Requirements Engineering, Publisher: Springer-Verlag London Ltd
ISSN: 0947-3602 (Paper) 1432-010X (Online)
DOI: 10.1007/s00766-005-0017-2
Issue: Volume 10, Number 4
Date: December 2005 Pages: 289 - 303
The original publication is available at www.springerlink.com.

Improving the Detection of Requirements Discordances Among Stakeholders

Haruhiko Kaiya‡, Daisuke Shinbara†, Jinichi Kawano†, Motoshi Saeki†
†Graduate School of Information Science and Engineering,
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN
E-mail: saeki@se.cs.titech.ac.jp
‡Faculty of Engineering, Shinshu University
4-17-1, Wakasato, Nagano City 380-8553, JAPAN
Phone: +81-26-269-5469; Fax: +81-26-269-5495
E-mail: kaiya@cs.shinshu-u.ac.jp

Abstract

This paper introduces a technique to identify requirements discordances among stakeholders. This technique is validated in experiments. An extended version of the goal-oriented requirements elicitation method, termed AGORA (Attributed Goal-Oriented Analysis), and its supporting tool are used. Two types of requirements discordances among stakeholders are defined: the first arises from the different interpretations by the stakeholders and the second is the result of different evaluations of preferences. Discordances are detected by the preference matrices in AGORA. Each preference matrix represents both preferences of each stakeholder and the estimated preferences of other stakeholders. A supporting tool for the AGORA method was developed. This tool is a groupware that seamlessly combines face-to-face meetings for goal elicitation and distributed individual sessions for scoring preference values. The experimental results showed that the proposed classification of discordances was sound and that the occurrences of the requirements discordances could be detected by preference matrices.

Keywords: Requirements Elicitation, Goal-Oriented Analysis, Requirements Discordances.

1 Introduction

In order to efficiently develop high-quality software, the elicitation of all potential software requirements of the stakeholders is necessary. Several methods and techniques for supporting requirements elicitation such as goal-oriented analysis [1] and scenario analysis [2, 3, 4] have been developed and implemented. Several successful case studies that assess these methods and techniques have been reported [5]. The tasks involved in requirements elicitation are essentially collaborative, i.e., various stakeholders participate in these tasks, and it is necessary to obtain all the potential requirements from the stakeholders. In order to realize high-quality requirements specification, mutual understanding and agreement among stakeholders regarding the elicited requirements are indispensable. However, in several practical requirements elicitation processes, the mutual understanding and agreement among stakeholders are observed to be insufficient, because the knowledge and/or interests of stakeholders differ with respect to the system to be developed.

Consider a simple example of a conflict in stakeholders' interests. While a customer desires a high-performance product that has multiple functions but a low price, the developers prefer an expensive product that requires less development efforts. In this example, the stakeholders exhibit varying evaluations of the preferences of the requirements. Based on the evaluation criteria of "cost" (payment), the customer has a higher preference for low cost, while the developers do not prefer a product with a low price. Such contrasting preferences lead to conflicts in the elicited requirements. With regard to the conflicts of stakeholders' interests based on their requirements, several projects where the conflicts in quality requirements resulted in serious failures have been reported [6].

In the above-mentioned case, stakeholders have different preferences for an elicited requirement. A different issue arises due to stakeholders' understanding of an elicited requirement. Since a stakeholder's knowledge may differ from that of others, he may sometimes not understand or may misunderstand the requirement that other stakeholders specify. This also leads to low-quality requirements elicitation. Such a misunderstanding of requirements should be detected and resolved while the tasks involved in requirements elicitation are being performed.

In this paper, we focus on the conflicts of interests and the misunderstandings that occur with respect to a requirement, and we term these as *requirements discordances*. We cannot detect discordances without the participation of the stakeholders in the requirements elicitation processes, i.e., stakeholder participatory processes are essential for eliciting requirements without discordances. To elaborate, the detection is not possible until a stakeholder conveys his opinion and understanding of the requirement to the other stakeholders. Therefore, the tool for supporting discordance detection can be considered to be a type of groupware tool, which enables a stakeholder to clearly communicate his intent to others, and it should be used in requirements elicitation processes. EasyWinWin is a groupware tool for requirements elicitation whose practical usage has been reported [7]. However, its goal is to support negotiation among stakeholders after requirements discordances are detected, and not to support discordance detection.

In this paper, we propose a technique to detect discordances while performing the requirements elicitation tasks. To elaborate, we discuss the technique to detect both requirements misunderstandings arising from the different interpretations by stakeholders and requirements conflicts arising from stakeholders' varying evaluations of the preferences. We adopt the extended version of a family of goal-oriented analyses techniques [8, 5, 9] referred to as AGORA (Attributed Goal Oriented Requirements Analysis), which provides syntactic constructs for attaching attribute values to a goal graph in order to evaluate the quality of the requirements [10]. In this approach, each stakeholder can attach his preference value and the estimated preference values of other stakeholders to the elicited goals. The variance of the preference values of the stakeholders enables us to detect the discordances resulting from a misunderstanding of goals and conflicts of interests among stakeholders.

In order to develop a supporting tool based on our approach, we consider the manner in which the stakeholders should collaborate with each other in such processes. A face-to-face session in the presence of a facilitator is preferable in order to elicit goals and develop a goal graph, while the activities for grading the preference values to the elicited goals are a part of distributed individual tasks. Thus, we require a groupware tool that seamlessly supports these two different types of collaborative activities [11]. In this paper, we report the development of a supporting tool for AGORA. This tool is considered to be a groupware tool, and we assess it using three different case studies.

This paper is organized as follows. In the next section, we discuss and clarify requirements discordances. Section 3 is a brief introduction to AGORA based on [10], which addresses the detection of the occurrence of discordances. We also discuss how the variance of preference values attached to a goal graph can reveal the types of requirements discordances, i.e., misunderstanding of goals and conflicts of interests. In section 4, we illustrate a supporting tool for AGORA, which is used in the case studies mentioned in section 5. Section 5 presents the experimental results and findings obtained from the three case studies that detect the occurrence of requirements discordances by using AGORA and the supporting tool. Sections 6 and 7 present related works and concluding remarks, respectively.

2 Discordance among Stakeholders

2.1 Characteristics of Discordances

This section first discusses the relationships among requirements regarding the different views that stakeholders potentially may have. We also provide a definition of requirements discordances in terms of misunderstandings and conflicts of interest. Finally, we discuss how we can detect discordances.

Figure 1 shows three possible problems that may occur when two stakeholders have separate views on the same requirement. These three problems may be described subsequently as follows:

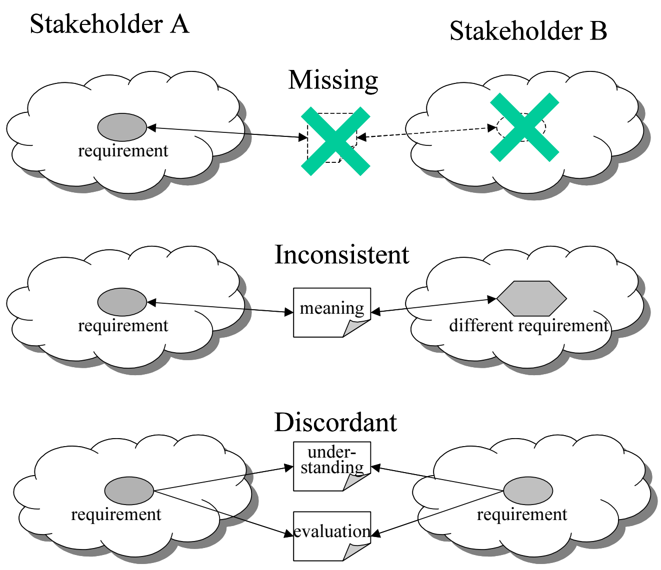

Figure 1. Problems from Two Stakeholders' Views

1. Missing : Stakeholder A has a requirement, while stakeholder B does not. The two stakeholders can detect the missing requirements if stakeholder A discloses all his requirements. The viewpoint approach can be used to detect the location of missing requirements [12].

2. Inconsistent: Although both stakeholders A and B have a requirement with regard to an item, A's requirement is inconsistent with that of B. For example, for the item "the color of the button," stakeholder A requires that "the color of the button should be red," while B requires that "the color of the button should be blue." These requirements of the stakeholders are inconsistent. The technique to detect such an inconsistency regarding multiple requirements, which are represented using graphs, by employing a graph matching algorithm is discussed in [13]. In [14], a requirements specification is defined using a state transition machine, and inconsistencies between two state transition machines are formalized using multi-valued logic.
3. Discordant: There are two cases. (1) Two stakeholders may interpret a requirement differently and (2) their evaluation of preferences of the requirement may differ. We refer to the former type of discordance as *discordance in interpretation* and the latter as *discordance in evaluation*. For example, suppose that a requirement "data exchange should be secure," exists. Two engineers X and Y are the stakeholders. Engineer X is aware of a cheap COTS for secure communication, while engineer Y is not. In such a situation, X would consider it easy to satisfy the requirement, while Y would consider it difficult to implement this functionality from scratch. Thus, X evaluates this requirement to be more preferable but Y does not. This is an example of a discordance in evaluation arising from knowledge gap between X and Y.

As mentioned above, many approaches have been reported in literature in order to resolve the issues in categories 1 and 2. However, to the best of our knowledge, the issues in category 3 have not been addressed, although discordance is a critical problem in requirements elicitation. In this paper, we focus on the issue of discordance among stakeholders, and on the technique to detect discordance in particular. The technique allows us to overcome the problems of mutual understanding and agreement for each requirement.

We now consider the mechanism of the occurrence of discordances. Figure 2 depicts a model that explains the reason for the occurrence of discordances from the viewpoint of stakeholders' communication of their requirements, in particular, from the viewpoint of a stakeholder receiving a requirement and his thought process. When a stakeholder receives a requirement from another stakeholder, he interprets and understands it based on his underlying knowledge. The case wherein his understanding is different from that of the other stakeholders is described in case (1) above. Therefore, he has an interpretation that is different from that of others. Subsequent to understanding the requirement, the stakeholder evaluates it. Case (2), i.e., different evaluation

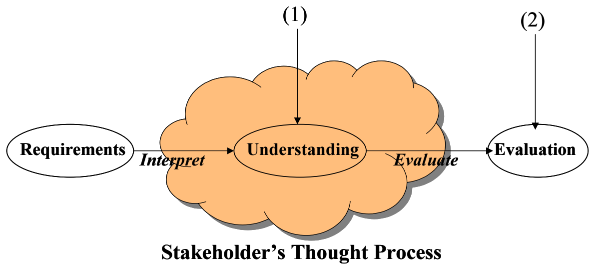

Figure 2. Thought Process of a Stakeholder

of preferences, may occur if he has a different evaluation mechanism although his understanding of the requirements is identical to that of the other stakeholders.

2.2 A Method to Detect Discordances

It is necessary to detect discordances in interpretation and evaluation. In order to detect these discordances, information regarding the stakeholders' understanding and evaluation of a given requirement should be gathered, and this information should be capable of being processed by a computer. Our approach uses a scoring technique to express the information regarding the understanding and evaluation of requirements. We revert to Figure 2 and assume that the two stakeholders A and B understand a requirement and evaluate it. Each stakeholder is allowed to assign a score to his evaluation result of the requirement. These evaluation results are then compared. If A's evaluation result differs from that of B, we assume that there exists a discordance in evaluation of the requirement between the two stakeholders. In other words, stakeholders A and B have a conflict of interest on the requirement.

We now aim to detect discordances in interpretation between A and B. The evaluation result depends on the "understanding" as well as on which stakeholder's preference is evaluated, as shown in Figure 2. Figure 3 demonstrates the basic idea for detecting discordances in interpretation. We assume that stakeholder A can estimate the preference degree not only for himself but also for B. If A's "understanding" of the requirement is different from that of B, the preference degree of B as estimated by A would be different from the degree that B would assign for himself. Since A's understanding is different from that of B, the evaluation result obtained by A should be discordant from that obtained by B. The occurrence of discordances in interpretation suggests that some stakeholders may misunderstand a requirement.

In order to detect discordances, each stakeholder is asked to grade a preference value for a requirement by applying the AGORA approach, which will be explained in detail in the next section.

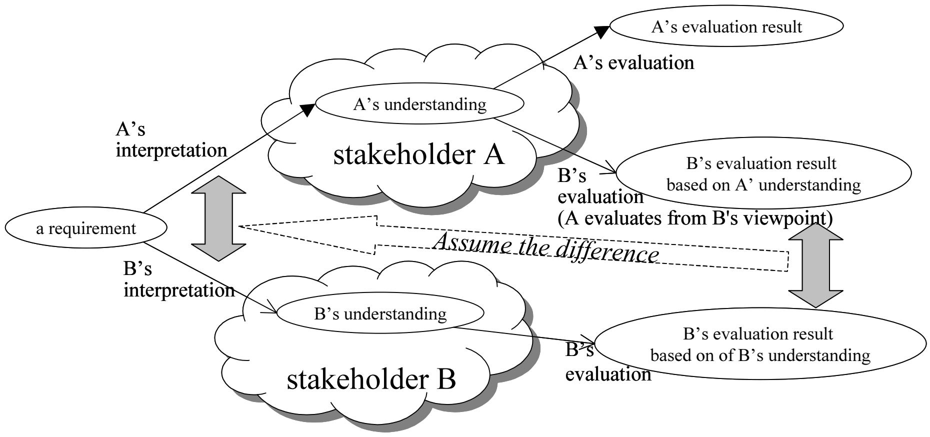

Figure 3. Detecting Discordances in Interpretation

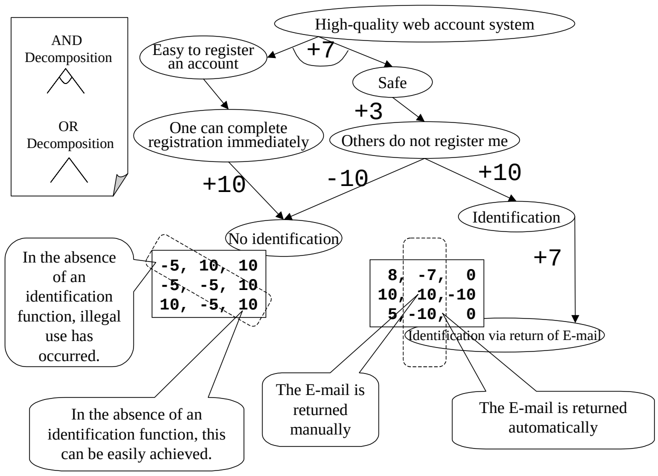

Figure 4. An Example of a Goal Graph in AGORA

3 Attributed Goal Oriented Requirements Analysis

3.1 Overview of AGORA

AGORA is an extended version of the goal-oriented requirements analysis method. Goal graphs are constructed in AGORA, in a manner that is similar to that followed in other goal-oriented methods as shown in Figure 4. Two attribute values, namely contribution values and preference matrices, are introduced in order to enhance requirements analysis using AGORA.

An analyst attaches contribution values and preference values to the edges and nodes of a goal graph, respectively, during the process of refining and decomposing the goals. The contribution value of an edge indicates the degree of contribution of a sub-goal to the achievement of its parent goal, while the preference matrix of a goal represents the preference of each stakeholder for the goal. These values can aid an analyst in selecting and adopting a goal from various alternative goals, recognizing conflicts between the goals, and analyzing the impact of changes in requirements.

According to [10], the values in a preference matrix suggest the occurrences of requirements discordances, therefore, a detailed explanation of the preference matrices will be presented. A preference matrix is attached to a goal, and the matrix indicates the degree of preference of each stakeholder for a goal. Each value assumes an integer between -10 and 10. Each stakeholder not only assigns a preference value for himself but also estimates the preference values for other stakeholders. Hence, the preference for a goal is represented in the form of a matrix. Each stakeholder estimates the preferences of other stakeholders by using preference matrices. This allows him to attain a greater awareness of the thought process of other stakeholders, and it aids the stakeholders' understanding of each other.

Figure 5 shows an example of a preference matrix. In this example, three stakeholders, a customer (C), an administrator (A) and a developer (D), participate in a requirements elicitation phase and estimate their preference values. Each value along the diagonal of the matrix, 8, 10, and 0, is the preference value that a stakeholder estimates for himself. The values in the first row of the matrix are attached by the customer. The customer estimates the preference values for himself, the administrator, and the developer as 8, -7, and 0, respectively. The stakeholder can attach the rationale for each preference value to the value itself.

In the face-to-face sessions involved in AGORA, a goal graph is constructed with the consent of all the participants, and their opinions are reflected in the graph. Therefore, omission of requirements can be avoided to the maximum possible extent.

Evaluator \ Evaluatee	C,	A,	D
C	8,	-7,	0
A,	10,	10,	-10
D,	5,	-10,	0

C = Customer
A = Administrator
D = Developer

Figure 5. An Example of a Preference Matrix

3.2 A Method to Detect Requirements Discordances using AGORA

As mentioned above, our goal is to identify two types of discordances among stakeholders, namely, "different interpretation" and "different evaluation of preference." We now demonstrate how preference matrices can be used to detect the occurrences of these discordances.

We first demonstrate the matrices can be used to detect the occurrences of varying interpretations. In order to identify these discordances by a preference matrix, we use the variances of each column of the matrix. These variances are referred to the *vertical variance* of a preference matrix. If the variance is zero or sufficiently low, we may assume that the stakeholders share a mutual understanding and a consensus regarding the goal. On the other hand, if the variance is sufficiently high, we may assume that some stakeholder's interpretation differs from that of the others because the grading of the preferences has been done for the same requirement and for the same stakeholder. In the example preference matrix in Figure 5, which is attached to the goal "Identification via return of E-mail" shown in Figure 4, an administrator assigned a value 10 to the goal. However, the developer estimated that the preference of the administrator for the goal was -10. It should be noted that both attached their rationale for assigning these values. This large difference suggests that the stakeholders have different interpretations. An administrator who assigned the value 10 may have a different interpretation from that of a customer and a developer who assigned the values of -7 and -10, respectively. The vertical elements include both a positive and a negative value. The calculation of the statistical variance of -7, 10 and -10 yields 116.3, which appears to be rather high. Furthermore the vertical elements corresponding to the customer are 8, 10 and 5, and are all positive values that appear to have a relatively low variance. Similarly, the variance for the developer is not very high. Based on this variance, we could conclude that the varying interpretations resulted from the administrator's view. As shown in Figure 4, the attached rationales reveal that the administrator considered automated return of E-mail for user identification while the developer considered manual return of E-mail. A combination of the preference matrices and rationales are helpful in recognizing and resolving the varying interpretations of goals.

In particular, a preference matrix is said to have *vertical conflict(s)* if both negative and positive preference values are attached in the same column. For example, the preference matrix in Figure 5 has a vertical conflict in the second column. We may assume that the stakeholders have different interpretations of a goal that has a preference matrix with vertical conflict(s).

Next, we demonstrate how preference matrices may be used to detect the occurrences of different evaluations. Since the preference matrix includes the preference degrees for each stakeholder, we can identify the occurrences of different evaluations of preferences, by examining the variance of the diagonal elements of the matrix. These variance are referred to the *diagonal variances* of a preference matrix. For example, the preference matrix on the left-hand side of Figure 4, which is attached to the goal "No identification," has three diagonal elements: -5, -5 and 10. The customer assigned his preference value as -5, while the developer assigned his preference value as 10. This implies that the adoption of this goal is not preferred by the customer, whereas the developer prefers it. In particular, a preference matrix is said to have *a diagonal conflict* if both negative and positive preference values are attached along the diagonal of the matrix. In the example of the goal "No identification" in Figure 4, we may assume that as compared with the developer, the customer and administrator had different evaluation of the preference for the goal, because the matrix attached to the goal has a diagonal conflict (the diagonal elements are -5, -5, and 10).

4 Supporting Tools

4.1 AGORA Supporting Tool

An AGORA supporting tool is used to input and edit goal graphs and attribute values. It may appear similar to a DAG (Direct-Acyclic Graph) editor. Both stakeholders and analysts can use the AGORA editor to input and to edit their preference matrices in particular. Figure 6 is a screenshot of the AGORA editor. This example was used in the experiments that will be described in the next section and in requirements elicitation for a management system for those who graduated from the same university, i.e., members of a certain alumni association. The system to be developed includes supports for automatic input of personal information and for sending-out bulletin kits, e.g., printing mail addresses of the members on envelopes to be sent out. As shown in the figure, a screen consists of two parts: the left part is a viewer for browsing a goal graph being constructed and the right part is used for input and editing the description of a goal and its attached attribute values, such as contribution values and preference matrices. In the left part, the goal graph is displayed as nested and indented text, but not as a two-dimensional visual graphic. The text denotes the goal label. For example, “Printing address seals for overseas member,” which appears highlighted since it has been selected by the analyst, is associated with the goal as a label. The analyst can input and edit a goal label using the “Caption” field on the right part of the screen. A detailed explanation on the goal may be provided in the field “Description.” The rationale for creating the goal and assigning the attribute values can be provided in this field.

The nesting of text in the left part of the figure represents sub-goal relationship in the goal graph, and the nested text is a sub-goal of the text placed above it. In the example shown in the figure, the goal “Managing a name list of alumni association,” which is the top goal, has 14 direct sub-goals in the browser. “Automatically putting a ZIP code on inputting...” and “Identifying the member himself...” are examples of its sub-goals. This form of representation can only be used to define a tree structure. In order to represent a graph structure, we can add links denoting sub-goal relationships between different lines of text on the browser. The analyst clicks on a source goal and a destination goal in the browser after clicking on the “Make Link” button in the command menu on the second line of the screen.

The right part of Figure 6 includes the preference matrix of the goal, and the analyst can input and edit the preference values in the matrix. In this example, there are three stakeholders: a user (shortened to “Use...” on the screen), a developer (“Dev...”) and a member of the alumni association (“Mem...”). The background of the second column of the matrix appears grayed to indicate that a vertical conflict occurs. In this example, the user estimated that the developer would prefer to implement the goal “Printing address seals...,” whereas the alumni member assumed that the developer would not prefer to implement it. The AGORA tool detects and displays vertical and/or diagonal conflicts based on the distribution of the preference values.

4.2 Collaborative Requirements Elicitation by AGORA

This section describes how the editor can be used in collaborative tasks. In a requirements elicitation task using the AGORA method, a facilitator and the stakeholders perform the task in a face-to-face session. Figure 7 is a snapshot of the collaborative requirements elicitation task using the AGORA editor.

Each participant uses a computer connected to a LAN, and the facilitator’s computer is connected to a projector to enable all of the stakeholders to view his computer screen. An AGORA editor runs independently and concurrently on the participants’ computers. In the snapshot, the person seated near the bottom left corner of the projector screen is the facilitator and his AGORA editor screen is being projected. The facilitator encourages discussion among the stakeholders and interviews them in order to construct a goal graph. The facilitator alone can input and edit the goal graph using his AGORA editor and may have a secretary to assist in the inputting and editing of the goal graph. On the other hand, each stakeholder can input and edit only the preference values assigned by him through his computer. A stakeholder cannot view the preference values assigned by the other stakeholders. In other words, each AGORA editor has been customized such that each stakeholder can view and input only his own preference values. To summarize, all participants can view the goal graph, but a facilitator alone can update it. A stakeholder cannot view the preference values assigned by other stakeholders. The facilitator can view all preference matrices, but cannot update them.

Figure 7. A Snapshot of a Collaborative Requirements Elicitation Task

As shown in Figure 8, a collaborative requirements elicitation task session comprises two types of sub-tasks: a cooperative task and a distributed individual task. In the cooperative task, the facilitator interviews the stakeholders and constructs a goal graph using his AGORA editor. The graph is projected onto the projector screen, as shown in Figure 7, and the stakeholders can observe it being constructed. This encourages the stakeholders to incrementally construct a goal graph during face-to-face discussions. The stakeholders can also view the goal graph through their AGORA editors. The goal graph is constructed with the consensus of all participants.

After completing the construction task, the facilitator distributes the graph and encourages the stakeholders to input their preference values through their AGORA editors. In Figure 8, the arrow labeled “1” between the facilitator and the stakeholders indicates that the stakeholders receive the goal graph. At this point, as the preferences begin to be graded by the stakeholders, the session shifts from a cooperative task to a distributed individual task. In this distributed individual task, a stakeholder is unaware of the preference values assigned by the other stakeholders and cannot discuss how to grade the goals. For example, stakeholder S4

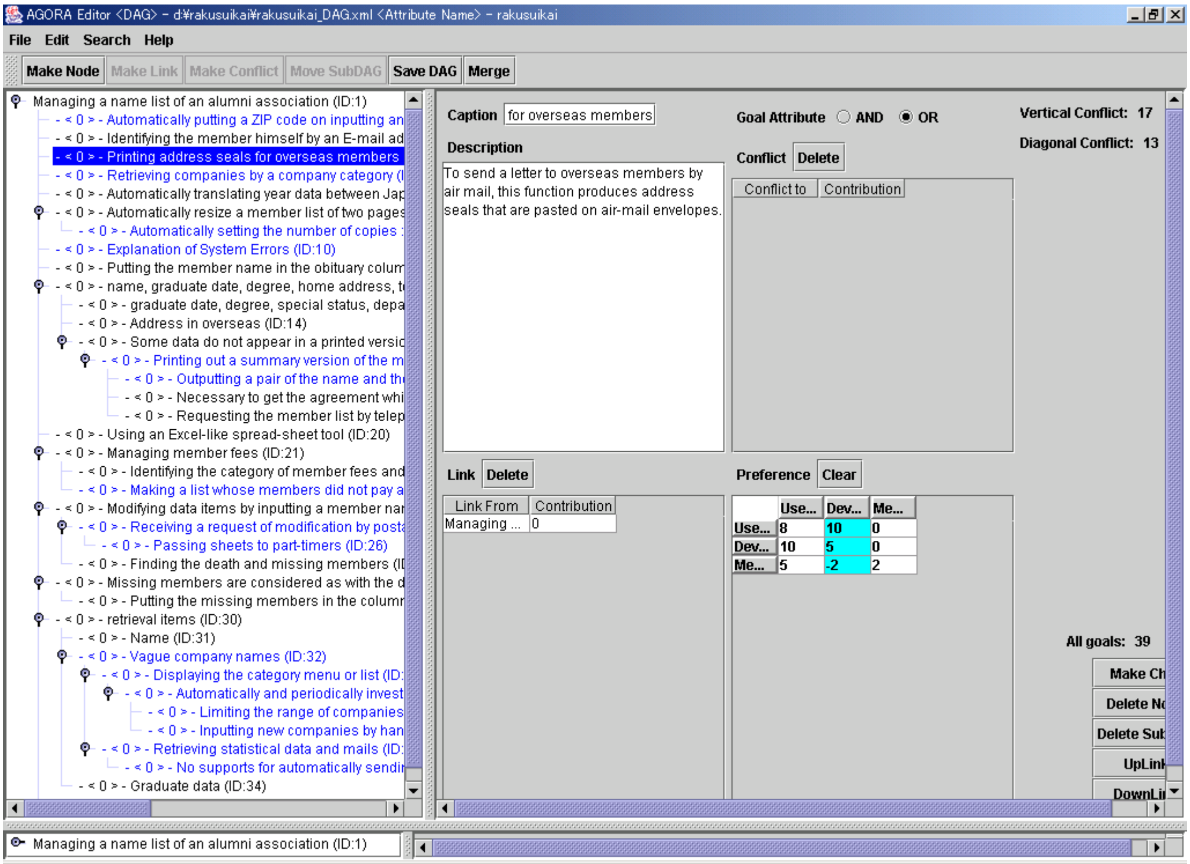

Figure 6. A Snapshot of a Screen in an AGORA tool

assigns preference values of 6, −5, −4, and 5 to goal D, and based on these values, he assumes that S2 and S3 are not satisfied if goal D is implemented. He is not permitted to discuss this scoring with the other stakeholders S1, S2, and S3.

After the grading of the goals is completed, the facilitator's AGORA editor automatically collects and merges the preference values. The arrow labeled "2" in Figure 8 denotes this process of collecting the preference matrices. The AGORA editor also detects the vertical and/or diagonal conflicts by examining whether positive and negative values are both present in a vertical and/or diagonal line in each preference matrix. If the facilitator detects vertical and/or diagonal conflicts as requirements discordances, he may initiate a negotiation process in order to resolve these conflicts or proceed with the evolution of the goal graph (arrow 3 in Figure 8). These processes are cooperative tasks; thus, our requirements elicitation task proceeds by interleaving and iterating through the two types of sub-tasks, namely, cooperative and distributed individual tasks. The support for negotiation processes is beyond the scope of this paper.

5 Experimental Analysis

In this section, we describe and discuss experiments to validate our hypotheses presented in sections 2 and 4.2. To elaborate, our experiments aim to clarify the followings:

1. The technique described in sections 2.2 and 4.2, i.e., the use of preference matrices in AGORA to detect requirements discordances
2. The validity of the classification of the discordances mentioned in section 2.1, i.e., "different interpretations" and "different evaluation"

   We also explore the other types of requirements discordances if any.

In the subsequent sub-sections, we present our experimental procedure, results, and discussions of the results.

5.1 Experimental Procedure

In our experiments, we adopt a collaborative requirements elicitation technique by using AGORA, as stated in section 4.2. After the elicitation processes, we extract the occurrences of vertical and diagonal conflicts from the re-

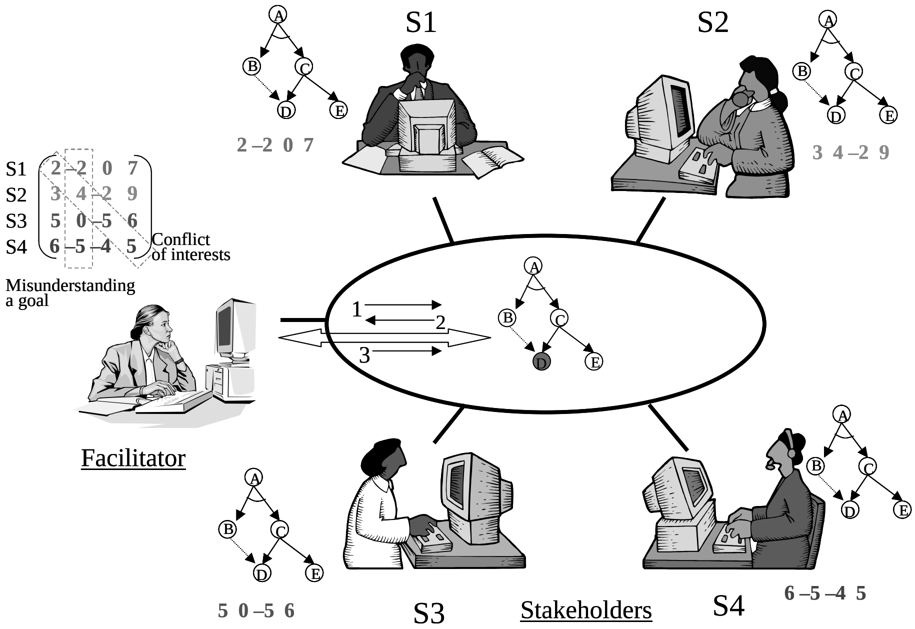

Figure 8. Requirements Elicitation Meeting using an AGORA Tool

sulting goal graphs, and conduct a more detailed examination of their characteristics and causes in order to determine whether they are truely discordances.

As stated in sections 2.2 and 4.2, we may assume that in a goal graph, requirements discordances occur at the nodes that have vertical or diagonal conflicts in their preference matrices. Since our classification is based on the reasons for the occurrences of discordances, we focus on these nodes and identify the reasons for stakeholders assigning conflicting values in the matrices. We can decide whether discordances occur at a node by analyzing the identified causes. Furthermore, we determine whether the identified cause results from different interpretations or different evaluations and classify the occurrences of discordances accordingly.

[Problems and subjects]
In order to effectively perform the experiments, we should carefully select the problems to be analyzed and the stakeholders who are to be the subjects of the experiments that discordances occur frequently. In addition, the stakeholders selected as the subjects should not have any political or personal relationships with each other that prevent fair discussions during the elicitation processes. In particular, the facilitator in the experiments should be carefully chosen such that he does not bias the construction of goal graphs or scoring of preference matrices.

Table 1 lists the contents of the experiments. In experiments 2 and 3, the customers and users had little knowledge of computers but were experts in the problem domains, i.e., they were extensively involved in their domains. In experiments 2 and 3, we estimated that discordances due to different interpretations may occur because the knowledge and experiences of the subjects differed. On the other hand, in experiment 1, we have chosen a problem wherein some stakeholders, namely, a teacher and a student, have conflicts of interests. We consider that this arrangement enables us to capture more occurrences of discordances of the "different evaluations" category. In all the experiments, the stakeholders assuming the role of "Developer" had a vast experience in developing software, particularly web applications.

[Procedure]
The experimental procedure consists of two stages. The first stage involves performing requirements elicitation processes and the second involves analyzing the candidates of occurrences of discordances, that are detected in the processes.

In the first stage, the stakeholders elicit requirements based on the AGORA approach using the supporting tool. The facilitator interviews the stakeholders to capture their requirements, and constructs a goal graph in cooperation with them during their face-to-face meeting, as shown in Figure 7. He controls the proceedings of the meeting. The stakeholders formulate their preference matrices either at their own discretion or based on the suggestions of the facilitator. They are permitted to discuss their requirements and ideas freely but are not allowed to discuss their preference values.

The second stage begins after all the AGORA tasks are completed, i.e., after the preference matrices are scored,

Table 1. Summary of the 3 Experiments

Experiment No.	Stakeholders	Contents
1	3: Teacher, Student (as users), Developer	A supporting system based on the WWW to facilitate electronic submission of the papers of lecture assignments
2	3: Administrator (as a customer & user), Alumni member (as a user), Developer	A management system for the list of members of an alumni association
3	2: Shop clerk (as a user), Developer	A supporting system for management of goods in a supermarket

and a facilitator interviews the stakeholders to determine whether the goals with vertical and/or diagonal conflicts are truly discordances and to classify the occurrences of discordances. To elaborate, the facilitator identifies the goals with vertical and/or diagonal conflicts by using his AGORA editor. For every candidate of requirements discordances in each of the identified goals, the facilitator asks the corresponding stakeholders (the providers of the preference values) regarding their rationale for providing values that resulted in conflicts in order to examine the reasons for the occurrences of discordances. Subsequently, he classifies the extracted causes into several categories. All stages, including the requirements elicitation stage, are recorded using video cameras and microphones, and an experiment analyst other than the facilitator reviews the classification of the discordances by viewing the recorded proceedings. If the results obtained by the facilitator and the analyst differ, they hold discussions to arrive at a consensus. Allocating two persons to fulfill the role of analyst enables us to obtain results that are more accurate. Figure 9 illustrates the flow of the experimental procedure.

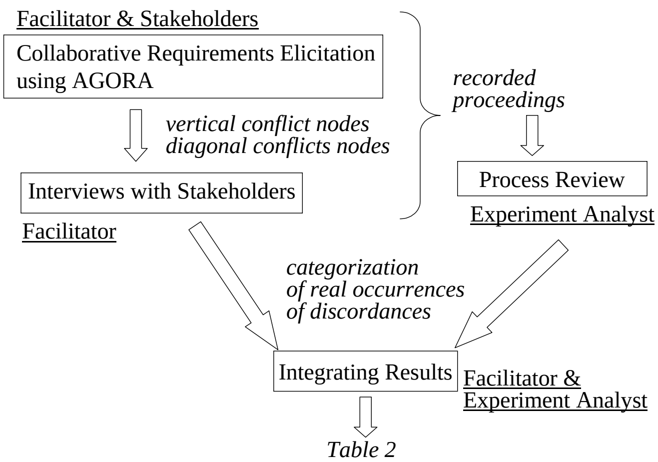

Figure 9. Experimental Procedure

5.2 Experimental Results

Table 2 summarizes the results of the three experiments. All facilitators classified the occurrences of vertical and diagonal conflicts into the following four categories based on the results of the interviews: misunderstanding goals (a), different criteria for evaluating goals (b), relative evaluation (c), and conflicts of interests (D). Although we initially considered only two categories of requirements discordances, as shown in Table 2, a new category (b) "different criteria for evaluating goals" emerged. This is one of the most significant findings of these experiments.

There were no significant differences between the classification of the results obtained by the facilitators and the experiment analysts, and they were able to arrive at a consensus very easily and quickly. All vertical conflicts resulted from (a), (b), or (c), while conflicts of interests appeared as diagonal conflicts.

In the problem of "relative evaluation," which occurred 7 times in experiment 1, as shown in Table 2, the stakeholder did not clearly understand the meaning of the term "preference values". He misinterpreted the preference values to be relative. In order to explain this concept, we consider the following situation as an example. Initially, the stakeholder assigned a low positive value, e.g., 1 to a certain goal; however, when other goals of its brothers in an OR decomposition were relatively less preferrable to him, he had to grade them with lower scores, i.e., less than 1, although he had a positive preference for these goals. As a result, he was forced to assign negative values to these goals. These 7 occurrences in experiment 1 resulted due to such situations, and these may be considered to be unsuccessful in detecting requirements discordances. On the other hand, the occurrences belonging to categories (a), (b) and (D) were determined to be discordances by both the facilitators and experiment analysts.

The following are concrete examples of the discordances obtained from experiment 2.

1. Misunderstanding goals

Table 2. Experimental Results

Experiment No.	1	2	3
Time Expired (hours:minutes)	2:45	2:30	1:30
Total number of elicited goals (A)	41	38	20
Vertical Conflicts (B = a + b + c)	16	13	3
misunderstanding goals (a)	2	7	0
different criteria for evaluating goals (b)	7	6	3
relative evaluation (c)	7	0	0
Diagonal Conflicts (Conflicts of Interests) (D)	10	12	4
Requirements Discordances (Req. Discordances) $(\mathrm{B}+\mathrm{D}-\mathrm{c})$	19	25	7
% of Req. Discordances of all elicited goals $((\mathrm{B}+\mathrm{D}-\mathrm{c})/\mathrm{A})$	46%	66%	35%
% of Req. Discordance of vertical and diagonal conflicts $((\mathrm{B}+\mathrm{D}-\mathrm{c})/(\mathrm{B}+\mathrm{D}))$	73%	100%	100%

The goal "Printing address seals for overseas members" corresponds to the function of producing address seals that are pasted on air-mail envelopes in order to send letters to overseas members. For this goal, the developer and administrator of the alumni association considered printing in English only, while the alumni member considered the possibility of using other languages, such as the Thai language, which do not comprise the symbols of the English alphabet. The alumni member had a different perspective because he had lived in a non-English speaking country. This is a typical example of different interpretations of goals resulting from a knowledge gap.

2. Different criteria for evaluating goals
For the goal "Automatically resize a member list of two pages into one page," the alumni member considered it difficult to automatically resize and re-format a retrieved member list to an A4-size page. On the other hand, the administrator of the alumni association, who has little knowledge of text processing using a computer, considered it to be easy. Both understood the goal correctly, i.e., they shared an identical interpretation of the goal, but their knowledge gap resulted in vertical conflicts in the preference matrix.
3. Conflicts of interests
The goal "Automatically and periodically investigating company names and their categories" implies that the system periodically, e.g., on a monthly basis, accesses internet search engines to locate the home pages of relevant companies and mines the company name and its category from its home page. Although the administrator was aware that the implementation of this goal would be difficult, she expressed a strong preference for this function because of the great convenience it offered. However, the developer did not prefer the implementation of this goal because it required high-level techniques for natural language processing and data mining, and he did not possess the necessary skills to implement these techniques. This is an example of stakeholders having different interests.

The contents of Table 2 will be discussed in detail in the next sub-section.

Although the stakeholders were permitted to provide matrices during the construction of the goals, they provided the matrices after completing the goal graph in experiments 1 and 2 with encouragement from the facilitator. The detection of the causes and their classification during the interview stage were performed with the mutual consent of the stakeholders, and these sessions lasted for approximately 1 hour.

5.3 Discussion

In this sub-section, we primarily discuss the achievement of our goal based on Table 2 (sections 5.3.1 and 5.3.2). In our experiments, we demanded that the facilitators do not play the role of requirements analysts and prohibited them from guiding the stakeholders in their scoring matrices. In the section 5.3.3, we discuss the consequences of this restriction not being imposed, i.e., the issue on group dynamics. Section 5.3.4, presents issues on the scaling of preference values from -10 to 10, closely related to the problem of "relative evaluation."

5.3.1 Detecting Requirements Discordance by using AGORA

As shown in Table 2, in experiment 1, 73% $(19/(16+10))$ of the goals with vertical or diagonal conflicts were actual requirements discordances. On the other hand, all occurrences of vertical or diagonal conflicts in experiments 2 and 3 were actual requirements discordances. Thus we can conclude that the preference matrix method in AGORA is able to detect requirements discordances. However this does not imply that the AGORA approach is capable of detecting all the requirements discordances. In this sense, experiments 2 and 3 resulted in *positive true* only. Ideally, in order to validate the completeness of our approach, we should have explored goals that have discordances but do not have vertical or diagonal conflicts. We should consider another method to determine whether the AGORA approach overlooked the actual requirements discordances.

As shown in the table, 46, 66, and 35% of the elicited goals included requirements discordances taken from experiments 1, 2 and 3, respectively. The presence of this large number of discordance occurrences might lead to the development of software having serious faults. One of the ways to reduce discordance occurrences is the support in cooperative sessions while constructing goal graphs. We can consider methods to reach a consensus or an agreement in face-to-face meetings in order to apply our AGORA method. However, this topic is beyond the scope of this paper and can be considered as a study for future works.

As mentioned above, we observed that some of the participants improperly graded preferences, i.e., "relative evaluation," as shown in Table 2. We had to facilitate their understanding of our method before starting with the meetings.

5.3.2 Classification of Discordance

On exploring the reasons behind the occurrences of vertical conflicts, we came upon some interesting findings. Figure 10 summarizes the resulting classification of requirements discordances and their relationships with vertical and diagonal conflicts. In section 2, we divided the requirements discordances into two categories; the stakeholders have (1) different interpretations and (2) different evaluations of preference. The first category of the discordances appeared in the vertical conflicts as "misunderstanding goals" ((a) in Table 2 and Figure 10).

The second category, "different evaluations of preference" can be further divided into two categories; the first is the case where the stakeholders have different criteria for evaluating their preferences (b), and the second consists of stakeholders having differing interests, i.e., conflicts of interests (D). Categories (b) and (D) appeared in the vertical conflicts and the diagonal ones, respectively, as shown in Figure 10. The "different evaluation criteria (b)" and "misunderstanding goals (a)" resulted from the knowledge gaps among the stakeholders. For example, stakeholders with a limited knowledge of computers frequently considered that some of the functions, which were practically difficult to implement, could be easily implemented. Table 2 shows that 2 vertical conflicts from experiment 1 arose due to misunderstanding goals (different interpretation), 7 arose from different criteria for evaluating goals (b), and all the 10 diagonal conflicts appeared to be conflicts of interests (D).

At the end of this sub-section, we discuss the distinctive features observed among the experiments. Since the stakeholders in experiment 1 had sufficient knowledge about web applications, the total number of occurrences of discordances detected were not as high as those obtained in experiment 2. (The reason behind experiment 3 having a smaller number of requirements discordances will be discussed later.) The conflicts of interests (D) observed in experiment 1 were more than the other types of discordances and they resulted from actual conflicts between the teacher and the student. On the other hand, misunderstanding goals appeared more frequently in experiment 2 than in other experiments since one of the stakeholders (the administrator of the alumni association) did not have enough knowledge about computers. Experiment 3 took on a different style of requirements elicitation. The stakeholders in this experiment discussed goal decomposition and goal contents during the task of scoring in the preference matrices. This is why experiment 3 had fewer occurrences of discordances despite the stakeholder (store clerk) having little knowledge of computers. On the contrary, in experiments 1 and 2, the stakeholders entered the preference values after completion of the goal graph, i.e., after the discussion on the goals; this increased the occurrences of requirements discordances. It can thus be inferred that controlling cooperative tasks (face-to-face meeting to construct a goal graph) is one of the key factors in reducing the occurrences of requirements discordances.

5.3.3 Group Dynamics

In these experiments, the facilitators did not provide the stakeholders with the contents of the goals, but controlled the flow of their discussions, i.e., suggested the goals that were to be discussed. Only the stakeholders discussed the goals and the goal graph. The facilitator asked the stakeholders if they could determine which points under discussion were sub-goals and correspondingly their parent goals. In addition, the facilitators kept track of the goal discussion time. Besides, we did not observe any bias or pressure from the facilitator in the face-to-face sessions.

In these experiments, there was no hierarchy between the stakeholders, such as a superior-subordinate relationship in an organization. We attempted to avoid any bias or pressure in the face-to-face meeting sessions by restricting the role of the facilitators and by carefully selecting the stakeholders. In fact, we did not observe any biased discussions in these experiments. The participants were not biased in any decision power in the discussions, and they could not observe the preference values of the other stakeholders while scoring their own preference values. Thus, the ideas of each stakeholder were precisely noted down, without being influenced by other participants. This led us to conclude that the diagonal and/or vertical differences in the preference ma-

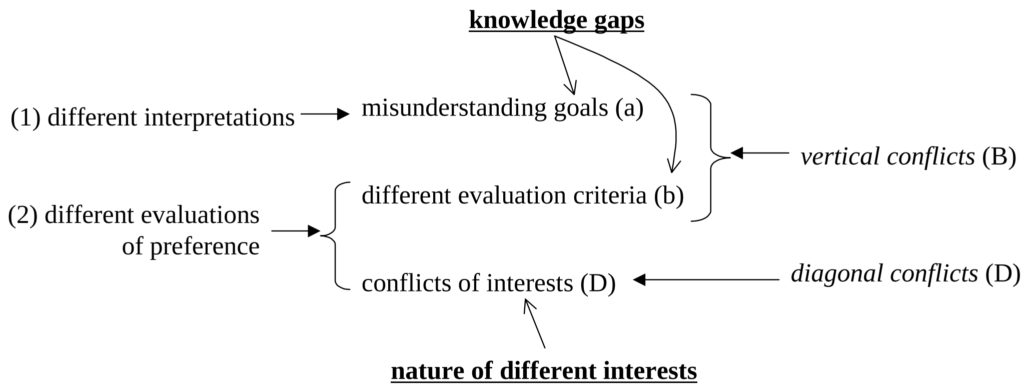

Figure 10. Classification of Discordance and its Relationships

trices resulted from a discordance in the stakeholders' understanding and interests. The following are the possible scenarios if some of the participants had a stronger decision power than the others, i.e., if some of them were biased:

1. A facilitator has the stronger decision power.
   Since the facilitator takes the initiative in deciding goals during the goal-oriented analysis phase, he may create goals reflecting his intents only. This may create goals that do not encompass the intentions or ideas of the stakeholders. Consequently, the essential goals to stakeholders may be absent in the constructed goal graph.
2. A specific stakeholder has the stronger decision power.
   The stronger stakeholder may influence the others to decide on goals that are preferable to him, and may also pressurize them to assign preference values in his favor. This will lead to confusion among the stakeholders. As a result, the discordance will not be detected. In a face-to-face session, the stakeholders can assign preference values to a goal whenever they desire. While the stronger stakeholder expresses his biased opinion, the others tend to assign preference values similar to him, and that causes biased preference values. The influence of the stronger stakeholder can be reduced by avoiding the real-time scoring of preference matrices, e.g., adopting a distributed meeting style and separating the scoring processes from the goal-elicitation ones.

To summarize the above discussion, the bias of decision power among the participants makes discordances impossible to detect. Thus, we must carefully avoid situations involving biased decision power.

5.3.4 Scaling of Preference Values

In AGORA, we can adopt three-level scaling, (e.g., $+1$, 0, -1) for preference values instead of scaling in the range from -10 to 10. In fact, in order to detect discordances, we re-scaled the range from -10 to 10 to comprise three levels (negative, zero, and positive); This approach is essentially similar to the three-level scaling. Although the three-level scaling seemed to be simpler, a stakeholder could not make up his mind whether to assign a score of +1 or 0 to a particular goal in the case that it was preferable but at the same time weak. In the case of the preferable sub-goals being disjunctively connected, i.e., they were derived from OR-decomposition, a positive preference value in the range from 1 to 10 could be used to select a sub-goal as an adopted requirement. The stakeholders could prioritize the sub-goals having positive preference values and adopt a sub-goal having the highest preference value. In fact, some of the participants commented that scaling should have been more flexible and refined. For example, consider a case in which the stakeholder has assigned the scores 3 and 2 to the goals A and B, respectively, and he needs to assign a score to a new goal C, whose preference value lies between that of A and B. In this situation, instead of changing the scores of A or B, he will assign a score of 2.5 to goal C. This is because a change in the scores of A or B could relatively affect the score of the other goals. The current version of AGORA still uses the range from -10 to 10 as a preference value scale, in order to assess the quality of the elicited goals [10]. However, it is uncertain whether this scaling technique of AGORA is suitable for detecting discordances. This issue should be clarified through additional experiments.

6 Related Work

6.1 Cooperative Decision-making Process

From the viewpoint of decision-making processes, requirements elicitation can be regarded as an ill-structured decision task carried out by a group; this implies that the members of the group do not decide on the problems in advance but locate the problems while carrying out the task. In addition, a heterogeneous group is structured so as to include wide variety of statuses in the group, such as social status, position in the organization, diversity of skills, etc. This situation poses an element of risk to the members as it can lead to a biased discussion result. In particular, a member with a lower status would tend to hesitate about disclosing his ideas to the other members, and frequently expresses negative evaluations than he really wants to in response to the results [15]. These biased results allow us to detect discordances in evaluation that are not actual discordances, i.e., *positive false* occurrences may be detected. With regard to the discordances in interpretation, since our approach adopts the technique of scoring goals to detect them, it may also result in incorrect detection. Let us assume that among a group for requirements elicitation there are two stakeholders – a weaker developer and a stronger customer. The weaker developer may think that his customer is not interested in an expensive product, and therefore, he may hesitate in assigning higher preference values to the goals that are expensive to be developed, even though they are preferable for him. In this case, it would have been detected as a discordance in evaluation, i.e., conflicts of interests. This is not actually so because the developer, against his preference, deliberately assigned low score. Some references to contributions in the field of group dynamics [15] can be useful in order to avoid biased discussions in requirements elicitation.

The adoption of "anonymity" is one of the solutions to avoid the biases when the heterogeneity of members' status cannot be excluded. Anonymity has been studied in the field of Group Decision Support Systems (GDSS) [16], [17], and its empirical investigation had been applied in software inspection activities [18]. In [18], anonymity is defined as "an environment in which team participants are unaware of the identity of other team members and their actions." However, according to [15], anonymity could reduce group efficiency because it prevents stakeholders from getting to know one another. Furthermore, our technique is based on the assumption that stakeholders are mutually aware of their evaluation criteria so as to evaluate goals with regard to other stakeholders. Strictly speaking, we adopted a method wherein stakeholders inserted scores while carrying out tasks, and our experiments were not entirely anonymous in nature but had partial anonymity. In fact, in the experiments, each stakeholder was informed that if his score includes vertical or diagonal conflicts in relation with the others, his score would be accessible to the others on completion. In the event that a conflict arises, some amount of negotiation will be necessary to come to agreements or to resolve misunderstandings. Although the degree of anonymity in our approach can be increased, we should carry out more experiences and analyze the influence of anonymity on group dynamics and decision power of the group. The "EasyWinWin" methodology mentioned below also uses a technique to establish anonymity in requirements elicitation [19].

6.2 Detection of Conflicts and Misunderstandings

Several approaches can be employed for specifying, detecting and resolving conflicts among stakeholders. One of the most widely used approaches is WinWin [6], wherein a stakeholder negotiates with other stakeholders having conflicts for trade-offs among their preferences. In this approach, the preferences are referred to as "win conditions." EasyWinWin methodology [7, 19] is a combination of the groupware and WinWin approach. It comprises the sub-activities of gathering, elaborating, prioritizing, and negotiating requirements. Additionally, in order to avoid the occurrence of discordances in interpretation, EasyWinWin includes the "capture a glossary of terms" sub-activity wherein stakeholders can define and share the meaning of important terms and words appearing in the requirements statements. Our approach adopts a scoring technique that initially focuses on vertical conflicts in preference matrices in order to systematically find discordances in interpretations. In addition, stakeholders should attempt to understand other stakeholders' evaluation criteria so as to grade their preference values in the matrices. It is useful to prioritize win conditions because it helps stakeholders to understand the reason that other stakeholders assign a higher priority to a particular condition. Thus our technique can support several activities through EasyWinWin.

A tool called QARCC (Quality Attribute Risk and Conflict Consultant) provided in the WinWin approach, systematically provides suggestions to analysts and stakeholders about the possibilities of potential conflicts by using a knowledge base. In the knowledge base, pairs of conflicting quality attributes, e.g., assurance and performance, are stored. However, the success of this approach largely depends on the quality of the knowledge base and, in general, it is difficult to build such a knowledge base. On the other hand, our approach does not require such a knowledge base in advance.

Several techniques to detect and resolve conflicts of multiple requirements have been studied. A tool called CORA (Conflict-Oriented Requirements Analysis) [20, 21] was used in order to detect and resolve conflicts resulting from the requirements of different stakeholders. In CORA, the requirements are decomposed and structured following the requirements ontology (which has been constructed in advance). The potential for a conflict among requirements to arise can be identified by the structural differences of their decomposition such as differences in types of constraints and constraint parameters. The guidelines of transformations for re-structuring the decomposed requirements are provided in order to resolve their conflicts. KAOS [22] and CREWS-SAVRE [23] also aim at the detection and resolution of conflicts arising from multiple requirements. For resolving conflicts, inference rules written in a temporal logic are used in KAOS and a simulation technique for event sequences specified in scenario descriptions is used in CREWS-SAVRE. These approaches deal with the conflicts arising from multiple requirements and not with requirements discordances among stakeholders (misunderstanding a requirement and conflicts of interests with respect to requirements). In AGORA, "contribution values" can be used to detect the conflicts and inconsistencies among multiple requirements; however, this topic is beyond the scope of this study.

The PRIME-CREWS tool [24] deals with conflicts in the perceptions of different stakeholders. By using this tool, each stakeholder attaches his degree of preference to a goal and/or to a relationship among goals. This tool can detect the occurrence of conflicts by comparing one's degree of preference with that of the other stakeholders. This technique is similar to our diagonal conflict approach. However, our approach can deal with discordances in interpretation by focusing on the vertical conflicts in a preference matrix.

There are several approaches that enable us to model the conflicts among stakeholders. For example, KAOS [22] has a concept "agent," which represents the responsibility to achieve goals. Similar to that of KAOS, Tropos [25] and i* [26] also have a concept called "actor" or "agent." If we can model actors and their intentions in a goal model, then it would be possible to identify conflicts among stakeholders by means of analyzing relationships among agents or actors on a goal graph. Accordingly, our approach can adopt this technique to elaborate on the detection of conflicts underlying the relationships among stakeholders.

Conflict resolution in goal-oriented analysis techniques have been studied [27], [28] and their results can be combined with our approach. The results can be applied to resolve the detected conflicts in a goal graph after detecting conflicts of interests by using our approach. In addition, AHP (Analytic Hierarchy Process) [29] has been frequently used for resolving conflicts in decision-making processes, for example, to prioritize requirements [30]. After detecting conflicts in the stakeholders' interests, AHP can be applied to select the requirements by taking into account the priority degrees calculated by it. In [31], AHP is embedded with i* [26] for prioritizing alternatives, i.e., a set of OR-decomposed goals to achieve a parent goal.

DDP (Defect Detection and Prevention) [32] is also an approach for prioritizing requirements from the viewpoint of the risk of loss of requirements. It includes an analysis on the impact of failure modes if a requirement is not satisfied by the developed system. DDP estimates the degree of reduction of the failure modes when an analyst adopts a technique to mitigate them. This degree can be used to prioritize the requirements and select the necessary ones. Although DDP is applicable for selecting a requirement from a set of conflicted ones, it does not deal with conflict detection.

There are several approaches to dealing with issues of misunderstandings among stakeholders. As discussed in sections 2 and 5, if each stakeholder interprets goals differently, they cannot mutually understand others' goals. Thus, we have to support stakeholders to have shared interpretations. Some formal methods such as Formal Tropos [33] are useful in attaining this end since the descriptions written in a formal language contain rigorous and unambiguous semantics that leave no rooms for differing interpretations. Some formal methods and their supporting tools can simulate or animate the behavior of a system.

Another approach, although not so rigorous, is based on a domain-specific glossary dictionary or thesaurus [34], [35]. By using these, stakeholders can share domain-specific knowledge for understanding requirements. The usage of multimedia artifacts can also enable mutual understandings among the stakeholders. For example, PRIME-CREWS [24], a goal-oriented analysis method, enables stakeholders to attach video recordings of stakeholders' activities to their goals. The hyper-minutes system can transcribe the stakeholders' utterances and link them to a part of a requirements document by using the hyper text technique [36].

7 Conclusion

In this paper, we have proposed a technique to detect requirements discordances among stakeholders by using preference matrices in AGORA. We assumed that this technique could detect two types of discordances, namely, discordances in evaluation and discordances in interpretation. We also assumed that such discordances were reflected in both vertical and diagonal conflicts in a preference matrix of a goal.

This technique was applied into three experiments in order to confirm the following two hypotheses: (1) preference matrices in AGORA can be used to detect requirements discordances, (2) instances of discordance can be classified with respect to the differences in interpretation and evaluation. Discordances were detected by referring to the vertical and diagonal conflicts, both of which were structural characteristics of a preference matrix. Discordances in interpretation were detected by vertical conflicts, and discordances in evaluation were detected by both vertical and diagonal conflicts. We also found that discordances could not be accurately detected by using this technique if the stakeholders did not clearly understand the meaning of preference values. Our classification for discordances seemed to be sound; discordances in evaluation can be further divided into two categories: different evaluation criteria and conflicts of interests.

In the next step, we have to clarify how to resolve these requirements discordances. A promising solution for discordances in evaluation is the EasyWinWin methodology [19] because it allows stakeholders to compromise on their requirements. The goal-oriented methods essentially contribute in reducing misunderstandings about goals since goal hierarchy provides contexts for understanding each goal. In general, goals at the top level in a goal hierarchy are easily misunderstood because the contents of these goals are extremely abstract; further, they do not have enough sub-goals that can provide contexts in the initial phase of a requirements-analysis. Therefore, stakeholders have to locate misunderstandings by the use of preference matrices and decompose the misunderstood goals until they can mutually understand the goals with each other. Recording and managing the evaluation criteria of each stakeholder can mitigate the discordances in evaluation. These records will also contribute to a consistent evaluation of each stakeholder.

The application of AGORA and its supporting tool(s) will be extended to solve the requirements discordances mentioned above, and they can be used synchronously and asynchronously over the Internet, which allows for interoperability with tools for other development phases such as design and implementation tools.

Acknowledgements

The authors would like to thank all the reviewers and participants in REFSQ'04, especially Dr. Jorge J. Garcia-Flores, Prof. Daniel M. Berry, Dr. Charles B. Haley, Dr. Steven J. Bleistein, Dr. Erik Kamsties, Dr. Björn Regnell, and Dr. Vincenzo Gervasi for their discussions and insightful comments on the REFSQ workshop version of this paper.

References

[1] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed Requirements Acquisition. *Science of Computer Programming*, 20:3–50, 1993.

[2] C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyte, A. Sutcliffe, N. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois, and P. Heymans. A Proposal for a Scenario Classification Framework. *Requirements Engineering*, 3(1):23–47, 1998.

[3] J. Leite, G. Hadad, J. Doorn, and G. Kaplan. A Scenario Construction Process. *Requirements Engineering*, 5(1):38–61, 2000.

[4] A. Sutcliffe. Scenario-Based Requirements Analysis. *Requirements Engineering*, 3(1):48–65, 1998.

[5] A. Anton and C. Potts. The use of goals to surface requirements for evolving systems. In *Proc. of 20th International Conference on Software Engineering*, pages 157–166, 1998.

[6] Barry Boehm and Hoh In. Identifying Quality-Requirement Conflict. *Software*, 13(2):25–35, Mar. 1996. IEEE.

[7] Barry Boehm, Paul Grunbacher, and Robert O. Briggs. Easy Win Win: A Groupware-Supported Methodology for Requirements Negotiation . In *23rd International Conference on Software Engineering*, pages 720–721, May 2001.

[8] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Academic Publishers, 1999.

[9] A. van Lamsweerde, R. Darimont, and P. Massonet. Goal-Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt. In *Proc. of 2nd IEEE International Symposium on Requirements Engineering*, pages 194–203, 1995.

[10] Haruhiko Kaiya, Hisayuki Horai, and Motoshi Saeki. AGORA: Attributed Goal-Oriented Requirements Analysis Method. In *IEEE Joint International Requirements Engineering Conference*, pages 13–22, Sep. 2002.

[11] M. Saeki, S. Sureerat, and A. Tanaka. Supporting Distributed Individual Task in Cooperative Specification Development. *International Journal of Software Engineering and Knowledge Engineering*, 10(3):319–344, 2000.

[12] B. Nuseibeh, J. Kramer, and A. Finkelstein. Viewpoints: Meaningful Relationships are Difficult! In *Proc. of 25th International Conference on Software Engineering*, pages 676–681, 2003.

[13] G. Spanoudakis and A. Finkelstein. A Semi-Automatic Process of Identifying Overlaps and Inconsistencies between Requirements Specifications. In *Proc. of 5th International Conference on Object-Oriented Information Systems (OOIS'98)*, pages 405–425, 1998.

[14] S. Easterbrook and M. Chechik. A Framework for Multi-Valued Reasoning over Inconsistent Viewpoint. In *Proc. of 21st International Conference on Software Engineering*, pages 411–420, 2001.

[15] Lisa Troyer. Incorporating Theories of Group Dynamics in Group Decision Support System (GDSS) Design. In *International Parallel and Distributed Processing Symposium*, page 108b, Apr. 2003.

[16] T. Connolly, L. Jessup, and J. Valacich. Effects of Anonymity and Evaluative Tone on Idea Generation in Computer-Mediated Groups. *Management Science*, 36(6):689–703, 1990.

[17] L. Jessup, T. Connolly, and J. Gallagher. The Effects of Anonymity on Group Process in an Idea-Generating Task. *MIS Quarterly*, 14(3):313–321, 1990.

[18] Padmal Vitharana and K. Ramamurthy. Computer-Mediated Group Support, Anonymity, and the Software Inspection Process: An Empirical Investigation. *IEEE Transaction on Software Engineering*, 29(2):167–180, Feb. 2003.

[19] Barry Boehm, Paul Grunbacher, and Robert O. Briggs. Developing Groupware for Requirements Negotiation: Lessons Learned. *IEEE Software*, 18(3):46–55, May/Jun. 2001.

[20] W.N. Robinson and S. Volkov. Conflict-Oriented Requirements Restructuring. GSU CIS Working Paper 96-15, Georgia State University, Atlanta, GA, Sep. 1996.

[21] William N. Robinson and Slav Volkov. A Meta-Model for Restructuring Stakeholder Requirements. In *Proc. of 19th International Conference on Software Engineering*, pages 17–23, May 1997.

[22] A. van Lamsweerde and E. Letier. Integrating Obstacles in Goal-Driven Requirements Engineering. In *Proc. of 20th International Conference on Software Engineering*, pages 53–63, 1998.

[23] Alistair G. Sutcliffe, Neil A.M. Maiden, Shailey Minocha, and Darrel Manuel. Supporting Scenario-Based Requirements Engineering. *IEEE Transaction on Software Engineering*, 24(12):1072–1088, Dec. 1998.

[24] Peter Haumer, Klaus Pohl, and Klaus Weidenhaupt. Requirements Elicitation and Validation with Real World Scenes. *IEEE Transaction on Software Engineering*, 24(12):1036–1054, Dec 1998.

[25] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Sytems*, 8:203–236, Kluwer Academic Publishers 2004.

[26] Eric S. K. Yu. Modelling Organisations for Information System Requirements. In *1st IEEE International Symposium on Requirements Engineering*, pages 34–41, Jul. 1993.

[27] I. Dimitromanolaki and P. Loucopoulos. Goal-Based Conflict Management in Scenario Analysis. In *11th International Workshop on Database and Expert Systems Applications*, pages 831–835, Sep. 2000.

[28] William N. Robinson and Suzanne D. Pawlowski. Managing Requirements Inconsistency with Development Goal Monitors. *IEEE Transaction on Software Engineering*, 25(6):816–835, Nov./Dec. 1999.

[29] T.L. Saaty. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.* RWS, 1990.

[30] Joachim Karlsson and Kevin Ryan. A Cost-Value Approach for Prioritizing Requirements. *IEEE Software*, 14(5):67–74, Sep./Oct. 1997.

[31] N. A. M. Maiden, P. Pavan, A. Gizikis, O. Clause, H. Kim, and X. Zhu. Making Decisions with Requirements: Integrating i* Goal Modelling and the AHP. In *REFSQ'02 Proceedings*, pages 24–35, Essen, Germany, Sep. 2002.

[32] Steven L. Cornford, Martin S. Feather, John C. Kelly, Timothy W. Larson, Burton Sigal, and James D. Kiper. Design and Development Assessment. In *Proceedings of the Tenth International Workshop on Software Specification and Design (IWSSD'00)*, pages 105–114, 2000.

[33] Ariel Fuxman, Lin Liu, Marco Pistore, Marco Roveri, and John Mylopoulos. Specifying and Analyzing Early Requirements: Some Experimental Results. In *Proceedings of 11th IEEE International Requirements Engineering Conference*, pages 105–114, Sep. 2003.

[34] Julio Cesar Sampaio do Prado Leite and Ana Paula M. Franco. A Strategy for Conceptual Model Acquisition. In *First IEEE International Symposium on Requirements Engineering*, pages 243–246, 1993.

[35] Junzo Kato, Motoshi Saeki, Atsushi Ohnishi, Morio Nagata, Haruhiko Kaiya, Seiichi Komiya, Shuichiro Yamamoto, Hisayuki Horai, and Kenji Watahiki. PAORE: Package Oriented Requirements Elicitation. In *Proceedings of 10th Asia-Pacific Software Engineering Conference*, pages 17–26, Dec. 2003. IEEE Computer Society Press.

[36] Haruhiko Kaiya, Motoshi Saeki, and Koichiro Ochimizu. Design of a Hyper Media Tool to support Requirements Elicitation Meetings. In *Proceedings Seventh International Workshop on Computer-Aided Software Engineering (CASE'95)*, pages 250–259, Jul. 1995. IEEE Computer Society Press.