The Role of the Decision Maker in DSSs and Representation Levels

P. Lévine and J-Ch. Pomerol

LPAC and LAFORIA-IBP, P. et M. Curie University
4 place Jussieu, 75252-PARIS 5-FRANCE, email: pomerol@laforia.ibp.fr

Abstract: We claim that the role of the decision maker in a DSS is to control a heuristic search. In this context, the models involved in a DSS naturally appear as evaluation functions driving the end-user control. This interpretation, which seems to be natural for Humphreys' lower representation levels, highlights the links between models and control and explains the prominent role of the "what if" concept in the DSS literature of the seventies. The focus now however, is rather on modelling, and will turn to the control which is related to flexibility. Hence, our analysis paves the way to the design of more flexible knowledge-based or "intelligent" DSSs. We moreover examine the difficult question of the decision support for higher representation levels.

Introduction

In this paper we address, from a theoretical point of view, the question of the role of the decisionmaker (DM) in a DSS. We argue that, when using a DSS, the DM performs a heuristic search in a problem space. More precisely we will show that, in a DSS, models appear as functions defined upon the data, these functions being used as "evaluation functions" in a user-controlled heuristic search. This new interpretation is illustrated by the case of spreadsheet-based DSSs. Putting this idea in the perspective of soft system thinking [5], we will discuss how it is adapted to the framework of designing DSSs relying on high levels of representation.

In other papers [23, 30, 31] we have advocated that the model(s) embedded in a DSS is (are) necessarily incomplete. In other words, it remains some part of the model which is not programmed, otherwise the DSS would become a decision automaton consequently excluding decision, because we think that "decision" is a concept which is essentially contradictory with programming. Hence the main role of the user of a DSS is to complete the model, i.e. to tell the system what to do when there is some gap in the program. In [24, 25] we have also shown that the intervention of the decision maker may be interpreted as a controlling action in a heuristic search. In other words the many information states generated by a DSS may be regarded as states in a problem space. From this point of view, the role of the DM is to evaluate each state and then to decide what will be the next states to develop. In fact, the DM performs a heuristic search [28, p. 101] in his problem space.

We cannot conceal that for the soft system defenders who share the ideas of Checkland [5], the problem solving paradigm suffers a major flaw. It presupposes a specified set of final states to be looked for during the heuristic search [28, p. 810, point 4]. It also supposes an initial well-defined state. The purpose of the search is to link the initial and final states, the solution being the path linking these two states. From an organizational point of view, within a process of progressive specification and development as studied by Humphreys and Berkeley [15] and Humphreys [14], it is clear that neither the states nor the solutions can be regarded as given data in the problem handling and solving process. But while the development process progresses, various states of reflexion-specification are reached and then frozen. Thus
at some low levels of the development it is not difficult to show that decision makers really are heuristic searchers and that the DSS and the decision maker form a united problem solver.

What happens at the higher levels is more subtle because the representations used by the decision makers belong to the psychological realm rather than to computing. Does it makes sense to talk of heuristic search in a state space when the decision maker does not even know in which world to navigate? The first task of the decision maker, or more precisely of the decision owners [5], is precisely to think about the state of the universe and to shape his "small world", to use Humphreys and Berkeley’s expression [15, p.272]. However, we consider that the problem solving paradigm also applies at this level in people’s minds. Individuals use it to determine the framework in which lower level decision process will be formalized.

In order to facilitate the discussion of the above points and to introduce our views we will devote the next section to some definitions and comparisons relative to the various objects and conceptualization levels used during the problem handling process.

2. Structuration, representations, models and problem handling representation levels

2.1 Structuration and representations

It is a cliche to say that DSSs address semi-structured or ill-structured tasks. Nevertheless, it is clear that, when a task is performed by a system, it has been previously formalized and programmed (in the computer sense). Thus a DSS always processes structured data: representations and models [33].

In the last section we used the term representation, as is usual in the DSS framework, to mean a structure designed for receiving data. For example a table with columns displaying months, and lines displaying expenses and sales, is a budget representation whether or not it contains any data. A map is a representation of territory. An investment may be represented as a vector, each coordinate of which is the value of the investment according to a predefined attribute. In order to avoid confusion with the psychological notion of representation we will talk of representation s.s. (stricto sensu) in this case. In DSSs, the representations s.s. are used to apprehend the environment; a representation s.s. is a static structure of information which serves to interpret and record external data or stimuli.

The word representation or prototype is used in a broader sense in the psychological field, where it means an instantiated entity representative of a class. Here representation l.s. (latu sensu) denotes a coherent description of an individual or object representative of a class. These representations l.s. are more or less comprehensive and symbolic. This is the reason why Berkeley and Humphreys [15] introduced the term "representation level". A representation level is characterized by a level of abstraction and a way of thinking, and so at a given representation level the representations l.s. are comparable from the point of view of abstraction and formalization. Humphreys [14] views problem solving as a development process passing through five representation levels, from more to less abstract, the process being somehow reverse to Piagetian cognitive development or genetic epistemology [15]. A level of representation characterizes a cognitive level adapted for thinking about and structuring a problem. "Each of these representations can then form a semantic primitive in its entirety (i.e. a single node or reference point) within the structure developed through operations at the next higher level" [14].

2.2 Models

In any DSS, the computation is performed according to a program, i.e. a sequence of elementary logical or arithmetical operations. Generally, the whole program driving the processing is structured into a number of sub-programs, each of which is the software translation of an abstract model. Thus a model defines the computing operations to be performed on the data collected and recorded by means of the representations s.s. The model tells the elementary processors what they have to do. For example, in Operational Research (OR), a vector \( x \in \mathbb{R}^n \) is a representation s.s. of the production of a plant, whereas the mathematical program (P):

\[
\max F(x) \text{ s.t. } Ax \leq a \text{ and } x \geq 0
\]

is a model of the calculus to be performed on \( x \), in order to "optimize" production. Beyond the computation induced by the program (P), one may also add that (P) is a model of the production process, a representation l.s. of the reality of the plant.

In the framework of expert systems, objects or frames (as used by Minsky [26]) are representations s.s. which will be instantiated into facts, whereas the rules or the methods contained in the objects or the daemons involved in the frames are parts of models. Rule bases or semantic networks are qualitative models. In what follows, it is important to note that, in our view, a model necessarily performs an operation (in the mathematical or logical sense).
2.3 Problem handling representation levels

Equipped with the above definitions, let us now return to decision problem handling. For a long time, a rather naïve view of decision problem solving has prevailed: accordingly the problem should be a well-defined object, imposed externally by an immanent divinity, to anyone dealing with it and, in the same vein, the notion of solution should be shared and clearly settled for anyone confronted with the problem. It is supposed that "problems arrive on the scene ready-structured" [15]. In this framework the states have an objective existence and are defined from the outset of the problem emergence. This implies that the means of identifying a solution are known from the beginning of the process. In this case most people would agree that the problem solving paradigm applies.

Unfortunately, in reality things are not so simple and authors with a large organizational experience such as Vickers [37], Checkland [5], Landry et al. [20] have underlined various aspects of the decision which do not fit in with the preceding view. In particular, they stress the high level of maturation in the decision maker's mind which is necessary before the formalization of the problem [5], the progressive construction of the problem [20], and the recursive choice of the representations I.S. [27], can begin. It is clear that all the reflection needed before the hard modelisation stage is an integral part of the problem solving process [14, 27].

To enter into the discussion of this real question, we will adopt Berkeley and Humphreys' five representation levels describing the problem handling process.

The highest level (level 5) consists of the more or less conscious expression of the problem [15]. The representations I.S. which are used at this level are mainly cultural and psychological. It is what Checkland [5] call the "Weltanschauung", defining a holistic "Gestalt". At this level, most of the representation is beyond language [14]. It is at this level that the problem is shaped (or constructed, [20]).

On the fourth level, the representations I.S. become explicit and the problem owner can formalize various frames (to use Humphreys' meaning); the process of specifying submodules of the decision problem then begins. At this level, the structuration remains mainly substantive rather than analytical, the representations I.S. being, for example, focused on the production process or on the marketing process rather than sustaining an analysis of the functions involved in these processes. In the same way the decision owner will talk of "marketing people" without distinguishing between the various categories. We are not yet at the level of the modelling in our sense, except as an outline of a semantic network representing the links and exchanges between the frames.

On the third level, we enter into the realm of "hard system thinking". At this stage, the structure within the frames is defined by the decision maker. With our notation we would say that the representations s.s. and the models are now built. Models and representations s.s. remain unchanged at the two lower levels. On level two, the models being fixed, the decision maker can perform sensitivity analyses by changing the data used in the models; thus the function is defined but the value of the arguments may change. In the problem handling process the decision maker will define the parameters and data most likely to be used. On the last level the decision maker chooses the value of the data to be entered in the system.

The problem handling process as conceived by Humphreys is a top-down process in which passing from a high level to a lower one entails a refining and a structuration of the concepts established at the next higher level. "The result of the operation carried out on a particular level constrains the way operations are carried out at all lower levels", [14, p.33]. This is a rather general approach for coping with complexity, reminiscent of [32, p. 217 et seq.].

To summarize these problem handling and information processing levels, we have drawn up a table (Table 1) in which we compare our views with Humphreys' views and the data structures used in database management systems (DBMS) as described, for example, by Brodie [4].

It is interesting to examine this table, both in terms of operations to be performed to pass from one level to another, and in terms of the degree of freedom for the decision maker.

The fifth level is the level of the problem emergence, and we agree on this point with Checkland [4] and Humphreys and Berkeley [15]. It is, of course, a very important step for the rest of the decision process [20, 27]. At this moment absolutely everything seems available for the decision maker (or problem owner): the problem, the frames, the representations, the models. What is not free is either psychological (unconscious) or cultural (in particular the educational background).
### Table: Problem Handling and Information Processing Levels

<table>
<thead>
<tr>
<th><strong>DATA BASE PERSPECTIVE</strong></th>
<th><strong>PROBLEM HANDLING PERSPECTIVE</strong></th>
<th><strong>INFORMATION PROCESSING PERSPECTIVE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Need for data storage</td>
<td>problem emergence</td>
<td>problem emergence</td>
</tr>
<tr>
<td>Data model (e.g. relational, hierarchical, network)</td>
<td>frames, problem structuring languages</td>
<td>theories to be handled and definition of the subdomains of application</td>
</tr>
<tr>
<td>conceptual schema</td>
<td>structuration and modelling within each frame</td>
<td>fixing the representation s.s. and building the models in subdomains</td>
</tr>
<tr>
<td>instantiation of the schema (record, segments, tuples)</td>
<td>parametering the models (sensitivity analysis)</td>
<td>processing with various data (sensitivity analysis)</td>
</tr>
<tr>
<td>data entry</td>
<td>assessment of the values</td>
<td>data entry</td>
</tr>
</tbody>
</table>

### Figure 1 Problem Handling and Information Processing Levels

In order to attain level four, a problem owner now clearly identified and self-committed begins to concentrate on the decision problem, in other words a decision maker appears starting to reason about and circumscribe the problem. This circumscription concerns the frames [14] as well as the subdivision of the problem and the theories and methodologies which qualitatively seem to be adapted to each subdomain (e.g. OR, sociology, decision theory, soft system thinking, expert systems, and so on). Various paths are open to the decision maker and numerous means of tackling the problem are available. In the framework of data storage, this is the stage of the choice of the DBMS model. At this level the nature of the problem is frozen, the problem owner is certain of its nature. The conceptual framework emerges, substantive rather than analytical [14]. When entering this level toward and moving on to the third one, the frames are also fixed [14], as are the theories and subdomains of applications [23]. Briefly, entering the third level supposes that the decision owner has clearly defined what the problem is and which framework is to handle it, the representations I.S. being frozen.

Now we enter "hard system thinking". In DBMS we define the conceptual schema of the base. At this level the decision maker must definitively fix the representations s.s. and model the computation process [23, 24]. For Humphreys [14] the problem handling consists of structuring each frame and the calculi within it. When entering the second level, the models and the representations s.s. remain fixed, the processing and the various networks are completed. The only degree of freedom that remains is to test and define various model parameters and to change the data (sensitivity analysis). In DBMS, the records which are instances of the objects contained in the base are filled. The "what if " analysis may be regarded as requests for various views of these records [23, p.45]. The last level corresponds to data entry in all the contexts. Humphreys insists on the reasoning about the values to be introduced in the system (assessment, [14]).

We should note that levels 4 and 5 are generally considered as strategic decision levels handled by executives, whereas the three lower levels correspond to more operational levels. To understand the general misuse or non-use of DSSs at high levels, we believe that it is important to introduce and to compare the conventional information processing view with the ideas of Checkland and Humphreys [5, 14]. We agree with Humphreys and Berkeley [15, p.276] that existing DSSs at most only address the first three levels. The main difference between the Humphreys' approach and ours is that the latter is much more processing oriented (in the sense of Simon...
and computer science, processing means computation along prescribed programs) whereas that of Humphreys is organizationally oriented as a development process. It means that we try to separate representation s.s. from models (functions from data) in order to present a picture which makes sense for a computer scientist, whereas the more psychological and organizational point of view draws a less analytical methodology of problem handling as a systemic integrated process.

One of the advantages of our approach is that it paves the way for an interpretation of the control and of the decision maker role in the utilization of DSSs, whereas in the problem handling framework, the decision maker appears as a modeller whose interaction with the computer aid is not analyzed.

Nevertheless, the three lower levels are very similar and are more easily conceived by computer educated people because they rely on fixed representations s.s. and models which have various counterparts in computer science (programs, semantic networks, conceptual schema) for models, (object, frames (in Minsky's sense, [26]), tables, records) for representations s.s. Thus it is not surprising that the problem solving paradigm seems to be well-suited to these levels, because it relies on the information processing concepts which are also emanating from Simon's world. However, we believe that it is also relevant for the higher levels, as will be discussed in Section 5.

3. A definition for low level DSSs

In a recent paper, Keen [17, p.253] addressses the question: "What is the definition of DSS - and does it matter that after over ten years of sustained research and practice there is still no established definition?"

We believe that the problem solving paradigm provides the tool for such a definition. Let us remember that one of the most important words in various attempts to describe DSSs is the word "flexibility" which is often associated with "user-driven" and "user-friendly" [7, p.347]. However it is not easy to understand why and how a DSS would be more flexible than a classical information system [23, p.113].

3.1 Heuristic Search

Our first step towards a definition is to observe that each DSS is an Information Processing System tailored in such a way that the decision maker can perform a heuristic search using it. The main point is that the heuristic search is user-controlled in the AI sense, which means that, at each state of the problem space, the user has to decide [28, p.101] whether he wants to continue or advance or backup.

To continue means that he chooses another operator at the state he has attained, advancing signifies that he changes his current state to a more promising state (with regard to the success of the exploration) and backup implies that he returns to a previously studied state.

Let us look more closely at the problem solving paradigm for the three lower levels of problem handling. Starting from a state $S_0$, where the problem and the framework have been fixed, the decision maker has a finite number of models $M_1$ (Figure 2). For each model regarded as a state, the decision maker can "continue", which means feeding the model with various sets of data; in short he proceeds to a sensitivity analysis ("what if").

"what if" analysis

enter various set of data

Figure 2 Problem solving at low representation levels

In the state $S_0$, the decision maker may "continue", i.e. try several models or modify an existing one. If he "advances" he fixes a model used during the sensitivity analysis. In any heuristically driven process, the system makes the choice between "continue-advance-backup" at the view of its evaluation of the current state of the search.

In a DSS where the heuristic search is user-driven this choice is therefore made by the decision maker. What is the evaluation of the decision maker at each state?

We consider that the decision maker's evaluation is essentially made according to the results of the model processing. In other words, when proceeding with his DSS the decision maker is, at any given moment, working on a model. This model is generally translated into a sub-program, we call this computer sub-program for a given model a state-processor. Thus at each state, the decision maker makes his evaluation according to the information provided by a model which is formalized by a computer subroutine. It implies that the models in a DSS may be regarded as evaluation functions (in the AI sense). More precisely the evaluation results from a dialectic process between the unformalized information in the decision maker's mind and the model processing. This evaluation function may of course change, depending on the state of the process, just as in a heuristic search.
Our interpretation gives a clear status and emphasizes the role of the models as the main tool for a decision maker's control. Again as in [33, 34], and in the Management Information System (MIS) tradition, the prominent role of the models is underlined, but with a new interpretation as an evaluation function for the heuristic search.

In this framework the notion of "what if" which is a leitmotiv of the DSS field [17, p. 257] is easy to interpret. At each state the "what if" is nothing more than the choice of a new operator, i.e. the "continue" in the heuristic search (see above). Most of DSSs allow such a large number of variations around a given state. This "what if" possibility is justly considered by Keen [17, p. 257] as a passive support which is not sufficient to significantly improve the decision maker's effectiveness. Actually, our interpretation shows that this is not the most important step of the interactive process, the most important step is the decision maker's evaluation according to the models and the other resulting movements between the nodes: advance and backup.

3.2 The spreadsheet-based DSSs

As an illustration, we will continue this section by re-examining a typical DSS. A spreadsheet DSS such as e.g. MULTIPLAN, LOTUS 1-2-3 or EXCEL and the DSSs built from them correspond to a theory (according to Figure 1) which we call "accounting theory". It presupposes that a decision problem can be solved by a accounting process. The representation s.s. is fixed, it is the usual accounting matrix representation.

Now each model is generally designed by the user and is formalized by the operations which are defined on the rows and the columns of the spreadsheet. Once this model is fixed (state-processor) the decision maker fills the unprotected cells. He tries different values (operators) around a state of his reflection and information.

For each modification of a cell, the decision maker gets a new value of, for example his net profit. According to this obtained value (evaluation) he might change his advertising budget, and thus see the effect on sales and the return on net profit. If he is not satisfied with this new value, he will make further changes; we are in the "what if" process (heuristic search on the data). The operators are the keyboard operators which allow cell modifications and the state is all the numerical information displayed by the spreadsheet. The view of these screens and the assessment of the result of the computation process (model) implies the acceptance or the reject of the decision maker (evaluation). Hence the heuristic search process is entirely controlled by the user.

The evaluation is made on the model level (level 3) and the "what if" results from using the "continue" operators applying on the models.

In a spreadsheet we can recognize the architecture of Figure 2 [23, p.43]. Hence, the model may be changed, for example adding or deleting a column or a row relative to the advertising budget, the currency exchange risk or anything else. On this level, a heuristic search process could also be followed by the decision maker, but the evaluation should be at the theory level (level 4 in Figure 1), it should consist of evaluating the validity of the model with regard to the problem. In our example, the decision maker has to assess the necessity of adding the currency exchange risk according to the type of business he makes. Here the evaluation function is a qualitative function about the relevance of the models.

Forgetting this fourth level, there are two levels of exploration, therefore two levels of flexibility: models and cells. The models are the states of level 3 (Figure 1), while the filled sheets are the states of the second one. From the user point of view they are felt as two degrees of freedom which makes this type of DSS so popular. From our point of view this two levels of freedom explain the popularity of these DSSs which are, as is well recognize, the most used, if not the only used, for strategic planning in the firms.

4. Perspectives for High Level DSSs

4.1 Definition

We now arrive at the general definition that we propose. A DSS is an Information Processing System which is designed so that the decision maker can perform a heuristic search at any representation level. At each state S_n of level n, the evaluation is made by the decision maker, this evaluation mainly relying on the results of the processing of S_n on the items of the next lower level. In other words, if we interpret S_n as a function applying to the next level down, the evaluation is mainly based on S_n(S_{n-1}). As a consequence of the evaluation, the decision maker chooses between the operators continue, advance (to S_{n-1}) or backtrack, if this evaluation is not sufficient. In short the system formed of the decision maker and the DSS is a problem solver whose evaluation function is partially or totally contained in the decision maker's mind.

In parenthesis, this definition solves the decision support paradox. Let us first explain the paradox. As stated by Doukidis [7, p.346], "unstructured tasks are those that have no specific procedure to deal with", and he adds that, in this case, the task cannot be automated and consequently cannot be supported by a DSS unless somebody has modelled it. When the task is structured and
successfully modelled, it "simply" suffices to computerize the whole process to observe that the decision is made by the machine. A totally modelled task (decision or otherwise) can be automatically performed and we can no longer talk of a decision aid. The paradox is that to design a DSSs, it is necessary to model, so eliminating the decision maker and the decision support concept. It is exactly this that is generally observed in reality, with the DSS, used either to make the decision and replace the decision maker when successful, or abandoned after a certain time.

To escape from this paradox we must consider that the search is not modelled, and so the metamodel (i.e. the way of sequencing the states: models, representations, data) is not entirely modelled and is not contained in the DSS. The DSS is an incomplete artefact [30, 31], because the metamodel resides in the decision maker's mind. It also explains why the interactivity in a DSS is essentially different from the interactivity in any accounting system (for example). In a non-supporting system the interactivity is only useful for data entry and reading the output of the systems; in a DSS, the interactivity is moreover necessary to control the heuristic search, and it is the reason why it has to be much more valuable.

4.2 Perspective for high level problem solver

We believe that our definition also applies to high representation levels. The starting point is now the vague feeling of the existence of a gap between an actual state of affairs and a preferred state. We will call this gap. A figure similar to Figure 3 may be envisaged here, where the models are replaced by the problem constructs P.

It is, of course, much more difficult to bring any computer support at these levels. This seems to be unrealistic at the fifth level for the problem construct. According to [14], the fourth level is the realm of problem structuring languages. From a DSS point of view, we have advocated [23, p.115] the use of a "tool box" providing various structuration frameworks, in order to support a decision maker or owner during this phase of pre-modelling. It is clear that these levels being situated before the modelling level 3, the use of a computer is not clear to conceive. We have to invent tools or languages accepting unstructured reflection and facilitating the progressive emergence of a structured solving environment. It is clear that few people have yet considered this question. We must mention the tentative of Humphreys [16] whereas the language of Geoffrion [11, 17] is more a modelling language applying to the third level than a structuring language. We have advocated a more pragmatic approach by designing multi-models and multi-experts systems [2, 22, 23]. It is also clear that allowing the decision maker to perform a heuristic search among many models, even if they are not completely specified, is not the total freedom of structuration that should exist at representation level 4. The decision maker is therefore too constrained and a pure structuring language would be preferable. The difficulty lies in designing such a language.

As is now clear after twenty years of experience in the field, the design of tools allowing heuristic search at least up to the third level is the only way of holding the interest of live decision makers. It is now necessary to attack the higher levels to meet the needs of high level executives. This notion of heuristic search is clearly tied to the important notion of the degree of freedom for decision makers. To summarize this idea we can figure in a table the nodes which are fixed, at each level, together with the meaning of the "continue" or "applying the operators" at this node (Figure 4).

The notion of heuristic search which encompasses backtracking is consistent with many of the observations which have shown that the decision making process is totally unlinear, particular at the higher levels. Nappelbaum [27] emphasizes this point, explaining how the decision maker may backtrack to the problem definition or to the determination of the representations. We also think that our approach reconciles the analytical stream with the systemic approach [e.g. 21, Ch. 6].

Figure 3 High level heuristic search
Proceedings of the 28th Annual Hawaii International Conference on System Sciences — 1995

"continue" operators | states
---|---

| Problem construction | Envisaging the gap |
| Framing, splitting, structuring languages, qualitative reasoning | Problems |
| Frames, submodules theory, solving environments and methods | Models and Representations s.s. |
| Model building | Data |
| Sensitivity analysis | |

Introducing the data

Figure 4 The nodes and the operators applying to it

Conclusion

In this paper we have reexamined the contribution of the problem solving paradigm to the theory of DSSs. We believe that this framework is still useful for understanding both the structure of DSSs and the role of the DM. The problem solving paradigm is perfectly adequate for describing low level DSSs, even handling various types of models, and it moreover opens up some new perspectives for attacking higher decision levels (strategic levels).

For the three lower Humphreys representation levels, the problem solving paradigm provides a new DSS definition, defining it as a system supporting heuristic search in various problem spaces, the main point being that the heuristic search is user-controlled according to the evaluation provided by the state examined at a given moment. We remember that Keen and Scott Morton [18] gave as one of the three main targets for DSSs: "The relevance for managers is the creation of a support tool, under their own control, which does not attempt to automate the decision process, predefine objectives, or impose solutions".

At the third representation level, the interpretation of the model as an evaluation function makes clear the link between modelling and control, which are two of the main components of the Representations, Operations, Memory aids and Control mechanism (ROMC) framework [33]. It also enables us to understand why, as stated by Keen [17], that "what if" processing does not capture the very nature of active DSSs.

The identification of the status of the models, in a DSS, as a fundamental tool for driving the interactive heuristic search is a new contribution to the discussion of the role of the ES in DSSs (complementary vs. symbiosis, see e.g. [1,6, 8, 9, 10, 13, 19, 23, 29, 35, 36] and the references therein). Actually, from our definition we can infer that ES shells or spreadsheet-based DSSs in which the second level of search (on the models) is also user-controlled are two species belonging to the genus of three level DSSs. In other words, a knowledge base is nothing more than a particular model which is used by the decision maker to perform an exploration. Sprague and Carlson [33] have rightly stressed the possibilities (even the necessity) for a DSS to encompass several models, so
that a knowledge base is just one of the possible models of the system. Note that such a DSS enters into the definition of Waterman [38, p. 231 for a knowledge-based system: "a program in which the domain knowledge is explicit and separate from the program's other knowledge".

The natural destiny of DSSs being to cope with the so-called unstructured tasks, they will obviously involve knowledge bases and expert systems for performing various, mainly qualitative, subtasks, because designing a knowledge base is the only realistic way we know to modelize an a priori unstructured task.

It is noticeable that at the model level, the decision maker's demands for explanation, modification, interpretation, aid and so on, are very pressing, as experienced in [25]. Thus, one of the efforts of the designers should be now to provide the same flexibility on the models that is already attained on the lower level (data).

At upper representation levels, the model can also be changed by the user (using language system [3], or by structured modelling [11, 12], or a structuring language [14]). Up to now in many systems, even for small modifications, the end-user is not supported by the system [7]. This proves that the task remains difficult and it is far from easy to aid the decision maker at the fourth representation level during the framing or pre-modelling phase. However, we do believe that precisely at this level the notion of heuristic search and exploration are fundamental.

References


