Metaplanning Model Based in an Organizational Memory

José Bernardo Parra Victorino[1] *Member, IAENG*, Raúl Morales Carrasco[2] and Efrén Armando Osorio Ramírez[3]

***Abstract*— When problems are present in production, equipment or the process which are critical for organizations, many management personnel and experts have to meet to propose a solution. Often, the solutions proposed don't have the best results because of management influence on the opinions of the people at the meeting; the experts withhold and don't use their valuable information, product of years of work.**

This model presents a way to obtain solutions for these kinds of problems in any organization. First, it is necessary to build an organizational memory where the experts register good problem solutions in a specific domain. In this way, when a new problem is presented, the model searches the desired goals in the organizational memory data base and compares them to the past actions registered as good solutions based on experience. If an action has been registered by the majority of the experts then this action will be part of the solution. This point of view is equivalent to a vote in a meeting to obtain agreement for a consensual solution. The metaplanning model was implemented with Visual FoxPro Software in a simple domain, to facilitate the understanding of the proposed solution. We did this because in an organization we don't have the availability of confidential information from enterprises. However, it is easy to extrapolate the model behavior to any domain.

***Index Terms* — metaplanning, decision rooms, decision support systems, agents, groupware.**

I. Introduction

Leaders and managers of major international businesses constantly search for innovations to improve efficiency and reduce process costs in a group; with a focus on productivity and global competition, the average company requires the elimination of the kind of statistics exemplified in the following [1]:

1) In some organizations, workers spend as much as 40% of their time personally surfing the web.

2) About 14 million daily meetings take place in the U.S., more than 50% are considered a waste of time by the assistants.

3) At least 31 hours a month are lost by professionals that spend their time uselessly (more than 4 days).

The most critical problems in an organization are usually solved in managerial meetings. In these business meetings the opinions of experts have a greater influence to solve the problems. The other assistants analyze the possible solutions from different points of view to discuss the best option financially, time-wise or for the future.

These meetings are carried out in an environment of urgency, because the problem is present in strategic equipment or a process that requires a quick and exacting solution. When the problem is present without a solution the equipment or process is stopped. This situation produces loss, so the solution has a great transcendence. Here, as a guide, consider a definition of the problem:

"*The Problem is the event that has the objective of searching unknown things taking into account other known things*"

To get a solution, from this point of view, the first step is to use tools to control the process of taking decisions since there are factors that affect a decision. To reduce some factors that affect a decision there are organizational models of meetings for decision support: teleconferencing rooms, group net, collaboration labs and decision rooms [2].

The model and the software here described remove factors that affect the problem solution, such as:

1) Influence of organizational position opinion
2) Influence of natural leaders opinion
3) Influence by personality
4) Large interventions with few contributions
5) Size and cohesion of the group

In the beginning, a work groupware model was designed as a tool for GDSS (Group Decision Support Systems). In groupware the meetings are made using hardware with distributed processes. With this design the software was built; to service a group of people working in the same domain participating in trying to solve the problem. The process establishes a deadline for registration; when they finish, the moderator, who is the person that issues the call, gives a menu of options to the people where it is possible to:

a) Release a new problem
b) Propose a solution plan
c) Give a contribution to support a project solution
d) Give a contribution to dispute a project solution
e) Motion to clear something in the process
f) Vote

The moderator gives the floor to each participant, coordinates the solution plan and counts the participations of each participant. Once there are enough contributions there is the option to vote; when there is an even number of participants the moderator votes in order to have an odd majority.

This kind of work in groupware typifies the interventions, removes some problems of meetings in person and contributes with:

1) The removal of large interventions by the participants
2) Enough time for every one

This work has been done at DEPI of Instituto Tecnológico de Puebla in Puebla, México under partial support of Instituto Tecnológico de Puebla, PROMEP, and Mexican Government (CONACYT, SNI).

[1]José Bernardo Parra Victorino, Centro Interdisciplinario del Postgrado, Universidad Popular Autónoma del Estado de Puebla, calle 21 sur 1103 Puebla, México (52+ (222) 229 94 00; bernardoparra@hotmail.com;)

[2]Raúl Morales Carrasco, División de Estudios de Posgrado del Instituto Tecnológico de Puebla, Av. Tecnológico 420 Puebla, México (52+(222) 229 88 24; rmc@itpuebla.edu.mx)

[3]Efrén Armando Osorio Ramírez, División de Estudios de Posgrado del Instituto Tecnológico de Puebla, Av. Tecnológico 420 Puebla, México (52+(222) 229 88 16; efrenosorio@yahoo.com.mx)

3) The participants know how many interventions and which type have been proposed
4) The interventions are concrete because they are written rather than spoken
5) They don't require an actual meeting of people in a specific place
6) There are registers of what has taken place and results of the interventions and votes.

This model was implemented in an AS400 IBM Computer in C language under the Unix Operating System.

Taking this system as a platform, a model was developed in which the interventions were not made in person and were not interrelated. For this, it was necessary to refer to the experiences of the experts when they solved problems in a specific domain. The objective was to reduce the influence characteristics, keep the information of some participants and other factors as mentioned before.

One of main problems is to eliminate incompleteness, because some participants confront factors of forgetfulness, secret solution methods, and mutual influence.

To reduce incompleteness, after every implementation of a strategic problem solution in a domain, this solution is kept in an Organizational Memory to increment the Data Base of solutions. Consequently we have a model with a mixture of previous experiences for the creation of a solution plan.

For voting, plan actions are taken by making a comparison of the plans in the Organizational Memory. If there is a majority action from previous experiences, the action is added to the solution plan. While this model was being developed a new mixture method appeared in which the judgment for selection of the element in the array was obtained by majority in the arrays of the mix [3].

II. CONFORMATION PLAN GLOSSARY

Experience. A way an agent has solved a problem.
Problem to solve. A set of goals that need to achieve a good solution
Agent. Person that previously solved a problem in a domain
Goals. Set of actions to reach a specific objective.
Actions. Specific activities, integrated in steps, to reach a goal. An action may or may not have parameters.
Parameters. Way to effect an action.
Bias. Parameters that aren't wanted in a solution
Chain actions. Actions that are bound together
Ponderous Action. The priority of an action to not be excluded in the solution
Ponderous Arguments. The priority of an argument no to be excluded in the solution
Alternative Actions. Set of actions which constitute an alternative method, operating as a goal inside another goal.
Synonymous. Equivalent actions
Precondition. State that should be reached before the current action
Postcondition. State after the last effected action
State. General environment in which a process is found

III. EXPERIENCE GENERATION

The experience of the experts is built with different problems previously solved. This experience should be registered. For this, the model allows the creation of goal, action and argument catalogs.

For example we select Desserts as a domain, and then more specifically Gelatin. This was decided based on two facts; first the information of strategic problems in the enterprises is confidential and unreachable, and second to better explain the model.

IV. METHOD OF MIXING ACTIONS

If there is a variable number of samples from 1 to W of agent experiences, each sample has **X** sets of **M**, without a specific order (the goals of the agent's experience)

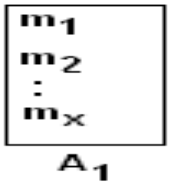

Fig. 1. Sample of sets M.

Each set has **y** elements of **A** (actions of a goal)

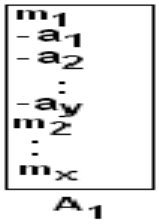

Fig. 2. **Y** elements in a each set **M**.

Then the **W** samples, each one with **M** sets and each set with **y** elements would look like figure 3.

It is necessary to compare each element **A** of set **M1** of the sample **A1** with each element of the set **M1** from the next sample, **A2**. The comparison is made with all samples. When the elements are equal to the majority ([w/2+1] times) of the samples **A**, we will retain it in the outlet of sample **B** as can be seen in figure 4.

To avoid another comparison of elements that were already taken for the outlet sample **B**, the elements are marked as equal. The process is repeated taking as a base of comparison each unmarked element of each set, of each sample; to compare with the remaining elements without marks from the rest of the samples, as can be seen in figure 5.

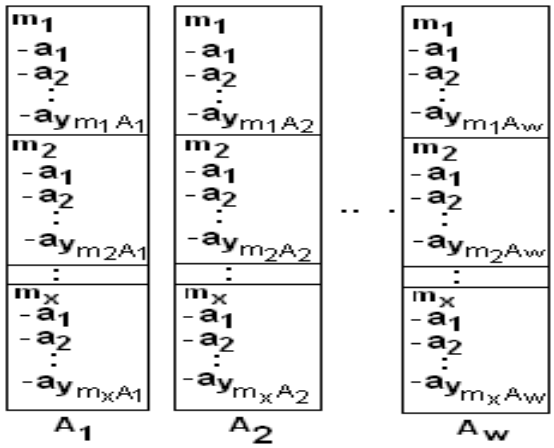

Fig. 3. Set of Whole Samples

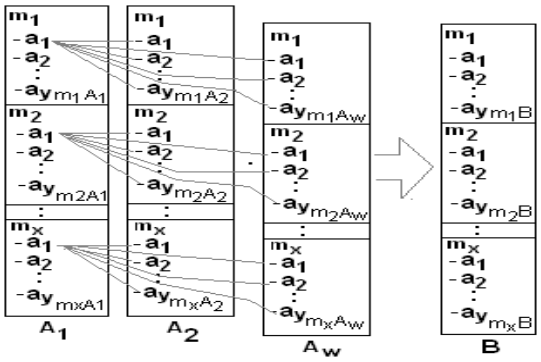

Fig. 4. Set of comparisons

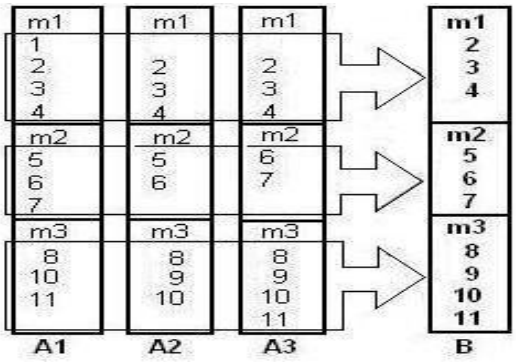

Fig. 5. Example of mixture

The comparison function can be expressed as is shown in the next equation:

$$\left[\left[\left[a\right]_{i=1}^{i=y}\right]_{j=1}^{j=w-1}\right]_{k=1}^{k=m} \xrightarrow{compare} \left[\left[\left[a'\right]_{i=1}^{i=y'}\right]_{j=j+1}^{j=w}\right]_{k=1}^{k=m'} = \left[\left[B\right]_{i=1}^{i=y}\right]_{j=1}^{j=m}$$

iff maj $\left[a = a'\right]$

Where **maj** = [(**w**/2)+1] times or more are equal in the majority of comparisons.

Observing the example of the mixture shown in figure 5, the result of sample **B** has in the first set **M1**, the elements 2, 3 and 4. This happens because 1 is not in the majority of the samples. From set **M2** the elements 5, 6 and 7 are in the majority of the samples. In the set **M3** the elements 8,9,10 and 11 are in the majority of the samples.

To define formally the comparative quantity realized in a set we have the following formula where maximum comparisons are represented by **μ,** see (1).

$$\mu = (w-1)(y-y') + [w(w+1)]y' + y'w(y'-1)(w-1) \qquad (1)$$

Where:

y-y' Lines with majority

w-1 The possible comparisons of a line with majority

[w(w-1)]y' Comparison of each line without majority with the additional elements of its line

y'w(y'-1)(w-1) Comparison of each element of a line with out majority with the elements of the rest of the lines, except the elements of the same column.

The possible cases of comparison with **w>=3** have the following activities:

α When an absolute majority exists in a line, we do **w-1** comparisons and the equal elements are marked for no further comparisons in the future.

β Each element of each line is compared with the rest of lines, except with elements of the same column.

⌐ The lines without majority require comparison with the same line **w(w-1)** times.

Here following are shown the possible cases of comparison.

Case 0: No one element has majority. Each element is compared with the balance of elements in the line.

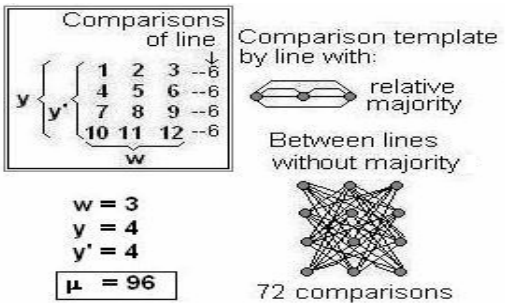

Fig. 6. No one Element has majority

Case 1: One line has majority.

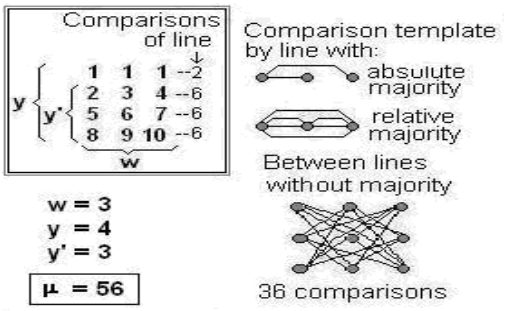

Fig. 7. One line has majority

Case 2: Two lines have majority.

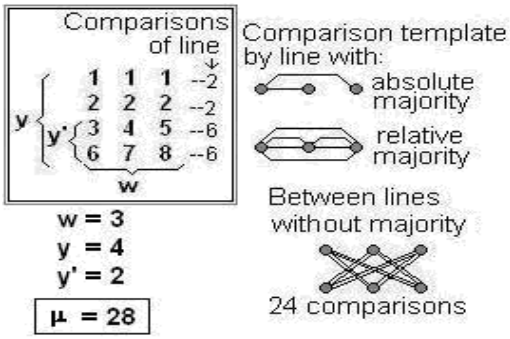

Fig. 8. Two lines have majority

Case 3: Three lines have majority.

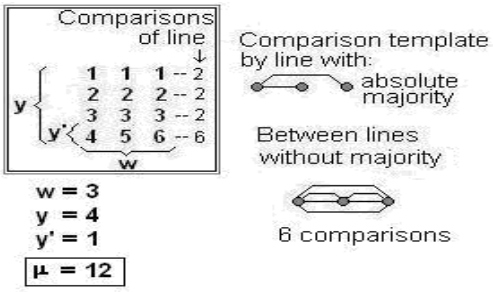

Fig. 9. Three lines have majority

Case 4: Four lines have majority.

However these aren't all the situations, sometimes we have majority without all equal elements. For example:

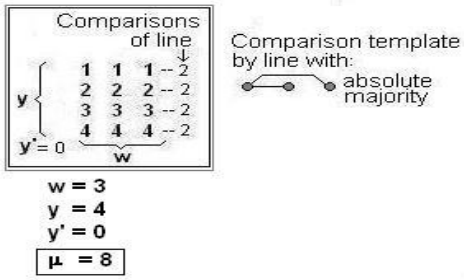

Fig. 10. Four lines have majority

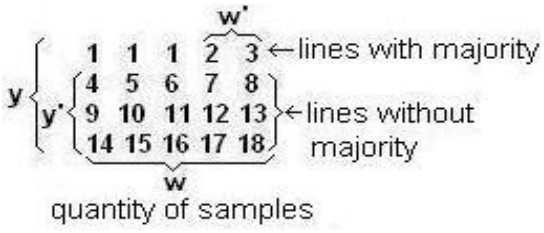

Fig. 11. Lines without absolute majority

These considerations make it necessary to add a set of comparisons, for which the quantity is described in (2);

$$\nu = w'(w'-1) + 2w'(y^*+y'-1) \quad (2)$$

where:

w' = quantity of elements with majority in a line with majority.

w'(w'-1) = comparisons of elements without majority (not marked) in the line to compare

y*+y'-1 = lines which are compare with **w'**

Then adding **μ+ν,** we have the number of comparisons which are described in (3);

$$\beta = (w-1)(y-y') + [w(w+1)]y' + y'w(y'-1)(w-1) + w'(w'-1) + 2w'(y^* + y'-1) \quad (3)$$

The meaning of the formula of comparisons is

$$B = (w-1)(y-y') + [w(w+1)]y' +$$

Comparison of elements with majority | Comparison in lines without majority

$$y'w(y'-1)(w-1) + w'(w'-1) +$$

Comparison between lines without majority (except the column of current element) | Comparisons of rest elements in lines with majority

$$2w'(y^*+y'-1)$$

Comparisons between lines with rest elements of majority lines against lines without majority

And the set of arrays to handle this has the following form;

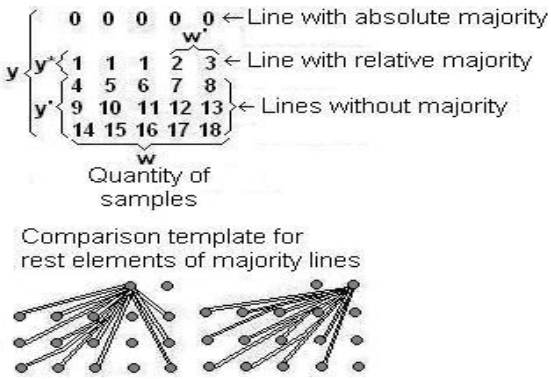

Fig. 12. Set of arrays with several lines

V. MODEL BEHAVIOR

The model was implemented as an application, using Visual FoxPro because this Data Base Manager allows: handle files without array restrictions, the search doesn't require explicit comparisons, the count of elements isn't explicit, and results of searching are kept in temporary files without waste of space.

The metaplanning system has the following work options:

a) update of catalogs
b) update of experiences
c) generate a solution
d) view of metaplanning solutions

Update of catalogs consists of: Adds, Drops, Updates and Queries of:

1. Agents
2. Goals
3. Problems
4. Actions
5. States

The information handled with agents has generic data of people that have solved problems in the past in this domain.

The goals retain the goal key and its description so that when the experience is registered no one needs to select the description.

The problem has its identification, description and the set of goals that make up the problem.

The actions contain key, description, cost, time, precondition, postcondition, id of mark, in addition to other details.

The State has an id and description. The State is the stage reached after the last of a series of actions.

Then the State can be reached considering three possible cases; after the last action of a goal, after a single action, after a serial of actions, or in the middle of a goal. It is necessary to observe that not all actions receive a State.

Consequently the actions have preconditions stating which State should be reached before effecting the current action.

Updates of experiences have different agent actions: adds, drops, updates and queries. During this stage it is necessary to register the agent solution for a problem in a domain; clarifying which goals were reached, with what type of actions, and which characteristics (arguments) were considered, such as cost and time, or if there is an alternative action that was used in some previous case of majority, ponderous action (importance an agent assigns to the action) in addition to other characteristics.

The generation of a solution requires:

- The id of the problem to solve.
- Selection criteria between equal actions considering minimum or maximum cost and time of actions.

The solution is obtained by the comparison of actions in the set of similar goals to those of the problem. For these reasons the solution contains actions of several agents. When the solution is shown to the user he has the following options:

- view the solution with all descriptions.
- view the high ponderous actions without there being part of the solution.
- generate another solution with only desired arguments.
- obtain an alternative solution if an agent contains alternative methods in its actions.

When the user takes a solution plan from the metaplanner, the implementation should be registered as part of the Data Base experience including real activities applied to the problem. Sometimes a plan is implemented with some differences *in situ*. In this way the user makes a new solution that should be registered in the domain.

VI. RESULTS

The implementation was made describing the gelatin dessert preparation because is not necessary to have much knowledge to verify if the solution can be implemented.

The information registered was captured as different recipes to make gelatin, which were obtained from some cookbooks and the Web.

These recipes were captured like experiences of fictitious agents and each problem with its goals was captured too. When several possible solutions were tested, it was observed that if there was a goal in a new problem for which agents didn't give a majority, this goal wouldn't have any action. This situation forced us to consider actions with relative majority; this being to add actions that don't have absolute majority (the middle plus one) but at least more that one agent having this action. This idea doesn't modify the count of comparisons but only modifies the point of view to select the majority. To understand the solution obtained, the experience content and the result are described.

Twelve different ways to prepare gelatins were registered. A new problem was set, "How to obtain a way to prepare Gelatin Salad". If no agent has previously prepared this recipe when the metaplanner selects the goals that should have a solution, some experiences have these goals. Then the solution will generate a plan that integrates the actions that the majority of agents have registered.

An additional test to reduce bias consists of not considering

actions with milk in their argument, supposing the user is allergic to this element. The solution obtained will be correct and verified analyzing the content of experiences registered. Some test variations were realized to calculate new results and the solutions were correct. The results are considered satisfactory because we have a plan with only the suitable actions. The next step is to select another domain and metrics to test new results that permit us to compare them with other tests in this area.

In conclusion the principal contributions are:

- A new method of mixture of arrays (majority)
- A model with a complete assortment of solutions (it considers all registered knowledge)
- A model with unconditional support to the user (given solution with realized goals for another problem)
- Solutions that supports absolute or relative majority in the experience

VII. RELATED WORK

Many universities and companies are interested in creating tools for decision support, because the transcendence seen in the decisions of human and business activities.

Some examples are: course evaluation in universities and trade research [4]; companies like Group Systems gives support to clients since 15 years ago for decision support [5]

The anonymity gives many advantages. Laboratory studies had showed that when a GSS is used to generate ideas, many ideas of high quality are produced using techniques of standard meetings.

In these studies it was proven that anonymity in GSS had better ideas when the people made positive and negative comments. The anonymity gives the opportunity for people to explore or criticize ideas without intimidation from partners or management. This encourages people to participate generating ideas without inhibitions [6]. Also the group work in teaching gives facilities to learn [7]. GSS allows the generation of ideas or solutions in big work groups like the seminar “Estrategias y detonadores para un cambio de modelo económico el caso Monterrey ciudad internacional del conocimiento 2004” [8] (Strategies and triggers for an economic model change, the case of Monterrey international city of knowledge in 2004).

In short, decisions in human and business activities, anonymity with positive and negative comments, and contributions without intimidation are characteristics that the metaplanner contains and it is possible to get results like the event mentioned [8].

In another comparison, the metaplanner effects planning in like fashion to the Kambhampati [9] in which two plans are compared and analyzed to see if there is a conflict or if they are different. After this, the plan is compared with all other plans, one by one in the same way.

The metaplanner evaluates the plans at the same time which is very important because many actions can be compared with all other actions of the same goal from other agents.

VIII. DIRECTION FOR FUTURE RESEARCH

The decision rooms like the Decisionarium [4] handle many extend services for individual and group decisions with ponderous ranking of people preferences and statistics for reaching decisions. If these kind of features could be incorporated in metaplanner, it would generate descriptions of the environment in which the decision was made and also could made other choices in agreement with the preferences of the user.

This kind of software will extend the scope of decisions in planning and the user can select in advance which kinds of experiences are desired as factors in the formulation of the problem solution applied to new problems.

REFERENCES

[1] Solis Luis F., *How to Improve Your Organization’s Group Intelligence*, CEO/President of Group Systems Corporative 2006

[2] Marakas, George M. *Decision Support Systems in the 21st Century*. Kelley Scholl of Business Indiana University. Prentice Hall 1999.

[3] Knuth, Donald E. *The art of computer programming*, Vol. 3 Sorting and Searching. Addison- Wesley 1972

[4] Hämäläinen:] R.P *Decisionarium - Aiding Decisions, Negotiating and Collecting Opinions on the Web*, Journal of Multi-Criteria Decision Analysis, Vol. 12, No. 2-3, 2003, pp. 101-110.

[5] *GroupSystems*, 520 Zang Street, Suite 211 Broomfield, CO 80021 USA 2006.

[6] Nunamaker Jay F., Jr., Robert O. Briggs, Daniel D. Mittleman and Douglas R. Vogel, *Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings* 1996

[7] Faulkner Rob, *Wired class lets everyone talk at once*. Mohawk software study intended to speed idea exchange. Local news at ASU Arizona State University 2005

[8] González Bernardo, Aréchiga, Guillermina, Benavides Briceño, Alejandro, *Estrategias y Detonadores para un Cambio de Modelo Económico: El caso Monterrey, Ciudad Internacional del Conocimiento*.2004.

[9] Kambhampati, S., *A Validation Structure-Based Theory of Plan Modification and Reuse*, Artificial Intelligence 55 1990