Traceability Editing and Checking Functions for Requirement Management Education Support Tool REMEST

Naoki Maeda
Department of Information Science
Saga University
Saga, Japan

Tetsuro Kakeshita
Department of Information Science
Saga University
Saga, Japan
kake@is.saga-u.ac.jp

Abstract— **Requirements definition and management are very important in modern software development. We are developing a requirement management education support tool named REMEST. REMEST is based on REBOK (Requirement Engineering Body of Knowledge) and provides instructions to the students about current and next RE tasks and student errors contained in the specification which the student is developing. We propose the traceability editing and checking functions for REMEST in this paper. The new functions make use of the correspondence matrix to represent traceability relationship. By conducting and evaluation experiment, we find that the manipulation time to develop traceability matrix becomes 50% less compared to the case without the proposed functions.**

Keywords—Requirements engineering (RE), RE education, Software tool, REBOK, Traceability

I. INTRODUCTION

The importance of software requirement specification and requirement management are widely recognized [1-3]. There is a high social demand for effective requirements engineering (RE) education at universities majored in IT as well as at IT industry. However such education is not enough due to the shortage of teaching staff specialized in RE education. Thus we are developing a RE education support tool named REMEST [4].

REMEST is developed as a plug-in of astah* professional [5] and monitors student activities to develop mind map and activity diagram which represent software specification. REMEST guides a student based on the standard process defined by REBOK (Requirement Engineering Body of Knowledge) [6,7]. When REMEST finds a student's mistake, it will quickly notifies the student about the mistake for the recovery in order to facilitate the RE learning process.

In this paper, we propose a new function to edit and check traceability [8,9] within software specification. Traceability is quite important to maintain consistency within the specification and to improve understandability of the specification by providing reasons of each specification item. Traceability is also important to clarify the scope which require modification within the specification due to a modification at some part of the specification.

The traceability editing and checking functions utilize correspondence matrix to represent traceability relationship. REMEST automatically checks consistency of the corresponding matrix. A student can easily review and edit the matrix by clicking arbitrary cells of the matrix. We compared the proposed functions with the traceability editing and checking function provided by the original astah* professional. The manipulation time to develop a traceability matrix becomes 50% less than the original function. We also have many positive comments from the students about the superiority of the proposed functions through an evaluation experiment.

This paper is organized as follows. We shall introduce REMEST and REBOK in the next section as basic concepts. The traceability editing and checking functions are proposed in Section III. The evaluation experiment and the observation are discussed in Section IV. The related work will be explained in Section V. The last section is devoted to the conclusion and future work.

II. THE REQUIREMENT MANAGEMENT EDUCATION SUPPORT TOOL REMEST

A. REBOK (Requirement Engineering Body of Knowledge)

REBOK [6,7] provides a common BOK shared and used by various types of stakeholders such as users and developers, and for both of enterprise/business software system and embedded software system. Although domain specific knowledge is essential to develop software specification at the domain, REBOK does not contain such domain specific knowledge. This is because REBOK is developed as a common BOK for all domains.

REBOK provides appropriate knowledge to all stakeholders involving in requirements engineering process at appropriate levels of expertise. REBOK also defines the following four main processes as a fundamental RE process.

1. Requirements elicitation
2. Requirements specification
3. Requirements analysis
4. Requirements verification, validation and evaluation

978-1-5386-7447-5/18/$31.00 ©2018 IEEE
DOI 10.1109/IIAI-AAI.2018.00046

The requirements elicitation process is composed of the following eight activities among them. REMEST provides training facilities for these activities since the upper process greatly affects the lower processes.

1. Stakeholder identification
2. Understanding of the current system (As-Is)
3. Modeling of the current system (As-Is)
4. Identification of problems and their causes
5. Goal analysis towards problem solving
6. Identification of means to achieve the goal
7. Modeling of the developing system (To-Be)
8. Requirements identification and specification

B. REMEST

REMEST is designed mainly for beginners of requirements engineering such as student and IT engineer at introductory levels. Hereafter we call them students. REMEST guides the students using the standard process of REBOK. Currently REMEST supports the requirement elicitation process and the corresponding requirement verification process.

REMEST is developed as a plug-in of a UML modeling tool Astah* professional. An astah* professional user can create and edit various types of UML diagrams and other diagrams including mind map and activity diagram. REMEST utilizes mind map for most of the activities listed in the previous section, while it utilizes activity diagram for the modeling of the current and developing systems.

REMEST provides the following four types of major functions.

1. Checking Function of Student's Diagram

   REMEST maintains a set of rules defined as REBOK guidelines. These rules are used to check structural consistency of the mind map and activity diagram. REMEST automatically checks the diagrams when a student modifies a node and/or an edge of the diagram.

2. Recognition Function of the Student's Progress

   Each rule maintained by REMEST corresponds to a certain step of the requirement process. REMEST recognizes the progress of the student through the checking results of the rules.

3. Guide Function for the Student

   REMEST provides guide information to the student by utilizing the recognition function of the student's progress. The provided information is as follows.

   - Explanation of the current activity and step
   - Guide of the next step which the student should work on
   - Result of the checking function
   - Warning of the operation at the wrong part of the diagram
   - Progress of the student

4. Comparison Function with the Right Answer

   REMEST compares the mind map which the student creates and the right answer which the teacher provides. A student can use the comparison function to check the mind map from semantic viewpoint. REMEST also provides editing function of the right answer to the teachers.

Fig. 1 illustrates the main window of REMEST. The major functions can be executed at this window.

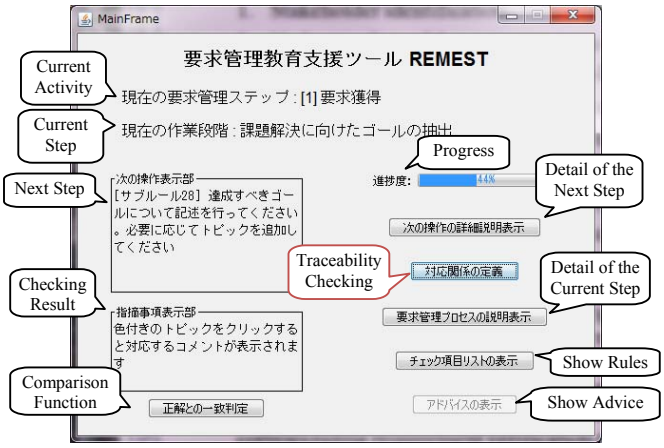

Fig. 1. REMEST Main Window

III. The Traceability Editing and Checking Functions

Astah professional represents traceability using a set of links between topics of the MindMap as illustrated in Fig. 2. However the checking and maintenance of the links are quite complicated as the readers can understand from the figure. The traceability editing and checking functions are designed to overcome these difficulties.

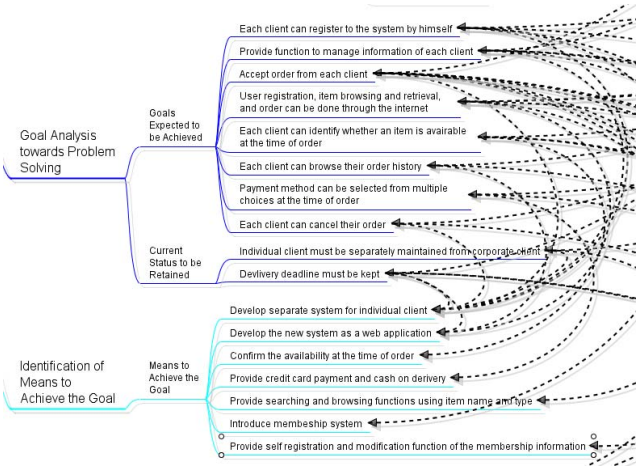

Fig. 2. A MindMap Representing Requirements and Traceability Relationship

A. Traceability Checking Function

It is important to define relationship between the related elements within the requirement. Such relationships are defined for the following four cases. We extract these

relationships by analyzing the REBOK process. We define traceability of the requirement by the set of these relationships.

1. Between the stakeholders and the identified problems
2. Between the problems and the goals to be achieved
3. Between the goals and the means to achieve the goal
4. Between the goals and the extracted specification

Traceability of the requirement is important in order to identify responsibility of each stakeholder for the proposed requirements. Traceability is also important to identify influence to the overall requirements by changing an identified problem, goal, mean and/or specification.

There must be an onto mapping for the both direction between the related elements. For example, each stakeholder must be related to at least one identified problem. Otherwise the unrelated stakeholder cannot have any requirement. Each of the identified problem must have at least one stakeholder who raised the problem and is responsible for the problem.

We illustrate each relationship by a traceability matrix. An example of the traceability matrix for the case of relationship between the stakeholders and the identified problems are represented in Fig. 3. The rows and columns represent the stakeholders and problems. The relationships are illustrated using a "○".

The traceability checking function verifies the onto mapping between the rows and columns. If a column is detected which does not have a related row, the column is identified using a red line showing the outer edge of the column as illustrated in Fig. 3. The row which does not have a related column is identified similarly.

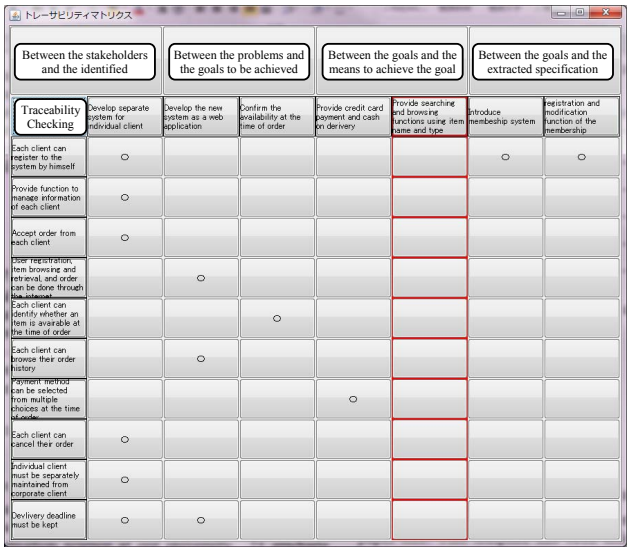

Fig. 3. Traceability Checking Function

B. Traceability Editing Function

The traceability editing function can be executed at the traceability matrix as illustrated in Fig. 2. A user can click a cell at the cross point of the desired row and column. If no relationship is defined between the specified row and column, then a relationship is created. If a relationship is already defined between them, then the relationship will be removed.

IV. Evaluation Experiment of the Traceability Editing and Checking Functions

A. Plan of the Evaluation Experiment

The evaluation experiment was carried out at a course named "advanced topics in software design" at the graduate school of information science at our university. 21 graduate students took the course and joined the experiment. These students are familiar with Astah professional during the course. The evaluation experiment was carried out as shown below.

1. We first distributed REMEST plug-in and REMEST user manual to the students. Students install the plug-in to Astah professional and read the user manual before the actual experiment.
2. We next distributed an Astah professional file representing requirements to the students. The requirement file does not contain traceability information so that we provided the information using a paper-based handouts to the students. This is because we intended to evaluate the difference for the checking and editing process of the traceability. The number of relationships which are missing in the requirement file is represented in Table I.
3. We then assigned two tasks A and B to the students to add traceability information and to check the correctness using two different ways. Task A is to add the information to the requirement file using our traceability checking and editing functions, while task B is to add the information to the original requirement file using the traditional Astah professional functions using links between topics. In order to eliminate the effect of the order of the two tasks, students are assigned the order of tasks so that half of the students work on the task A before task B. The remaining half of the students did the task B first.
4. We also asked the students to measure the elapsed times for the two ways.
5. We asked the students to respond to a survey questionnaire after the two tasks.

TABLE I. Number of Correspondences in the Requirement File

Traceability Relationship between	Number of Correspondences
Stakeholders and the identified problems	21
Problems and the goals to be achieved	11
Goals and the means to achieve the goal	14
Goals and the extracted specification	19
Total	65

As a result, we collected feedbacks and comments from 18 students (85.7% of the students).

B. *Analysis of Student Feedback*

Fig. 4 represents the required times of the proposed method using traceability matrix and the traditional method using links between topics. All students finish the editing of the correspondence more quickly using the proposed method. The ratio of the required time of the proposed method divided by that of the traditional method is distributed in the range between 26% and 81%. The average required time is 48.9% less in the case of the proposed method.

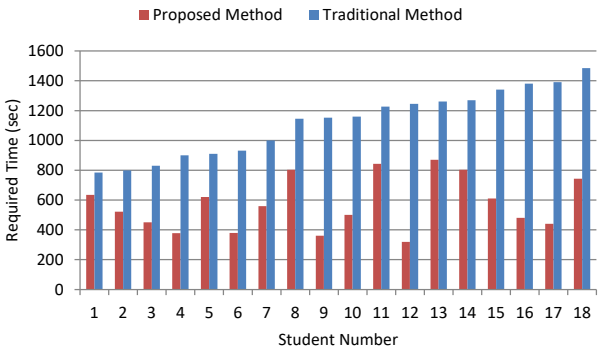

Fig. 4. Comparison of Required Time

Table II represents student's evaluation of the usability of the two methods. All students evaluate that the proposed method is better than the traditional one.

TABLE II. USABILITY COMPARISON OF THE TWO METHODS

Student Evaluation	Number of Votes
The Traditional Method is better	0
The Traditional Method is slightly better	0
Same Usability	0
The Proposed Method is slightly better	4
The Proposed Method is better	14
Total	18

These two facts clearly state that the proposed method using traceability matrix is better than the traditional method.

C. *Analysis of Student Comments*

We also collected comments from the students about usability of the proposed method. The strong points of the proposed method are as follows. The comments are sorted in the descending order of the number of support students.

- Since the traceability matrix can be browsed without scrolling, the proposed method is better to overview the entire relationship. (11 students)
- It is difficult to find appropriate topics in the mind map particularly when the mind map size is large. (7 students)
- Editing and checking of the traceability matrix is significantly easier compared with the traditional method. (3 students)
- The proposed method is better since editing of a relationship can be done by clicking of a cell. (2 students)

The points which require improvement are listed below. We are currently working to improve the traceability editing and checking functions considering these comments.

- Stakeholders are shown instead of the identified problems in the traceability matrix between stakeholders and the identified problems. As a result, same stakeholder appears multiple times in the column of the matrix. (4 students)
- There is a case when the " ○ " representing relationship is not shown in the matrix when the width of the traceability matrix is not enough. A user needs to expand the window size to show the relationship. (4 students)
- REMEST only runs under Windows OS and does not run under Mac OS.

V. RELATED WORK

It is recognized that there are not so many research contribution about requirement traceability as surveyed in [10]. However, there are some works on automatic generation of traceability such as [11]. REMEST focuses on requirement engineering education so that the purpose of our research is different from that of [11].

There are some works on requirement engineering education particularly focused on traceability issue. One example is [12]. However this work focuses on the education process and does not focus on the tool to support education. An effective education process will change through a support of a software tool.

VI. CONCLUDING REMARKS

We propose the traceability editing and checking functions for requirements engineering education in this paper. The functions are integrated into our tool named REMEST and utilize traceability matrix. We observed that the manipulation time to develop traceability matrix becomes approximately 50% less compared with the case using the traditional method using links between topics. Although there are some points that need further improvement, all students responded that the proposed method is better than the traditional method.

As a future work, we are planning an exercise to modify a part of an established requirement. Then the related portion of the requirement must be modified to maintain consistency. Such relationship can be defined using traceability. It is also important to collect and analyze student's learning process data for a teacher to recognize understanding of the students. This

means an extension of REMEST as a tool to support learning analytics.

ACKNOWLEDGMENT

We are appreciate the students who participated in the evaluation experiment of the traceability editing and checking functions. This research is supported by the JSPS Kakenhi Grant numbers16K01022 and 17K01036.

REFERENCES

[1] B. W. Boehm, P. N. Papaccio, "Understanding and controlling software costs", *IEEE Trans. on Software Engineering*, Vol. 14, No. 10, pp. 1462-1477, Oct. 1988.

[2] D. Leffingwell, "Calculating the return on investment from more effective requirements management", *Cutter IT Journal*, Vol. 10, No. 4, pp. 13-16, 1997.

[3] B. B. Roberts, et al., "The Benefit of Integrated, Quantitative Risk Management", International Council on Systems Engineering (INCOSE), 2001.

[4] T. Kakeshita, S. Yamashita, "A requirement management education support tool for requirement elicitation process of REBOK", Proc. 3rd Int. Conf. on Applied Computing and Information Technology (ACIT 2015), Software Engineering Track, pp. 40-45, July 2015.

[5] Change Vision, Astah* professional, http://astah.net/

[6] M. Aoyama, T. Nakatani, S. Saito, REBOK manifest: Towards a Requirements Engineering Body Of Knowledge, *Proc. IEEE International Requirements Engineering Conference* (RE2010), pp. 383-384, Sept. 2010.

[7] JISA REBOK WG, Requirements Engineering Body of Knowledge (REBOK) Version 1, Kindai Kagaku Sha Co., Ltd., 2011. (in Japanese)

[8] K. Takahashi, S. Yamamoto, "An analysis of traceability in requirements document", *IEICE Trans. on Information and Systems*, Vol. E78-D, No. 4, pp. 394-402, 1995.

[9] B. Turban, Tool-Based Requirement Traceability between Requirement and Design Artifacts, Springer Vieweg, 2013.

[10] A. Vegendla, et al., "A systematic mapping study on requirements engineering in software ecosystems", *Journal of Information Technology Research*, Vol. 11, Issue 1, pp. 49-69, 2018.

[11] R. Mordinyi, S. Biffl, "Exploring traceability links via issues for detailed requirements coverage reports", *Proc. IEEE Int. Requirements Engineering Conf. Workshops* (REW 2017), pp. 359-366, 2017.

[12] O.C.Z. Gotel, S.J. Morris, "Case-based stories for traceability education and training", *Proc. IEEE Int. Workshop on Requirements Engineering Education and Training* (REET 2012), pp. 1-8, 2012.