they must be considered as arbitrary for our logical expert system. But these mechanisms and patterns are present in the existing experts' mind and therefore experts know how to perform them. A main programme that would supervise the general diagnosis and leave the user of this expert system to provide these decisions from outside whenever needed. In fact the diagnosis consists of a rather fixed sequence of things and choices which have to be made on each diagnosis and the general programme appears to be a good solution to overcome the lack of knowledge regarding decision making. Such a programme is appealing since it provides a tool for chaining the different fragmentary expert systems and for getting rid of the frustrations issued from the transcription exercise

Attempts were made to develop such a tool in C (Annex 6) that would enforce a chaining on the different sub-expert system under particular objectives given by the user.

The organisation of this supervising diagnostic programme consist in seven subprogrammes which carry out the following activities:

(i) prompt the user to select the leading field (irrigation system, agricultural system, economy or sociology)

(ii) select the subfield priorities according to the objectives

(iii) identify diagnosis framework and constraints bearing on the diagnosis (time, money,...)
(iv) identify the type of system
(v) prompt the user to give the selected reference system for the system (selection of a set of performance criteria or priorities)
(vi) execute the expert system corresponding to the field selected (deep level) and whenever required asking for information from the other fields (low level). (vii) give elements of conclusions in a report form.

A skeleton of the programme that is required to monitor Nexpert from outside and to chain a different knowledge base is given in Annex 6. The type of programming required is quite heavy but once the skeleton is built the development of subroutines is much easier.

The reconnaissance of the system which aims at understanding the overall system and finding out where the main problems are would require that a set of possible problems were identified (a tank with a water resource decreasing over years, a system with increasing population, a system with no maintenance,...) as well as a set of possible structures for the system. (a tank system with sugar cane as a main crop, a tank with two communities, a tank system with no external supply of water,...). Were that the case, then the identification of the type of the system could be worked out. (Experts systems have been developed for bird identification, and they work quite well but in this case; the varieties of birds are well defined and whichever bird does not fit in a predefined category is considered an unknown variety and rejected). But indeed the variety among tank irrigated systems is great (because of the number of possible variations (fig No.5.17-18-19)). And therefore the possible classes can only be grouped into bigger groups under restrictive considerations or objectives.

The execution of the different modules can be launched from inside the supervisor programme, but the decisions on parameters to be changed and conclusions to be made must be prompted to the programme by the user. The final conclusions are the reports issued on the different parameters selected. It is up to the user to select which solution he wants.

Therefore this programme is quite restricted with regard to the flexibility and possible combinations of situations, criteria and strategies that were found in the real situations and it cannot perform any diagnosis in a satisfactory manner. The elementary parts of the diagnosis are quite often straightforward (execution of the four modules) but the initial selection of criteria relevant to the objectives and to the system is the deciding part of the diagnosis which has to be left out. The identification of the constraints is logically evolved from the values of criteria and conclusions are drawn on possible alternatives. Yet the interesting part of the diagnosis does not lie in these intermediary conclusions but rather in the feasibility, cost and interest of the different alternatives. This latter part depends on the links between the different fields and the objectives. It is a subjective resolution evolved through a learning process that cannot be performed by the supervisor programme.
### SOCIAL SYSTEM

**WELFARE**
- Increase welfare
- Time variation of welfare
- Spatial variation of welfare

### ECONOMIC ASPECTS

**WEALTH**
- Increase wealth
- Ratio of economic production
- Productive area

### AGRICULTURAL SYSTEM

**PRODUCTION**
- Increase general production
- Cropping seasons
- Cultivated area

### IRRIGATION SYSTEM

**WATER**
- Water resource
- Irrigation period
- Command area

<table>
<thead>
<tr>
<th>SUBSYSTEM</th>
<th>RESOURCE</th>
<th>QUANTITY</th>
<th>DISTRIBUTION</th>
<th>QUALITY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>total amount</td>
<td>in the area</td>
<td>timeliness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>time period/year</td>
<td>in course of time</td>
<td>adequate amounts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>spreading area/ha</td>
<td>yearwise reliability</td>
<td>nature quality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/decade/inhabitants</td>
<td>seasonwise</td>
<td>cost efficiency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>kind/type or origin</td>
<td>among social groups</td>
<td>losses</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>extra benefits</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUBSYSTEM</th>
<th>IRRIGATION SYSTEM</th>
<th>AGRICULTURAL SYSTEM</th>
<th>ECONOMIC ASPECTS</th>
<th>SOCIAL SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOURCE</td>
<td>WATER</td>
<td>PRODUCTION</td>
<td>WEALTH</td>
<td>WELFARE</td>
</tr>
<tr>
<td>QUANTITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total amount</td>
<td></td>
<td>Increase general production</td>
<td>Increase wealth</td>
<td>increase welfare</td>
</tr>
<tr>
<td>time period/year</td>
<td></td>
<td>Cropping seasons</td>
<td>Ratio of economic production</td>
<td>Time variation of welfare</td>
</tr>
<tr>
<td>spreading area/ha</td>
<td></td>
<td>Cultivated area</td>
<td>Productive area</td>
<td>Spatial variation of welfare</td>
</tr>
<tr>
<td>/decade/inhabitants</td>
<td></td>
<td></td>
<td>/ha</td>
<td>/ha</td>
</tr>
<tr>
<td>kind/type or origin</td>
<td></td>
<td></td>
<td>/decade</td>
<td>/decade</td>
</tr>
<tr>
<td>rain</td>
<td></td>
<td></td>
<td>/inhabitants</td>
<td>/inhabitants</td>
</tr>
<tr>
<td>tank</td>
<td></td>
<td></td>
<td>rice</td>
<td>money</td>
</tr>
<tr>
<td>wells</td>
<td></td>
<td></td>
<td>groundnuts</td>
<td>immobilisations</td>
</tr>
<tr>
<td>intake channels</td>
<td></td>
<td></td>
<td>millets</td>
<td>debt repayment</td>
</tr>
<tr>
<td>head/middle/tail access to water</td>
<td></td>
<td>head/middle/tail production</td>
<td>head/middle/tail economics</td>
<td>head/middle/tail welfare</td>
</tr>
<tr>
<td>long term trend on water resource</td>
<td></td>
<td>LT trend on agric.production</td>
<td>LT trend on econ.production</td>
<td>LT trend on welfare</td>
</tr>
<tr>
<td>climatic fluctuations</td>
<td></td>
<td>climatic impact on production</td>
<td>climatic impact on economy</td>
<td>social effects of bad/good years</td>
</tr>
<tr>
<td>dry seasons</td>
<td></td>
<td>dead seasons</td>
<td>unproductive periods</td>
<td>seasonal crisis and tensions</td>
</tr>
<tr>
<td>various communities' water access</td>
<td></td>
<td>various communities' Ag. prod.</td>
<td>various communities/Eco. prod.</td>
<td>social equity</td>
</tr>
<tr>
<td>timeliness of water delivery</td>
<td></td>
<td>seasonal grain shortages</td>
<td>cash flow problems</td>
<td>conflicts</td>
</tr>
<tr>
<td>enough qty of water</td>
<td></td>
<td>enough qty of Ag production</td>
<td>enough qty of wealth</td>
<td></td>
</tr>
<tr>
<td>chemical quality</td>
<td></td>
<td>crop quality</td>
<td>$, Rs</td>
<td>social costs</td>
</tr>
<tr>
<td>water temperature</td>
<td></td>
<td>pests attacks</td>
<td>cash, added-value</td>
<td>institutional costs</td>
</tr>
<tr>
<td>cost effectiveness of delivery sys.</td>
<td></td>
<td>crop margin</td>
<td>economic/financial interest</td>
<td>losses of power</td>
</tr>
<tr>
<td>tax recovery rate</td>
<td></td>
<td>financial input/financial output</td>
<td>interests/loans</td>
<td>losses of liberty</td>
</tr>
<tr>
<td>losses</td>
<td></td>
<td>storage losses</td>
<td>governmental hidden costs</td>
<td>increase of social disparities</td>
</tr>
<tr>
<td>percolation</td>
<td></td>
<td>losses on the field</td>
<td>shadow prices/financial prices</td>
<td></td>
</tr>
<tr>
<td>evaporation</td>
<td></td>
<td>transport losses</td>
<td>financial products</td>
<td></td>
</tr>
<tr>
<td>leaks</td>
<td></td>
<td>byproducts (for livestock)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>water reuse</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIG No. 6.5: Possible performance criteria**
There is a sort of contradiction in trying to develop a supervisor programme based on algorithms where an expert system has proved inefficient. How could sequential procedures deal with reasoning strategies which are not based on formal logics? The fig No.6.5 gives the set of priorities that have to be selected for each field. The problem here is that generally the priorities cannot be set down in black and white. The diagnosis priority is generally a subtle mixture of technical, productivity, political and humanitarian considerations which cannot be clearly presented by the experts. This must be the human side of the diagnosis business! Therefore picking criteria to give each of them a fixed priority is not likely to appear satisfactory in the end even for the expert who selected them. What should be done is to determine the possible types of objectives (as viewed by expert) and to have an expert system identifying which of the predefined classes of diagnosis this one under study belongs to.

6.5. Interface and system functionalities

One of the positive aspects of the transcription exercise has been the access to the data base and external programme. Up to now, engineers mastering technical aspects of irrigation have written many useful programmes concerning the hydrological cycle, the water available in the tanks and in the aquifers, the consumptive water needs of the crops, the water balances, the fertilising aspects, yields as well as market and farming system economy. Nexpert based expert systems allows to use such programmes as well as to retrieve data from external bases and to transfer parameters from the knowledge base to the programme and from the programme to the knowledge base.

Interactivity with Nexpert allows to get answers and to introduce elements from outside which is a means to introduce arbitrary decision in the reasoning process.

6.6. Interests, hindrances

There are a number of technical limitations in the transcription of a reasoning process due to the particular type of knowledge representation selected in the Nexpert shell. Particularly, since a conclusion is considered as false if no rule pointing to this conclusion can be triggered (Horn's clause), there is an imperative necessity to explicitly give all possible combinatorial rules. This prevents a progressive development of the rule base.

The way the inference engine deals with the strategies and the competitive goals on the agenda are not always satisfactory and seldom clear whenever complex cases are on the agenda. Therefore all the possible reasoning paths have to be identified right from the beginning and the strategies set to ensure a satisfactory behaviour of the inference engine. This kills the myth of an expert system in which all elementary chunks of knowledge are translated into rules and jumbled into the knowledge base.
Another important limit of the Nexpert shell is the fact that a list definition is only valid within the rule in which it has been defined. Since the transmission of lists from one rule to the other is not possible, the progressive selection of elements or hypothesis within a list is not possible if more than one rule is required to proceed to the selection. It is difficult to apply a method that would consist in a limited sub-expert system to different elements successively.

The constraints linked with the shell are numerous but do not seem to be only due to the use of Nexpert; they exist on all shells. This tool enables some expert system development provided the problem and the knowledge can be adequately represented with the above system.

The output of this transcription phase is altogether quite positive. A fully formalised reasoning process can be translated and then performed by an expert tool. Similar knowledge base chunks can be chained in a systematic process and decisions introduced from outside if no rule pre-exists to deal with the choices required. This assumes that a number of limited alternatives is provided in the knowledge base, among which an arbitrary choice is prompted from outside.

Even assuming that a well defined problem has been transcripted into an expert system, it is clear that not all the tasks required for the diagnosis will be performed by the tool. Data collection (and to some extent its analysis and interpretation) is one of the tasks that is to be performed outside of the diagnostic expert tool. This process of data collection depends heavily on human factors; the collected of information is often biased (§2.5). An element given by an informant depends on his perception of the expert asking the question. The answers obtained are not only qualitative, but also subjective; that means that the value given to the answer changes with the knowledge that the expert gets from this informant.

Another way of collecting information and gaining understanding of a system is achieved through spatial recognition. This important information cannot be achieved through a mere expert system tool; other AI tools would be required. Yet it appears that this is an important drawback to the present attempt to automate experts' actions. Image processing is a powerful technique used by experts to characterise the system under analysis. The quantity of information accumulated and processed by the expert's brain in a simple glance at the system is important, and this process is faster, far more flexible and far less tedious than any possible set of questions.

The nature of the diagnosis as identified (performing a sequence of expert diagnoses independently in a sequence which is neither logically driven nor systematic) requires external monitoring. But anyway the arbitrary and political choice have to be made to jump from the rule based diagnosis to the final diagnosis. The main tutor programme for which a development skeleton has been elaborated tries to overcome this difficulty. It attempts to chain external
decisions and choices with the inmutable rule based expert systems. Yet to some extend, this idea is a myth rising from the frustration evolved by the nature of software and the eagerness of coming out with a final diagnosis tool at any rate. On general grounds, such a monitoring programme should be even worse than expert systems since it tries to hide the required decisions that have to be made behind a fixed sequence of actions. These decisions that we have called political decisions are based on many elements and considerations that should be thoroughly studied before rules can be identified and applied. The time required is far beyond that was planned for this research work. And in the present stage of knowledge these decisions have to be considered as non logically inferred from identified fixed political rules.