ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **4**, April **2016** pp. **865–870**

HEDGEHOG: TEAM BUILDING SYSTEM ESTIMATING EFFECTIVENESS OF TEAM

Shuhei Miyake and Hajime Murao

Graduate School of Intercultural Studies
Kobe University
1-2-1 Tsurukabuto, Nada, Kobe 657-8501, Japan
{ shuhei.miyake; hajime.murao }@mulabo.org

Received September 2015; accepted December 2015

Abstract. *This paper proposes a team building system, called Hedgehog. Hedgehog estimates effectiveness of team for each task and recommends combinations of team members by using machine learning. Users do the task by recommended team and register the result score. Then, Hedgehog is trained by receiving combination of quantified personality of the team member as team's characteristic and result score as supervised data. Besides, Hedgehog generates maps of personalities for each task using Self-Organizing Map (SOM) and identifies them. In this paper, we generate maps of 2 tasks, and evaluate the validity.*
Keywords: Team building, Group formation, Machine learning, Self-organizing map

1. **Introduction.** We work as some teams in business, research, studying or many other situations. To achieve better results, effective team building system is important. Wilde proposed Teamology [1]. He assumed that teams whose members have a wide variety of personalities become effective, and proposed techniques analyzing people's cognitive modes quantitatively and building teams which have variety members. The technique is applied to build teams of flagship graduate course students for their projects in Stanford University. Then they won three times of Lincoln Foundation graduate design prizes after application. However, it is just an anecdotal evidence and he did not evaluate the effectiveness of teams. In a field of collaborative learning, there are many researches on team building [2-5]. Most of them analyze just two types of teams, homogeneous and heterogeneous teams in terms of students' learning style, grade or so forth. They do not consider likelihood of other types of teams. From preliminary researches, it appears that assuming effective teams is especially difficult. To assume the effectiveness of teams, understanding the task and necessary skills for it is inevitable. Huang *et al.* [6] claimed that the importance of task-oriented group formation and investigated group activities in an online game. However, the results are limited just in online-game, and they need more considerations to apply them to physical world's tasks. Analyzing real world's tasks is too complicated because the factors determining them intertwine with each another and cannot be observed entirely. Besides, every task has their own characteristics and cannot determine universal. Previous researches lack consideration of tasks' complexity. This paper introduces a team building system, called Hedgehog. Using machine learning or statical techniques, Hedgehog recommends effective team constitutions depending on each task by training from previous team's results. In the following section, we introduce the structure of Hedgehog, and Section 3 shows outlines and results of experiments. Finally, examine the contributions and future prospects of this study.

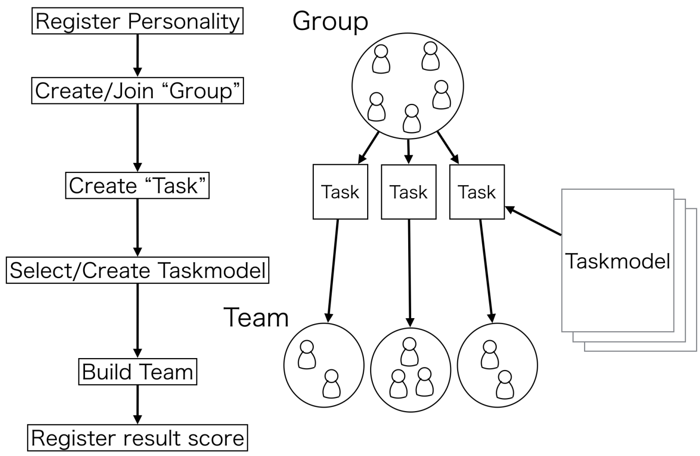

FIGURE 1. Hedgehog flow chart

2. Hedgehog.

2.1. **Components.** Figure 1 shows a flow-chart and model-figure of Hedgehog. Candidates of team member and their personalities obtained beforehand are registered to "Group" in advance. And users register tasks they want to do. Hedgehog chooses members of a team for a task from registered candidates so as to maximize the fitness, which is estimated using taskmodels. The taskmodel is a map from a combination of personalities to fitness. Each taskmodel corresponds to a group of similar tasks. It is trained through tasks. Taskmodels are also used to classify tasks. That is, a task is classified to a taskmodel for which a distribution of fitness for various teams for the task is the most similar. Hedgehog has trainers for each taskmodel and trains for each. In that way, when users want to do a task, they can select a taskmodel the most similar to it. If they cannot find the one, they can register a new one. Taskmodels are shared in users. Thereby users can use various taskmodels and get recommendations from well trained trainers. Users register a task, and then Hedgehog recommends a team, picking out members of the group. After the team has done the task, they quantify the result of the task and register it. Then, Hedgehog estimates which taskmodel the task belongs to. After several times of building teams, Hedgehog determines the taskmodel and the trainer is trained. Once a task belongs to taskmodel, Hedgehog recommends with the trainer and trains it. Hedgehog consists of two parts, personality analysis and recommendation taskmodel. We are developing Hedgehog as a web-based application with FuelPHP 1.7 [7] and Python 2.7 [8].

2.2. **Personality analysis.** In personality analysis part, Hedgehog analyzes candidates of team member cognitive mode scores by Jung personality quantitatively [9]. It measures subjects' personalities quantitatively as eight cognitive mode scores. Users answer the questionnaire of Iino and Wilde [10] at the time of registration. The questionnaire is the one translated Wilde's into Japanese. Eight cognitive modes of "Experiment", "Ideation", "Organization", "Community", "Knowledge", "Imagination", "Analysis" and "Evaluation" are denoted by "Se","Ne","Te","Fe","Si","Ni","Ti" and "Fi" respectively. They are scored between −20 to 20 for each. "Se" and "Ni", "Ne" and "Si", "Te" and "Fi", "Fe" and "Ti" are contrary to each other. For example, if Ni score is 13, Se score is −13. The mode scores of each users are stored with a database called "Personality DB". Teams have characteristic which is common to cognitive modes. We call it "team mode". Team mode scores are determined by member's mode scores. Each team's mode scores are the greatest ones among the members' mode values. Table 1 shows some participants'

Table 1. Participants' and team's mode scores

name	Se	Ne	Te	Fe	Si	Ni	Ti	Fi
Allen	10	6	−6	2	−6	−10	−2	6
Bob	0	−4	−6	−2	4	0	2	6
Carry	13	−7	−1	−5	7	−13	5	1
Team mode	13	6	0	2	7	0	5	6

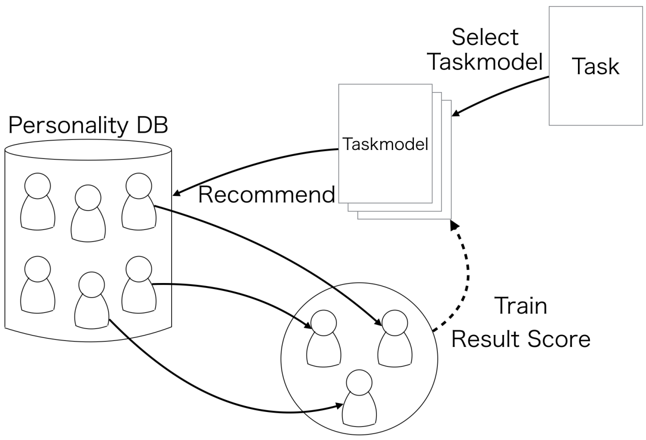

Figure 2. Recommendation system

mode scores and their team's mode score. For example, they have 10, 0, 13 Se scores for each, and then the team's Se score becomes 13. If all the members have negative scores, corresponding team score becomes 0. In Table 1, all the participants have negative scores for Te, and then the team's Te score becomes 0. Therefore, team mode scores have 20 grades each.

2.3. **Recommendation.** Figure 2 shows an overview of Hedgehog's recommendation system. Taskmodels recommend teams by using trainers corresponded to it. The inputs are 9 dimensions, 8 mode values of the team and the result score of the taskmodel. The outputs are the same as inputs. We think that using some neural networks or statical methods, it is expected to estimate the best result score for the taskmodel and the team's mode scores corresponded to it. After it calculates the best team's mode scores, Hedgehog connects to Personality DB, and picks out a combination of the users whose mode scores fit the best team's mode scores. Finally, Hedgehog recommends the combination of the users. And Hedgehog trains ideal team mode scores for each taskmodel from previous teams' results of them. After the team has done the task, they register the result score. Then, Hedgehog gives the trainer the combination of it and the team's mode scores as a supervised data. Repeating recommendation and training, the trainer will be more accurate. Besides, using SOM (Self-Organization-Map), taskmodel can be used for classifying tasks. Tasks are different from each another, and therefore their map must be unique. Comparing taskmodels' SOM, we can estimate which taskmodel some task belongs to.

3. Evaluation.

3.1. **Taskmodel.** We use torus typed SOM for taskmodel. It has 20×20 neurons as a map, and each neuron has 8 team mode scores and the result score of the task as inputs. Sometimes SOM is said to be low repetitive analysis technique due to dependence on initial state. However, using torus structure, almost same shapes can be got by SOM with some rotations. We made some teams with Hedgehog and train 2 taskmodels, "Escape from the desert" and "HANABI". Participants are 23 Kobe university students (11 males

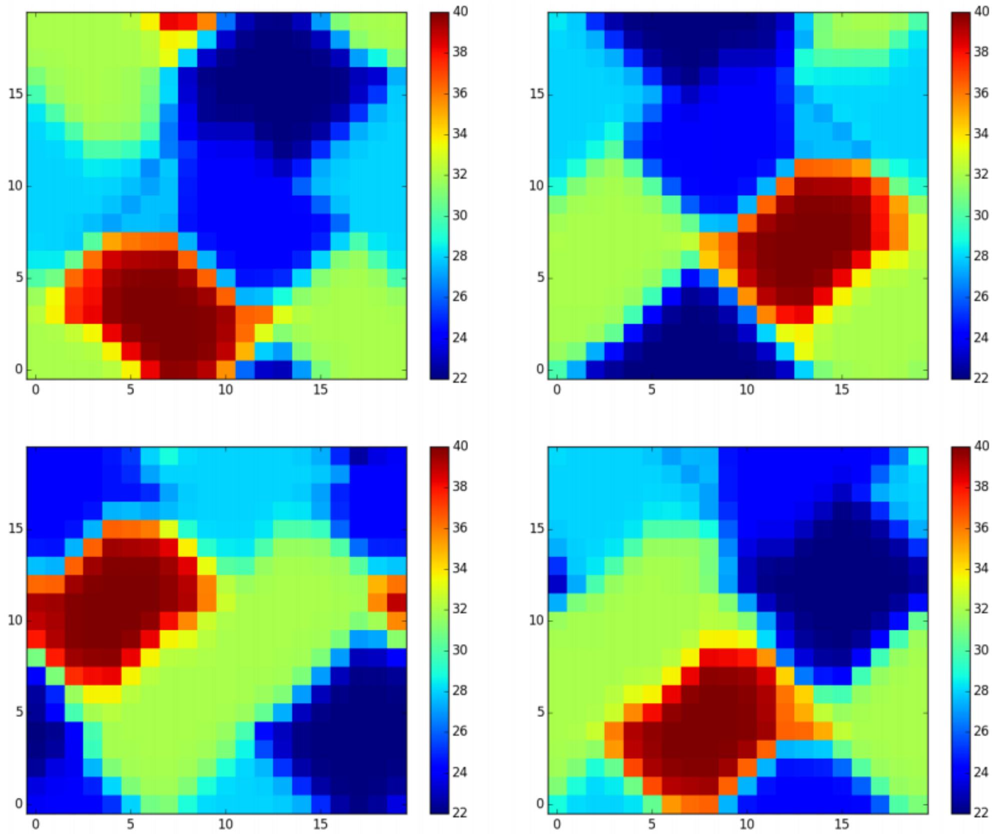

FIGURE 3. Results of desert SOM

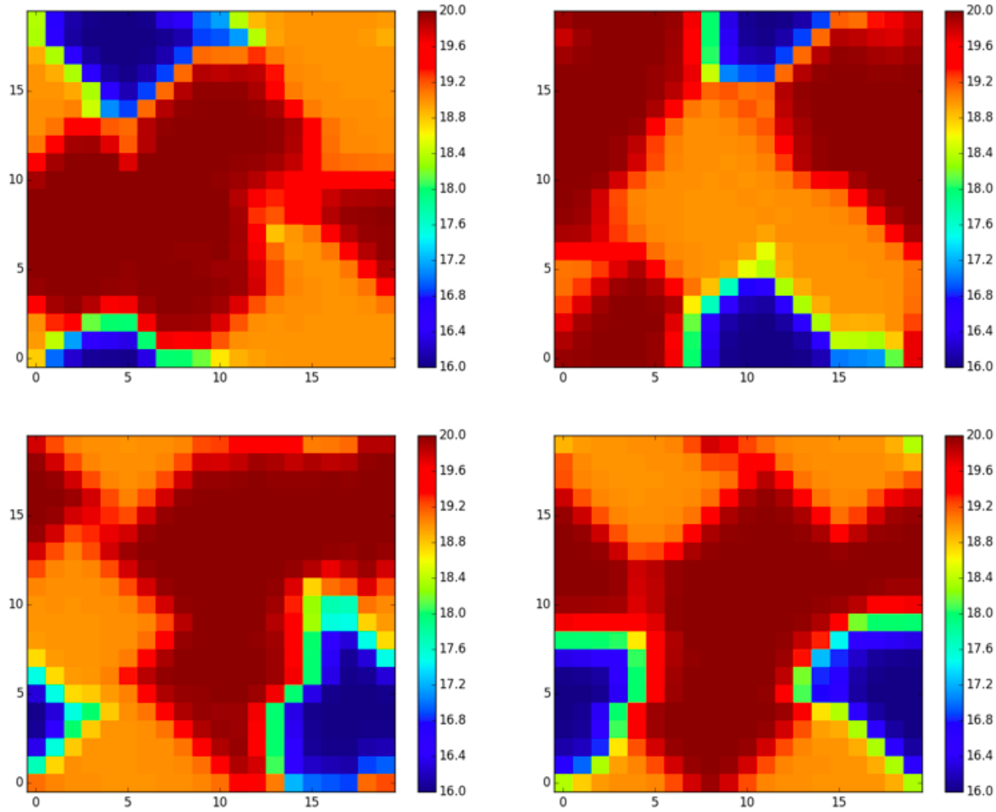

FIGURE 4. Results of HANABI SOM

and 14 females). We built 6 teams (It has 4 members in 5 teams and the other has 3) but the combination of the members is different between "Escape from the desert" and "HANABI". They play both of them once for each. Then, as supervised datasets, 6 result scores and teams' mode scores are trained for 2000 times with 0.01 for training rate. Figures 3 and 4 show 4 results of each SOM of them with different initial states. The color shows estimated score of the taskmodel. It is observed that both of them have each own specific patterns. Therefore, it appears to be able to identify a taskmodel by SOM. Besides, it is also observed that specific area has better scores or worse scores. Hence, taskmodels have individual effectiveness of team.

3.2. **Escape from the desert.** "Escape from the desert" is another version of the task of Hall [11]. Participants play roles of victims of plane crush in the middle of a desert. There are 12 stuffs left in the plane, and they must discuss and rank the priority of them by use of survival in 20 minutes. NASA has presented the suggested answer, and the sum of differences between team's answer and suggested one is the team's score. The score they can get is 0 to 72. The less the score is good, but to make clearly understandable, we use the score subtracted from 72. The more the score is, the better the team has done. As Hall said this score can be decision adequacy index, this taskmodel needs the skill about decision making. Besides, to image the situation of desert and creativity to think up the use of items are also important.

3.3. **HANABI.** The other taskmodel is called "HANABI". HANABI is a cooperative card game by Bauza [12]. There are 3 of 1 rank cards, 2 of 2 to 4 rank cards and just a 5 rank card for each 5 suits, Red, White, Green, Blue and Yellow. The purpose of the game is to line up from 1 to 5 ranks cards for each 5 suits. Players hold 4 or 5 cards in their hands with right side out not to see ranks and suits of their own cards. They can only see other players' cards. In turn of a player, he can do a move from among 3 kinds. First one is telling someone a piece of information of his/hers cards in exchange for a hint token. S/he can tell about only one kind of suits or ranks. For example, when some player has 3 of red, 4 of green, 4 of red and 2 of white, other player can tell him/her "This and this are 4." or "This and this are red". Players have 8 hint tokens first, and if they spend all of them, they cannot do this move. Second move is to discard a card. Player can discard a card without seeing them. Then the player adds a card in their hands, and gets back a hint token. Third move is to play a card. Player can take a card from his/her hands and put it in table without seeing it. If it is tied to a card already played or first 1 rank cards of the suit, the player adds the card to line and player gets 1 point. When the player plays inappropriate cards, they discard it and they are put out. Three outs, the game is over. And if there is no card to add, one more turn for every player and the game is over. After game is over, they get the score till then. As there are 25 kinds of cards, they can get 0 to 25 points. One of this game's difficulty is management of cards and hint tokens. To tell the information is needed to play cards, but players have only 8 hint tokens. If all hint tokens are gone, players should discard unnecessary cards but to do so, they need information. To get better score, planning is inevitable. The timing of spending hint tokens, getting back it and playing cards are important.

3.4. **Validity of taskmodels.** Table 2 is estimated best team mode scores of the 2 taskmodels. From this result, 6 for Se (Experiment) and 8 for Ti (Analysis) modes

Table 2. Taskmodel's best estimation score

Taskmodel	Se	Ne	Te	Fe	Si	Ni	Ti	Fi
Escape from the desert	6	3	5	1	3	2	8	4
HANABI	5	3	5	1	7	2	8	4

appear to be important for "Escape from desert". Wilde writes Experiment as "Discovers new ideas and phenomena by direct experience" [9, p46]. It appears to enhance teams' creativity to think up the use of items as our assumption. And Analysis is referred to as "Logically improves rational performance" [9, p46]. It might be useful for ranking order, but anyway more consideration is needed for this taskmodel. And for HANABI, 7 for Si (Knowledge) and 8 for Ti (Analysis) modes appear to be important. Knowledge mode is referred to as "Physically self-aware, values practice and technique" [9, p46]. Analysis mode might be useful for planning and management of tokens. However, a relation between knowledge mode and this taskmodel is not clear. This part needs further consideration.

4. **Conclusion.** In this paper, we introduced the system, Hedgehog, and it trained 2 tasks experimentally. Maps of them seem to be identical and be utilized for estimating tasks. It shows that each task has unique patterns of effectiveness. As a result, Hedgehog is efficient because it trains effectiveness of tasks for each, regardless of the form of the task. However, more developments and evaluations are needed; Hedgehog must have great potential. It can be used for analysing the task itself with SOM. Users can find the critical point of the task and spare more resources for it. Furthermore, Hedgehog can be extended to accept more inputs. For example, users want to build a team for collaborative learning on English, and they can add test score of English to the inputs. It covers some aspects of teams which cannot be determined by only personality. In this paper, we just introduce Hedgehog and evaluate the validity of the concept. Then we must examine whether Hedgehog improves results of tasks as a further research.

REFERENCES

[1] D. J. Wilde, *Teamology: The Construction and Organization of Effective Teams*, Springer, 2009.
[2] S. Liu, M. Joy and N. Griffiths, An exploratory study on group formation based on learning styles, *IEEE 13th International Conference on Advanced Learning Technologies*, 2013.
[3] C. E. Christodoulopoulos and K. A. Papanikolaou, A group formation tool for e-learning context, *The 19th IEEE International Conference on Tools with Artificial Intelligence*, 2007.
[4] A. Washio, A group formation method for inreasing the effectiveness of group learning, *Bulletin of Takada Junior College*, 2012.
[5] H. Inoue and K. Hanyu, The method of group formation on the learner's characteristics – The group formation supporting system for effective group decision-making in distance learning environment, *The Special Interest Group Technical Reports of IPSJ (GN)*, 2001.
[6] Y. Huang, M. Zhu, J. Wang, N. Pathak, C. Shen, B. Keegan, D. Williams and N. Contractor, The formation of task-oriented groups, *International Conference on Computational Science and Engineering*, 2009.
[7] *FuelPHP*, http://fuelphp.com.
[8] *Python*, https://python.org.
[9] D. J. Wilde, *Jung's Personality Theory Quantified*, Springer, 2011.
[10] K. Iino and D. J. Wilde, Science of design, *Mathmatics for Team Vuilding*, Yokendo, 2012.
[11] J. Hall, The effects of a normative intervention on group decision-making performance, *Human Relations*, vol.23, no.4, 1970.
[12] A. Bauza, *HANABI*, 2011.