Requirements Engineering Education using Role-Play Training

Taichi Nkamaura
National Institute of Informatics
Tokyo, Japan
nakamuratc@nii.ac.jp

Yuki Tachikawa
NEC Solution Innovators, Ltd
Tokyo, Japan
g211301392@edu.teu.ac.jp

Abstract**—This paper proposes a new method for use in requirements engineering education at universities. The method is based on two components: (1) group-work role-play training to elicit customer's real requirements; (2) a software agent to play the role of a domain expert supplying business knowledge to learners. Learners who have had no previous practical experience in defining requirements may follow a recommended procedure in order to elicit requirements. We have implemented two types of role-play training - with and without software agents' support. Twelve undergraduate students from the School of Computer Science at the Tokyo University of Technology participated in an experimental assessment of these two types of role-play training. The results of a subsequent questionnaire survey and portfolio evaluation of the goal model diagrams developed by the students showed that their requirements modeling skill levels had improved. However, because some students may have had no prior business experience, each role-play scenario needs to be carefully designed so that all necessary information regarding the strategy or vision in the problem domain is clearly laid out.**

Keywords—requirements enigineering ; KAOS; goal modeling;

I. INTRODUCTION

Throughout Japan, there has been considerable investment in computer systems to enable companies to survive in the current commercial environment. One of the greatest concern of the top executives in these companies is how much profit a new computer system earns. However, the executive's demand, such as launch new services and bring reform business operations are not always satisfied. A failure to respond to customer expectations can often result from a lack of requirements definition and inadequate analysis skill by system engineers in both the development company and the customer company [1]. The information industry in Japan has, therefore, been requiring universities to provide the education bridging between software engineering and business analysis requiring the skills to not only analyze the requirements presented by a customer but also to identify genuine customer requirements, which even the customers themselves may not yet recognize.

In order to find customer's requirements derived from both practical problems in the business field and the corporate strategy, an engineer takes actions, as follows:

(1) Read a Request for Proposal (RFP) provided by a customer to extract customer needs and interviewing the customer;

(2) Extract the relationships between requirements, their priority, and the development object;

(3) Create a requirements specification document;

(4) Check the consistency between the requirements and the satisfaction of customer needs [2].

In addition, both facilitation skills and organization skills are required in order for a system engineer to progress through these four actions successfully. To provide practical experience for learning these skills in a university, we have proposed a method of requirements engineering education using a combination of two key concepts [3].

(1) In order for students to have a simulated experience of requirement elicitation processes, role-play (RP) training in an on-line group-work training environment, in which each student makes use of their knowledge of requirements elicitation and analysis derived from classroom lectures, can provide the practical situation even in a university, and can obtain the behavioral track records of each student to evaluate the educational effectiveness.

(2) In order for students to be able to collect the domain knowledge during role-play, a domain expert system for questions and answers playing the role of a customer and other domain expert responds to students' questions concerning the customer's business matters.

We carried out the requirement engineering education using the proposed method based on the combination of RP training and the domain expert system for questions and answers for the graduate students. According to the result derived from analyzing the behavioral track records of each student, students could not even ask an appropriate question to obtain the customer's needs because they have no prior experience in developing a commercial system. Then they could not collect the domain knowledge to elicit customer's requirements [3].

In order for students to be able to ask questions about basic business matters underlying the virtual project during RP, the domain expert system is required to monitor the chat messages exchanged among students as a part of the students' behavior, if students do not exchange any messages for a little while, invite students to ask a question, when necessary, provide the domain knowledge.

The on-line group-work training environment, which cooperate with the software agent system as an alternative to the domain expert to provide RP training to learners. The software agent system plays the role of an adviser or a mentor, monitors the behaviors of each student, whenever necessary, give an advice and hints to solve problems caused in a virtual project, so that a student who has no previous experience of system development can arrive at the correct decision in an orderly manner.

Section II explains how the RP training system for requirements engineering education is now used, with a software agent system acting as an alternative to the domain expert. Section III describes related work carried out with this system. Section IV describes the experiments undertaken in order to assess the effectiveness of the proposed method. Section V discusses the results of these experiments. Finally, section VI provides conclusions and looks to the future of requirements engineering education involving RP training.

II. Requirements Engineering Education using RP Training in An Oneline Group-Work Training

A. Goal-Oriented Requirements Analysis Technique

In the requirements analysis process, the structuring of requirements obtained from a customer is an important task in clarifying the consistency and dependencies among requirements, and in detecting lack of requirements or ambiguity in requirements [9]. KAOS (Knowledge Acquisition in autOmated Specification) is a commonly used method for goal-oriented requirement analysis [8]. The goal model diagram it produces is represented by a directed acyclic graph.

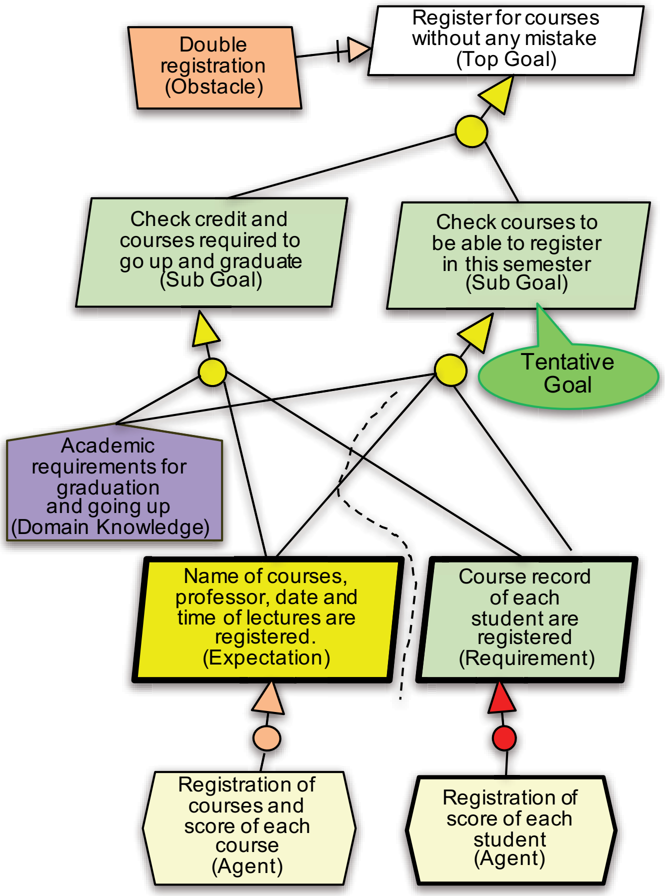


Fig.1. An Example of KAOS Goal Model

Fig. 1 is an example of the structured requirements shown by a KAOS goal model of a software system to be developed for solving issues in a complicated registration of courses in a university. If "Check courses to be able to register in this semester" is used as a tentative goal which a student wants to achieve, then "Register for courses without any mistake" can be used as the reason for the existence of the tentative goal.

The identified goal, stating a genuine requirement of the student related to the registration, is then allocated as the top-goal of the software system to be developed. In order to satisfy the top-goal, a finer-grained goal can then be represented as "Check credit and courses required to go up and graduate", and the tentative goal should be achieved. In order to achieve the tentative goal, both "Course record of each student are registered" and "Name of courses, professor, date and time of lectures are registered" are required as sub-goals. If the sub-goal cannot be decomposed into any finer-grained goals, then this goal is allocated as a requirement.

There should be only one agent, the "Registration of score of each student" which is responsible for achieving the requirement, and this is the system that needs to be developed. Otherwise, the "Name of courses, professor, date and time of lectures are registered" should be achieved to satisfy the tentative goal by only one agent, "Registration of courses and score of each course" too. However this agent is the software system implemented for the instruction section. Therefore the software system to register score of each student expects the agent, "Registration of courses and score of each course" to be responsible for achieving "Name of courses, professor, date and time of lectures are registered". This sub-goal is distinguished from requirements. This is named expectations. If an obstacle such as "Double registration" is found as a cause that prevents achievement of the top-goal, the cause is allocated as an obstacle in the structured requirements. "Academic requirements for graduation and going up" is allocated as domain knowledge in order to satisfy the sub-goal.

B. Role-Play in an Online Group-Work Training

RP training has several significant advantages over the on-the-job training (OJT): 1) a learner can experience management from various different perspectives, not only that of a project manager in a vendor company but also that of a customer; 2) a learner can participate in many more simulated experiences than with OJT; and 3) a learner can take part in extremely difficult projects and can practice RP exercises which target specific skills for requirements engineering.

Since 2007, we have been developing an online group-work training environment named PROMASTER (Project Management Skills Training Environment) in order to collate the behavioral track records of each student, and evaluate the educational effectiveness of project management education [4][5]. We are now trying to apply PROMASTER to requirements engineering education in order to provide training in the human-related skills needed in the requirements elicitation process. Group-work is also necessary in order to elicit requirements and analyze them from various different perspectives.

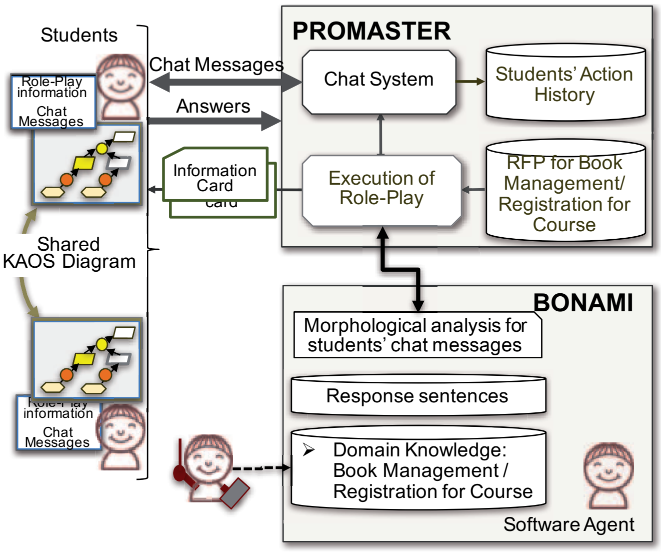
The software agent system BONAMI (an agent system Based ON Aggregated Mentoring and expert Intelligence for project management) plays the role of a mentor - providing a depth of experience in developing systems [6][7]. It works with PROMASTER to support students by giving advice, as necessary, so that they can take appropriate and desirable action. To fulfill this role, the software agent is required to encourage the students to adopt the roles of the stakeholders in a RP exercise, to focus on the exercise problems, to collect information, to move the exercise forward, to participate in discussions, and to help wrap up their discussions.

Domain knowledge including constraints on developing the system such as regulations affecting a customer's business, dependency relationships among stakeholders, interests, and the causes of problems are all examples of business domain-specific information which can be implemented in BONAMI as messages spoken by the agents.

C. RP Training for Requirements Engineering Education

The RP training environment, in which PROMASTER is used in conjunction with BONAMI to respond to students' questions concerning business matters during RP, is shown in Fig. 2. All the information necessary to run a RP in PROMASTER is described in a RP scenario, which is developed in advance. Each scenario describes a virtual project, exercises related to the skills to be practiced, and details of how the RP is expected to proceed. The scenario is composed of a set of information cards. PROMASTER sends the information cards to the students' PCs. Students use two PCs, one for progressing RP training and another for sharing the KAOS diagrams that they create.

Two students receive information regarding the state of the virtual project from the information card displayed on the screen of the PCs, and hold discussions by exchanging chat messages in order to progress role-play training. In order to refine the requirements from various perspectives, two students share a developing KAOS diagram using another PC display screen. The software agent, BONAMI, acts as the domain expert and monitors and analyzes the chat messages and provides advice or hints, as necessary.



PROMASTER : PROject MAnagement Skills Training EnviRonment
BONAMI :agent system BasedON AggregatedMentoring and expert Intelligence for project management

Fig.2. Role-Play Environment for KAOS Goal Modeling

III. Related Work

The work presented in this paper is focused mainly on requirements engineering education using role-play training. The key skills required by a requirements analyst are those of modeling and interviewing. Requirements engineering training provides a learner with the following exercises: carefully reading a document describing the information concerning a customer's business domain; identifying problems in the current operating environment; the abstraction of underlying problems; and developing the structure of the requirement. Tsumaki points out that in order to acquire the modeling skills required for the requirements analysis process, a member of the teaching staff should provide an unknown subject and the learner should then deal with the abstraction of that problem [10]. The main difficulty in modeling usually arises from the abstraction process, which is the mainstay of modeling but also depends on the individual's own cognitive style.

In past software development aimed at operational efficiency improvement, the product or service, which is the target for modeling, had usually already been abstracted because the business regulations applying to the product or service had generally also been established. A system engineer playing the role of a requirements analyst had no opportunity to model the product or service, even if he or she was capable of doing so and had the necessary requirements analysis skills [9]. However, in the modern, ever-changing information society, the target for modeling is becoming increasingly uncertain. In other words, the scope for development is now not always obvious. Abstraction is of increasing significance in the requirements elicitation and analysis process. Therefore, as part of requirements engineering training, there is a need to provide an exercise in which a learner can challenge existing assumptions, because the outcome of abstraction is dependent on the individual system engineer. Yamamoto points out that past requirements specification training using a model-based technique provided only one correct answer, so that there is now a dissociation between actual system development and the training course for model-based requirements techniques. He confirms that it is important to include a question about diversity in modeling and an interest in what type of model is developed [11].

IV. Experiments for Developing KAOS Diagrams

In order to implement experiments to assess the effectiveness of the techniques described above, RP scenarios describing a RFP for a virtual project and stakeholders were required, along with students to participate in RP training designed to elicit the requirements. This section explains the design of the experiment used to assess the effectiveness of the proposed method for acquiring these requirements eliciting and defining skills.

A. Role-Play Scenarios

In order to evaluate the educational effectiveness of the proposed method, some means of comparing the users' newly acquired skill levels with those derived from the RP training previously carried out is necessary. When students participate for a second time in RP training, they will already know the content of the RP exercise so their acquired skill level may be higher than before. Two different scenarios are, therefore, required in order to avoid repetition. The two RP scenarios used were as follows:

(1) In the research laboratory of a university, many books, magazines, references, papers submitted by research members and design documents for a system developed in the laboratory are kept unfiled. In order to make further research progress, the literature and deliverables need to be organized. Three stakeholders are involved. These include a professor, a student in the laboratory, and a student assigned to manage the literature and deliverables. Two students participating in this experimental role-play scenario play the role of the student and the student assigned to the manager, respectively, and together they then proceed to develop the structured requirements model in order to define the requirements and expectations needed to achieve the real aim. This role-play scenario is called "Book Management".

(2) In the School of Computer Science at a university, the course curriculum is overly complicated for student use. In order for students to graduate, a certain number of credits from the fundamental education course, the basic professional education course, and the professional education course are all needed. Every year, some students fail to graduate due to a lack of the required credits. Whenever a new semester starts, many students crowd into the instruction section of the university to ask what courses they have to resister for again in order to obtain the necessary credits. A better registration system is required, in which both students and office staff can easily check the number of credits already obtained and calculate what courses still need to be completed. This role-play scenario is called "Registration for the Course".

There are two versions of each scenario. One involves assistance from the software agents playing a role of a domain expert, while the other does not involve any assistance.

B. *Conducting Experiments*

Twelve undergraduate students in the School of Computer Science at the Tokyo University of Technology participated in the experiments. Nine of the 12 students were third-year undergraduate students who were considered suitable for practicing requirements engineering since they had already completed some professional education courses. Two of them were highly motivated first-year students, but with no technical knowledge. One was a fourth-year undergraduate student.

The RP scenarios required participants to cooperate in teams of two. Six RP teams were selected, and these were then divided into two halves:group-A and group-B.

Before starting any RP experiments, we presented a 20-minute lecture to all 12 students on requirements engineering, and described how to create KAOS goal modeling diagrams. Each RP experiment took about 90 minutes.

Students used two PCs with a 21-inch LED display as shown in Fig.2. One was used for RP, connected to the PROMASTER RP training system through a LAN set up on the university campus. Another was used to draw a shared KAOS diagram. Each team had to find a top-goal, refine sub-goals, elicit requirements, and draw KAOS diagrams in keeping with the RP scenario.

The first experiment was carried out on August 7, 2014. Three teams belonging to group-A practiced RP training based on the "Book Management" scenario - without the assistance of the agents. The other three teams belonging to group-B practiced RP training based on the "Registration for the Course" scenario - without the assistance of the agents.

In order to avoid improvement by practice due to repetition of the same RP training, 40 days later, on September 16, 2014, a second experiment was carried out. Group-A practiced RP training based on the "Registration for the Course" scenario - with the assistance of the agents. Group-B practiced RP training based on the "Book Management" scenario - with the assistance of the agents.

V. RESULUTS OF EXPERIMENTS

A. *Questionnaire Survey*

In order to investigate how students' self-evaluation of their ability to elicit requirements had changed after experiencing the two different types of RP training, we asked them 7 questions after every RP training exercise.

The results of this questionnaire survey are shown in TABLE I. The numbers presented in the column labeled "Number of responses" are the number of responses received to each question. In order to measure the effectiveness of the proposed method, the different replies were given different scores, as shown in the column labeled "Numerical rating scale used for scoring". Multiplying the "number of responses" by the corresponding "numerical rating scale" produced the scored values shown in the column labeled "Score on responses". A subtotal was then calculated for each question, giving an overall score for each of the 4 types of RP experiment.

In the first RP experiment, three teams from group-B did not respond to Question 5, 6 and 7 because they did not find the questions printed on the back of the questionnaire sheet.

TABLE II shows these scores normalized by the range of total values for each question. The maximum value of this range was 12, because if all 6 students responded with the same answer having the maximum numerical rating score of 2, the total would be 12. Likewise, the minimum value of the range was -12 in the case of Question 1, 2, 3, 5 and 7. Therefore, a value of 25 (12 + 1 + 12) was used to normalize the score. In the case of Question 4 and 6, this value was 19 (6 + 1 + 12). The two right-hand columns of TABLE II show the two groups' progress rate in response to each question, expressed as a ratio. Fig. 3 shows these progress rates in bar graph form. The results show that the progress rate for all questions showed an improvement, except for Question 4. Overall, students felt that they had attained the necessary skills for eliciting requirements when using software agents as the domain experts.

The progress rate shown for Question 7 suggests that, after their two experiences of eliciting and defining requirements in

TABLE I. RESULTS OF QUESTIONNAIRE SURVEY

	Software agents support role-play training		No software agenets		Numerical rating scale for scoring	Software agents support role-play training		No software agenets	
	Book Management: Group B	Registration for the Course: Group A	Book Management : Group A	Registration for the Course : Group B		Book Management: Group B	Registration for the Course: Group A	Book Management : Group A	Registration for the Course : Group B
	Number of responses					Score on responses			
Q1. Were there things about the scenarios for “Book Management” or “Registration for the Course” that you didn’t understand?									
Definitely no	0	0	0	0	2	0	0	0	0
Not sure	3	2	2	1	1	3	2	2	1
No	1	1	0	1	0	0	0	0	0
A few	2	1	2	2	-1	-2	-1	-2	-2
Some	0	2	2	1	-2	0	-4	-4	-2
					Q1 subtotal	1	-3	-4	-3
Q2. Could you easily come up with a first requirement?									
Found it easily	1	1	0	0	2	2	2	0	0
Found it	2	2	1	2	1	2	2	1	2
Found it after reading the description and instructions again	2	1	2	1	0	0	0	0	0
Confused, but found it eventually	1	1	2	0	-1	-1	-1	-2	0
Confused	0	1	1	2	-2	0	-2	-2	-4
					Q2 subtotal	3	1	-3	-2
Q3. Could you collect information from one of the shareholders assigned to a software agent?									
Collected enough information	0	0	0	0	2	0	0	0	0
Collected some information	3	3	0	0	1	3	3	0	0
Not sure	1	2	2	1	0	0	0	0	0
Only collected a little information	2	0	3	3	-1	-2	0	-3	-3
Did not collect any information	0	1	1	1	-2	0	-2	-2	-2
					Q3 subtotal	1	1	-5	-5
Q4. Did you find “Expectations” to achieve a top “Goal”?									
Found a few expectations	1	4	4	4	2	2	8	8	8
Found one expectation	5	1	1	1	1	5	1	1	1
Did not find any expectations	0	1	1		-1	0	-1	-1	0
					Q4 subtotal	7	8	8	9
Q5. Could you define a software system responsible for achieving goals?									
Easily defined	0	0	1	0	2	0	0	2	
Defined	5	3	0	0	1	5	3	0	
Defined, but with uncertainties	1	3	2	1	0.5	0.5	1.5	1	
Difficult to define	0	0	1	1	0	0	0	0	
Could not define	0	0	1	0	-2	0	0	-2	
					Q5 subtotal	5.5	4.5	1	
Q6. Could you find “Obstacles” that would prevent the achievement of goals?									
Found obstacles and their risk factors, and refined them	0	3	0	0	2	0	6	0	
Found obstacles	5	3	4	1	1	5	3	4	
Could not find any obstacles	1	0	1	1	-1	-1	0	-1	
					Q6 subtotal	4	9	3	
Q7. How did you deal with conflicts of opinion between you and your partner in the team?									
Stuck to my own opinion	0	0	0		2	0	0	0	
Mediated	4	5	3	2	1	4	5	3	
Suggested alternatives and mediated	0	0	0		0.5	0	0	0	
Agreed with my partner’s opinion	1	1	2		-1	-1	-1	-2	
Compromised with my partner	0	0	0		-2	0	0	0	
					Q7 subtotal	3	4	1	

teamwork situations, the students felt that they have also learned mediation skills.

B. Portfolio Evaluation

Twelve KAOS goal modeling diagrams were developed by the students during their role-play experiments. Each team developed one diagram supported by agents, and another without agents.

We evaluated these diagrams in terms of whether or not the students had successfully used the techniques required to draw a comprehensive diagram correctly. The evaluation items were as follows [12]:

(1) Has an appropriate top-goal (i.e., the genuine need of a customer) been correctly defined?

(2) How many sub-goals have been refined in order to comprehensively describe the requirements?

TABLE II. PROGRESS RATE

	Software agents support role-play training		No software agents		Progress rate	
	Book Managemen t: Group B	Registration for the Course:	Book Management : Group A	Registration for the Course:	Group A	Group B
Q1. Were there things about the scenarios for “Book Management” or “Registration for the Course” that you didn’t understand?						
	13.50	9.50	8.50	9.50	**1.12**	**1.42**
Q2. Could you easily come up with a first requirement?						
	15.50	13.50	9.50	10.50	**1.42**	**1.48**
Q3. Could you collect information from one of the shareholders assigned to a software agent?						
	13.5	13.5	7.5	7.5	**1.8**	**1.8**
Q4. Did you find “Expectations” to achieve a top “Goal”?						
	16.50	17.50	17.50	18.50	**1.00**	**0.89**
Q5. Could you define a software system responsible for achieving goals?						
	18.00	17.00	13.50		**1.26**	NA
Q6. Could you find “Obstacles” that would prevent the achievement of goals?						
	13.50	18.50	12.50		**1.48**	NA
Q7. How did you deal with conflicts of opinion between you and your partner in the team?						
	15.50	16.50	13.50		**1.22**	NA

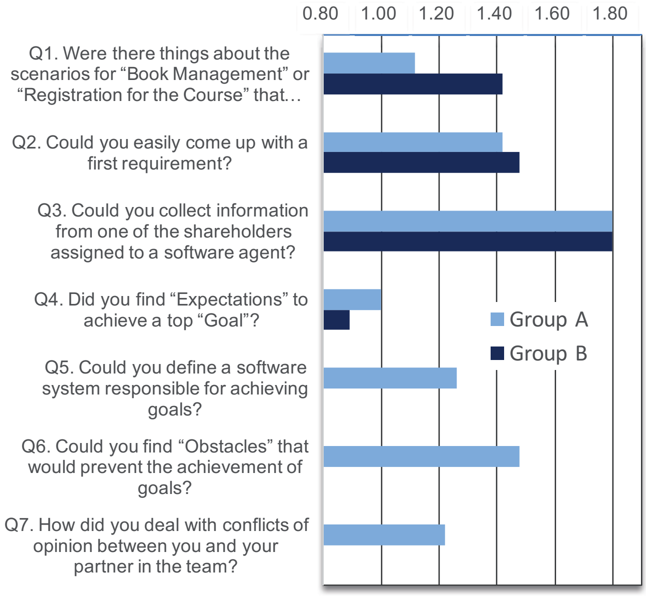


Fig.3. Progress Rate based on Students Responses

TABLE III. VALUATION OF KAOS GOAL MODELING DIAGRAMS DEVELPOED BY EACH TEAM

	Date for Experiments	16-Sep-14						7-Aug-14					
	Experiment Group	Group A			Group B			Group A			Group B		
	Role-Play Team: 2 students	Team	Team	Team 3	Team	Team	Team 6	Team	Team	Team	Team	Team	Team
	Student Year	st, 4th	1st, 3rd	1st, 3rd	3rd, 3rd	3rd, 3rd	3rd, 3rd	st, 4th	1st, 3rd	1st, 3rd	3rd, 3rd	3rd, 3rd	3rd, 3rd
	Theme and Type of Role-Play	Registration for the Course, Agents Support			Book Management, Agents Support			Book Management			Registration for the Course		
1	An appropriate top goal is refined	1	1	0	0	1	0	1	0	1	1	1	0
2	How many sub-goals are refined?	3	3	5	2	7	7	1	3	3	1	2	3
3	How many requirements are extracted?	3	4	4	3	2	2	3	2	3	2	4	1
4	Is there only one agent responsible for requirements?”	1	1	4	1	1	1	1	1	1	1	1	1
5	How many expectations are refined?	2	1	4	2	1	1	3	3	4	1	2	0
6	Is there only one agent responsible for expectations?	1	1	2	2	1	1	0	3	3	1	2	0
7	How many obstacles are refined?	2	1	1	3	2	1	2	1	1	1	1	0
8	How many new requirements avoiding obstacles are refined?	2	2	1	3	2	1	0	1	1	1	1	0
9	Is there only one agent responsible for new requirements?	2	1	1	0	0	0	2	0	1	0	0	0
	Total score for each team	**17**	**15**	**22**	**16**	**17**	**14**	**13**	**14**	**18**	**9**	**14**	**5**

(3) How many requirements have been extracted, and are they sufficient?

(4) Has only one agent been responsible for these requirements? If not, sub-goals might need to be refined into finer-grained sub-goals or requirements;

(5) How many expectations have been extracted? In order to realize the downsizing of a developed software system, external resources should be used as much as possible;

(6) Has only one agent been responsible for expectations?

(7) How many obstacles have been refined? This question is important in order to avoid the influence of risks;

(8) How many new requirements avoiding obstacles have been refined?

(9) Has only one agent been responsible for new requirements?

TABLE III shows the evaluation results for the 12 KAOS goal modeling diagrams, and each team's score for every pair of experiments.

The improvement in requirements eliciting skills acquired after experiencing two RP training scenarios can be assessed by comparing the score results obtained from the second experiment implemented on September 16, 2014, with the score results from the first experiment implemented on August 7, 2014. Fig. 4 shows the rate of improvement for the skill levels of each team. This is shown in terms of a ratio comparing the scores obtained in the second experiment with those obtained in the first one.

As shown, all teams were able to raise their requirement eliciting skill levels.

This result suggests that the proposed method, in which the software agents acting as the domain experts provide the timely information necessary to create KAOS goal model diagrams, is a useful technique for acquiring requirements eliciting skills.

The rate of improvement in requirements eliciting skills shown by Team 6 was especially high (2.80). However, their score was actually the lowest (19). Since there was an overseas student in Team 6, he may not accurately read and understood the scenario description and instructions, which were written in Japanese. On the other hand, Team 3 obtained the highest score (40), but only achieved a rate of improvement of 1.22. Thus, Team 3 was not able to improve their requirements eliciting skills as much as the other teams. This could be because, in the second experiment, Team 3 was unable to identify an appropriate top-goal. They defined the specification of the registration system as the top-goal, even though the specification should be defined only after refining the goal modeling process.

C. *Educational Effectiveness of Scenarios*

To investigate whether the two role-play scenarios described above were suitable for training students in requirements eliciting skills, the total scores of the different test groups were compared. The total score earned by group-B practicing RP training based on the "Book Management" scenario, with agent supports, was 47 (16 + 17 + 14), as shown in TABLE III. The total score earned by group-A practicing role-play training based on the "Book Management" scenario was 45 (13 + 14 + 18). The ratio of the total score of group-B to the total score of group-A was used to represent the educational effectiveness of the role-play training. For the "Book Management scenario, with agent support", this value was 1.04. In comparison, the educational effectiveness for the "Registration for the Course scenario, with agent support" was 1.93. Based on these results, we can conclude that the "Registration for the Course" scenario is suitable for training requirements eliciting skills, while the "Book Management" scenario is not so suitable and may need to be redesigned.

D. *Pros and Cons of Teams*

Half of the 6 teams participating were able to understand the explanation of requirements engineering given in lecture form before starting role-play training, and were able to construct an appropriate KAOS modeling diagram.

However, Team 2, 3, 5 and 6 could not find the correct top-goal. They may not have understood that the customer's desired outcome is the top-goal in any such diagram, and that the differences between the sub-goal and the requirements are extracted by refining the top-goal and sub-goals. In order to improve educational effectiveness, the students need to be able to identify a genuine goal by interactively exchanging questions and answers with the software agent system, obtaining the necessary domain knowledge, and refining the goal model in order to achieve the top-goal.

Some teams did not understand the basic concepts of requirements engineering. Additional pre-lecture instruction may need to be provided, focusing on the use of KAOS goal modeling to create a diagram by asking questions such as "why this goal is necessary?" and "how can this goal be achieved?"

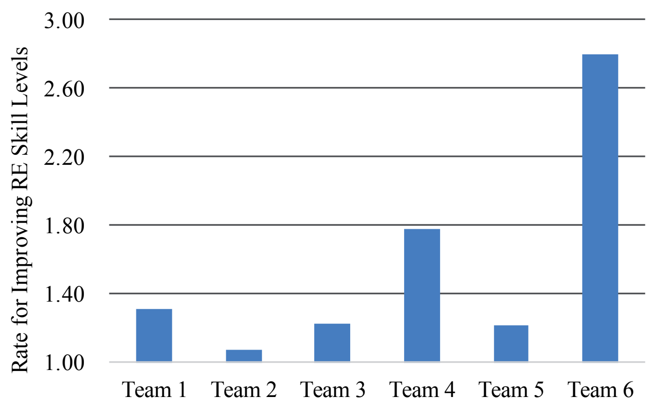


Fig.4. Improving Requirements Skill Levels

VI. CONCLUSIONS

We have proposed a new method for requirements engineering education using role-play training with software agents providing the domain knowledge related to the customer's business field. Portfolio evaluation results showed that students could improve their KAOS goal modeling skill levels using this technique. Questionnaire results from each student's self-evaluation showed that they felt they had improved their capability to elicit requirements, extract sub-goals and expectations, find requirements and the software system responsible for achieving the requirements, discover the risk factors causing obstacles preventing the achievement of goals, and define new requirements. Therefore, we conclude that the proposed method is a useful technique for practicing the techniques of requirements elicitation.

However, we expected students to exchange messages more often between themselves to determine the correct course of action needed in order to create the proper KAOS goal diagram. In future, the effectiveness of the software agents might need to be improved in order to increase the degree of interaction between the students and the software agents playing the role of a domain expert.

There may also have been important differences between the two role-play scenarios adopted. In the role-play training based on the "Book Management" scenario, it was difficult for students to extract the top-goal because the scenario required students to understand the strategy of a research laboratory and its vision for the future. Even the teaching staff could not create a comprehensive diagram describing the model answer. Scenarios requiring explicit goals, more familiar to the students themselves, will have to be developed.

ACKNOWLEDGMENT

This research is supported by JSPS KAKENHI Grant Number 26330408.

REFERENCES

[1] Elizabeth Hull, Ken Jackson and Jeremy Dick, "Requirements Engineering", Springer, 2002.

[2] "Requirements Engineering Body Of Knowledge (REBOK)" Version 1.0, Japan Information Technology Services Industry Association (JISA), Japan, 2011.

[3] Taichi Nakamura, Un Kai, Yuki Tachikaawa, "Requirements Enginering Education usin Expert System and Role-Play Training", IEEE International Conference on Teaching, Assesment, and Learning for Technology (TALE 2014), 2014.

[4] Taichi Nakamura, Yuko Kitaura, Hiroshi Maruyama and Akio Takashima, "Analysis of Learners' Behavior in Role-play Training for Project Management Education", The 9th IEEE International Conference on Advanced Learning Technologies (ICALT2009), pp.144-146

[5] Yuki Tachikawa， Akio Takashima, Hiroshi Maruyama, Taichi Nakamura，"An Analysis of the Relation between the Behavior of a Learner and Acquired Skill Level in Role-Play Training, 6th International Conference on Project Management (ProMAC2012), pp.113-120，2012.

[6] Taichi Nakamura, Akio Takashima, Akane Mikami, "The use of agents to represent learners in role-play training", IEEE 1st Annual Engineering Education Conference (EDUCON2010), pp.185-190, 2010.

[7] Taichi Nakamura, Akio Takashima, Yasuo Sambe, "Role play exercises for project management education that incorporate a software agent", IEEE International Conference on Teaching Assessment and Learning for Engineering 2012 (TALE2012), pp.242-247，2012

[8] Axel van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guide Tour", Proceedings RE'01, 5th IEEE International Symposium on Requirements Engineering, pp.349-263, 2001

[9] Eric Siu Kwong Yu, "Modelling strategic relationships for process reengineering", Ph.D. dissertation, University of Toronto, 1995

[10] Toshihiko Tsumaki, Junko Sirogane, "Introduction to Requirements Engineering", NII Japan, 2009

[11] Shuichiro YAMAMOTO, Masakazu KANBE, "An Experience of Open Exercise on Requirements Engineering Education", IEICE Technical Report KBSE2009-39, pp.49-53, 2009, Japan

[12] Respect-IT, "A KAOS Tutorial" V1.0 Oct.18, 2007, http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf , 30 April 2016