APPENDIX 2 : LONG TERM RESEARCH CONSIDERATIONS

In this appendix a detailed design is presented that concentrates on the specific problem of relating assessment methodologies to overall fisheries management and the data requirements that arise from this. The proposed goal-data design addresses many of the issues that have been raised from reviews of the completed products in a very rigorous and detailed manner. However the implementation of such a detailed design into a practical product would require the investment of many more resources than were available for the present study. The expert systems that have been produced though simpler than the design presented here, have answered real needs in the time frame that was available and many of the principles embodied in these products would also be applicable to any future work on this design proposal.

DEFINITION OF MANAGEMENT GOALS

Accepting that available assessment methodologies are to be used and that any given one is only appropriate to particular kinds of management strategy, it is vital that the aims of the fisheries managers can be formally defined to a particular goal. Then the correct form of assessments can be instigated to produce the information that is necessary to make the right decisions in progressing towards the chosen goal.

It must also be borne in mind that there is often a further constraint, being imposed from the 'bottom end' of the process, namely what data is available. Where resources and the administrative framework limit the kind, quality and quantity of data available this will be carried through to the kinds of assessments that are feasible and therefore the forms of management that can be envisaged.

The ability to clearly identify goals and then place these goals within the perspective of available assessments and their data requirements depends a great deal on the knowledge and experience of the fisheries managers and scientists involved. If this expertise is not available it is possible to collect data for the sake of data collection. These data are often not suitable and produce information which is of no use towards management requirements even if the later are known.

The examples of goal definition used by Shepherd (in Fish Population Dynamics ed by Gulland 1988) are taken as the starting point. The principles he develops serve as a strong framework which could be adapted for the management of non temperate species and fisheries. For example the requirements for a quick assessment of sustainable yield for a newly developing fishery on a flatfish, would be quite different from those for a forecast of the catch per unit of (fishing) effort for August next year in a fishery on anchovies. They would depend on the biology of the species concerned, the place, the time scale, and not least the purpose of the assessment. This would be affected by the social and political context, as this influences the kind of management envisaged, if any, this in turn affects the type of calculation which would be of interest.

The most fundamental decision to reach is whether the goal aimed at is long term or short term. A clear example of a management goal would be calculation of the minimum stock size allowable that ensured, within an acceptable level of risk, that a stock did not collapse. This would be classified as a long term goal. Another example would be the calculation of catch quotas by managers to allocate allowable catches to vessels and fleets fishing in their waters. This would require a short term forecast of the likely catch at a selected level of effort.

The other main goal criterion that needs to be defined is whether the exploitation of the fish is to be held constant or allowed to change. An example of constant exploitation would be forecasting of catch at various levels of fishing mortality. An example of changed exploitation pattern would be where mesh sizes are altered. The reason this distinction has to be made is that if exploitation patterns are changed then only size structured models should be used since the alternative 'production models' do not recognize age or size but simply treat the population as a single entity, and therefore cannot be used to assess effects of management options on the internal structure of this population.

There are a whole range of assessments that can be classified according to this framework:

SELECTION	TERM		
		LONG	SHORT
	CONSTANT		
	CHANGED		

and these are summarized along with their data requirements in Table 1.

Further details on how to classify management goals within this framework and choose the best suited assessment can be obtained from pages 39 - 46 of Fish Population Dynamics (Gulland (Ed) 1988)

The critical point is that this process of formalizing goals in order to match the correct assessments that they need is an area which is well suited to an expert system solution and probably one that follows the classic 'consultation' approach between the fisheries manager and the expert system. Care needs to be taken in choosing the form of knowledge representation (as production rules, in predicate logic, semantic networks, or as frame based systems) used in constructing the knowledge base. This will involve applying the various techniques of 'knowledge engineering, deciding on how such an expert system is to be implemented i.e. via a language, toolkit or shell and finally the particular development software to be used. The choice of software is limited by the first three, even so there are still a range of products available for each area. The useful limits of an expert system must be designated to decide which aspects of the overall management are better performed by a database management system (DBMS) or procedural algorithms. These aspects are considered in more detail in the following sections.

DEFINITION OF GOAL REQUIREMENTS

It is clear from Table 1 that once a given assessment procedure is settled upon, determining its information requirements can be performed as a simple routine procedure,i.e. looking up the table and will not require any of the inferential powers offered by an expert system. However where some of these type of judgements are clouded, the ease and speed with which additional peripheral facilities offered by many expert systems i.e., user interfaces, data validation and other standard routines can be developed, make them a more useful tool compared to traditional software.

With this proposed method for determining the assessments appropriate to a particular goal (expert system) and finding the information requirements of those assessments (some form of table look procedure) it is important to consider the data viewpoint.

Table 2 summarizes the relationship between the information sought and the data required to derive it, this being done through various standard fisheries calculations. The data requirements can again be determined by an analogous look up procedure.

A hierarchy now exists eg :

For example if an expert system has decided that to achieve the management goals the best kind of assessment is of a short term constant selection type requiring an assessment of general catch verses effort. Then the pattern of queries in the database would then be as follows :

Look up 1 determines that for this assessment we need to know current stock size and recruitment.

Look up 2a determines that to calculate current stock size we need research vessel fishing data that includes catch at age data and/or egg/larvae data.

Look up 2b determines that to calculate imminent recruitment we only need data from a research vessel fishing survey and catch at age data if available.

However, this hierarchical view assumes unlimited resources. In reality limitation will be imposed from below by the data, because :

(i) some may already be available so should be put to good use if possible.

(ii) it may only be feasible to implement collection of certain types of data.

A working system must be able to take such limitation's into account to produce the best acceptable match between what's desired in terms of management and what's feasible in data availability. The following system is proposed to accomplish this.

PROPOSED SYSTEM

Figure 1 represents the proposed design for an entire fisheries/management system. Processes labelled Expert Systems are those that would be particulary suitable or have to be implemented via truly expert modules. There are three main areas within the design:

(i) goal definition, assessment selection, data matching.

(ii) assessment.

(iii) comparison of results with goal.

Given this overall design, key areas need to be identified for the next design stage. The three most important are :

(1) EXPERT CAPABILITIES : the formal goal identifying module, which would need to be "expert". Decision trees and knowledge bases will need to be constructed.

(2) MATCHING CAPABILITIES : the design of lookup and lookdown and data computer modules.

(3) SYSTEM DATA MODEL :

The three can be looked at in conjunction to begin with. This is useful in deciding how to recognise where the boundaries of what needs to be truly 'expert' lie.

The entire process could be constructed from the management goal at the head of the hierarchy all the way down through Tables 1 and 2 until each and every data requirement had been specified by using nothing but "knowledge based structure". However this would be a very inefficient design requiring huge sections of a knowledge base each time used, the majority of which would be purely routine and not involve the true process of inference at all. The only possible justification is that many of expert system shells offer a certainty or probability facility which can weight each rule. This facility could be put to good use with the degrees of dependence and utility, attached to information and data respectively, as proposed by Sheperd (Gulland 1988).

A better design could be achieved by combining Tables 1 and 2 into a unified relational data model. This is illustrated in Figure 2.

Table 1. Dependence of Assessments on Information (from Shepherd in Fish Population Dynamics Gulland 1988)

Assessment type	Current stock size	Current fishing mortality	Imminent recruitment	Natural mortality	Exploitation pattern	Long-term recruitment (SRR)
SHORT-TERM						
Constant selection						
Catch rate/SQC/stock size	***[1]	*	***	–	–	–
General catch versus effort	***[3]	**	***	*	–	–
Changed selection						
Catch rate/SQC/stock size	***	*	***	*	***	–
General catch versus effort	***	**	***	*	***	–
LONG-TERM						
Constant selection						
MSY, etc. (via *S/P* model)	*	*	–	–	–	***
MSY, etc. (via YPR, SRR, etc.)	–	*	–	*	**	***
Target *F* (via YPR)	–	*	–	**	**	–
Target *F* (via BPR)	*	*	–	**	**	***
Changed selection[2]						
MSY, etc.	–	*	–	*	***	***
Target *F* (via YPR)	–	*	–	**	***	–
Target *F* (via BPR)	*	*	–	**	***	***

–, no dependence; *, weak dependence; **, medium dependence; ***, strong dependence.
[1] Recent data on quantity of interest is adequate.
[2] Only possible via YPR/SRR approach.
[3] Recent catch data are adequate.

Table 2. Information Content of Data (from Shepherd in Fish Population Dynamics Gulland 1988)

Data	Current stock size	Current fishing mortality	Imminent recruitment	Natural mortality	Exploitation pattern	Long-term recruitment (SRR)
Aggregated catch data	*	*	*	—	—	*
. . . + age/size composition	*	*	**	—	**	** [1]
Effort data	—	** [2]	—	—	—	—
Catch *and* effort (CPUE)	** [2]	** [2]	—	—	—	* [4]
. . . + age/size composition	** [2]	** [2]	**	*	***	***
R/V fishing survey[3]	** [8]	**	***	—	*	*
. . . + catch-at-age data	***	***	***	* [6]	***	***
Egg/larvae survey	***	—	—	—	—	—
Acoustic survey	** [7]	* [5]	?	—	—	—
Tagging (occasional)	*	*	—	—	—	—
Tagging (regular)	**	**	—	*	*	—

–, no utility; *, low utility; **, medium utility; ***, high utility.

[1] Using VPA, etc.

[2] Only medium because catchability may vary.

[3] Age/size composition assumed available.

[4] Using stock/production models.

[5] Relative effort may be deduced from Z.

[6] Provided catchability does not vary with age.

[7] Only relative unless retrospective calibration possible or TS known reliably.

[8] Relative only.

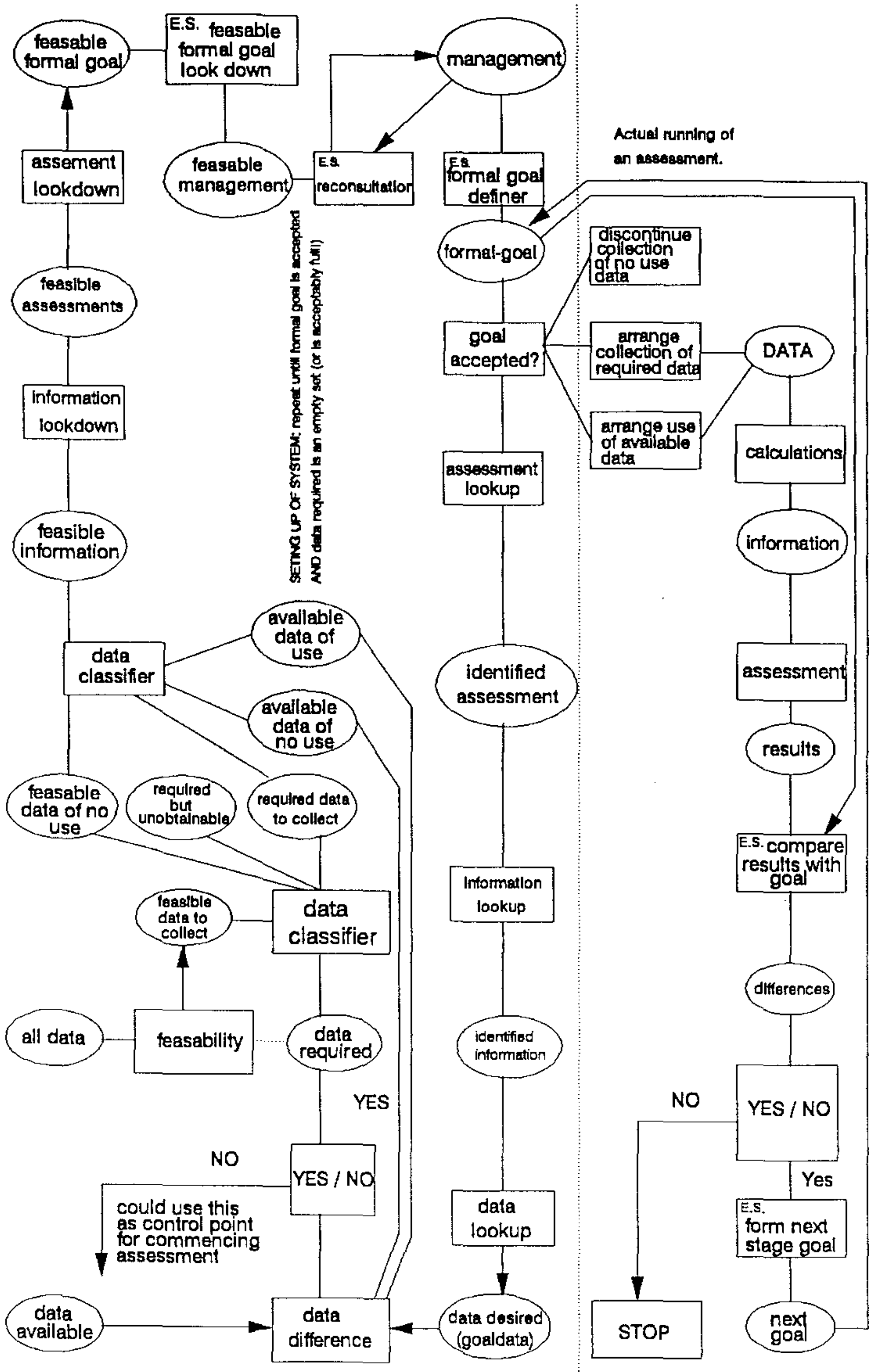


Figure 1. Data Flow and Functional Decomposition for Fisheries Management System

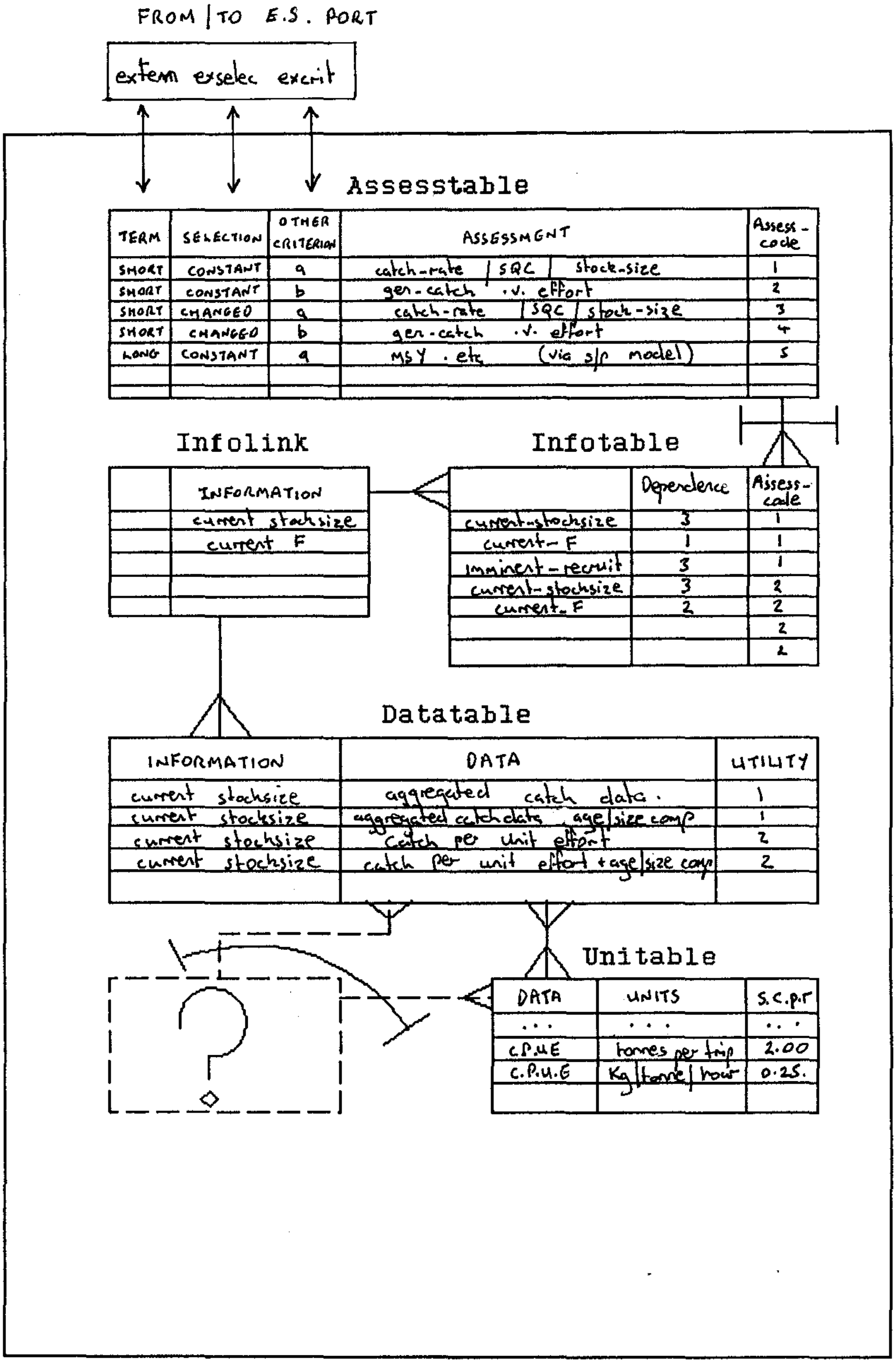


Figure 2. Entity Relationship Attribute Analysis for Fisheries Management Data Model.

The considerable advantages gained from this approach are :

(i) much more efficient searching for information and data querying.

(ii) any future changes and additions to the fishery management system by way of : goal criteria; assessment and models; information derived; new types of data utilized for new type assessments or existing ones, can then be simply incorporated into the system by a straightforward update of this database.

(iii) The actual measured field data itself would be best handled by a relational database since much of the data would be shared and used on a regional as well as local basis. Furthermore, the use of a relational database could facilitate rapid development of the wide range of applications required. The later point is tempered by decreased performance of such general software as compared to applications written in a third generation language with dedicated information. Also, several of the fundamental models used to calculate growth parameters from length composition data, which then use these parameters to make short term forecasts of catch rates have already been written in turbo pascal.

In order to achieve this the first Table in Figure 2, has to be flattened because of its non 'first normal' form. There is still a problem of uniquely identifying the records because of the multiple use of the same assessments in different contexts. This can be overcome by labelling them with an assessment_context_code. This code then has a one to many relation with the second table of the assess_context_code key field to the assess_context_code foreign field in the related table 'infotable'.

Major design difficulties are encountered when incorporating the dependence (and later on from Table 2 the 'utility') attributes, since the way these are combined in Tables 1 and 2 with the information and data is complex. The main problems are sorting out identification dependencies and correctly normalizing the tables. Further work needs to be carried out in this area. The data model presented is a first 'cut'.

The reason for incorporating dependencies and utility as a numeric field is that weightings can then be attached to query evaluations to decide which are the best assessments to use and which is the most important data to collect (quality and quantity). The 'infotable' attribute 'information' has a many to many relation to the 'datatable' 'information' attribute. Removing this is one of the more problematic areas. Introducing a new identity type that has associative identification dependence on 'infotable' and 'datatable' does not seem possible because of 'infotable's' characteristic identification dependence on 'assesstable'. The introduction of table 'infolink' seems to provide a solution but needs to be checked to see if it is conceptually and logically correct.

The data model can be further extended to 'unitable' which differentiates between the various units that can be used for the same type of data. e.g. catch per unit effort (CPUE) might be in (i) tonnes per boat trip, or much more detailed as (ii) k.g. per vessel-tonne-hour. Such refinements of the data can in suitable circumstances lead to improved assessments but the cost of obtaining such refined data is likely to be greater. Therefore a standard cost per record (scpr) field is attached to each of the 'unitable' records which can then be used in costing an assessment on its data requirements. Again the many to many relationship needs to be ironed out.

EXPERT CAPABILITIES

Production rule symbolism is used to illustrate this phase. The top down route progresses from the management goal as far as the definition of the formal goals. From there on down the DBMS outlined previously takes over to derive the suitable assessment type, its information requirements and the data requirements by way of query evaluation.

In the future it may be possible to have as illustrated in Fig 2 an expert 'bottom up' route which can determine what feasible goals there are from data already (or potentially) available.

An alternative future approach could be the use of a 'reconsultation module' (see Fig 2) that will allow management to balance their goals and the data they are willing to collect in order to come to a working compromise. This reconsultation facility would have as a large part of its input both the calculated relative certainty factors and costs for each management option worked out respectively from the utility/dependence and standard cost per record attributes of the database. Such a facility could be implemented with a specially designed user interface that can make the goal definition process very flexible with good feedback to the user of cost/certainty factors for each combination of goals and data.

Goal Definition Top Down

In order to interface correctly with the DBMS the end result of the expert consultation needs to be defined values for the 'term', 'selection' and 'othercriteria' attributes. Other criteria is a single attribute which will suffice to distinguish a unique assessment for this theoretical approach. It will have to be extended to several attributes in a practical system. The principle would be the same, and the basic decision would be as follows :

At the simplest level one could have three variables, three questions and <u>NO</u> rules.

Variables :

exterm
exselec
excrit

```
Is your goal a short term or long term one ?  -> exterm
Is your selection to be kept or changed ?     -> exselec
Is the other criteria a b c or d ?            -> excrit
```

A knowledge base would not be necessary, a database query could be performed along the lines of (using SQL in this example) :

```
CREATE VIEW Goaldata AS
  SELECT data from datatable
  WHERE datatable.utility >          = 2
  AND datatable.information          = infolink.information
  AND infolink information           = infotable.information
  AND infotable dependence           >= 2
  AND infotable assess code          = assesstable. assess-code
  AND assesstable.other_criteria     = excrit
  AND assesstable.selection          = exselec
  AND infotable.term                 = exterm
```

* (Note the search to evaluate the query might be much more effective if the select clause is completely inverted.)

In the simplest case available this view would be displayed as the data requirement for that goal.

To execute the assessment each of the goaldata view attributes could be interpreted as table names which could have agreed standard structures into which any measurement data would be loaded. The assessment models which are written with the table structures in mind then run accessing the actual data in the final structures to perform their calculations and produce the results.

This example has not been devolved to the lowest table 'unitable' which is what would occur in practise so as not to cloud the principle being illustrated here.

If desired there could be two tiers of assessment the lower tier calculating the 'information' which could be stored in tables, again which would be accessed by a top tier of assessments to produce the final results. Such a two tier arrangement would allow more flexibility in interpreting the data.

The subject of coping with data availability and how this interacts with goal definition is examined in the following section on data matching.

Why the Need for an Expert System ?

The answer is that an expert system is needed to help in defining the formal goal variables (exterm, exselec, excrit) correctly so they can be used for querying the database. In the above example their values are simply entered ie. the user is assumed to be expert enough to understand exactly how to formulate their formal goals from their management aims, which they also clearly know.

In reality a user would often need help to :

(i) clearly define their management aims.

(ii) interpret these properly into the correct formal goals required for the choice of a suitable assessment.

This is where the expert system will be truly useful. A brief example of a consultation is given written as production rules (Page 54). It only goes as far as determining whether the formal goal is long term or short term and whether selection is constant or changed. For the purposes of this example we will say that it is all needed to settle on an assessment ie. other_criterion is not needed. There is a body of knowledge which consists of a hierarchy of rules and facts which are used to determine the values of the formal goal variables, excrit, exselec, exterm.

The user interface is composed of a series of intermediate goals which have to be evaluated first. These are then used to evaluate the end goal, exterm. If the user does not know the answer to one of the immediate goals the system steps back yet another level to ask the user a further set of intermediates which can then be used to evaluate the answer to the one the user was unsure of. Such structures cannot be represented by simple trees, since these become fragmented so separate sub trees have to be evaluated the results of which are passed back to the 'parent' tree, as illustrated in Fig 3.

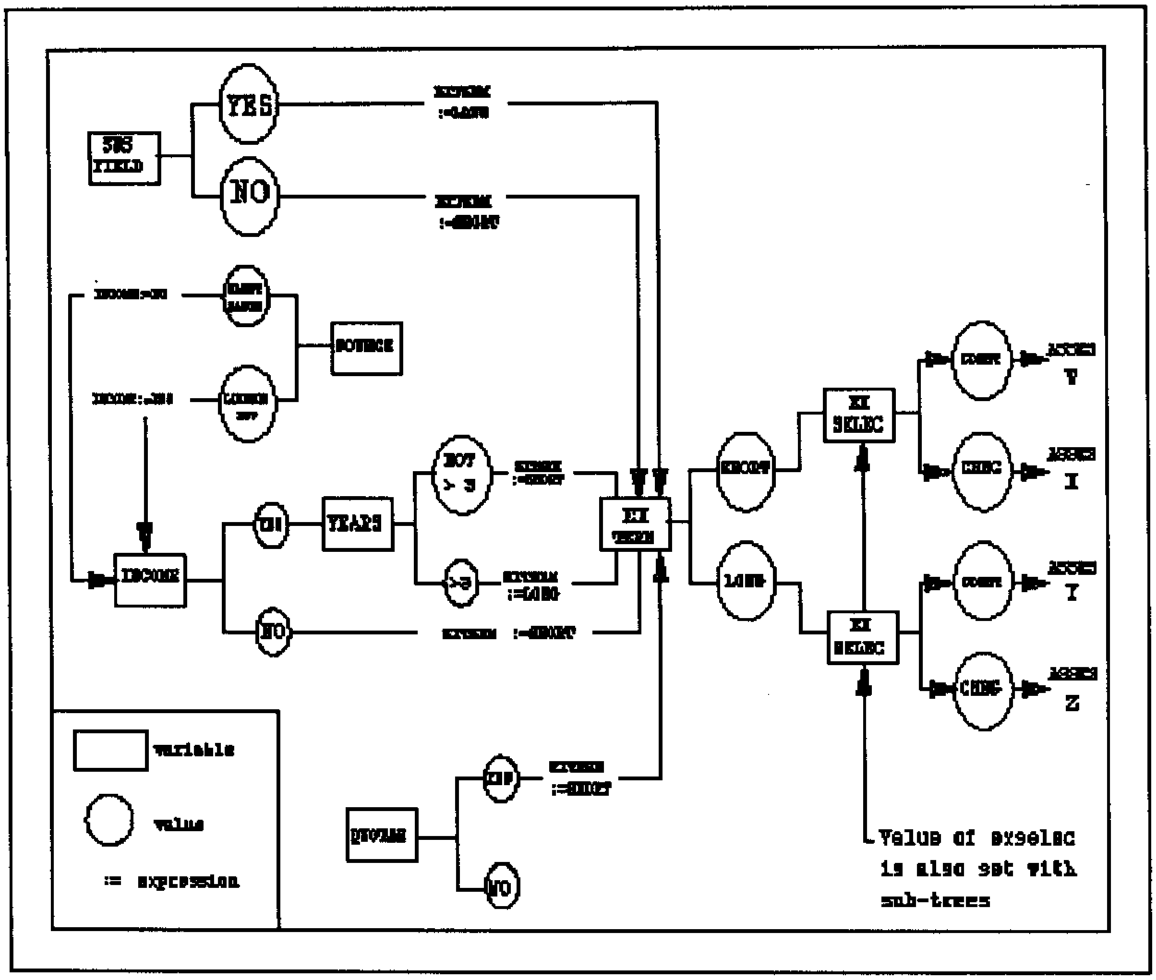


Figure 3. Expert System part of decision tree to determine assessment type.

Example Consultation :

User interface :
User supplies data by selecting menu option →→→→→
↓
↓
↓

REQUEST FOR DATA	**MENU OPTION**	**VARIABLE**
1b is the term of your goal	LONG/SHORT/EXPLAIN	->exterm
2b Do you have to calculate quotas next year	YES/NO	->quotas
2b Do you want to calculate a sustainable yield	YES/NO	->sus_yield
2b Do you need to know government income	YES/NO/EXPLAIN	->income
3b Which is the most important	YOUR OWN FLEETS CATCH / LICENCE REVENUE FROM FOREIGN FLEETS	->source
2b Please enter no of years hence you wish to calculate income for ?	____________ ?	->years

Secondly to determine the value of exselec. 'Again' the user could enter its value directly if the question is fully understood or explain to devolve responsibility to the next level where the question is broken down to intermediates in order to correctly determine the 'parent' formal goal.

REQUEST FOR DATA	**MENU OPTION**	**VARIABLE**
1b Is selection	CONSTANT/CHANGED/EXPLAIN	->exselec
2b Are you planning to use closed areas for management	YES/NO/EXPLAIN	->area
3b Will you have restrictions to which grounds vessels have access to	YES/NO	->grounds
3b Does the species concerned frequent these grounds	YES/NO	->sp_area
2b Are you planing to use closed seasons	YES/NO	->seasons
2b Are you planing to vary mesh sizes in regulations	YES/NO	->mesh

This is the actual knowledge base :

```
1a IF exterm       = SHORT and exselec    = CONSTANT    do assessment W
   IF exterm       = SHORT and exselec    = CHANGED     do assessment X
   IF exterm       = LONG and exselec     = CONSTANT    do assessment Y
   IF exterm       = LONG and exselec     = CHANGED     do assessment Z
     IF quotas     = YES THEN exterm                                      := SHORT
   2a IF sus_yield = YES THEN exterm                                      := LONG
      IF income    = YES AND Years > 3 THEN exterm                        := LONG
      ELSE exterm                                                         := SHORT
       3a IF source = YOUR OWN FLEETS CATCH THEN income                   := NO
          IF source = LICENCE REVENUE FOR GOVT THEN income                := YES
   2a IF area      = YES OR seasons = YES OR mesh = YES THEN exselec      :=CHANGED
     ELSE exselec                                                         := CONSTANT
       3a IF grounds = YES AND species_area = YES THEN area               := YES
```

The numbers on the left of the rule base and user interface correspond to the order in which the inference engine would control both the processing of the rules and requests to the user for more data.

In a goal driven backward chaining system it would try to evaluate the rules first and only request data where it was required to satisfy the conditions to prove the conclusion. In a data driven forward chaining system it would request the data first i.e. the b sections and then test the condition to move forwards towards the conclusion.

The user knowledge base and user interface do not depend on the symmetry shown in this example. Rules and data supplied by the user can be re-used in many different areas in any logical combination. The symbolic plan here is presented to emphasize how the inference engine would work. It would only proceed to the deeper nested levels or 'intermediate' rules and their associated requests for data if and only if the 'parent' rule could not be evaluated directly; provided, that is, the developer has provided the correct 'intermediate' rules. The inference engine will find and evaluate them whenever necessary. If the inference engine used forward or backward chaining the knowledge engineer and application builder would need to decide what order the rules will be tested in. This example uses a breadth first search though it could be built using depth search first. Indeed a depth first design may be better because it can be more efficient, and users are disconcerted by questions which jump all over the domain which is what happens with a breadth first design.

An inference engine also controls 'backtracking' which is necessary when a line of reasoning turns out to be a dead end. A good design of the knowledge base by the developer will help minimize this, e.g. by asking general questions which are applied in many of the branches of the decision tree so that they don't have to be asked repeatedly. If they are asked early on their answers can be applied wherever required in the knowledge base.

Data Matching

Having considered the design from the top end,i.e. progressing from goal to data, the next step is consideration of data or bottom end, in order to see how these two approaches can be integrated. In this proposal this was considered early enough to contribute to the choice of the data model used for the goal end of the design.

It is necessary to categorise the data* without having endless decisions trees. The application of set theory is used because the principles derived can be translated directly onto the relational data model (see Fig 4).

* (data referred to is the data entity types and not the actual values they contain which would be the field measurements of biological characteristics and the values for commercial data etc, which would be held in a whole series of other tables as an extension to the 'data model' described in this report.)

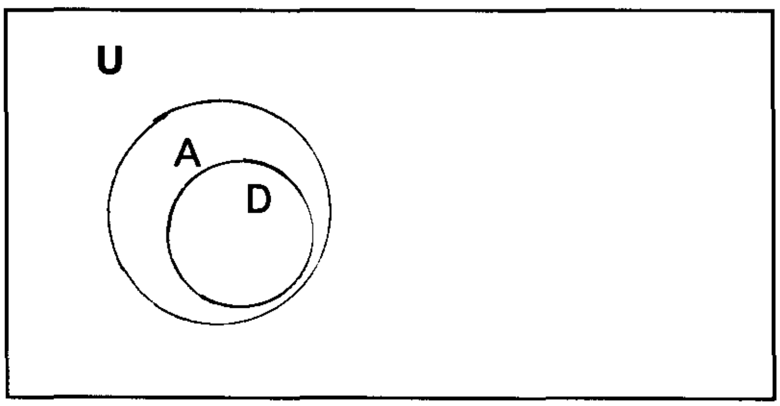
The set D (data desired) is defined by the formal goal chosen, the mapping being done by the view goal_data defined in the relational model earlier.

The set A is available data i.e. that which is already collected and recorded.

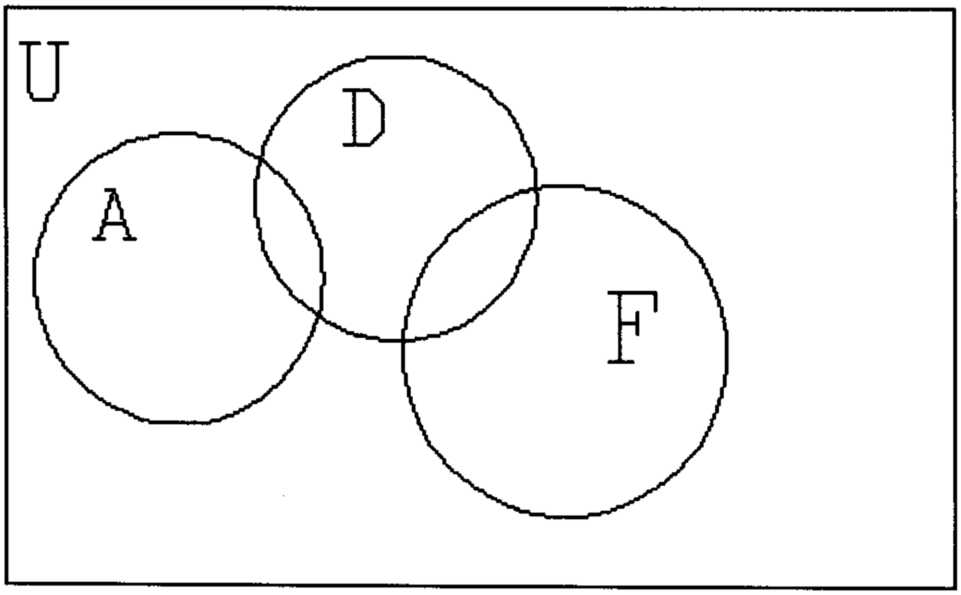
Set F is the data that managers are willing to consider feasible to collect i.e. within their budget and resources.

By definition A and F never intersect but D can intersect with A and F to any degree and is free to move around within U excepting the restriction on A and F.

This set theory and its relational interpretation provides the tool for categorizing all data.
eg.



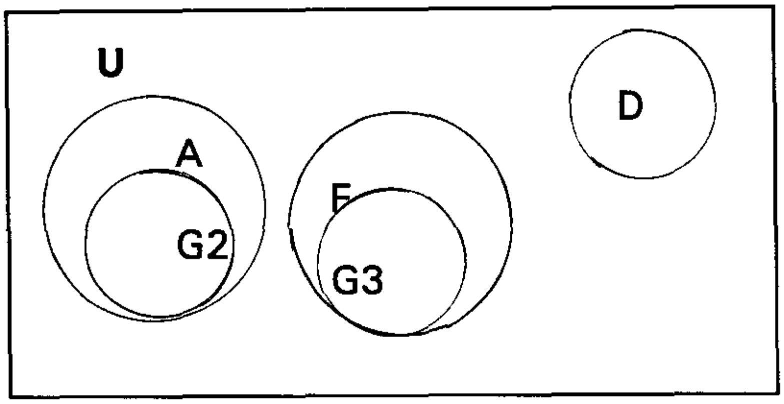
The formal goal that defined the desired data set D is entirely feasible since $D \subseteq A$. Also A-D is data that is wasteful to collect because it is of no use for the goal that defined D.



U	universal data set
A	available
D	desired (or goal_data)
F	feasibile
$A \cap D$	use
D - A	required
A - D	stop collecting
$F \cap D$	start collecting
F - D	dont start collecting
(D - F) - A	required but not accepted as feasible

Figure 4. Data sets and their properties.

Another example :



The formal goal desired is totally unfeasible but goal G2 is achievable with available data and G3 is feasible with the data that is feasible to collect, etc, etc.

As long as all the data sets are defined for each goal as demonstrated by CREATE VIEW goal_data earlier on, then simple boolean tests can be made to see which of these various goal sets intersect fully with the sets A or F or the union of A and F. e.g :

IF $D - (A \cap D) = \emptyset$ the empty set
THEN goal is achievable
ELSE goal is not achievable

IF $D - (F \cap D) = \emptyset$
THEN goal is feasible
ELSE goal is not feasible

IF $D - ((F \cup A) \cap D) = \emptyset$
THEN goal is feasible
ELSE goal is not feasible

The central benefit to the user of this method is that the user simply defines his formal goal through the expert system and enters which data is available and which data is feasible. Then a readout can be obtained of whether the chosen goal is achievable or feasible and what alternative goals are achievable/feasible. They will also have tables of which data presently being collected is of no use and what needs to be selected from that data declared as feasible to collect, and finally, a list of data required for the chosen goal which will have to be collected but is presently neither available or declared to be feasible. See Fig.4 for the definition of these sets.

The E.S. reconsultation module will then allow the user to 'tweek' desired goals and feasible data values until an acceptable compromise is reached. Costing can be attached to each goal by performing the necessary calculation by summing the products of the standard cost per record attributes by time period/ amount of records required to bring their variance within acceptable limits.

A novel design of screen interfaced is proposed so that the user can make best use of the flexibility that is available. This could be presented in the form of a interactive control panel with the user adjusting input factors such as management objectives, available data etc, it would then show simultaneous feed back of required management measures, resources, costings etc.

Implementation of data categories in 'Structured Query Language' (SQL)

The following demonstrates how the previous set operations would actually be implemented in practise via a standard related database language.

SQL for evaluating booleans for Goal Testing

This is implemented by performing the necessary SQL on the tables and views just created. i.e.

Goal_data (or Desired)
Available
Feasible
Use
Start

Only the first example will be encoded here

IF $D - (A \cap D) = \emptyset$ THEN formal_goal_test := T
ELSE formal_goal_test := F

And this is written in SQL as :

```
IF SELECT * data
  FROM  Goal-data
   WHERE data NOT IN
    (SELECT data
      FROM available
        WHERE data IN
         (SELECT data
           FROM Goal-data)
IS NULL

                    THEN formal-goal-test:=T
                    ELSE formal-goal-test:=F
```

The desired data set D is implemented by the code for CREATE VIEW Goal_data given earlier.

The TABLE of available data A is created using the SQL INSERT operation at a record at a time. e.g.

```
INSERT  into Available
VALUES ("Aggregated catch data").
```

The user would obviously do this via a menu and not directly through SQL.

A similar insert operation would be carried out for a table feasible.

Both the above tables would only have one attribute 'data' which holds the names of the available data. Instruction of which data can be used from that available would be given by implementing the intersection of A and D ie. A D . This would be done by creating a view using 'IN'.

```
CREATE VIEW Use AS
  SELECT data FROM Available     (the name of that table)
    WHERE data IN
      (SELECT data FROM Goal_Data)
```

NB note Goal_data would be better off being called 'desired' in order to keep the nomenclature uniform.

Data required is D-A which is the difference operator available by the NOT IN function of SQL.

```
CREATE VIEW Required AS
  SELECT data
      FROM Goal_data (or Desired)
        WHERE data NOT IN
          (SELECT data FROM Available)
```

The inverse view ie. D-A is created to inform the user which data he/she can stop collecting. Call this view 'Stop'.

Data to start collecting from that which has been said to be feasible is given to the user by F $\cap$ D using the same code as for CREATE VIEW Use except table Feasible is substituted for table Available. Call it 'Start'.

FURTHER EXPERT SYSTEM MODELS

A complementary approach would be to view a fishery in terms of information structure and control.

The entire fishery is viewed as having an inherent information content and that named items of 'information types' and the 'values' these information types contain describe the 'state' of the fishery at any one time. Any one objective can be represented as a sub set of 'information types' that hold the calculated 'ideal values' that should prevail when the objective has been satisfied. Logically the sub set of 'information types' describes the objective that should be actually measured in the real fishery to provide 'real values'. From the differences between the 'ideal values' and 'real values' of the information sub set it is possible to deduce what 'actions' are required to 'move' the fishery towards the objective.

The overall 'state' of the fishery covers all aspects including economic, commercial, and social considerations, not only the biological state of the stocks concerned. The sub sets of 'information types' representing each objective will overlap in many areas. This is an advantage when it comes to data collection of 'real values' since the data can be shared for each objective being considered. However a fundamental problem arises when separate objectives might require widely different 'ideal values' for any one of the 'information types'. In the worst case these 'ideal values' may lie either side of the existing 'real value' if this happens then directly opposing management 'actions' may be recommended as a result of these conflicting 'objectives'. With luck the 'ideal values' would lie on the same side of the 'real value' for a given information type, then the 'direction' of the actions would correspond even if their required magnitude did not. In the latter case some optimum intermediate 'ideal value' could be settled upon.

The above considers only one single 'information type' in isolation. Usually an objective will be linked to more than one information type, then interactions between objectives become complex to manage because some objectives may agree in the direction of movement for the 'value' for one of their common 'information types' but disagree for one of the other common 'information types'. In this complex situation the best solution for a practical system would be to identify objectives that 'overlap' at one of the early 'stages', and warn the user if they are potentially conflicting and be likely to require opposing management actions. This solution is implemented as part of stage 2 of the overall system plan presented here, it is hoped that this will pre-empt any major management conflicts before the list of objectives that is passed on to stage 3, process 4, is finalised.

The structure and management of a fishery viewed in these terms is illustrated in Figure 5. The plan following this figure describes in detail each of the stages involved.

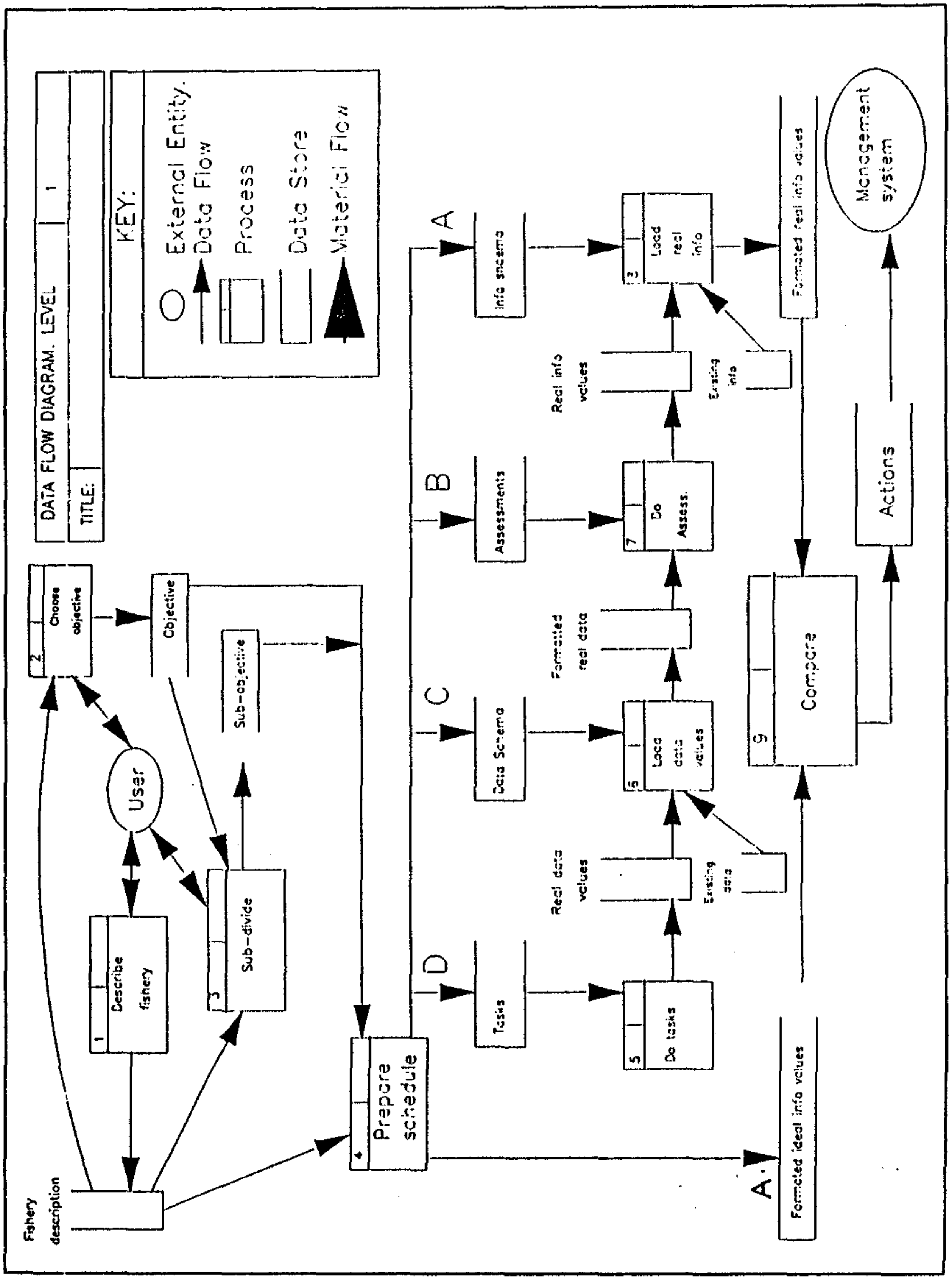


Figure 5. Data Flow and Functional Decomposition for the Proposed Fisheries Management System

OVERALL SYSTEM PLAN

Each 'stage' represents a natural break in time in the management process, where processing would be suspended whilst questions/suggestions were considered and ancillary tasks carried out e.g. a sampling program, or the running of complimentary software or preparation of data for input. Each 'process' within the proposed system is in accordance with functional decomposition and would all (as it happens but not necessarily) be carried out sequentially. One or several processes can be carried out within each stage.

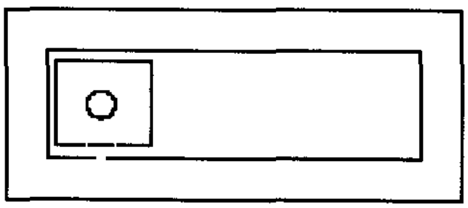
STAGE 1

Process 1 : Describe fishery.

a) What kind of fishery do you have?
Guide user in their responses with use of an expert sub-module.
* size.
* value.
* species.
* geographical limits.
* commercial framework.

If user cannot provide this information then this stage offers advice on how to go about finding the information and in what format to prepare it for answering these questions the next time stage 1 of the programme is run.

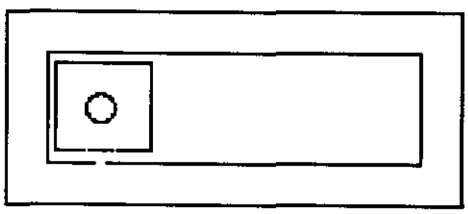
b) Find out type, quality and quantity of fisheries data that is already available :
* stock identity
* catch / effort
* age data
* length data



<--------------

SAVE 'STATE' TO LOGFILE

STAGE 2.



-------------->

READ 'STATE' FROM LOGFILE

Process 2 : Choose objective.

Determine management objectives for this fishery?

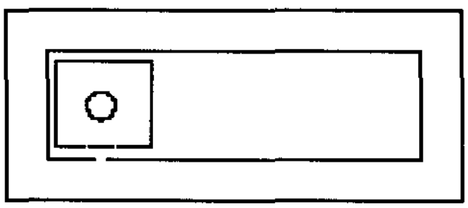
* Present a full range of all potential management objectives.
* Have hypertext explanation of their respective requirements and implications.
* Guide user in their responses with use of an expert sub-module.
* The expert sub module will compare the stated objectives with the 'type of fishery'

as defined in stage one and read in from the log file and advise as to which objectives are realistic (given this comparison) for a particular fishery.

* It could enlarge on the likely requirements (studies including surveys and assessments, resources including expertise, manpower, plant etc.) and implications, as a result of this comparison between type of fishery and each of the intended management objectives.
* It should include advice as to whether any of the 'data already available' as stated in stage one is of use for any of the assessments advised as being part of the likely requirements to meet a given objective.
* Advise on any likely interaction between management objectives where several are chosen.e.g.
    * Where two objectives require conflicting management strategies.
    * Where tasks to achieve different management objectives have the same data requirements.

Process 3 : Sub divide objectives.

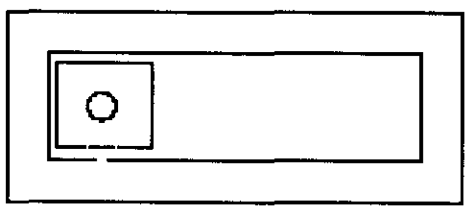
* Where necessary break down an objective in to a series of sub- objectives and repeat the above analysis on these. Format these lists of main objectives and associated sub objectives for storage to the log file so that their arrangement is understood when read in to stage 3.



<--------------

SAVE 'STATE' TO LOGFILE

STAGE 3.



-------------->

READ 'STATE' FROM LOGFILE

Process 4 : Prepare schedule.

Explore in detail the requirements of each of the individual management objectives that were settled upon in stage 2. These objectives would be processed one at a time for each run of stage 3.

This stage forms the main part of the information structuring and control activities described above and illustrated in Figure 5 (labelled activities A --> D).

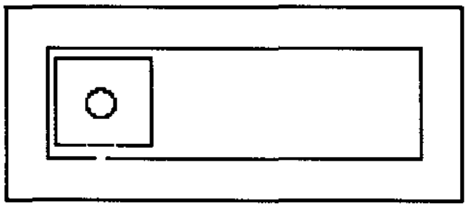
This stage should be able to bear in mind the 'type of fishery' as described in stage 1 and shown by the following process numbers in Figure 5.

A) Frame objective in terms of both 'information type' names and estimate 'ideal values' for these 'information types' when objective is achieved.

B) Provide list of 'assessments' that it will be necessary to carry out to provide 'real values' for the same 'information types' as framed in A in order to represent the real present state of the fishery.

C) Provide list of 'data types' that will be required for input into the 'assessments' given by B.

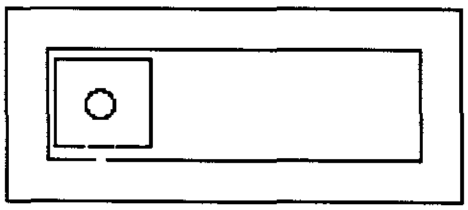
D) Provide list of 'tasks' required to provide the 'data values' for the 'data types' given by C.



<--------------

SAVE 'STATE' TO LOGFILE

STAGE 4.



-------------->

READ 'STATE' FROM LOGFILE

Process 5 : Do tasks.

Carry out the list of tasks provided by D in stage 3.

STAGE 5.

Process 6 : Load data values.

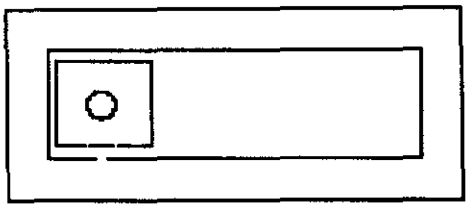
Assemble all of the results in the correct format i.e. fill in the 'data types' with the 'data values' produced from the 'tasks' in E.

Process 7 : Do assessment.

Run the 'assessments' using output from process 6 as input.

Process 8 : Load real information.

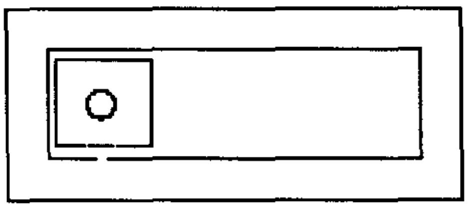
Assemble all of the results in the correct format i.e. fill in the 'information types' with the 'actual information values' produced from the 'assessments' in process 7.



<--------------

SAVE 'STATE' TO LOGFILE

STAGE 6.



-------------->

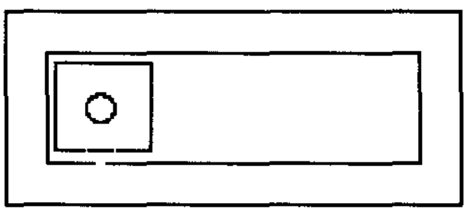
READ 'STATE' FROM LOGFILE

Process 9 : Compare.

Compare the two lots of 'information values' produced for the 'information types':

(i) 'Formatted ideal info values' produced by A in process 4.
(ii) 'Formatted real info values' produced by process 8.

Depending on the differences between the two lots of values decide on the required management 'actions' for moving the fishery from its present 'real' state (ii) to the 'objective' state (i).



<--------------

SAVE 'STATE' TO LOGFILE

Data Modelling

Care has to taken here to have both at once a data model that :

(i) is rigorous enough to implement the above design for information structure and control in a practical system without causing conflicts in data integrity etc.

(ii) allows the expert system to be flexible enough to be useful for a wide range of fishery types and management strategies.

The model must be strong enough not to fail as a result of a certain combination of user input and be able to cope with missing information. A model with this kind of integrity will demand a order of magnitude of greater effort in the design phase, compared to ad hoc systems created by expert users to address specific questions in a single given fishery.