2015 International Conference of Cognitive Science (ICCS), Institute for Cognitive Science studies, Iran

Simulation of Pair Programming using Multi-Agent and MBTI Personality Model

Fariba Noori, Mohammad Kazemifard*
Department of Computer Engineering
Razi University
Kermanshah, Iran
fa.noori@pgs.razi.ac.ir, m.kazemifard@razi.ac.ir

***Abstract*—with the advent of agile software processing methods and needs to write the programs and software with the best quality in the shortest possible time, unorthodox practices in computer programming are becoming more and more common. One of these practices is pair programming characterized by two programmers sharing the same computer for collaborative programming purposes. Pair programming implies a psychological and social interaction between the participating programmers. The goal of this paper is an evaluation of the pair programming to determine the influence of programmers' personality and problem difficulty on efficiency of pairs. An agent-based system is used to simulate a pair and Myers-Briggs Type Indicator (MBTI) is used to measure personality of each member of the pairs. This paper presents, suggests, and evaluates the role of personality in formation and the utility of a pair.**

Keywords-Pair programming, Personality, Collaborative programming, Multi-Agent System, Simulation, Myers-Briggs Type Indicator (MBTI).

I. INTRODUCTION

Nowadays, having an agile team is significant element for success of software project [1]. On one hand, finding new and creative solution for the problem of today's organizations needs team working [2]. On the other hand, in doing projects, personal and cooperative characteristics and personality of team members play a key role in achieving goals [1]. Thus with suitable team, the efficiency of a team can be significantly improved and the projects can be finished sooner. The focus of this study, from many factors causing a team to be successful, is formation of teams based on personality of individuals and problem difficulty. The most popular of agile software process paradigm is eXtreme Programming (XP) and pair programming is one of the core practices of it.

"Pair programming is a practice in which two programmers work side-by-side at one computer, continuously collaborating on the same design, algorithm, code, or test" [3]. One programmer, named the driver, has the control of the keyboard/mouse and actively implements the program, and the other programmer, named the navigator, directs the work of the driver by identifying tactical (syntax, spelling, etc)[3-7]. The two programmers can switch their roles since they work equally in developing the software [6]. "Advocators of pair programming claim that it has many benefits over individual programming when applied to new code development or maintaining and enhancing existing code" [7]. There is synergy in the efficiency of a pair [3-5]. Stated benefits of pair programming include higher-quality of coding and product in about half time of an individual programmer, happier programming, enhancing team working and learning, improving knowledge transfer, faster problem solving of a pair, less training time, and lower cost of software production [3, 4, 7]. But this approach has its own problems. Suppose, if in a pair, a programmer is not willing to work with the other programmer or is absent, the other programmer cannot continue and the project will be delayed [8].

As mentioned, the characteristics of team members are important for team working. And according to [9, 10], personality can influence performance of a team by the way that the team members can communicate and interact. Additionally, teams often are formed from people with different personalities. We measure personality by using "Myers Briggs Type Indicator" (MBTI) [11, 12], In this study, the relationship between all scales of the MBTI personality, problem difficulty and pair programming for doing project in shortest time are measured. We would not test our hypothesis on the real world, because strongly believe that before testing, we must find the best condition for experiment. In addition, the cost of experiment could be significantly decreased if a way existed for investigating whether the hypothesis is true or false. One of the most important and remarkable subjects in fields of research and creating applications is simulation of human behavior [13-15], and one way for this purpose is Multi-Agent Systems Simulation [16]. Hence, firstly we can do a simulation and after getting preliminary results, we can do the experiment on the real world for validation. In this study, fuzzy agent and multi-agent systems have been used to simulate personal characteristics and interactions in the pair programming team.

The idea of fuzzy and multi agent simulation is inspired from our previous study in [1], in which we showed that besides the project characteristics, considering the communication skill, personality, mood, and capabilities of team members can improve the performance of a team. They used OCEAN personality model [17] and a macro view about interactions in a team; Here, we used MBTI personality model in a pair and a micro view about interactions in a pair. The result of this study can help project manager to find the best individuals for pairs before establishing a team.

We organize the rest of the paper as follows: section II describes the background of the study including software agents, fuzzy systems, the role of MBTI on pair

* Corresponding author. Tel.: +98 83 34283265; fax: +98 83 34283265.
E-mail addresses: m.kazemifard@appliedemotion.com, m.kazemifard@yahoo.com

978-1-4673-7481-1/15/$31.00 ©2015 IEEE

programming and related works; section III explains the agent model of our simulation; section IV describes the fuzzy system used by agents; section V presents the evaluation and the final section presents the conclusion and perspective of future work.

II. BACKGROUND

A. Software Agent

Agents are characterized by autonomy, proactivity, and communication ability [18]. "Being autonomous, they can independently carry out complex, and often long-term tasks. Being proactive, they can take the initiative to perform a given task even without an explicit stimulus from a user. Being communicative, they can interact with other entities to assist in achieving their own and others' goals" [18]. Ghasem-Aghaee and Ören [13] defined fuzzy agents as agents that can perform qualitative uncertainty reasoning with incomplete and fuzzy knowledge in environments that contain linguistic variables. A multi-agent system consists of multiple interacting agents to achieve a common goal. "Multi-agent systems have the capacity to play an important role in developing and analyzing models and theories of interactivity in human societies" [19]. A more detailed discussion of AOP appeared in [20].

B. Fuzzy system

The term fuzzy logic emerged in the development of the theory of fuzzy sets by Zadeh [21]. The fuzzy set theory provides a natural method for dealing with linguistic terms (i.e. easy, normal, and hard) of the linguistic variables (i.e. problem difficulty). A general fuzzy system includes the following elements [22]:

1. Fuzzification converts the crisp value of input variables to fuzzy inputs using fuzzy membership functions (MF).
2. The Knowledge base contains fuzzy rules based on the domain of the expert's knowledge.
3. Fuzzy inference converts fuzzy inputs to fuzzy outputs using the knowledge base.
4. Defuzzification converts fuzzy outputs to crisp values.

There are three widely used fuzzy inference system types: Mamdani, Sugeno, and Tsukamoto that are different in the fuzzy inference mechanism and the defuzzication [23]. In this paper, we have used Tsukamoto inference system with two kinds of MFs: triangular and trapezoidal.

C. MBTI

MBTI was developed by Isabel Briggs Myers and Katherine Briggs [11, 12, 24]. MBTI is used as a test for personnel selection. "Myers believed that different occupations favored different personality orientations" [25]. The MBTI personality assessment identifies an individual's type with four basic preference scales with opposite poles, and provides a common language around how people interact with others and the world around them. The four scales are: (1) extroversion/introversion, (2) sensate/intuitive, (3) thinking/feeling, and (4) judging/perceiving [11, 12, 24, 25].

According to [9], the personality type of a pair influences programming performance. It can be said that mostly programmer's interaction is influenced. The problem difficulty cannot be ignored too. The influence of MBTI scales and problem difficulty on pair programming is discussed in the following paragraphs.

(1) Extroversion/introversion: Most of the conscious perceptions and reactions to the world are determined by extroversion and introversion [2]. According to [2, 7, 9, 11, 12], intra team interactions influence team working. A person must be able to share information with others. This scale represents communication of a person with the outside world. Introverted people are not tending to interact and share information with others because they cannot transfer information and they are often shy and self-focused. Unlike introverted people, extroverted ones are more receptive and socially aggressive. Hence, it can be said that introverted people can have a negative and extroverted ones have a positive effect on the intra communication of a pair.

(2) Sensate/intuitive: According to [2, 7, 9, 11, 12], sensate people can perceive and organize facts step by step. Hence, if the problem is simple and does not need innovative solutions, they can solve the problem well. They do not waste time to looking for new ideas. Hence, in hard problems needing new ideas and innovative solutions, these people have negative effects on the pair. Intuitive people can give new ideas since they love to learn new skills. They always try to solve problems with new ideas and offer creative solutions. Therefore, intuitive people can have negative effects on the pair when the problem is simple, but they can have positive effects on the pair when the problem is hard and needs creative solutions.

(3) Thinking/feeling: According to [26], Herbert and Bradley conducted a trial to show the productivity difference between two teams, in which personality types of each team are different. Results showed that the team with 40% feelings completed the task sooner than the team with 20% feelings. Feeling people envisage the influence of their decision on other team members. Teamwork and team cohesion are important for them; hence, they can have a positive effect on the pair programming. Unlike feeling people, thinking people do not care the effect of their decision on the team members. Logic and justice are important for them; hence, thinking people have a negative effect on pair programming.

(4) Judging/perceiving: According to the trial of Herbert and Bradley [26], the second team was completely judging while %70 of the team one was judging. Judging people are ordered and ethical in their work. The main idea or theme of judging people is "work first, play later." They pay attention to time and consider it; so, these people have a positive effect on sooner completion of projects. Perceiving people love freedom. And their slogan is "play first, work later." They believe that time is a renewable resource, which is not important but how to do work matters for them. Since time is an important issue in the projects, perceiving people can have a negative effect and

judging people can have a positive effect on pair programming.

D. Related Work

Several experiments on comparison of pair and solo programming have been reported. Lui and Chan [27] measured relationships between human experience and programming productivity. Arisholm et al [28] evaluated the effect of pair programming on duration, effort and correctness. In addition, they investigated the relationship between system complexity and duration, effort and correctness, and assessed the effect of programmer expertise on duration, effort and correctness. Meanwhile, Hanny et al [7] presented a meta-analysis of these experiments. The Hanny's analysis showed that pair programming has a small positive effect on qualifying, a medium positive effect on duration, and a medium negative effect on effort. Bradley and Hebert [26] conducted an experiment and found the effect of personality type on team working. Some other worked on the effecting of personality on pair programming (There is a summary of studies on personality and pair programming in [10]). Choi et al [9] considered the second and third scales of the MBTI and the results showed that groups with diversity in MBTI types exhibit better productivity. In the mentioned studies, all items of personality, problem difficulty and pair programming were considered. In this study, via the fuzzy agent system simulation, we are going to find the effect of personality and problem difficulty on pair programming for doing the project in shortest time.

III. AGENT MODEL

The main concern of pair programming simulation is team's characteristics and problem difficulty. The simulation results can help to create the best formation of a pair. This pair can finish the project sooner. In this study, the interactions between the team members with all scales of MBTI are considered.

There are two kinds of agents in this simulation: "Team Member Agent" (TMA) and "Simulator Agent" (SA)[1]. TMA is a fuzzy agent for the simulation of a team member. Two TMAs can simulate a pair. The TMAs communications can reflect intra communication of a team. Fig 1 shows a communication graph of a pair (two programmers): TMA_1 and TMA_2. The team's communications are shown as a directed graph in which the links show the communications. With respect to this graph, the programmer 1 communicates with the programmer 2; and the programmer 2 communicates with the programmer 1.

A TMA sends two kinds of messages to other agents in this simulation: an "effect message" to the other TMA and a "result message" to the SA. The "effect message" is the influence of a member on the teammate, and the "result message" is the result of the influence. A sample communication between TMA_1 and other agents with FIPA-ACL[2] in one cycle of the simulation is as follows:

```
(Inform
        : Sender TMA_1
        : Receiver SA
        : Content
                (
                Type is result
        Final UF is 85
        Average UF is 83
        Id is 1
                )
        : Language s1
)

(Inform
: Sender TMA_1
    : Receiver TMA_2
    : Content
(
                Type is affect
      UF_Contrary is 56.88
      Id is 1
            )
    : Language s1
)
```

We defined Utility Function (UF) as the ability of a person to solve problems and finish projects sooner. UF_Person is the UF of an individual. "Type" is utilized to identify one message from other messages in the system. SA stores the "UF_Final" that is final UF and "UF_Average" that is the average of UFs of a pair's individuals during the cycle of simulation. TMA agent uses "UF_Transmission" to effect the teammate. "UF_Contrary" is UF of the other agent.

The intra-structure of a TMA is illustrated in Fig 2. Each person has one of the 16 individual's MBTI types [11, 12, 24, 25]. A software agent needs three basic aspects to match with a real person: each person receives some information from the other person, analyzes this information at the end, and sends some information to other persons[1]. It can be said that first scale of individual type, introversion, and extroversion, illustrates the social characteristics and with this scale people can send or receive information. With other scales, i.e. sensate/intuitive, thinking/feeling, perceiving/judging, people can analyze the problem. Problem difficulty can influence our analysis too.

Hence, the TMA includes five internal variables: introversion/extroversion, sensate/intuitive, thinking/ feeling, perceiving/judging, and problem difficulty (which will be discussed in the next section). At the beginning of the simulation, a message from SA initializes the internal variables of each TMA, assigns an "id" to each TMA, and sends the initial information that TMA needs for its computation. A sample initialization message between SA and TMA_1 is as follows:

```
(Inform
        : Sender SA
        : Receiver TMA_1
        : Content
        (
        Type is Initial
        Introversion/Extroversion is 60
                Sensate/intuitive is 80
                Thinking/Feeling is 40
```

[1] It us inspired from [1].

[2] Foundation for Intelligent Physical Agents (FIPA) works on developing standards for agent systems and Agent Communication Language (ACL) specifications. We used a semantic language (sl) to explain the messages between agents. It is not the real language used in the FIPA-ACL.

Perceiving/Judging is 90
Problem Difficulty is 60
Id is 1
)
: Language s1
).

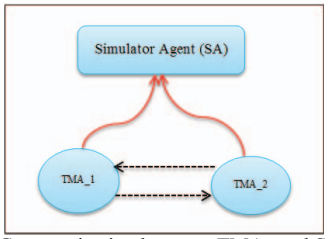

Fig 1. Communication between TMAs and SA.

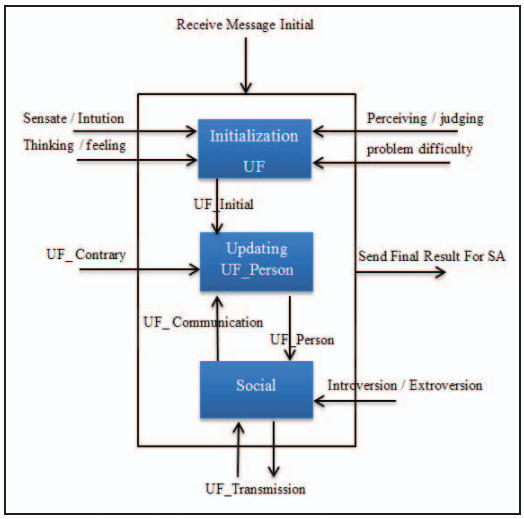

Fig 2. The intra-structure of a TMA.

IV. FUZZY INFERENCE

We have utilized the Tsukamoto fuzzy inference system. Tsukamoto aggregates each rule's output with a weighted average method [23]. For the variables, which need superior transmission, UF_Transminision, and UF_Initial, we have utilized the triangular membership function. We have utilized trapezoidal membership function for personality characteristics (introversion/ extroversion, sensate/intuitive, thinking/feeling, perceiving /judging), and for other variables (UF_Communication, problem difficulty, UF_Contrary, UF_Person).

A. UF Initialization element

This element contains four fuzzy variables as inputs and one fuzzy variable as output. The inputs are as follows:

- Sensate/intuitive was discussed in section II.C.2 (Fig 3).
- Problem difficulty related to how much the problem is hard (Fig 4).
- Thinking/feeling was discussed in section II.C.3 (Fig 5).
- Perceiving/judging was discussed in section II.C.4 (Fig 6).

The output is UF_Initial whose fuzzy set is illustrated in Fig 7. The effect of inputs on UF_Initial is according to the rules in tables I, II and III. Since problem solving and problem difficulty is related to sensate/intuition, we have considered effect of both sensate/intuitive and problem difficulty on UF_Initial. A union of these effects are used to create UF_Person. For example, the first row of the table I is interpreted as follows:

IF	*sensate/intuitive*	IS	SLow
AND	*Problem Difficulty*	IS	Easy_PD
THEN	*UF_Initial*	IS	ISP

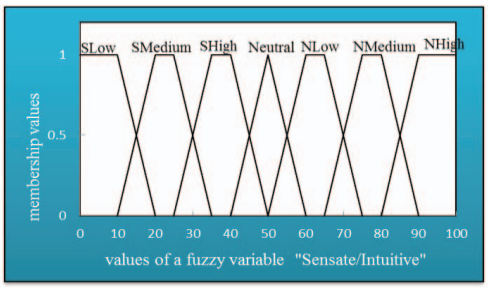

Fig 3. The fuzzy membership function of the sensate/intuitive (The character 'S' indicates Sensate and 'N' indicates iNtutive; i.e., SLow is Sensate Low)

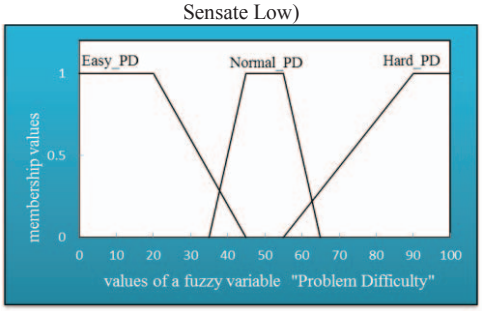

Fig 4. Fuzzy membership function of the Problem Difficulty (The 'PD' indicates problem difficulty; i.e., Easy_PD is Easy Problem Difficulty)

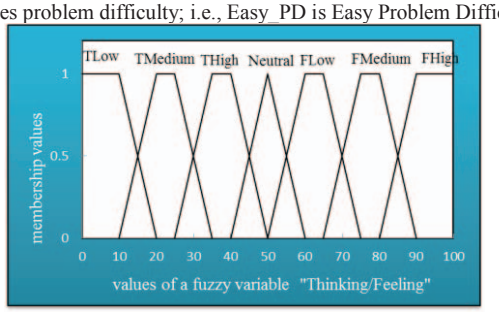

Fig 5. The fuzzy membership function of the thinking/feeling. (The character 'T' indicates Thinking and 'F' indicates Feeling; i.e., TLow is Thinking Low)

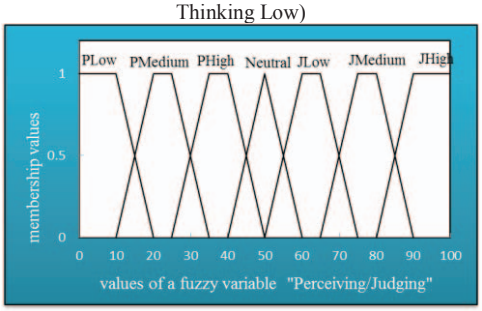

Fig 6. The fuzzy membership function of the perceiving/judging. (The characters 'P' indicates Preceiving and 'J' indicates Judging; i.e., PLow is Preceiving Low)

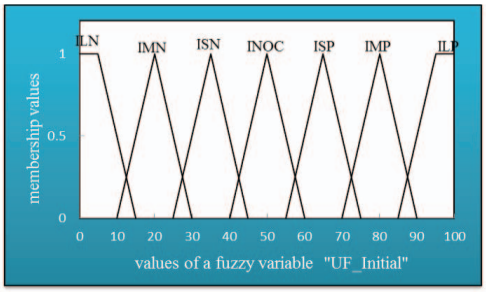

Fig 7. Fuzzy membership function of the UF_Initial (The character 'I' indicates UF_Initial; L, Large; M, Medium; S, Small; P, Positive; N, Negative; NOC, NOChange; i.e., ILN is UF_Initial Large Negative).

TABLE I. DIFFICULTY ON UF_INITIAL

Input		Output
Sensate/iNtuitive	***Problem difficulty***	***UF_Initial***
Slow	Easy_PD	ISP
Slow	Normal_PD	INOC
Slow	Hard_PD	ISN
SMedium	Easy_PD	IMP
SMedium	Normal_PD	ISP
SMedium	Hard_PD	ISN
SHigh	Easy_PD	ILP
SHigh	Normal_PD	IMP
SHigh	Hard_PD	ILN
Neutral	Easy_PD	INOC
Neutral	Normal_PD	INOC
Neutral	Hard_PD	INOC
NLow	Easy_PD	ISN
Nlow	Normal_PD	INOC
NLow	Hard_PD	ISP
NMedium	Easy_PD	IMN
NMedium	Normal_PD	ISP
NMedium	Hard_PD	IMP
NHigh	Easy_PD	ILN
NHigh	Normal_PD	IMP

TABLE II. EFFECT OF THINKING/FEELING ON UF_INITIAL.

Input	Output
Thinking/feeling	***UF_Initial***
TLow	ISP
TMedium	INOC
Thigh	ISN
Neutral	INOC
Flow	ISP
FMedium	IMP
FHigh	ILP
TLow	ISP
TMedium	INOC

TABLE III. EFFECT OF PERCENTAGE/JUDGING ON UF_INITIAL.

Input	Output
Percentage/judging	***UF_Initial***
Plow	ISN
PMedium	IMN
PHigh	ILN
Neutral	INOC
JLow	ISP
JMedium	IMP
JHigh	ILP
Neutral	ILN
JLow	INOC

B. *Updating UF_Person element*

This element contains three fuzzy variables as inputs and one fuzzy variable as output. The inputs are as follow:

- UF_Contrary is the UF of the teammate without considering introvert/extrovert value of an individual. (Fig 8).
- UF_Communicate (which will be discussed in the next section).
- UF_Person initializes with the UF_Initial, at the first cycle of the simulation. It is changing during simulation (Fig 9).

The output:

- UP_Person is updated by adding UF_Change to current value of UF_Person (Fig 10). UF_Change is computed by aggregation of UF_Person, UF_Contrary, and UF_Communication.

Table IV contains 18 rules to compute the value of UF_Change. For example, the first row of the table is interpreted as follows:

IF	*UF_Person*	IS	Low
AND	*UF_Communicate*	IS	Low
AND	*UF_Contrary*	IS	Low
THEN	*UF_Change*	IS	CNOC

C. *Social element*

Each agent sends UF_Transmission to the other teammate. This element consists of two fuzzy variables as inputs and one output.

Inputs:

- UF_Person (which was discussed in the previous section).
- Introversion/extroversion was discussed in section II.C.1 (Fig 11).

UF_Transmission is the affect message sending by teammate.

Output:

- UF_Communication is multiplication of received UF_Transmision and deffuzified value of Transmission_Percentage (Fig 12). Transmission_Percentage filters UF_Transmision with respect to introversion/extroversion value of the agent. Hence, UF_Communication can be less than or equal to UF_Transmision (Fig 13).
- UF_Transmission is the multiplication of UF_Person and deffuzified value of Transmission_Percentage (Fig 12).

The rules of Transmission_Percentage generation are shown in Table V. This table contains seven rules to compute the value of UF_Communication in each cycle of simulation. For example, the first row of the table is interpreted as bellow:

IF	*Introversion/Extroversion*	IS	*ILow*
THEN	*UF_Transmission*	IS	VVSome

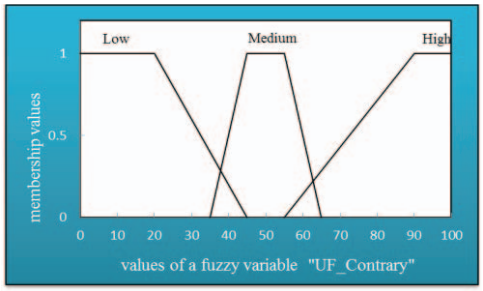

Fig 8. Fuzzy membership function of the UF_Contrary.

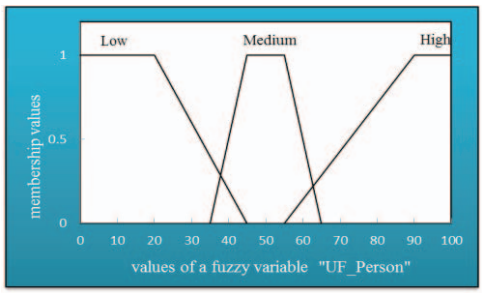

Fig 9. Fuzzy membership function of the UF_Person.

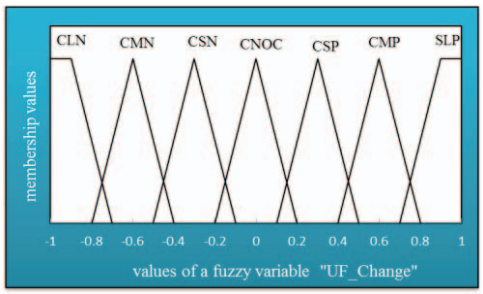

Fig 10. Fuzzy membership function of the UF_Change (The character 'C' indicates UF_Change; L, Large; M, Medium; S, Small; P, Positive; N, Negative; NOC, NOChange; i.e., CLN is UF_Change Large Negative).

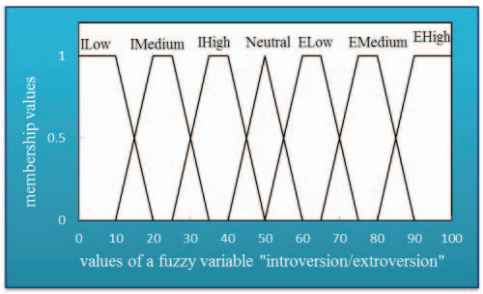

Fig 11. The fuzzy membership function of the Introversion/Extroversion. (The character 'I' indicates Introversion and 'E' indicates Extroversion; i.e., ILow is Introversion Low)

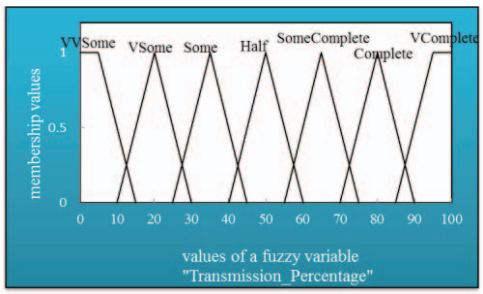

Fig 12. Fuzzy membership function of Transmission_Percentage. (The character 'V' indicates Very; i.e., VVSome is Very Very Some).

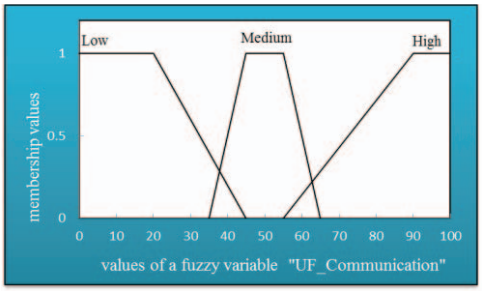

Fig 13. Fuzzy membership function of the UF_Communication.

TABLE IV. DIFFICULTY ON UF_INITIAL

Input			Output
UF_Person	***UF_Communication***	***UF_Contrary***	***UF_Change***
Low	Low	Low	CNOC
Low	Low	Medium	CNOC
Low	Medium	Medium	CSP
Low	Low	High	CSP
Low	Medium	High	CMP
Low	High	High	CLP
Medium	Low	Low	CSN
Medium	Low	Medium	CSN
Medium	Medium	Medium	CNOC
Medium	Low	High	CSP
Medium	Medium	High	CMP
Medium	High	High	CLP
High	Low	Low	CMN
High	Low	Medium	CMN
High	Medium	Medium	CSN
High	Low	High	CSP
High	Medium	High	CMP
High	High	High	CLP

TABLE V. EFFECT OF PERCENTAGE/JUDGING ON UF_INITIAL.

Input	Output
Introversion/Extroversion	***Transmission_Percentage***
ILow	VVSome
IMedium	VSome
IHigh	Some
Neutral	Half
ELow	Some Complete
EMedium	Complete
EHigh	VComplete
Introversion/Extroversion	Transmission_Percentage
ILow	VVSome

V. RESULT AND SIMULATION

This simulation uses fuzzyJ package [29] to implement the fuzzy inference system and JADE [30] to implement the multi agent system (MAS). This simulation runs in five modes: (1) sensing/intuition variation, (2) thinking/feeling variation, (3) perceiving/judging variation, (4) introversion/extroversion variation, (5) all the parameters variation. The simulation runs for one workday, and it is supposed that each pair is effected every 30 minutes. Hence, this simulation runs 16 cycles for each pair. The simulation runs for one hundred thousand pairs in each mode.

A. Sensing/intuition variation

In this mode, sensing and intuition are randomly variable and problem difficulty includes three values: easy (0), normal (50), and hard (100). The other parameters are constant (around 50). The results showed when both individuals in a pair are sensing and problem difficulty is

easy, in 6% of pairs UF_Average is greater than UF_Initial. When both individuals are intuition people and problem difficulty is hard, UF_Average of 6% pairs is greater than UF_Initial. There is no significant value in other cases. As we discussed in section 2.3.3, sensing people influence pairs positively in the easy problems and intuition people influence pairs positively in hard problems.

B. *Thinking/feeling variation*

In this mode, only the thinking and feeling parameter is randomly variable and the other parameters are constant (around 50). The results showed when both individuals in a pair are feeling, in 7% of pairs UF_Average is greater than UF_Initial. There is no significant value in the other cases. As we discussed in section 2.3.4, feeling people influence pairs positively and thinking people influence pairs negatively.

C. *Perceiving/judging variation*

In this mode, only the perceiving/judging parameter is randomly variable and the other parameters are constant (around 50). The results showed when both individuals in a pair are judging, in 10% of pairs UF_Average is greater than UF_Initial. There is no significant value in the other cases. As we discussed in section 2.3.5, judging people influence pairs positively and perceiving people influence pairs negatively.

D. *Introversion/extroversion variation*

In this mode, introversion/extroversion parameter is randomly variable and UF_Initial includes three fuzzy values: low, medium, and high. The UF_Initial was used to evaluate the skill sharing intra a pair. When UF_Initial of both individuals is high, the results showed if both individuals are introvert or both are extrovert, in 25% of pairs UF_Average is greater than UF_Initial. If one individual is introvert and the other is extrovert, in 50% of pairs UF_Average is greater than UF_Initial. As discussed in [7], if we see UF as expertise, “There are also expectations with respect to the benefits and drawbacks of various kinds of pairing, e.g., that expert–expert pairings seem to be especially accelerated” [7]. Results of introvert-extrovert pairs were the best since regarding to the trial conducted by Herbert and Bradley [9], a team which both types have the better result. When large number of individuals are extrovert and expert, it influences pair programming negatively. When UF_Initial of an individual is low and the other is medium or high, the results showed if both individuals are extrovert, in 7% of pairs UF_Average is greater than UF_Initial. No significant result was obtained for the other cases.

E. *All the parameters variation*

In this mode all the parameters are randomly variable. The results showed that 16.8 percent of pairs were positively affected in comparing solo programming. In [7] pair programming was better than solo programming in about %8 and we got %16.8.

VI. CONCLUSION

Understanding the factors causing the success of a team is a step of team formation. Although there are many factors in this field, it is very important to create the best possible team as far as we can. Previous studies showed that personality influences pair programming. In this study, we investigated the role of personality as well as the projects’ parameters. We utilized the MBTI to identify and represent the personality of a person. Software agents were used for simulation of a pair. There were two kinds of agents in this simulation: "Team Member Agent" (TMA) and "Simulation Agent" (SA). TMA is a fuzzy agent for simulation of a team member. A team can be simulated by multi-TMA. Each TMA is characterized by five internal variables: introversion/extroversion, sensing/intuitive, feeling/thinking, perceiving/judging and problem difficulty. A multi-agent has been used to simulate intra-team communications. The model was evaluated with five modes. The results showed that personality plays an important role in the formation and utility of a pair. For example, when the expertise of both individuals are high, the best pairing is introvert-extrovert. When both individuals are extrovert, the best pairing is low-high or medium-high expertise.

In future work, the simulation can also be done for a team constituted of several pairs. Then, the project is divided into several parts and each pair works on a part of them. It can help to make the best formation of a team.

REFERENCES

[1] M. Kazemifard, A. Zaeri, N. Ghasem-Aghaee, M. A. Nematbakhsh, and F. Mardukhi, "Fuzzy Emotional COCOMO II Software Cost Estimation (FECSCE) using Multi-Agent Systems," *Appl. Soft Comput.,* vol. 11, pp. 2260-2270, 2011.

[2] M. Sadrinia, "Making a Successful Team; Different Compilation," *Tadbir Monthly Magazine,* vol. 190, pp. 30-34, 1386.

[3] L. A. Williams and R. R. Kessler, "All I really need to know about pair programming I learned in kindergarten," *Commun. ACM,* vol. 43, pp. 108-114, 2000.

[4] A. Cockburn and L. Williams, "The costs and benefits of pair programming," in *Extreme programming examined*, ed: Addison-Wesley Longman Publishing Co., Inc., 2001, pp. 223-243.

[5] S. Bryant, P. Romero, and B. d. Boulay, "Pair programming and the mysterious role of the navigator," *Int. J. Hum.-Comput. Stud.,* vol. 66, pp. 519-529, 2008.

[6] S. Bryant, B. d. Boulay, and P. Romero, "XP and Pair Programming practices," in *PPIG (Psychology of Programming Interest Group) Newsletter*, ed, September 2006.

[7] J. E. Hannay, T. Dyba, E. Arisholm, and D. I. K. Sjoberg, "The effectiveness of pair programming: A meta-analysis," *Inf. Softw. Technol.,* vol. 51, pp. 1110-1122, 2009.

[8] A. Safayee, "New Style of Programming – Extreme Programming," *Network Monthly Magazine,* vol. 74, 1385.

[9] K. S. Choi, F. P. Deek, and I. Im, "Exploring the underlying aspects of pair programming: The impact of personality," *Inf. Softw. Technol.,* vol. 50, pp. 1114-1126, 2008.

[10] J. E. Hannay, E. Arisholm, H. Engvik, and D. I. K. Sjoberg, "Effects of Personality on Pair Programming," *IEEE Trans. Softw. Eng.,* vol. 36, pp. 61-80, 2010.

[11] (20.02.13). *Personality Pathways*. Available: http://www.personalitypathways.com

[12] (20.02.13). *Myers-Briggs Type Indicator*. Available: http://en.wikipedia.org/wiki/Mbti

[13] N. Ghasem-Aghaee and T. Oren, "Towards fuzzy agents with dynamic personality for human behavior simulation," in *SUMMER COMPUTER SIMULATION CONFERENCE*, 2003, pp. 3-10.

[14] S. Sharma, H. Singh, and A. Prakash, "Multi-agent modeling and simulation of human behavior in aircraft evacuations," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 44, pp. 1477-1488, 2008.

[15] T. Oren and N. Ghasem-Aghaee, "Personality representation processable in fuzzy logic for human behavior simulation," in *Summer Computer Simulation Conference*, 2003, pp. 11-18.

[16] Faisal Alkhateeb, E. A. Maghayreh, and I. A. Doush, *Multi-Agent Systems - Modeling, Interactions, Simulations and Case Studies*: InTech, 01 April, 2011.

[17] P. T. Costa and R. R. McCrae, "Normal personality assessment in clinical practice: The NEO Personality Inventory," *Psychological assessment,* vol. 4, p. 5, 1992.

[18] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*: Wiley, 2007.

[19] J. Martínez-Miranda and A. Aldea, "Emotions in human and artificial intelligence," *Computers in Human Behavior,* vol. 21, pp. 323-341, 3// 2005.

[20] Y. Shoham, "Agent-oriented programming," *Artificial Intelligence,* vol. 60, pp. 51-92, 3// 1993.

[21] L. A. Zadeh, "Fuzzy Sets," *Information and Control,* vol. 8, pp. 338-353, 1965.

[22] J. Durkin, *Expert Systems: Design and Development*: Macmillan Publishing Company, 1994.

[23] P. P. Cruz and F. D. Ramâirez-Figueroa, *Intelligent Control Systems with LabVIEW,TM*: Springer London, 2009.

[24] N. L. Quenk, *Essentials of Myers-Briggs type indicator assessment*: J. Wiley & Sons, 2000.

[25] D. J. Pittenger, "Measuring the MBTI... And Coming Up Short," *Journal of Career Planning and Employment,* vol. 54, pp. 48-52, // 1993.

[26] J. H. Bradley and F. J. Hebert, "The effect of personality type on team performance," *Journal of Management Development,* vol. 16, pp. 337-353, 1997.

[27] K. M. Lui and K. C. C. Chan, "Pair programming productivity: Novice–novice vs. expert–expert," *International Journal of Human-Computer Studies,* vol. 64, pp. 915-925, 9// 2006.

[28] E. Arisholm, H. Gallis, T. Dyba, and D. I. K. Sjøberg, "Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise," *IEEE TRANSACTIONS ON SOFTWARE ENG,* vol. 33, 2007.

[29] (14.09.12). *Fuzzy J ToolKit*. Available: http://www.iit.nrc.ca/IR%20public/fuzzy/fuzzy-JToolkit2.html

[30] (10.07.12). *Java Agent Development Framwork*. Available: http://jade.tilab.com/