International Journal of Hybrid Information Technology
Vol.3, No.1, January, 2010

Multi-Agent/Multi-Goal Modeling Templates and the 1-1, n-1,1-n, n-n Deductions of Relationship-chain

Cai Zhiming, Yin Jun, and Zhou Yingjie

Faculty of Information Technology, Macao University of Science and Technology,Macao
zmcai@must.edu.mo, handsonyin2005@yahoo.com.cn
yjzhou.umac@gmail.com

Abstract

The Multi-Agent/Multi-Goal distributed selection modeling system supports experts at different sites to judge the solutions with many relationships in the visualized modeling. The implicit relationships in judgment modeling templates are hard to find by directly surveying the templates. The paper deducts the implicit relationship based on the dimensions established by AHP. Firstly, search for all the relationship-chains between any two objects; secondly, calculate the weight ratio of every relationship-chain by means of the weight of every object and every relationship. The detailed searching algorithm and calculation are given. Any relationship-chain between any two objects in one/more templates can be discovered, including 1-1, n-1, 1-n, n-n deductions.

Keywords: *Multi-Agent/Multi-Goal, Judgment Templates, Distributed Modeling, Relationship-chain.*

1. Introduction

The process to select the solution for a large and complicated issue is a multi-goal and multi-agent oriented, very complex judgment process. The agents (e.g. people, departments) must analyze many goals (e.g. cost, reliability) repeatedly, balance the advantages and disadvantages, then they can draw the final conclusion. The analysis process will last out among groups of agents, in different locations, with different phases, so it is always an intermittent, distributed and interlaced procedure.

The Multi-Agent/Multi-Goal judgment modeling is to build visual models of the objects and processes in the decision-making, it will give the agents a thinking-grid among a mass of goals, agents, solutions, conditions and others, so as to give advices and help about the final decision. [10,11]

Based on this background, the agents must collaboratively analyze and select solutions for complex issues in a distributed environment. The references [2-6] propose and implement a so-called MAMG distributed modeling system. With the support of the system platform, the experts at different sites may establish their judgment template, in a visualized modeling way, to represent various judgment models of analyzing objects and their relationships. Figure 1 displays a judgment template with judging on the relationships between conditions (Supply, Demand…), solutions (Gambling, Travel…) and goals (Productivity Increase…), for the issue “Pluralistic Selection on Future Industries of Macao”.

Judgment modeling is a continual finite process conducted by Baseline. The process of judgment and the result are stored in local, Intranet/Internet databases. Every expert can share their templates with others in different scopes, according to their own roles as well as priorities. Process conducting and alternation messaging are accomplished by collaborative

software agents (Master/slave software agent structure)[5]. After experts finish various judgment templates of an issue, the MAMG can use AHP[4][7], OWG[9] and SVM[18] to integrate the templates, in order to support the selecting of the optimal solution . This paper will focus on the deduction of implicit relationship in judgment templates during the integration.

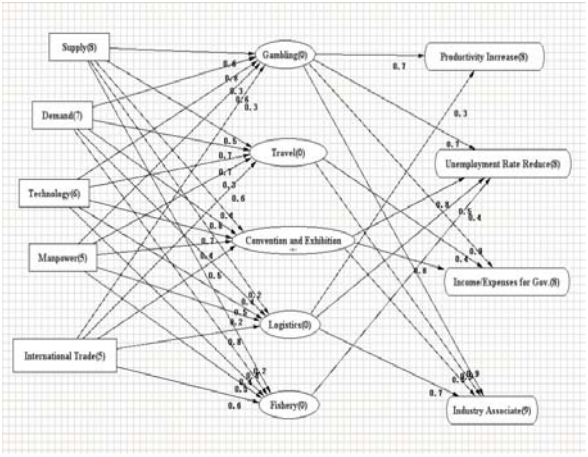

Figure 1. A Judgment Template

2. Group judgment modeling

MAMG modeling system has been constructed, following our modeling methodology and the principles of Group Decision Supporting. The distributed modeling development teams can use it to build visualized judgment models and evaluate solutions among goals, conditions, etc.

Since the users of our system would be in the different sites, the technology of distributed multi-databases and multi-agents are employed to organize the system. The user-agents denote members of the teams, which have been invited to develop the modeling. Each agent has its own workspace in database, and they can manage the work that they have done themselves. At the same time, many agents can cooperate in the modeling. They are connected by Internet/Intranet and do not need head meetings frequently, although they are not in the same place.

The system is made up of many “Soft-Agents” corresponding to “User-Agents”. The User-Agent indicates the users who may be the manager of the project/team, and also may be the team member. The Soft-Agent is a set of software with its own database. The projects' information, the models' information and the agent's information are saved in the database. The Soft-Agent can supply the help and conduct the User-Agent in judgment modeling. Each User-Agent can attend several projects and teams. Soft-Agents and User-Agents can keep in touch with each other frequently via Internet/Intranet.

The primary job of User-Agents is to analyze the solutions of the issue and create the judgment models. With the help of the system, the user can visualize their considering of solutions through the models. Besides this, the Soft-Agent can trace and control the User-Agent’s modeling process, such as: working status of each team and user, models audit and baseline conducting. Based on the information collected, the managers can discover the influencing factors during modeling, such as lack of consistency, limitation of time, effort,

manpower and so on. These will be very helpful to improve the modeling process and management.

The users are divided into three levels. The first level is the project manager, the second is the team manager, and the last one is the team member. System admin selects the project manager. All users are distributed on network. The project manager can create the project and decide who will be the team managers. The team managers can manage the members' creating, access right, baseline of the team, etc. When the project manager create the project, he/she should set up initial model "template" which presents how many objects the project has, for instance, which solutions, conditions and goals the project has. Base on the initial model template, the team members' working is to construct the models (, relationships, weight of objects/relationships, etc.) among many factors with their local database.

3. Objects and templates

The objects on the template include: Agents, Goals, Solutions and Conditions.

As mentioned above, the Agents denote two kinds of agents: user-agents and software-agents. User-agents represent the people or departments that will be involved in the selection process. Software-agents are running on the central and local platforms to conduct the modeling process. There are two kinds of user-agents: Key Agents (including project manager and team manager) and Common Agents. Key Agent has the higher right, it can create task database and judgment template, define other agents' layers and purviews, but it can't change other agents' contents of modeling templates and the data of their modeling processes. Common Agents can only work on its own template with the specified database, but it can check the contents of other relevant agent's templates according to its layer and rights. There are also two kinds of software-agents: Master and Slave software-agents, Master is with the central database and Slave ones are with Intranet database.

Goal is the objective that is wanted to achieve when the selection problem is being considered, such as profit, sociality, reliability, availability, etc. Goal can be further subdivided into sub-goals and part-goals.

Solution is the resolution that could be adopted to achieve the preconcerted goals. For example, the solutions of "Pluralistic Selection on Future Industries of Macao" can be "Gambling" solution, "Traveling" solution, "Logistics" solution, etc. Solution can be further subdivided into sub-solutions or part solutions.

Condition is the limitation that can limit the agents, goals and solutions, such as the limitations of time, money, human resource, etc. Condition can be further subdivided into sub-conditions and part-conditions as well.

The relationship between the ojects indicates the degree how much a object is supported or impaired by the other one. It has a power to illuminate the level of supporting. A positive value denotes support; a negative value denotes impairment or limitation.

Judgment modeling is a relationship evaluation modeling process in which each agent evaluates the relative relationships among the objects on its own judgment template. With the power of relationship, agent can express positive or negative evaluation and its degree of the relationship. The key agent will define the format and scope of the power when the first judgment template in the database is created, it is quite similar to the level of marking, 5 levels or 3 levels will be preferred. For example: 2, 1, 0, -1, -2, by which the 2 denotes very positive, 1 denotes medium positive, 0 denotes neutral, -1 denotes medium negative, -2 denotes very negative. The further explanation about some relationships can be linked to relevant documents. The evaluations made by one agent can only be modified by the agent

itself, other agents(including the agents with higher level) can just check them, modification is forbidden.

Judgment template can be categorized into overall selection template and local selection template. Overall selection template is a multi-dimensional interweaved relationship-template, which is formed by the collection of all the kinds of relationship models of different agents, it is impossible to be displayed by a two-dimension model on a plane. On the contrary, local selecting-template can be expressed by a two-dimensional graphic. Local selecting-template can also be a three- or multi-dimensional model interweaved by several relevant two-dimensional models. With the judgment templates, the potential path between any two objects or paths among several related points could be there. The path can be directly shown in the models. It can also be connected by many objects, even sub-agents, sub-solutions and sub-goals indirectly, which can't be found out by visible searching directly.

The templates of sub-models and classified-models can be extracted if the model is going to very large and complex; and Zoom in/out functions of the templates are supplied in MAMG system.

Actually, more than twenties notations have been proposed to present the objects and relations on the template. The detailed semantic notations and modeling process are depicted in other papers [2][3]. To be understood simply, only C-S-G judgment model is applied as sample template in this paper, which is consisting of condition, solution, goals and the relationships between them.

4. Explicit /implicit relationship

The term "explicit/implicit relationship", restricted to this paper, is defined as follows:

In the templates containing complex relationships, define all the relations as a set $\mathbf{R} = \{R^1, R^2, R^3, R^4 \ldots, R^n\}$, where the power of R represents the number of objects(conditions, solutions, goals) that have to be gone through from one object to another, that is, the steps of deduction to reach another object. If the power of R equals 1, which means two objects can get to each other using only one step of deduction, this situation is named as "shallow association" , i.e, "explicit relationship". However, if two objects need $\boldsymbol{n}$ steps of deduction to reach each other, we call this situation "$\boldsymbol{n}$-layer association". When $\boldsymbol{n}>2$, it is usually hard to find this kind of deep association by directly survey on the template, thus, we can say that an "implicit relationship" exists between these two objects. As you can see from Figure 1, can you judge the relationship between the condition "Demand" and the goal "Productivity Increase"? Since many paths through different inner nodes connect the two objects, it's really difficult to decide what their implicit relationship is, as well as how to compare this implicit relationship with other ones. For example, can you see which condition is more intimate with the goal "Productivity Increase", "Demand" or "Supply" or others?

Furthermore, for an issue, many experts will use MAMG platform to construct different templates according to their own understanding. With all these templates integrated, what are the exact implicit relationships of these objects, while the implicit relationships are included in different templates?

In the research of this paper, "implicit relationship" deduction does its calculation based on the dimensions established by AHP algorithm (see the quantity with each relationship in Fig 1, each quantity is hyper-linked to a document to explain the reason of the quantity). That is, find out the relationship-chains that mainly take conditions, goals and solutions as nodes, and quantify the weight of every relationship-chain in the relationship set according to the dimensions of AHP algorithm, and then deduct their implicit relationships. More information on the dimensions of AHP algorithm is provided by the reference [4].

5. The paths of relationship-chain

The "implicit relationships" deduction between objects mainly depends on the following two aspects:

Firstly, according to the starting object (e.g., condition) and the terminal object (e.g., goal) selected by a user, search for all the relationship-chains between them through the relationships described in the template.

Secondly, calculate the weight ratio of every relationship-chain by means of the weight of every object and every relationship.

With respect to searching relationship-chain, the dynamic programming and dynamic arrays/stacks are applied. The detail of the searching steps is as follows.

Step 1: Collect all the templates that participate in the implicit relationship, and then find out the objects within them as well as the code of each template;

Step 2: A user should select the start object and the terminal object. After requesting for their implicit relationship deduction, the software will firstly search for the corresponding code of the selected objects, as well as the code of the current template being calculated, and then these codes will be sent to the operation-FindObjectLine() to calculate.

Step 3: In the operation FindObjectLine(), establish a data structure necessary in the calculation of this algorithm: Firstly, create a stack and an array composed of many stacks; Secondly, for the data stored in the stack, create a class (Class ObjectData) to store the code, type, text and weight of every object.

Step 4: Initiate the first layer stack in the array and push all the objects that are related to the start object to this first layer stack.

Step 5: Expand the upper boundary of the array and initiate the second layer stack. Starting from the top element on the first layer stack, search for all the objects related to it and push them to the second layer stack.

Step 6: Recursively repeat step 5 and establish layer two, three … stacks. If the top element of a stack at a specific layer is the same as the terminal object, then a relationship-chain has been found out, the nodes of which is all the top elements of each layer's stack. After outputting the result, pop up the top elements of each stack and continue calculation. If a stack of a specific layer is empty, it means that this branch has been searched through, so minus the upper boundary of the array by 1 and start calculation of another branch. Doing this way, all the relationship-chains between the start object and the terminal object can be found out from the current template.

Step 7: Repeat step 4, step 5 and step 6, take different template's code as the parameter of the operation FindObjectLine(), and all the relationship-chains with the same start and terminal objects will be figured out in a series of templates.

From the steps of searching relationship-chain, we can see that the algorithm makes use of dynamic programming and stacks as well as dynamic arrays to solve the problem similar to "N-Queen" problem. The advantage of this algorithm is clear and relatively simple while the defect lies in that it requires huge amount of calculation. Besides, the structure of the database needs to be matched. In order to achieve this purpose, the design of various algorithms and their need to database must be taken into account as a whole. As for the database design of the system, please refer to the references [3,4].

6. The weight of paths

After finding out the relationship-chains, rational calculation of the relationship-chain, by making use of the weight of objects and relationships, needs to be carried out in order to provide a rational supporting for the observation and comparison of the implicit relationship as well as the selecting of the final solution. Since the weight of objects and relationships are based on dimensions given by the AHP algorithm, they can be used as a reliable basis of the deduction. Based on these facts, this paper proposes a method of calculating the relative weightiness ratio of relationship-chain, to expose the relative ratio of the relationship-chain found in a single template as well as a serial of templates, so that experts can understand relative weightiness ratio of each implicit chain with this quantified basis. Detailed calculation process is as follows.

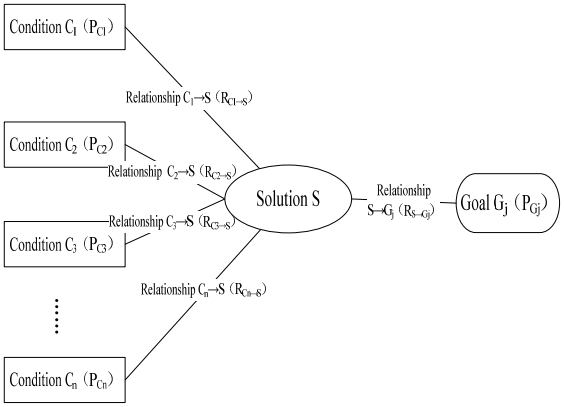

Figure 2. C-S-G Template

According to the judging of the conditions, goals and selectable solutions in a project, an expert can establish a template as shown in Figure 1. Figure 1 shows an example of a judgment template called “Pluralistic Selection on Future Industries of Macao”. Here a definition is made:

A template may contain multiple condition objects, all of which are defined as a set: $C_i = \{C_1, C_2, \ldots, C_n\}$ (see figure 2). The weight on each condition object is also defined as a set: $P_{Ci} = \{P_{C1}, P_{C2}, \ldots, P_{Cn}\}$. Besides condition objects, there also exists a goal object G_j, the weight of which is P_{Gj}, as well as a solution object S. Some relationships exist between these objects. Here we define the relationship between the condition object and the solution object as a set: $C_i \to S = \{C_1 \to S, C_2 \to S, \ldots, C_n \to S\}$, the weight of which is $R_{Ci \to S} = \{R_{C1 \to S}, R_{C2 \to S}, \ldots, R_{Cn \to S}\}$; the relationship between solution and object is defined as $S \to G_j$, the weight of which as $R_{S \to Gj}$.

In a complicated template, there may be lots of goals and solutions. To simplify the analysis of the relationship-chain ratio, we only refer to one solution and one object in figure 2.

With the above definitions, suppose the start object of the relationship-chain is a condition object C_i, and the terminal object is a goal object G_j, a formula calculating the relationship-chain ratio H is given:

$$H = \frac{P_{Ci} R_{Ci \to S}}{P_{C1} R_{C1 \to S} + P_{C2} R_{C2 \to S} + \ldots + P_{Cn} R_{Cn \to S}} \times P_{Gj} R_{S \to Gj}$$

In this formula, the product of a condition weight and its relationship-chain weight implies the supporting degree of this condition to the solution. Similarly, the product of a goal weight and its relationship-chain weight is defined as the supporting degree of the solution to the specific goal. Thus,

$$\frac{P_{Ci}R_{Ci\to S}}{P_{C1}R_{C1\to S} \div P_{C2}R_{C2\to S} + ... + P_{Cn}R_{Cn\to S}}$$

represents the supporting degree of the selected condition to a specific solution in its relationship-chain, and $P_{Gj}R_{S\to Gj}$ can be thought of as the supporting degree of a solution to a goal. Further more, the relationship-chain ratio H means the supporting degree ratio of this relationship-chain (from Ci to Gj) to all the corresponding relationship-chains in the template.

The several relationship-chains may exist for an implicit relationship in a template, from the start object to the terminal object. By summing up these relationship-chain H ratios, the relative weightiness of this implicit relationship can be gained in this template. The significance of an implicit relationship can be compared with others by the relative weightiness of this implicit relationship in the template. Likewise, if we expand from a single template to a set of templates, we can get the relative weightiness ratio of the implicit relationship and the significance of an implicit relationship in this set of templates.

7. The deduction

In MAMG system, there are four types of the deduction: (1-1), (n-1), (1-n), (n-n) deduction of implicit relationship.

7.1 (1-1) Deduction

(1-1) deduction is the deduction from a object to another object, Figure 3 shows a part of an implicit relationship (1-1) deduction results in ten templates (v1-v10, which are different templates of “Pluralistic Selection on Future Industries of Macao”, similar to the template in Figure 1.) from the object “Demand” to the object “Industry Associate”. Restricted to the paper’s length, only the deducted data of v7-v10 and the final “Total Relationship Degree” are shown. This picture illustrates the relationship-chains of the implicit relationship from the start object to the terminal object, as well as their corresponding H ratio (called "Relationship Degree” on the screen) in every template. Thus, in template 8, the implicit relationship from object “Demand” to object “Industry Associate” is the mostly supported (10.843), while template 10 is the least supported (7.9). The final “Total Relationship Degree” integrates the supporting degree (H ratio) of the implicit relationship in all ten templates. The relative supporting degree of the implicit relationship from object “Demand” to object “Industry Associate” is 84.884, the realistic meaning of which needs to be compared with other implicit relationships. As an example, we can further deduct the implicit relationship from object “Demand” to object “Productivity Increase”, the result of which is shown in Figure.4:

The final “Total Relationship Degree” is 37.354, which means, in integration of all the results of the ten templates, the supporting degree (84.884) from object “Demand” to object “Industry Associate” is more than twice of that (37.354) from “Demand” to “Productivity Increase”. That is, the previous implicit relationship is more important than the latter, which is very hard to find out by human intuition. Following this way, more implicit relationships and their H ratio can be calculated and then sorted so as to give precise basis for further analysis and decision-making.

Figure 3. (1-1)Deduction from “Demand” to “Industry Associate”

Figure 4. (1-1) Deduction from “Demand” to “Productivity Increase”

7.2 (n-1) Deduction

Based on the previous (1-1) deduction, n-1 deduction is the deduction from all the condition objects to the same goal object in the C-S-G templates, and this process is named as "the longest (n-1) path" deduction. There exists a longest path for each condition to a goal in a template. The "longest" path means the largest value of ($R_{C->S}$ + $R_{S->G}$), i.e. the relationship-chain weight. Users are not required to select condition objects. What they only need to do is to choose a goal object. Then the program will find out the longest path through the most frequently used solution object from all the conditions to the selected goal.

The meaning of this kind of deduction is that it aggregates all the paths from all the conditions to the same goal object and selects the optimal path from each condition to the goal. To better understand this concept, we'll give an illustration here:

Suppose the goal object is: Income/Express for Gov. and all the default condition objects are listed on the left side on C-S-G templates, and then the output can be derived in Figure 5:

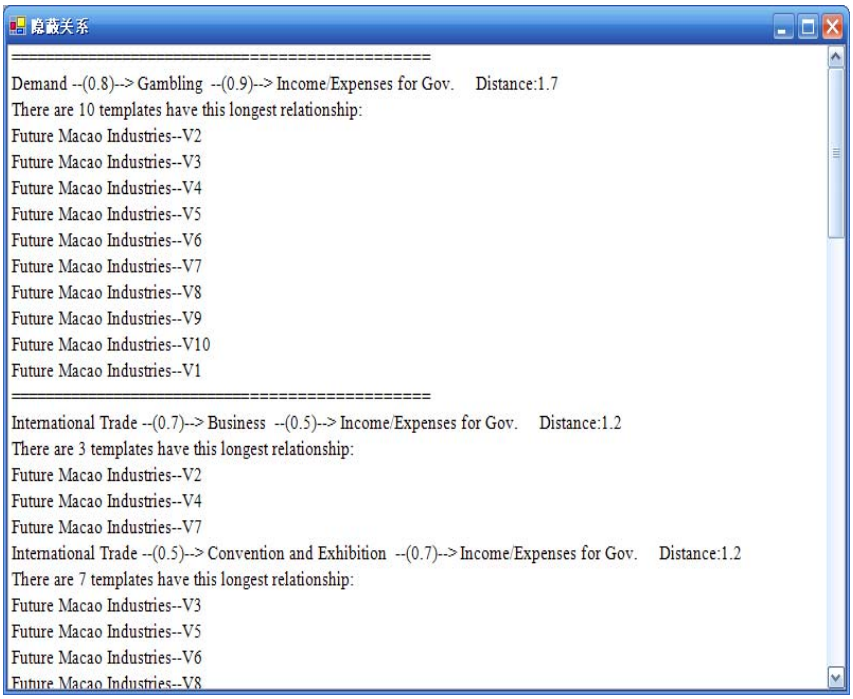

Figure 5. (n-1) Deduction Process (longest paths)

The Figure 5 shows: The times of each longest path linked by the solution in each template.

Roll down the scroll bar in figure 5, we'll see the following salient conclusions in Figure 6. What these conclusions imply is: it is prior to select the condition "Demand" for the goal "Income/Express for Gov.", and, it's the most feasible to implement the solution "Demand(0.8)→Gambling(0.9) →Income/Express for Gov".

It is also meaningful to calculate the shortest path in this way. We are enabled to select the least cost path to a specific goal from all the conditions. The example in Figure 7 demonstrates the process and the conclusion of this kind of calculation.

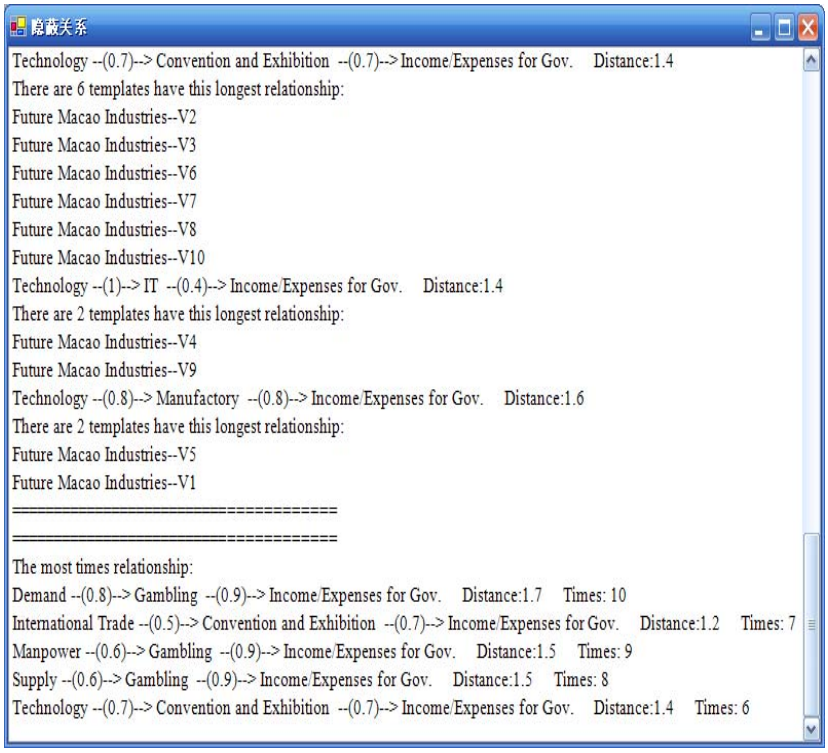

Figure 6. (n-1)Deduction Summary (longest paths)

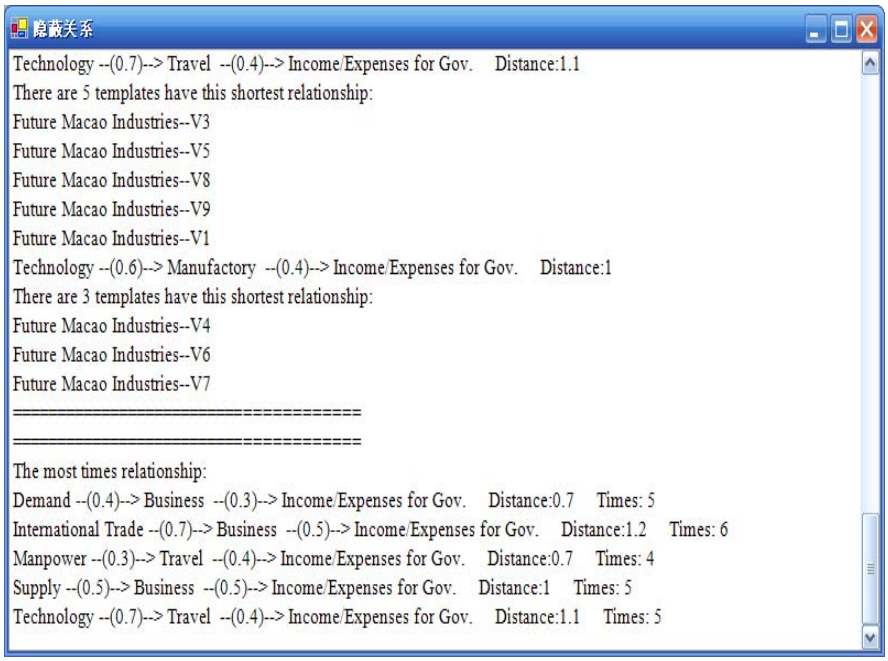

Figure 7. (n-1)Deduction Summary (shortest paths)

7.3 (1-n) Deduction

Figure 8. (1-n) Deduction Process (longest paths)

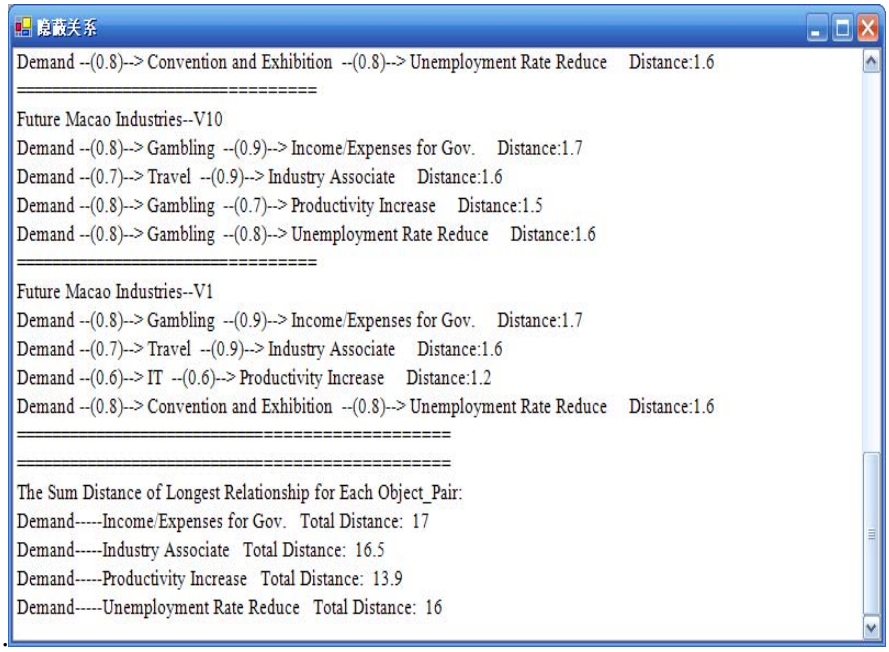

Figure 9. (1-n) Deduction Summary (longest paths)

Roll down the scroll bar in figure 8 and the conclusion will be shown after the two rows of the symbol "==" in Figure

(1-n) deduction is defined as the deduction from one condition object to all the goal objects, which is named as "the (1-n) longest path". When a user picks out a condition, calculation is default to take all the goals into account.

Figure 10. (1-n) Deduction Process (shortest paths)

Figure 11. (1-n) Deduction Summary (shortest paths)

The longest paths from a specific condition to all the goal objects are calculated. All the longest paths iteratively are added up in all the corresponding templates. The comparison of

the addition results will expose the support degree of a particular condition to different goals. Let's see an instance:

The condition chosen is "Demand" and all the goals are default. The longest path from "Demand" to all the goals is shown in Figure 8.

It is shown that "Income/Express for Gov." is most supported by "Demand".

Likewise, the shortest path from a condition to all the corresponding goals can also be calculated. This is called "(1-n) shortest path".

The calculation in Figure 10,11 indicates that the cost from the condition "Demand" to the goal "Productivity Increase" is the minimum.

7.4 (n-n) Deduction

(n-n) deduction is defined as the deduction from all the conditions to all the goals, which is called "(n-n) longest path aggregation". It collects the statistics of all the conditions and goals, and then selects the most feasible goal to each single condition to facilitate the decision making. The calculation of this process is relatively complicated since it's necessary to calculate the longest paths from all the conditions to all the goals.

Finally the conclusion can be summarized as: find out the most feasible path for every single condition to a goal.

Following is an illustration:

Selecting all the conditions and goals by default, the output of the program is shown in Figure 12.

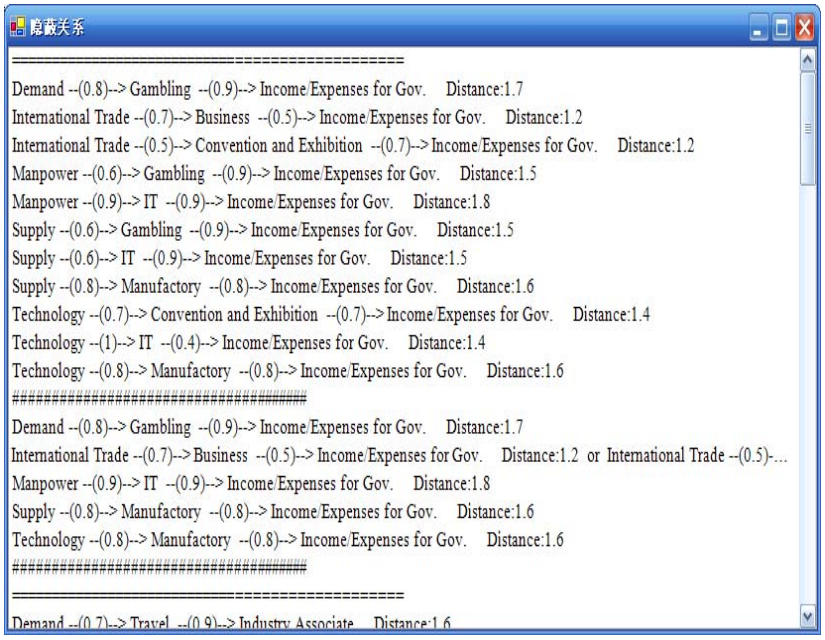

Figure 12. (n-n) Deduction Process (longest paths)

(n-n) deduction is the calculation of the longest path from all the conditions to all the goals. "All the conditions, all the goals" indicates three-phase longest path extraction. Firstly, calculate the longest paths from all the conditions to a specific goal and present the longest paths derived from different templates; Secondly, compare the longest paths from the same beginning condition to the same ending goal and choose the longest one (at this moment, M*N longest paths are left, where M is the number of condition objects while N is the

number of goal objects); Thirdly, find out the longest paths (from the same conditions to different goals), compare them and select the longest (meanwhile, M longest paths are left if no exception happens). “No exception” means no paths are of the same length. If such paths exist, they’ll be shown on the same row.

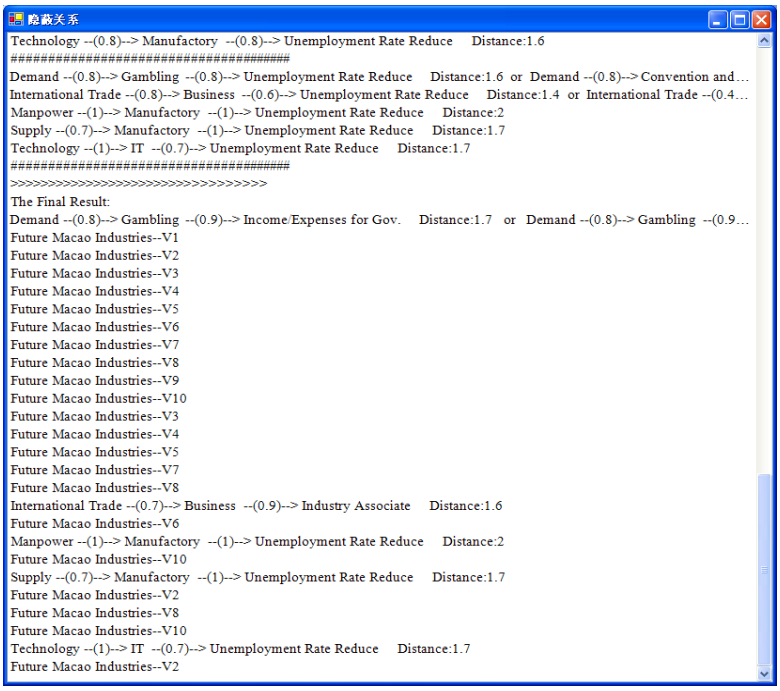

Figure 13. (n-n) Deduction Summary (longest paths)

In the summary of Figure 13, the templates IDs of each path are also listed to make users more convenient to check.

Similarly, n-n shortest paths can also be calculated. An instance is shown in Figure 14, 15.

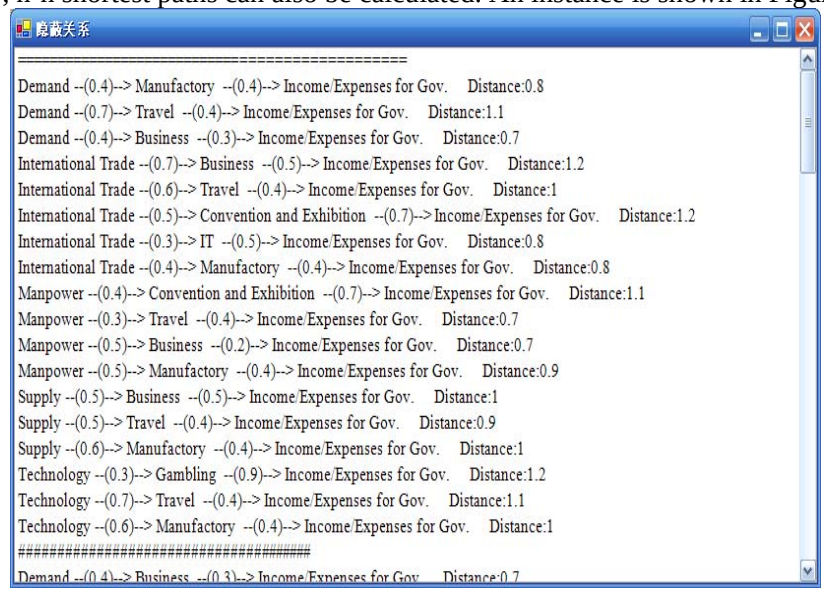

Figure 14. (n-n) Deduction Process (shortest paths)

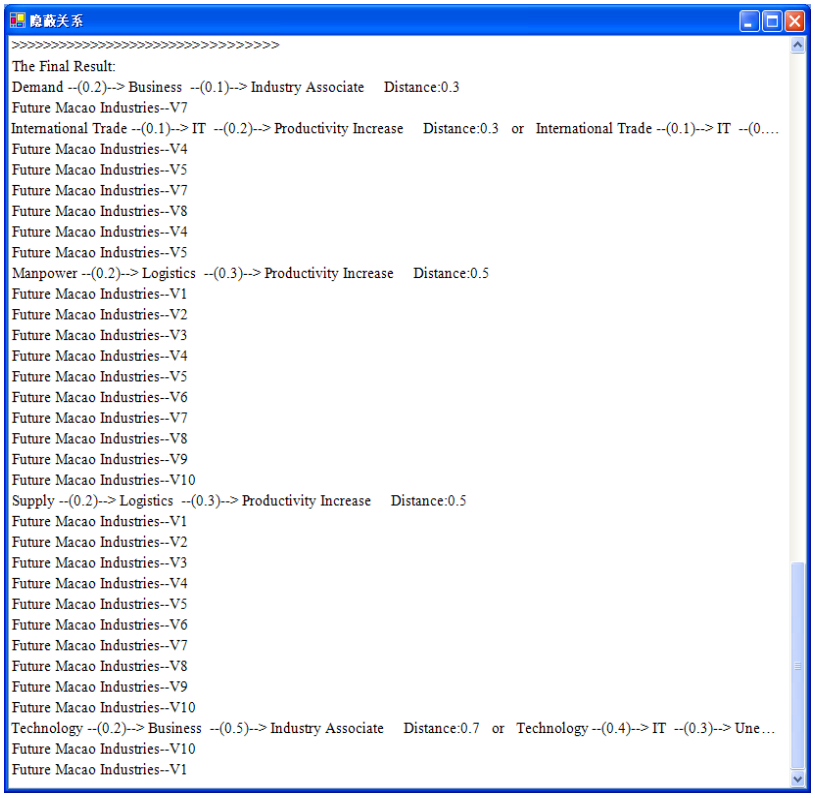

Figure 15. (n-n) Deduction Summary (shortest paths)

8. Conclusion

MAMG system provides a way to distributed group modeling for the judgment of solutions. It helps the experts to share, trace, review and conduct the modeling process, the members' work, the projects' baseline, and so on. It makes the modeling more visualized, traceable, controllable and efficient, especially for the organization and enterprise whose members are in the different sites.

With the implicit relationship deduction provided by MAMG system, any implicit relationship between any two objects in one or more templates can be found out and compared. For instance, we can seek the solutions that can achieve an expected goal with some concerned conditions by the deduction. We can also evaluate the relative significance of each implicit relationship-chain by calculating each one's relative *H* ratio. Thus, project experts get a measurable way to analyze and compare complicated implicit relationships that are hard to be straightforwardly observed. We will try to deduct more implicit relationships, for example, between any two objects in different templates. The other methods to compute *H* ratio will be researched further. 1-1, n-1, 1-n, n-n Deductions of Relationship-chain can supply the assistance to discover the longest and shortest paths between any condition and any goal in relative templates.

References

[1] Beroggi, G.E.G. “Visual-interactive decision modeling (VIDEMO) in policy management: Bridging the gap between analytic and conceptual decision modeling.”, *European Journal of Operational Research,128(2),* 2001, pp.338-350

[2] Cai, Z.M., Guo, X.Y., "Multi-Agent and Multi-Goal Oriented Decision Modeling and its Supporting System on Network." *International Conference on advances in Internet, Process, System and Interdisciplinary*, Italy, 2005

[3] Cai, Z.M., Guo, X.Y., "Distributed Modeling of the Solution Selection and Its Cooperative Agents." *Sixth IEEE International Conference on Electro/Information Technology*, USA, 2006

[4] Cai, Z.M., Huang, L.L. and Yin J., "The Concentration and Distribution of Visual Group Selecting Templates and Sorting Solutions by AHP.", *The Journal of MUST*. 2007, 1(2), pp.13-21

[5] Cai, Z.M., Yin, J., "The Process Conducting and Member Audit in the Distributed Enterprise Modeling." *2008 IEEE Asia-Pacific Services Computing Conference*. 2008, Taiwan

[6] Cai, Z.M., Yin, J., "The Selection Modeling System with Grouped Agents." *The IEEE 22nd Int'l Conf. on Advanced Information Networking and Applications.* 2008, Japan

[7] Dyer, D. F., Forman, E. H., "Group Decision Support With AHP." *Decision Support System,*1992,Vol.8, pp.991-1024.

[8] Earl, M. G., D’Andrea, R. "Modeling and Control of a Multi-agent System Using Mixed Integer Linear Programming" *Proc. IEEE Conf. Decision and Control,* Las Vegas, Neveda, 2002. pp.107-111.

[9] Fan, Z.P., Jiang, Y.P., "Approach to group decision-making with different forms of preference information based on OWG operators." *Journal of Management Sciences in China*, 6(1), 2002, pp.34-41

[10] http:// www.decisionmodeling.com.

[11] http://www.vanguardsw.com/solutions/application/modeling-and-simulation/

[12] Hu, W.B., Meng, B., "Research on the Key Technology of Distributed Intelligent Group Decision Supported System Based on Multi-Agent." *Computer Engineering,* 2006.3, pp. 23-28

[13] Kuipers, B. J., "Qualitative Simulation". *Artificial Intelligence*, Vol.29, 1986 pp.289~338.

[14] Miettinen, K., Lotov, A. V. "George K. Kamenev and Vadim E. Berezkin. Integration of Two Multiobjective Optimization Methods For Nonlinear Problems." *Optimization Methods and Software,* Vol.18, No.1,2003, pp.63-80.

[15] Tan, Y.J., *Quantitative Analysis Method.* China Renmin University Press, Beijing, 2002

[16] Tan, Z.D., Wang, W.P., Meng Q.S., "Analysis of the Relationship Evolution of Knowledge-sharing Among Networked Enterprise Clusters." *Chinese Journal of Management*, 2006 3(1), pp.12-16

[17] Turban, E., Aronson, J.E. *Decision Support Systems And Intelligent Systems*, 5thED. Prentice Hall, 1998

[18] Vapnik, V. N., Golowich, S., Smola, A., “Support vector method for function approximation, regression, estimation, and signal processing.” *Advances in Neural Information Processing Systems*. Cambridge, MA:MIT Press, 1997.

[19] Wang, Q., Shen, Y.P., Chen, Y.W., “A Method for Intelligent Decision Making Based on Support Vector Machines.”, *System Engineering*, 23 (10), 2005, pp.45-52

[20] Yang, S.L., *Intelligent Decision Making Method and Intelligent Decision Making Support System.* Science Press, Beijing, 2005

[21] Yager R R, “On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making.” *IEEE Transaction on Systems, Man and Cybernetics 18*, 1998, pp:183-190

[22] Zha X. F., “A knowledge intensive multi-agent framework for cooperative/collaborative design modeling and selection support of assemblies”, *Knowledge-Based Systems* 15, 2002, pp.493–506