Computer Science Journal of Moldova, vol.6, no.1(16), 1998

Syntesis of group project decisions in distributed computer decision support systems

E. Trakhtengerts

Increasing volume of information received by leaders and bosses, complication of problems to be solved, necessity of calculation of great numbers of arising problems and quickly changeable situations requires to use computers in decision making process. Therefore a new class of systems has appeared — decision making support systems (DSS).

Decision making support systems allow a decision maker (DM) to combine his personal judgement with computer output. This is especially important in emergency. But to do it the DM must formalize his judgement. The main purpose of the works in this field is to create methods which allow the DM to formalize and express his preferencies. This formalization is implemented in several stages. This paper briefly discusses major stages of formalization (some of them are new). Main attention is dedicated to the most interesting and complicated stage — computer analysis with dinamically changeable rules which allows the DM to estimate the consequences of his decision in emergency.

Computer means make it possible to formalize, analyze and ultimately optimize decisions to be taken. Although semantic problems have not been formalized and remain unresolved so far, a large number of procedures used to assess importance of criteria, to generate scenarios or otherwise to lend them to formalization, at least partially.

The sequence of decision making procedures is shown in Fig.1. The module numbers in Fig.1 indicate the order of decision-making stages.

The arrows reflect the cyclic nature of the process. If the results of evaluation of scenarios or their selection (modules 2, 5) do not satisfy the DM, "reappraisal" is performed. Then the control goes to module 1 for reassessing the weights of criterios, possible changes in their selection etc. Now let us discuss modules shown in Fig.1.

1 Establishing relations between operations and generation of scenarios

The DSS offers a set of possible operatons for each situation. If such a set has not been provided for, it can be created by DM. The task of DM is to indicate a sequence of operations and point to a possibility of their parallel execution. Based on this data the system automatically generates possible scenarios of decisions [1].

For scenario generation, we introduce the notion of parallel-time grammar. This includes six sets $G = \{V_N, V_T, P, S, T, F\}$, of which the first four sets have conventional meaning and sets F and T are related to parallelism and time characteristics of grammar rules modelling the simulated process.

Set T consists of elements $t_x \in T$, which determine the time of fulfillment of operation x of simulated process. Set $F = \{', \Lambda\}$ consists of apostrophe above the left part of a rule, for instance A', which means the beginning of parallel processes, and empty symbol above a left part of rule, for instance A, which means linear transition to the right part of the grammar rule.

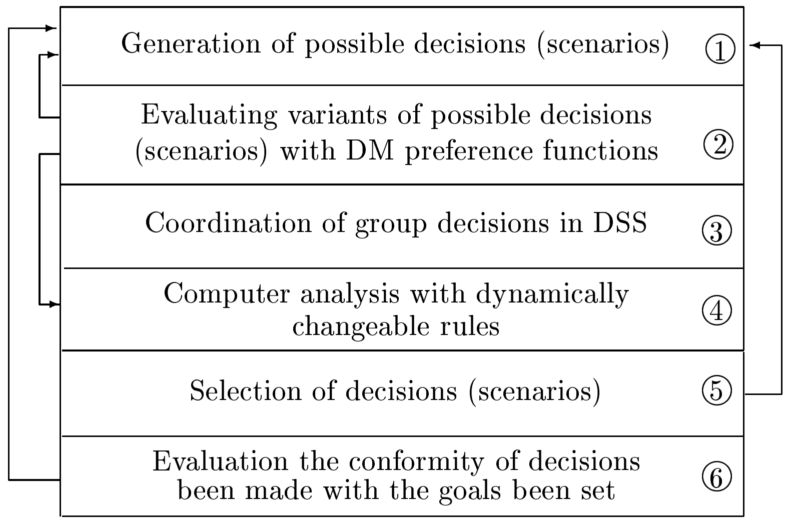

Fig. 1

The set of parallel-time grammar rules P is the following:

$$
\begin{aligned}
<\text{grammar rule}> &\to <\text{left part of the rule}> \\
&\to <\text{right part of the rule}>;
\end{aligned}
$$

$$
\begin{aligned}
<\text{left part of the rule}> \to\ &<\text{nonterminal symbol}> \mid \\
&<\text{nonterminal symbol}> \\
&<\text{description of the cycle}>;
\end{aligned}
$$

$$
\begin{aligned}
<\text{nonterminal symbol}> \to &<\text{nonterminal symbol}> \mid \\
&<\text{nonterminal symbol}'>;
\end{aligned}
$$

$$<\text{description of the cycle}> \to < C_n > \mid < C_m > \mid \Lambda$$

$$< C_n > \to < \ (\text{real number}) \ >;$$

$$< C_m > \to < \ (\text{integer number}) \ >;$$

$<$rignt part of the rule$>$ $\to$ $<$symbol$>$ $<$time of execution$>$;

$<$symbol$>$ $\to$ $<$nonterminal symbol$>$ $|$ $<$terminal symbol$>$;

$<$time of execution$>$ $\to$ $< (C_n) >$ $| \Lambda$

As usual in formal grammars under symbol is understood a line which is minimal in number of signs used in the given grammar or in the language described by this grammar and having an independent meaning.

The above shown grammar rules may be presented in three forms:

a) $A \to B(t_B)$, where t_B — the time of execution of the operation B.

b) $A' \to B(t_B)$, where A' shows that it is the beginning of parallel processing. That means that there exist at least another one rule $A' \to D(t_D)$ which is processed in parallel with the rule $A' \to B(t_B)$. These rules simulate parallel operations.

c) $Z(t_Z) \to B(t_B)$ where Z is the last symbol in cyclically repeated part of the line. Thus this type of a rule simulates cyclically executed operations.

DM must depict on the display connections between operations and duration of each operation. It may be done easily with the help of a table or a graph [1].

After that DSS creates a parallel-time grammar and all possible variants of decisions (scenarios) and indicates assumed time of their execution [1].

Let us point to the four features of the proposed grammar. It generates all regular language lines described by the grammar, i.e. all the scenarios, generates “parallel” lines, i.e. operations carried out in parallel, indicates the time for implementing each of the scenarios and shows cyclical operations.

2 Evaluating variants of possible decisions (scenarios) with DM preference functions

2.1 Selection of criteria and assessment of the situation

Selection of criteria depends on decision-maker's preferences only. The criteria can differ from conventional ones or be generally accepted and therefore the decision making support system can just offer DM a certain set of criterions leaving the choice to him.

Let us introduce three subspaces of the criteria spaces [2]: D, S and $H(t)$. D is subset of points characterizing, by DM assessments, the current states of the object for which the decision is made. S is the subset of points for which it would be desirable to have the state of the object, as estimated by DM, $H(t)$ is the subset of points characterizing, by DM estimate, the state of the object in time t, given no management measures. All these subspaces belong to the criterial space $\mathbf{R}^m$, where m is the number of criteria. Each of this sets can consist of one point which will be designated d_0, s_0, $h_0(t)$, respectively.

Let K_i^D, K_i^S and $K_i^{H(t)}$ be DM's estimates of the current and desirable states of the object and an estimate of the object state in time (t), given no management. Then the "weight" (significance) of the i-th criterion can be defined as $K_i = \gamma_i F\left(K_i^D, K_i^S, K_i^{H(t)}\right)$. Function F_i can have different forms, for example

$$F_i = \alpha_i \left(K_i^S - K_i^D\right) + \beta_i \left(K_i^D - K_i^{H(t)}\right),$$

or

$$F_i = \alpha_i \left(\frac{K_i^S}{K_i^D}\right) + \beta_i \left(\frac{K_i^D}{K_i^{H(t)}}\right),$$

where α_i and β_i are coefficients describing a relative importance of the difference (partial) of K_i^D, K_i^S, $K_i^{H(t)}$.

2.2 Formation of basic space as the basis for evaluation of decisions. Calculation of DM preference functions

To form the basic space we use the scale which establish a relation of DM criteria with parameters of clear physical meaning. In literature such a scale is often reffered to as basic. We consider the function of fuzzy set $\mu_A(x)$ [3], showing to what extent element x belongs to a given set A, to be linear. The criterion evaluations will be made through linguistic variables[3]. The presentation introduced the notion of linguistic interface[4] enabling DM to express his preferences in terms of quality: "better", "worse", "good", "bad" which are projected by DSS onto the numerical scale of fuzzy sets.

Estimation, selection and decision making is based on the DM preference function, the value of the function is determined by DSS based on basic scales formed by DM and "weights" of each criterion determined by DM and DSS.

The DM preference function can be written in the form similar to that of the utility function.

$$\pi : A \to \mathbf{R}^1, \qquad A \in \mathbf{R}^m,$$
$$\pi\left(\overline{X}\right) < \pi\left(\overline{Y}\right) \Leftrightarrow \overline{X} \tag{1}$$

In the majority of cases, the preference function is not known. To determine it an artificial procedure based on (1) is used.

Let π_i be the preference function built on the i-th basic scale (by the i-th criterion).

$$\pi_i = \frac{\chi_{ik} - \chi_{ik}^{\min}}{\chi_{ik}^{\max} - \chi_{ik}^{\min}} + \theta_{i,k}; \qquad i = \overline{1, m} \tag{2}$$

where $\theta_{i,k}$ is the numerical value of the k-th linguistic variable by the i-th criterion, $\chi_{ik}^{\max}$, $\chi_{ik}^{\min}$, χ_{ik} are its maximum, minimum and current values.

By combining all m of the basic scales into one space, we get the m-dimensional basic space. Thus the whole space of parameters is reflected onto the space of criteria of the same dimension. The criteria space is divided into linear subspaces by linguistic variables.

To evaluate and rank the effectiveness of decisions been made with the preference functions, the significance ("weight") of criteria should be taken into consideration.

The value of the DM preference function in the basic space can be determined from the relation:

$$\pi = K_1\pi_1 \oplus K_2\pi_2 \oplus \ldots \oplus K_m\pi_m, \tag{3}$$

where K_i is the estimate of "weight" (importance) of the i-th criterion; π_i is defined in (2). The symbol $\oplus$ means an operation (addition, multiplication etc.) with a pair from relation (3). If it is necessary it is possible to present (3) in nonlinear form.

3 Coordination of groop decision in a distributed decision support system

3.1 Statement of the problem

As was already emphasized repeatedly, decisions of various experts must be coordinated at various stages and levels of decision making. This coordination is required in many, if not all, areas of application of distributed decision making support systems, and it seems that its importance needs no substantiation.

We introduce the notion of "device state", which may be its coordinates in a flying vehicle, a version of design, etc. Each device state can be estimated by a scalar, if pessible, or a vector criterion such as weight, power consumption for forced air cooling,manifacturability, operational convenience, electromagnetic compatibility, etc.

Today coordination procedure hardly can be considered as completely automatic. It is an interactive man-computer procedure implementable as follows.

1. The designers define the set of permissible states of the devices being coordinated. A designer marks each state that is regarded

as permissible in the "common" window of the multiuser interface so that it can be seen by all other interested designers. Note that the question of choosing the optimal state arises not only upon completion of the design of devices being coordinated, but as the design progresses. To this end, the system offers the designers the opportunity of exchange the required information.

2. The system displays a menu listing the criteria of each state to be estimated. The list can be augmented at will. Each designer estimates each state by each criterion.

3. For each criterion of each state, a coordinated estimate is determined allowing for the opinions of all experts involved.

   There exist varios methods for determination of the coordinated values. The weighted mean estimate

$$\alpha^k = \sum_{i=1}^{N} \alpha_i^k x_i^k \Big/ \sum_{i=1}^{N} \alpha_i^k,$$

   where α^k is the "weight" of the i-th designer's estimate of the k-th parameter, seems to be the simplest one. If all experts are of the same importance $\alpha^k = 1$, then only the mean value is determined.

   Importance of "weight" α^i reflects the designer's competence and, in the case under consideration, degree of consciousness, because interests can conflict during design coordination. The "weight" of each designer can be evaluated by the DSS system itself. Let the relative designer's error in the j-th examination be

$$\varepsilon_{ij} = |T_j - T_{ij}| / T_j$$

   where T_j is the mean value in the j-th examination (making a decision) and T_{ij} is the estimate of the i-th designer in the j-th examination. Then [5]:

$$\alpha^k = \sum_{i=1}^{N} \left(\left(\sum_{s=1}^{t_j} \varepsilon_{js} \right) \Big/ K_j \right) \Big/ \left(\left(\sum_{s=1}^{t_i} \varepsilon_{is} \right) \Big/ t_i \right),$$

where t_i is the number of estimates given by the i-th expert.

Other estimates α_i^k are possible as well. At any rate, it is desirable that not only the "competing" designers take part in the examination, but also "neutral" ones who have expertise in the matter.

After averaging the expert estimates, we get the estimate vector of each state.

4. The obtained state estimate vectors are processed, and the optimal state is determined.

First, we consider the simplest case of scalar estimate.
Let

$$X_i = \begin{cases} 1, & \text{if } i\text{-th device must be coordinated with another device;} \\ 0, & \text{otherwise,} \end{cases}$$

and let β_{ik} be the "reduced price" or weight function of the k-th state of the i-th device.

We introduce a variable $\gamma_{ik} = \beta_{ik} x_i$, $i = 1, 2, \ldots, n$, and denote by $\gamma_{ik^o} = \min \beta_{ik} x_i$ the optimal state of the i-th device when its weight function reaches the minimum.

With coordination of the designs of the i-th and j-th units, the optimal state of the i-th unit can turn out to be compatible with the optimal state of the j-th, which is denoted by $\gamma_i = \gamma_j$. If this is the case the problem of design coordination is solved. If the i-th device is the optimal state k^o, but cannot be interfaced to the j-th, which is in its optimal state p^o (this case is denoted by $\gamma_i \neq \gamma_j$), a tradeoff must be sought such that $(\gamma_{ik} - \gamma_{ik^o}) + (\gamma_{ip} - \gamma_{ip^o}) \to \min$.

Thus, the following function is sought [6].

$$\begin{aligned} \delta_{ij} &= \min\left((\gamma_{ik} - \gamma_{ik^o}) + (\gamma_{ip} - \gamma_{ip^o})\right) = \\ &= \min\left((\beta_{ik} - \beta_{ik^o}) + (\beta_{ip} - \beta_{ip^o}) x_j\right) \end{aligned}$$

$$k,\ p,\ i,\ j = 1, 2, \ldots, n.$$

For three or more devices been coordinated, the function δ is determined similarly, for example, by enumerative search, because the number of possible states p and k is usually not too high.

Upon computation, δ_{ij} is displayed with the chosen states. If the designers are satisfied with the results, the procedure is regarded as completed; the values of β_{ij} can be revised and δ_{ij} determined a new.

If the state of a device is not describable by one criterion (scalar), it is described by two or more criteria (vector of values), thus creating the problem of ranking (sorting) a collection of objects represented by points in the parameter space. Instead of maximizing (minimizing) one or more numerical functions, the best elements of the set under consideration are isolated here. If for the scalar values of β_{ij} the optimal value of δ_{ij} was sought, no explicit vector counterpart of δ_{ij} will be sought for the vector (multicriterial) optimization. Instead, since there may be more than one state of the coordinated designs, a state will be sought that is at least not worse than the rest of then. The notion "not worse than the rest" should be in compliance with the system of preferences of the DM (here, the designer) who provides information about the quality of the design. In our case it may happen that one designer estimates the designs of one device and somebody else estimates another device and their preferences (generally, there are more than two designers) must be coordinated. Obviously, organization of the interaction of the experts working out a coordinated decision is the heart of the problem.

3.2 Interaction of experts during decision making

Traditionally, two types of decision coordination are distinguished:

(1) procedures based on interaction between the experts (in our case the information to be exchanged is displayed) and

(2) multilevel (iterative) procedures with control feedback but without interaction, which are executed by a dedicated software. The feedback information is displayed to the interested sides of the decision-making group.

The classic representative of the first-type procedure is the "round-table discussion", in which the participants express their opinions repeatedly taking into consideration other points of view. In our case, this is exchange of data via the computer network with the help of the multiuser interface. We again illustrate such a discussion as applicable to the case of computer-aided design.

Let two designers coordinating their designs find that a certain device can be fixed at four positions. To estimate the quality of decision, one has to determine the power of the cooling system and the weight of the mounts, which differ at different positions. The position must be chosen at which the total weight of the cooler and mount is minimal. The estimates of both designers made on their displays are shown in Tables 1 and 2. Let them agree to carry out coordination by averaging the estimates. The mean values computed by the system are shown in Table 3, where the roundoff is done to within the parameters of the corresponding devices, and as we consider the "weight" of all parameters are equal it is possible to use the Pareto estimates of the points plotted by the system. Table 3 is displayed to both designers.

Thus, point 3 is the best.

This purely illustrative example shows a primitive procedure of coordination. Note that the experts presented their answers only in the numerical form. The distributed decision support systems enable them to support their decisions by diverse graphic and textual information. Moreover, the algorithms of decision coordination may be modified during discussion. The experts coordinating decisions may either use the previous coordination procedures stored in the system library or develop new ones.

Let us consider another, more complicated algorithm for making optimal coordinated decisions. To estimate the optimality of a decision about coordination the two devices, their designers come to an agreement about the set of criteria and coordinate their estimates through the medium of the common context and common windows. There exist many methods for coordination of expert's estimates of criteria, including those based on the expert's "weight". However, they are not always applicable, because it is desired here that the designers themselves come to a common opinion rather than having the system "averages" somehow their estimates. To do so, they can make use of a common context to display in common windows those parts of the designs of units been coordinated that require discussion and to come to agreement about a common estimate of the criteria.

The following procedure can be proposed for choosing the optimal decision [7].

(1) Coordination of the estimates of the criteria.

(2) Determination of the domain in the parameter space where the optimal decision will be sought.

(3) Linear approximation of the indifference curves (surfaces) in the chosen domain, for which purpose the domain is decomposed into "cubes" (hyperspaces). If each cube must be decomposed into local classes by several equidistant indifference planes, a straight indifference line (plane) is determined in its center.

(4) Upon coordination of the criteria estimates, each device been coordinated is ranked with respect to them, if necessary.

(5) Ranking of all feasible designs.

Table 1

No. of points	1	2	3	4
Cooling power	15	40	20	30
Cooler weight	20	60	30	45
Mount weight	35	80	50	65

Table 2

No. of points	1	2	3	4
Cooling power	25	30	20	25
Cooler weight	40	45	30	40
Mount weight	60	70	45	55

Table 3

No. of points	1	2	3	4
Cooling power	20	35	20	25
Cooler weight	30	50	30	40
Mount weight	50	75	45	60

We illustrate this approach by an example.

(1) Let two designers coordinating their devices come to an agreement that 10 kilograms of the device weight "correspond" to 10,000 rubles of its production cost (this sum is used as a measure of manufacturability), that is, that each of them can "give up" 10 kg for 10,000 rubles. The parameter space domain where the optimal decision is sought (Fig.2) is defined by the minimal and maximal values of the parameters shown in Table 4.

(3) Figure 2 depicts a linear approximation of the indifference curves. It is assumed here that the norm of replacement for the i-th and j-th criteria is constant. Therefore, the "indifference curve" is a straight line passing through the diagonals of the rectangles. If the constancy condition were not satisfied, construction of an "indifference curve" and, the more so, an "indifference surface" in a multidimensional space would be much more difficult.

(4) Let each device be ranked by the coordinated criteria as shown in Table 4. For device A, Variants 1 and 2 are equivalent. Variant 2 weighs 20 kg more and is 5000 rubles less costly. Variant 3 is superior to Variant 2, and Variant 4 is inferior to Variant 3. For device B, Variant 2 is inferior to Variant 1, Variants 2 and 3 are equivalent, and Variant 4 is inferior to Variants 2 and 3.

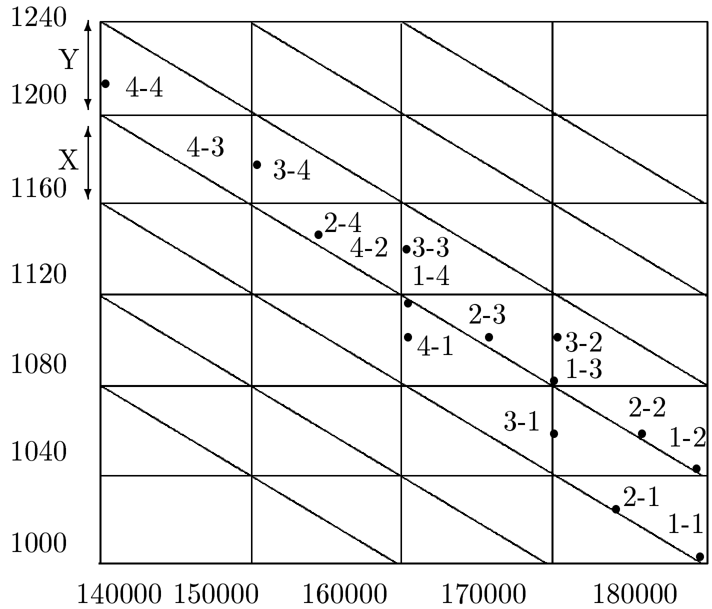

Fig. 2

(5) Now, the possible decisions are ranked (Fig.2). The diagonal bands define decision efficiency. In this example, all possible variants lie within two bands denoted by X and Y. For other data, the variants could be scattered quite differently. Here, the best decisions are 1–1 (Variant 1 of A and Variant 1 of B) and 2–1 (Variant 2 of A and Variant 1 of B); then, equivalent variants 3–1 and 4–1 follow. Next, the equivalent variants 1–2, 2–2, 1–3, 2–3, 1–4, and 2–4 follow. Finally, the worst combinations 3–2, 3–3, 4–2, 3–4, 4–3, and 4–4, which also are equivalent, follow. We note that the variants 3–3 and 4–2 lie at the same point of the parameter space, which means that they have the same weights and "costs". The same is true of combinations 3–4 and 4–3.

Now we pass to the second kind of procedures dating from the well-known "Delphi" method. The experts making decisions are isolated from each other, and the procedure is performed in several iterations. This procedure is popular for various kinds of examinations; therefore, in its description we will speak about experts and not DMs. The essence of "Delphi" is as follows. The experts are shown the estimated object. They are interrogated in several iterations. In the first iteration, each expert estimates the object numerically. Then, the averaged estimate and scattering index are communicated to all of them. The experts giving the extreme estimates are asked to substantiate their opinions in written form, and all other experts become familiar with them through the network. Next, the inquiry is repeated until a "sufficient" agreement between the expert estimates is reached. This original variant, called sometimes the standard "Delphi", was followed by many versions and modifications dictated mostly only by particular applications and formulations of the questions. Sometimes the experts give a fuzzy estimate and an interval or linguistic estimate.

Table 4

No. of variants of design decisions	Device A		Device B	
	Weignt	Price	Weight	Price
1	400	100000	600	80000
2	420	95000	640	80000
3	460	90000	680	70000
4	500	80000	720	60000

Sometimes, in order to bring together the initial estimates of the experts, they are paired arbitrarily and asked to exchange their estimates. One (any) expert in each pair revises his/her estimate; when this is done by all pairs, the next iteration is executed, new arbitrary pairs are formed, etc.

3.3 Rules of coordination of the decision

For efficient implementation of the coordination procedure, the involved experts should be provided with rules for compromise. The rules employed in practice are usually simple enough. Some of them are described below.

1. Method of the ideal point. A point is called ideal if it is optimal simultaneously by all criteria. As a rule, no such point exists even for the estimates of at least one presented object, but a compromise can be reached by minimizing the distance to the ideal point, which can be easily determined. We consider a simple example. For all points x^j characterizing the estimates of an object we define the Euclidean distance between the points x_i and a as
$$p(x,a) = \left[\sum_{i=1}^{m}(a - x_i)^2\right]^{1/2},$$
where m is the space dimensionality. Solution of this function reduces to that of an ordinary one-criterion problem of optimization.
$$p(x,a) \to \min.$$

2. The method of concessions consists in finding a compromise defining the "payment" for the gain in one or more criteria at the expense of another criterion (criteria). Illustration of input data for the method is depicted in Table 5.

Table 5

Criterion	State					
	1	2	3	4	5	6
Weignt of mount	40	25	50	10	10	30
Manifacturability	good	fair	good	good	good	good
Cooling air consumption	10	30	10	0	0	0
Operational convenience	good	good	good	good	bad	fair

3. Method of decision coordination by main criterion. Sometimes problems with several indices can be reduced to a problem with one index for which an extremum is sought, the rest of the criteria being bounded somehow. Then, coordination reduces to

    - identification of the main criterion,
    - coordination of the bounds imposed on the remaining criteria, and
    - determination of the compromise by the main criterion.

4. Method of decision coordination for lexicographic ordering. If the importance of the criteria can be determined, ordering can be first carried out by the most important criterion; then, if several states are equal by the first one, by the second-in-importance criterion; etc.

    For example, if the criteria in Table 5 are arranged in decreasing order of importance, states 4 and 5 are the best by the criterion of weight. Their second and third criteria also coincide, and 4 turns out to be superior to 5 only by the last, fourth criterion.

5. Conforming of criteria, their "weights" and basic scales.

    Sets of criteria of different DM during coordination may or may not coincide. In the last case it is necessary to come to an agreement to criteria which will be used for evaluation of possible decisions. If criteria of all DM coincide the situations which may occur during coordinating decisions of different DM in distributed systems are summarized in table 6.

Table 6

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Basic scales are</td></tr>
<tr><td>coincide</td><td>not coincide</td></tr>
<tr><td rowspan="2">“weight” of criteria are</td><td>coincide</td><td>selection of best coincident decision by (3)</td><td>coordinating basic scales and selection best decision on coincident basic scales by (3)</td></tr>
<tr><td>not coincide</td><td>coordinating “weights” of criteria and selection best decision on coincident “weights” of criteria by (3)</td><td>coordinating basic scales and “weights” of criteria and selecting best decision on coincident basic scales and “weights” of criteria by (3)</td></tr>
</table>

Method of coordinating of basic scales and “weights” of criteria described in [1].

4 Computer analysis with dinamically changeable rules

It is critical importance to provide DM with information about possible evolution of the situation, when a particular decision is made. Such information emerges in the process of a computer game modelling development of events after DM makes a decision and the counteracting environment responds. DSS enable DM to look at his decisions not only after the first “move”, but also to evaluate the response of the environment and effect of each next decision on the process of eliminating consequences of an extremal situation.

Let us divide all factors on which the success is dependent into three groups: prescibed known factors (we designate them α); decision elements dependent on DM which altogether form a set of decisions X and unknown factor ε. The effectiveness indicator W depends on all the three groups of factors $W = W(\alpha, X, \varepsilon)$. Since the value of

W depends on unknown factor ε it cannot be calculated or remains undetermined. Hence, we cannot write $W = \max W(\alpha, X, \varepsilon)$.

This is no longer a pure mathematical problem. It becomes a problem of selection of the decision under uncertainty and we will approach it using methods of computer games. The game evolution will be described by a tree. The nodes of the tree will be matched by real numbers. One number determines the value of DM preference function, the other, if necessary, the execution time of the scenario which is matched by the node. The roots of subtrees are matched by vectors characterizing subsets D, S, $H(t)$ and real number L. The arcs will show the relations between operations and/or scenarios. We will call it a game tree.

The weight of criteria K_i and estimates K_i^D, K_i^S, $K_i^{H(t)}$, basic scales and basic space can be changed during the course of the game; new scenarios and decisions can be added. Therefore, this is the game witn dynamically changeable rules.

The purpose of the game is to carry the situation from the initial $d_o \in D$ over to the ideal point $s^* \in S$ which can be unattainable as is any ideal point (goal), but which the active side is tending to. The success of the game is determined by distance $L = \pi s^* - \pi d_r$ between points s^* and d_r where d_r is a point characterizing the situation after the r-th iteration of correction of the decision,i.e. the objective of the game is $L_r = \pi s^* - \pi d_r \to \min$.

It should be emphasized that the movement is not just one operation, but a whole scenario. Therefore, an interrelated sequence of moves is possible, part of which can be executed in parallel, but this sequence is considered as one move.

In the game tree the subtrees of scenarios are normally turned to their terminal node which characterizes the scenario. The terminal nodes of scenarios are assigned with new marks which become marks (names) of the scenario.

In the process of decision making as a response to changing conditions in the course of situation evolution, the decision is made on a certain fixed situation characterized by subset D_r with consideration for a possible change in the situation characterized by subset $H_r(t)$.

Then the situation is controlled. This state characterized by estimates of each criteria will be considered as response to measures been taken.

The time of next evolution and of the situation and decision making after scenario correction or using a new scenario may depend on sudden change of the situation characterized by certain parameters.

These condition allow to introduce discrete time in decision making process and decision support and to approach methods used in game programming [1]. Let us note, however, that in computer games it is usually not permitted to introduce new rules (scenarios), new means and to change criteria for evaluating new situations.

To select the best decision, conditions should be created for a bilateral game in which "the adversary" of DM is the environment and source of emergency. In computer analysis of the decision two options are possible:

a) the program plays for "the adversary";

b) one of DM team plays for "the adversary".

It is clear that the game may be assymetric: the rules, resources and means of DM may be radically different from those of "the adversary".

The initial posiition of the analysis is generated based on: D_r, S and $H_r(t)$, parameters and criteria, formed basic space; DM preference functions and a set of operations which may be executed.

Unlike conventional computer games, in the DSS game it may happen that some scenarios will be banned. This ban can be initiated by one's own side. For example, some hardware has gone out of order and it appeared impossible to get some scenarios executed. On the other hand, it may become possible to execute unforeseen scenarios. For example, new hardware becomes available which is not supposed to be used.

Unforeseen developments, of course, can negate earlier estimates and predictions. But introduction of such situations in the game reflects the reality which frequently cancels the best calculated plans. So, the task here is to be able to take unforeseen situations into account at an earliest possible stage.

At the beginning of the game DSS builds the graph of the game; the root top of the game graph is characterized by vectors D_0, S_0, $H_0(t)$ and quantity L_0. Arcs A_0B_i; relate the node A_0 with nodes B_i which are terminal nodes of subtrees of the game graph scenarios. Each node B_i, is characterized by the preference function of the i-th scenario and, if necessary, its execution time. The system selects the node having the maximum estimate which means the proposal to choose the i-th scenario.

Now control passes to the game tree G' — game for "adversary". Its root A_0 is characterized by vectors D_0', S_0', $H_0'(t)$ and quantity L_0' estimating the situation after execution of scenario B_i, from the standpoint of "adversary". Among nodes B_i the node with the maximum estimate is selected and command is issued to execute the i-th scenario.

Control goes again to graph G. Node B_i characterizing the i-th scenario now becomes a root one. Vectors D_1, $H_1(t)$ and quantity L_1 is determined after execution of scenario B_i of the "adversary" and the best response scenario is selected. (It is assumed that the characteristics of subset S do not change, but this is not necessarily, goals can change in the course of the game and then characteristics S_r are determined)

It should be noted that DSS allows to change basic scales and scenario "weights" and thus various decision options can be played out. Besides, it should permit comparison of expected results of decisions with actual ones and if results are similar the characteristics of the situations and adequate decisions been found should be saved, whereas in the case of diverging results it should provide for analysis of causes of errors. In addition to quantitative estimation of possible decisions, it is advisable to use the so-called semantic models in which acceptance and effectiveness of moves $(A, B) \in G$ is determined based on quality analysis. For many applications the notion of unsteady situation makes sense. For an unstable situation, it is highly probable that the DM preference function deviates significantly from the true value. One can find out whether the situation is unstable using the analysis of attributes normally known to DM.

Thus using the game approach one can simulate model movement towards a set goal and by analysing different ways reaching this goal, the best way can be found.

5 Selection of decision (scenario)

To select the best scenario one needs to learn how to evaluate the proposed decision options. For this purpose we use the basic space and the method of calculation of DM preference functions. It should be borne in mind that earlier operations can affect the cost of operations to be conducted later. In this case, it is advisable to use a matrix, each element of which β_{kj} estimates the influence of execution of the k-th operation on the j-th $(0 \leq \beta_{kj} \leq 1)$. β_{kj} may be expressed in linguistic variables. If the values of β_{kj} are known, theh the weights π_i of the nodes of the scenario graph can be calculated as follows:

$$\pi_i = \sum_{j=1}^{J} \left[\theta(\alpha_j) - \sum_{k=1}^{K} \beta_{kj} \theta(\alpha_j) \right] X_{ij}$$

where $\theta(\alpha_j)$ is the weight of the j-th node (operation),

$$X_{ij} = \begin{cases} 1, & \text{if the node } j \text{ is included in the path} \\ & \text{of the } i\text{-th graph} \\ 0, & \text{otherwise} \end{cases}$$

Then the most preferable for DM scenario is

$$\pi = \max_i \left[\sum_{j=1}^{J} \left(\theta(\alpha_j) - \sum_{k=1}^{K} \beta_{kj}^{K} \theta(\alpha_j) \right) X_{ij} \right], \quad i = \overline{1, m}$$

Evaluation of decision options (scenarios) made by DM is, by far, not always the same as estimates of distibuted DSS made on the basis of preferences formulated by DM. This is because DM is not always aware of how preferences (formulation of basic scales, estimates of criteria weights, estimates of temporary characteristics etc.) affect ranking decisions. In this connection, we propose the following procedure for DM dialogue with DSS.

1. The system offers DM several options of decisions taking the upper lines of ranking.

2. If DM selects one of these decisions, the procedure of selection is considered to be finished, else one should move to item 3.

3. DSS proposes DM a sequence of possible decisions in the descending ranking order to the point when DM shows the decision which satisfies him.

4. DSS provides DM with possible characteristics of basic scales and estimates of criteria weights at which the decision selected by DM appears the best from the standpoint of DSS.

5. The values of utility function for decision options are recalculated and the selected decision option takes one of the upper ranking places.

6 Evaluation the conformity of decisions taken with set goals

Any system with feedback collects information about results of its own functioning. Information can be collected in different modes (continuosly, from information sources, at specified time moments etc.).

The data collection subsystem serves to process the incoming information and provides it to DM as graph, tables, charts, i.e. in the form most convenient for understanding. At the same time, smoothing and averaging of the incoming information is performed as is done in management of technical facilities so that random fluctuations do not cause unnecessary sudden management actions.

References

[1] E.A.Trakhtengerts. Computer support of decision making. Moscow, SINTEG-GEO, (Russian) in press.

[2] E.A.Trakhtengerts. Computer analysis in decision support systems. New Media for Education and Training in Computer Science. Russian-German Symposium Moscow, Russia, Nov. 23–28, 1996, p.187–193.

[3] L.A.Zadeh. Fuzzy logic // Computer 1, p.83–93, 1988.

[4] E.A.Trakhtengerts Generation, evaluation and selection of the scenario in DSS // Automation and Remote Control, v.3, part 2, 1997.

[5] I.M.Makarov, T.M.Vinogradskaya, A.V.Rubchinski, V.B.Sokolov. Choice Theory and Decision Making (in Russian). Nauka, Moscow,1982.

[6] E.A.Trakhtengerts Methods for generation, evaluation and coordinations of decisions in distributed decision support systems // Automation and Remote Control, v.56, No.4, part 1, 1995.

[7] E.A.Trakhtengerts. Design of distributed group design system // Automation and Remote Control, v.54, No.9, part 2, p.1431–1448, 1993.

Prof. E.Trakhtengherts
Institute of Control Sciences,
65 Profsoiuznaia str.,
Moscow, 117806, Russia

Received December 1, 1997