

The Agents Coordination and Templates Aggregation in Distributed Modeling

Cai Zhiming¹, Yang Zhe¹, Wang Menghan² and Yin Jiangling³

¹*Faculty of Information Technology,*

Macao University of Science and Technology, Macao

²*University of Florida, Department of Computer Information Science and Engineering, USA*

³*University of Central Florida, Department of Electrical Engineering & Computer Science, USA*

zmcai@must.edu.mo, menghan@cise.ufl.edu, jyin@eecs.ucf.edu

Abstract

To group the agents from various areas for the solution modeling of large-scale, sophisticated systems or issues, we have developed a distributed modeling methodology and its networked supporting platform [1]. The upcoming problems are, however, how to coordinate (organize, supervise, evaluate) such distributed modeling agents, and how to aggregate a number of modeling templates for the best solution(s). The Soft-Agents system is designed to perform coordinating user-agents teams. Such coordination enables judgment on working characteristics and modeling quality of each team and individual separately. To work out the best solution(s), the individual template is aggregated by using Analytic-Hierarchy-Process and multiple templates are aggregated by the Ordered-Weighted-Geometric algorithm.

Keywords: *Distributed Modeling, Agents Coordination, Templates Aggregation*

1. Introduction

The visual modeling of a complex system or issue usually involves many agents: experts, officers, system analyzers, and developers from many areas in different sites. Our distributed modeling methodology and its networked supporting platform enable these people to conveniently interact with each other and help them to build modeling templates by evaluating the solutions of the system or issue [1, 3, 4]. The difficulties are how to coordinate (organize, supervise, evaluate, optimize) such distributed teams of agents, and how to aggregate a number of modeling templates the agents produced to conclude the best solution or solutions.

Many visual modeling tools are there such as PowerDesigner [7], Rational Rose [8] and Eclipse [9]. However, these tools, lacking support of the distributed teams of agents and aggregating of templates, have the following problems:

(1) Due to the complexity of system modeling, the organizing of the agents is the first of all difficult problems. Who should participate? How to group the agents? How to assign appropriate roles on the basis of each agent's previous experience and performance? These problems all arise frequently.

(2) In distributed modeling, every agent will work according to his/her own time schedule and habit, which makes coordination difficult. For example, a particular agent's modeling work might be affected by delay of some other agent, because their modeling is related. Therefore, the following problems are proposed: how to share the working progress of each agent in a timely fashion? How to evaluate their progress of the modeling?

(3) The project manager (team leader) often develops a schedule with referencing his/her own considerations, requiring group agents to submit the corresponding template versions according to the schedule. But how does the manager know the specific situations under which each agent finish the required work? For example, is the submission an outcome after days of serious thinking, or a rush at the last minute? In addition, how to measure the quality of modeling of the entire team by summarizing the performance of each agent?

(4) For some heavy modeling and other work, even when the group agents have the intention to finish the work on time, they may still fail. Can the platform coordinate, conduct and organize group agents to complete their work orderly and timely, rather than rushing at the last minute? How the managers rethink and improve the schedule by supporting of the platform?

(5) How to summarize the various kinds of information of group agents' work? How to give appropriate and integrated evaluation of individuals and distributed teams, and make use of the information in future to construct suitable teams for similar projects?

(6) A lot of templates will be produced once every agent evaluated the solutions of the system/issue on their modeling templates. It's not possible to check out the conclusion directly that which solutions are best or better with so many objects, relationships and chains on so complicated templates [2]. How to find out best solution(s) by aggregating individual template and grouped templates?

The paper is to develop a unique way to coordinate the involved agents and aggregate the templates. The methodology and support platform introduced in [1] are extended to solve the above problems, which helps the organization, evaluation and optimization of the agents in distributed modeling. It also enables sharing and supervising, as well as tracking of each agent's information such as the modeling process, updating history, working intensity, etc. The templates are aggregated by making use of GDS (group decision supporting) algorithms.

2. The Soft-Agents

In order to help managing distributed agent teams, a Soft-Agents network [6, 10] has been constructed, which uses "soft-agents" software to collect and coordinate the modeling work of each user-agent. See Figure 1.

Once a user-agent is registered and logon, the corresponding soft-agent will automatically collect and process the information related to the agent. The soft-agent includes 4 components: The local database where the related data are stored, information processors of project, models and user-agents.

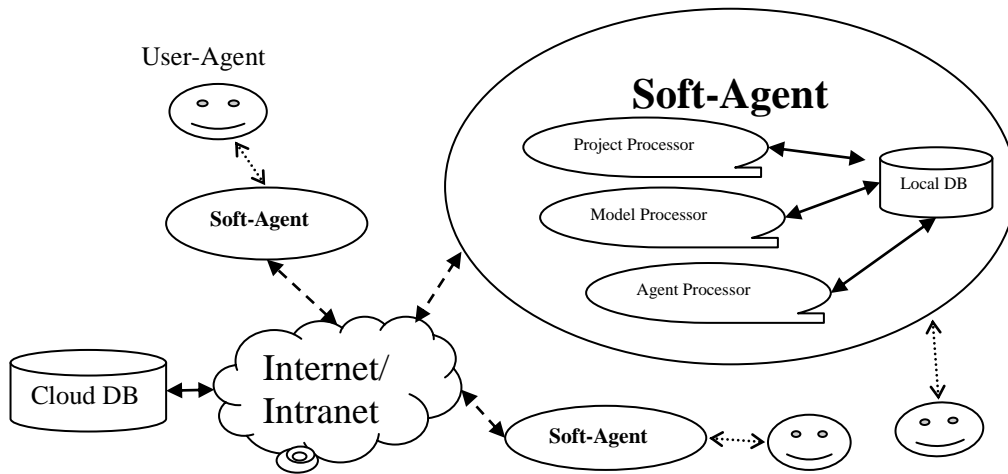


Figure 1. The Construction of Soft-agents

Any user-agent is working with the coordinating and supervising of Soft-Agents. For examples, the user-agents may issue many activities and deliver their outcomes: they may construct many own modeling templates. The modeling activities (like addition/deletion of an object and relationship) will be automatically announced to others in the agent team. Once updated version of a template is settled down by a user-agent, the template will be released and shared via soft-agent network, the other team agents can download, modify, re-submit the template. In order to ensure the work to be finished before a specified deadline, an automatic alert email for deadline will be sent by soft-agents.

3. The Modeling on Templates

3.1. The team constructing



Figure 2. The Team Constructing

The project manager, the leader of the agent team, may create an initial agent network like in Figure 2.

changes have been done. In addition, once any agent updates a template, the platform will notify other related agents in the team. The platform also has a conversation module supporting the agents to talk each other online.

4. Coordinating Agents

During the modeling, the templates will surely have many versions updated. Each version of template, together with related information, will be saved in databases for tracing and auditing. A diagram generated to check the versions of templates is showed in Figure 4.

The diagram indicates when the agent created the templates, how much time he/she spent for each version, and when the deadline (dashed red lines) of each version is. The managers can check the progress and supervise the agents following the information from the diagram. They can also review the modeling process step by step to evaluate the templates repetitively (see the button “Re-Play” on top-right corner on Figure 3). To coordinate the modeling properly, the soft-agents will release the reminding-bars to the related agents when they log in the system before deadline and haven’t submitted required version, or send reminding emails if the agent does not log in the system during this specific period.

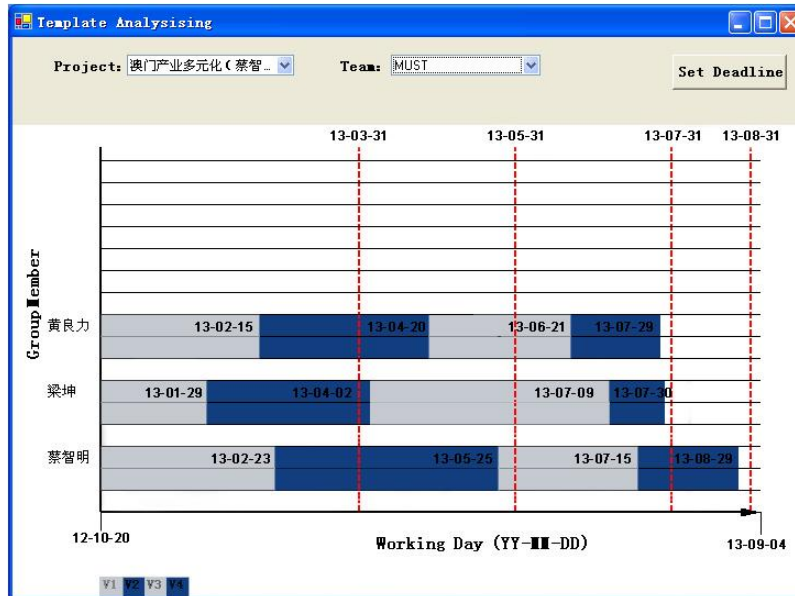


Figure 4. The Templates Audit Diagram

The soft-agents also trace the activities of each agent in the team during the modeling, and deliver the information to other team agents. With these supporting of soft-agents, the agents can learn about each other’s considering and progress in time, and promote mutual coordinating and supervising. (For example, one agent may be urged by the updated templates from other agents, and improve his/her own efficiency).

The “Online-Intensity” diagram in Figure 5 is showing the working intensity of each agent in each team. The y-axis represents the Working Time Duration and the x-axis indicates each Working Day. The time-slot and the team to display can be specified. By using the diagram, the managers can check how much time the agents work in each day during a specific period.

A “bar chart” showing detailed modeling activities can pop up by click “Team Detail” button. Each agent’s working style will be evaluated by the records in the database: lazy,

normal, or hard working. These records will be useful in selecting team agent for the future modeling.

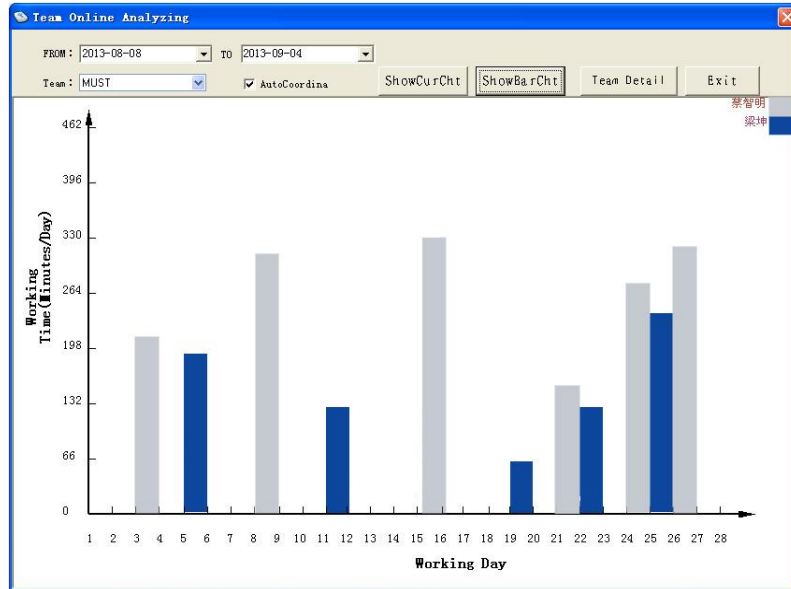


Figure 5. The “Online-Intensity”

Supervising is also an important issue of coordinating the networked-team in distributed modeling. A precise supervising for participants’ historical performance could be a good reference for future. For instance, once a new project is assigned to the team, the key manager/leader must decide who is the project manager, the project manager must select team managers, and team managers need to select the team agents. How do they decide which candidate in the team is suitable for which proper position? The decision can be much supported with the accumulated information of each team agent in our databases.

The agent’s performance is evaluated based on the modeling they have participated in the platform. The each agent will be judged from the team he/she joined, his/her working habit, specialty, working intensity, outstanding achievement and so on. With the information from above audit/tracing components of soft-agents, the platform can assist the managers to find whether the agent is suitable for a certain position.

Specifically, one of our supervising way is to compute each agent’s modeling sedulity and complexity, which come from data recorded, such as: number of templates (T) the agent created, objects(O) and relationships (R) on each modeling template, updating times (U), number of message (M) he/she sent and working time (W) for each template. The computing approach is:

$$\text{Score}[T_i] = \zeta * (O+R) * U * M / W$$

ζ is adaptive ratio of the score. The basic idea of the computing is from the Least-Recently-Used updating regulation employed in many operating systems.

Total Score of each agent is sum of Score $[T_i]$ with each template he/she built. Fig. 6 is an example of the score computing.

These information and assessment results will be used as reference to construct a suitable networked-team for the relevant organizing of agents for distributed group modeling in the future.

Member	Model	Create Time	Release	Rel-Line	Object	Updated	Messages	Working Days	Sedulity	Complex	Score
David	Future Macao Industries-V2	2008-4-1	08-04-02 12:3...	0	1	2	15	1	1.2166666	0.15	0.123287671232877
David	Future Macao Industries-V3	2008-4-1	08-04-01 08:5...	14	2	4	15	1	1.55	0.65	0.419354838709677
David	Future Macao Industries-V1	2008-4-1	08-04-02 08:1...	14	3	3	16	1	1.57	0.8	0.509554140127389
Jack	Future Macao Industries-V2	2008-4-2	08-04-03 03:2...	47	17	15	258	1	3.25	4.55	0.714285714285714
Jack	Future Macao Industries-V4	2008-4-2	08-04-03 03:2...	53	17	28	556	1	3.25	4.55	0.714285714285714
Jack	Future Macao Industries-V3	2008-4-2	08-04-03 03:2...	46	17	12	394	1	3.25	4.55	0.714285714285714

Figure 6. The Score of Team Agent

5. Templates Aggregation

Once every agent submitted their own evaluation on their modeling templates, a number of templates will be produced. Though each agent evaluated each solution by considering the relationships among objects on their templates, it's not possible to check out the conclusion directly that which solutions are best or better with so many objects, relationships and chains on so complicated templates [2]. Because AHP (Analytic Hierarchy Process) and Ordered Weighted Geometric (OWG) [9] have proven effective in group decision supporting, so AHP is employed to find out best solution by aggregating individual template and grouped templates are aggregated by OWG to work out the best solution or set of best solutions. The detailed information and algorithm are researched in paper [5, 3]. The following is the brief introduction.

For each individual template, the solutions on the template are ranked by AHP aggregating:

- (1) Each goal on the template is decomposed into its constituent parts, progressing from the general to the specific. In its simplest form, this structure comprises a goal, criteria and solution levels.
- (2) Each solution would then be further divided into detailed levels, recognizing that the more criteria included, the less important each individual criterion may become.
- (3) Assign a relative weight to each one. Each criterion has a local and global priority. The sum of all the criteria beneath a given parent criterion in each tier of the template must equal one. Its global priority shows its relative importance within the overall template.
- (4) Put the information into the template. Scoring is on a relative basis, not an absolute basis.
- (5) Compare one choice to another. Relative scores for each choice are computed within each leaf of the hierarchy.

(6) Scores are then synthesized through the template, yielding a composite score for each choice at every tier, as well as an overall score.

After ranking the solutions with AHP in each template, OWG arithmetic operators are used to reach an approximate conclusion among multi-templates, which is the best solution or which is not-worst solution.

Suppose the number of agents is l and the solutions are n . Every agent evaluates each solution on the templates. The aggregation can be formulized as following:

(1) Availing Value U

Here is $U = (u_1, u_2, \dots, u_n)$, $u_i \geq 0$, $\sum_{i=1}^n u_i = 1$

The larger u_i is, the local-better solution on the template is.

(2) Ranking-Relationship Value O

The value $O = (o_1, o_2, \dots, o_n)$, $o_i \in N$

The less O_i , the local-better solution is.

(3) Fuzzy Preference Relation Matrix

$P = (p_{ij})_{n \times n}$ is the fuzzy preference relation matrix and it presents the compare between two solutions. p_{ij} can be explained to which is better between the solution x_i and solution x_j . And $1 \leq i, j \leq n$, $p_{ij} \geq 0$, $p_{ij} + p_{ji} = 1$.

(4) Complementary Judgment Matrix

The matrix $A = (a_{ij})_{n \times n}$ is the complementary judgment matrix. It presents the compare of two solutions. The a_{ij} presents which is more important between solution x_i and solution x_j . And $1 \leq i, j \leq n$, $a_{ij} > 0$, $a_{ij} \cdot a_{ji} = 1$.

As per the above evaluating-format, the complementary judgment matrixes are produced and the group-concentrating vector can be computed. The OWG arithmetic operators are used to concentrate and gain the group preference. The weight of first-choosing OWG arithmetic operators can be figured out, which will make the best-solutions aggregate, the not-worst solutions aggregate and the approximate global-best solution(s) worked out.

Figure 7 gives an example, 4 agents built the templates (V1,V2,V3,V4) concerning the issue "The Diversification of Macao Industries". However, the agents can't see the conclusion which solution(s)-industries is/are best since so many objects/relationships on the templates and so much data with objects/relationships from statistics or agents' judgments. The aggregations by AHP and OWG can reason the templates to reach the conclusion. The computation of AHP/OWG is demonstrated on Figure 8, with the results of AHP ranking, best solution's aggregate, the not-worst solution's aggregate and the final best solution.

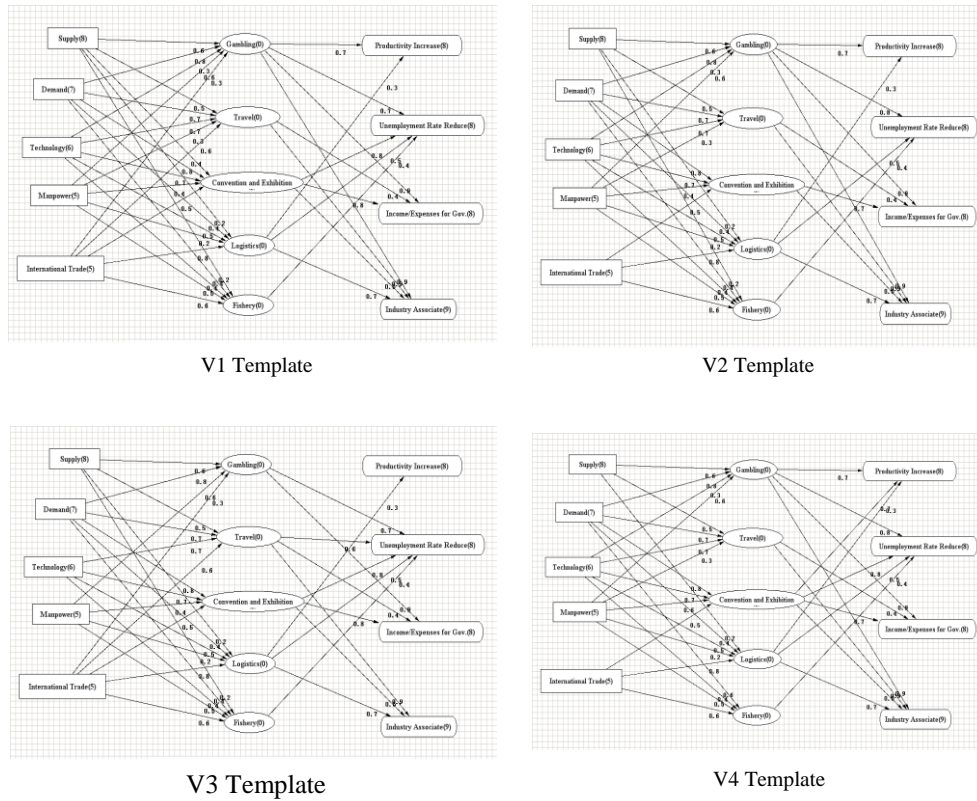


Figure 7. The Templates of “The Diversification of Macao Industries”

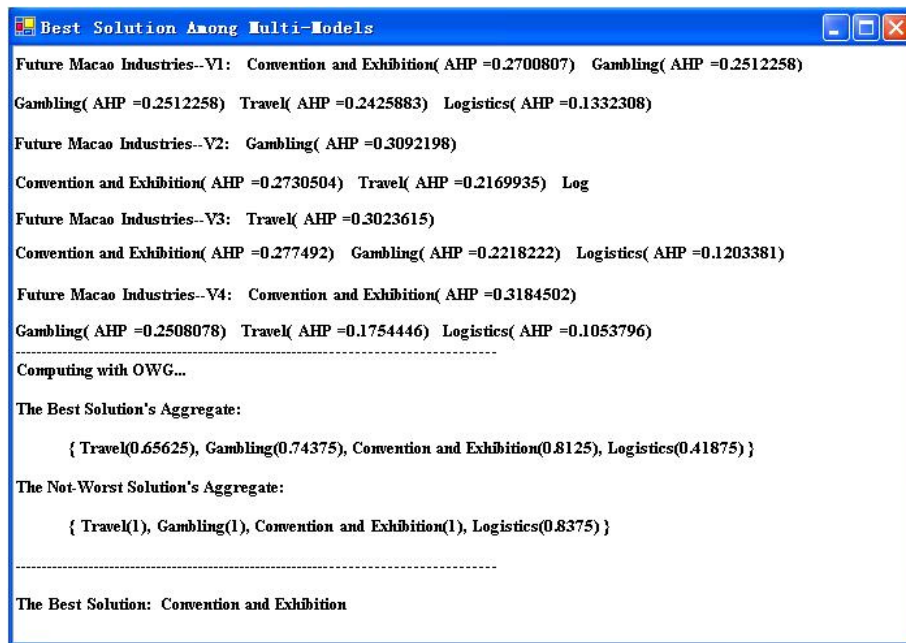


Figure 8. The Aggregation of AHP/OWG

6. Conclusion

The paper expands our distributed modeling methodology and platform with supporting agent's coordination and templates aggregation in distributed group modeling. The soft-agents are designed to coordinate the modeling agents. By collecting the working data of each individual and team, an optimized structure of agents can be organized on selecting the future agents and assigning a proper role for each agent in new projects. In order to ensure the distributed modeling to be worked with right cooperation, the platform allows the agents to share working procedures, modeling activities, updated versions of templates, etc. Also, a comprehensive evaluation method is proposed to judge the quality of agents and teams. Moreover, the templates aggregation by AHP/OWG will assist the agents and team to find out the best solution(s) among a great deal of templates the agents made. Based on these efforts, our distributed modeling methodology and platform provide much hand to agents in different sites to reach high quality of distributed group modeling and problem analysis.

The project is supported by Macao Fund of Development of Science and Technology.

References

- [1] Z. M. Cai, Z. Yang and M. H. Wang, "The Distributed Modeling Methodology and Platform", International Journal of Digital Content Technology and its Applications, vol. 7, no. 9, (2013).
- [2] Z. M. Cai, J. Yin and Y. J. Zhou, "Multi-Agent/Multi-Goal Modeling Templates and The 1-1,n-1,1-n,n-n Deductions of Relationship-chain", International Journal of Hybrid Information Technology, vol. 3, no. 1, (2010).
- [3] Z. M. Cai and J. Yin, "The Selection Modeling System with Grouped Agents", Proceeding of The IEEE 22nd International Conference on Advanced Information Networking and Applications, (2008) March 25-28, Japan.
- [4] Z. M. Cai and J. Yin, "The Process Conducting and Agent Audit in the Distributed Enterprise Modeling", Proceeding of 2008 IEEE Asia-Pacific Services Computing Conference, (2008) December 9-12, Taiwan.
- [5] Z. M. Cai, L. L. Huang and J. Yin, "The Concentration and Distribution of Visual Group Selecting Templates and Sorting Solutions by AHP", The Journal of MUST, vol. 1, no. 2, (2007).
- [6] Z. M. Cai and Y. Liu, "The Distributed Cooperative Work of Multi-Agent on Network", Proceeding of International Conference of CAID&CD'99, (1999) November 11-14, Thailand.
- [7] <http://www.sybase.com/products/modelingdevelopment/powerdesigner>.
- [8] <http://www-01.ibm.com/software/awdtools/developer/rose/>.
- [9] <http://www.eclipse.org/modeling/>.
- [10] C. Gutwin, R. Penner and K. Schneider, "Knowledge sharing in software engineering: Group awareness in distributed software development", Proceedings of the 2004 ACM conference on Computer supported cooperative work (CSCW '04), (2004) November 6-10, Chicago, USA.
- [11] E. Carmel and R. Agarwal, "Tactical Approaches for Alleviating Distance in Global Software Development", IEEE Software, vol. 18, no. 2, (2001).
- [12] F. Lanubile, D. Damian and H. L. Oppenheimer, "Global Software Development: Technical, Organizational, and Social Challenges", ACM SIGSOFT Software Engineering Notes, vol. 28, no. 6, (2003).
- [13] H. Wenbing and M. Bo, "Study on Distributed Intelligence Group Decision Support System Based on Multi-agent", Computer Engineering, vol. 32, no. 6, (2006).
- [14] K. Miettinen, A. V. Lotov, G. K. Kamenev and V. E. Berezkin, "Integration of Two Multiobjective Optimization Methods For Nonlinear Problems", Optimization Methods and Software, vol. 18, no. 8, (2003).
- [15] K. R. Al-asmari and L. Yu, "Experiences in Distributed Software Development with Wiki", Proceedings of SERP'06 in WORLDCOMP'06, (2006) June 26-29, Las Vegas, USA.
- [16] L. Gwanhoo, D. William and E. J. Alberto, "Ambidextrous Coping Strategies in Globally Distributed Software Development Projects", Communications of the ACM, vol. 49, no. 3, (2006).
- [17] M. G. Earl and R. D'Andrea, "Modeling and Control of a Multi-agent System Using Mixed Integer Linear Programming", Proceedings of 41st IEEE Conf. Decision and Control, (2002) December 10-13, Las Vegas, USA.