AGORA: Attributed Goal-Oriented Requirements Analysis Method

Haruhiko Kaiya
Dept. of Computer Science
Shinshu University
Wakasato 4-17-1
Nagano 380-8553, Japan
kaiya@cs.shinshu-u.ac.jp

Hisayuki Horai
Dept. of Computer Science
Tokyo Institute of Technology
and
Celestar Lexico-Sciences, Inc.
MTG D17, Nakase 1-3
Chiba 261-5801, Japan
horai@cl-sciences.co.jp

Motoshi Saeki
Dept. of Computer Science
Tokyo Institute of Technology
Ookayama 2-12-1, Meguro-Ku
Tokyo 152-8552, Japan
saeki@cs.titech.ac.jp

Abstract

This paper presents an extended version of Goal-Oriented Requirements Analysis Method called AGORA, where attribute values, e.g. contribution values and preference matrices, are added to goal graphs. An analyst attaches contribution values and preference values to edges and nodes of a goal graph respectively during the process for refining and decomposing the goals. The contribution value of an edge stands for the degree of the contribution of the sub-goal to the achievement of its parent goal, while the preference matrix of a goal represents the preference of the goal for each stakeholder. These values can help an analyst to choose and adopt a goal from the alternatives of the goals, to recognize the conflicts among the goals, and to analyze the impact of requirements changes. Furthermore the values on a goal graph and its structural characteristics allow the analyst to estimate the quality of the resulting requirements specification, such as correctness, unambiguity, completeness etc. The estimated quality values can suggest to him which goals should be improved and/or refined. In addition, we have applied AGORA to a user account system and assessed it.

Keyword : Requirements Elicitation, Goal Oriented Analysis, Quality Metrics

1. Introduction

A process of software requirements analysis consists of requirements elicitation and requirements description. Requirements elicitation is a phase where an analyst collects information from the stakeholders, clarifies the problems and the needs of the customers and users (simply we use "customers" from here), tries to find the best solutions, and makes its planning on what software system will be developed.

A family of goal-oriented requirements analysis(GORA) methods such as I*[11, 15], KAOS[12, 8] and GRL[10] is top-down approach for refining and decomposing the needs of customers into more concrete goals that should be achieved for satisfying the customers' needs. The resulting artifact is an AND-OR graph whose nodes represent identified goals. We can find several case studies and assessments of a goal-oriented method in [2, 17] etc. and it seems to be one of the promising methods for supporting requirements elicitation. However, it does not include the supports for facilitating the following activities;

1. selecting the goals to be decomposed,
2. prioritizing and solving the conflict of goals and the conflict of stakeholders on a goal,
3. choosing and adopting a goal out of the alternatives of the goals as a requirements specification,
4. analyzing the impacts when requirements change,
5. improving the quality of the requirements based on measurement of the quality of the artifact developed by the method.

To solve the above problems and strengthen goal-oriented methods, we have extended them into Attributed Goal-Oriented Requirements Analysis method AGORA, where attribute values, e.g. contribution values and matrices of preference values, are added to AND-OR goal graphs. An analyst attaches the contribution values and preference matrices to the edges and the nodes of a graph respectively during the process for decomposing and refining the goals. The contribution value of an edge stands for the degree of the contribution of a sub-goal to the achievement of its parent goal, while the preference matrix of a goal represents the preference of the goal for each stakeholder. These attributes can help an analyst to choose a goal out of the alternatives, to recognize the conflicts among the goals, and to analyze the impact of requirements changes. AGORA provides quantitative analysis techniques on goals. On the other hand, Chung's approach[5] to non-functional requirements provided qualitative attributes (e.g. positive contribution or negative one) in a goal-oriented method and it cannot handle with goals from quantitative view.

Another contribution of our AGORA is to providing a technique how to estimate the quality of requirements specifications from intermediate artifacts produced during a requirement elicitation process. Almost all of the existing methods are just for supporting the creation of artifacts, neither the measurement of their quality nor their improvement based on the quality measurement. In AGORA, these attached attribute values and the structural characteristics of a graph allow an analyst to estimate the quality of the resulting requirements specification, such as correctness, unambiguity, completeness etc. The estimated quality allows the analysts to recognize which goals should be improved and/or refined. This technique suggests to us how we embed quality measurement into the existing methods and AGORA is one of the examples where a framework of measuring quality is embedded. We can apply this technique, i.e. attachment of attributes to artifacts, to the existing methods.

The rest of the paper is organized as follows. We illustrate AGORA by using a user-accounting system in the next section. Section 3 introduces the quality factors of a requirements specification, which were defined in [9] and IEEE 830 Standard[1] and presents how we can estimate these quality factors from AGORA quality metrics. The AGORA quality metrics can be calculated from the attribute values on an AND-OR graph and its structural characteristics, and we also formally define the expressions for calculating the AGORA quality metrics.

2. AGORA Method

2.1. Overview of AGORA

To make up the support functions that the existing goal-oriented method does not have, an AGORA goal graph is the extended version of AND-OR goal graphs, more concretely it is an attributed AND-OR goal graph. Figure 1 depicts the model of the AGORA goal graphs. The extended parts of AGORA goal graphs can be listed up as follows;

1. Attaching **attribute values** to nodes and edges. These attribute values, in addition to the structural characteristics of the graph, allows us to estimate the quality of the requirements specification that is produced from the graph, as mentioned in the section 3. We have two types of the attributes in the following;

    - **Preference matrix**: It is attached to a node, i.e. a goal, and stands for the degree of preference or satisfiability of the goal for each stakeholder.

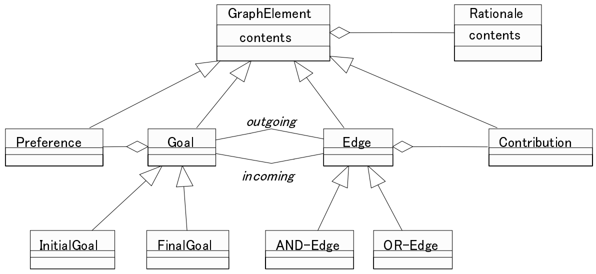

Figure 1. Model of AGORA Goal Graphs

    - **Contribution value**: It is attached to an edge and expresses the degree of the contribution of the goal to the achievement of its connected parent goal.

2. **Rationale**: It can be attached to an attribute as well as a node and an edge. It represents the reasons why an analyst decomposed the goal into such sub-goals, why he chose the sub-goal at the OR branch and why he attached such attribute values to the edge or the node. Attaching rationales is very helpful to maintain the AGORA goal graph.

The procedure to construct an AGORA goal graph is as follows;

1. establishing initial goals as customers' needs,
2. decomposing and refining goals into sub-goals,
3. choosing and adopting the goals from the alternatives of decomposed goals,
4. detecting and resolving conflicts on goals.

In the following sub-sections, we illustrate the details of AGORA method by using a user-accounting system on a Web as an example. Figures 2, 4 and 5 show a series of the snapshots on the process where the analyst is developing the AGORA goal graph of the example problem.

2.2. Establishing Initial Goals

Initial goals can be considered as the needs of the customers, and at first an analyst puts them on the root nodes of the AGORA graph. Figure 2 depicts the first snapshot of the AGORA graph for our example, an user-accounting system on Web. In the example, the initial goals are "high quality", "the purpose of the system is for international and worldwide use" and "every user has an E-mail account", and these are put on the root nodes of the graph as shown in the figure.

2.3. Decomposing and Refining Goals

The second step of the AGORA method, one of the most significant steps, is for decomposing and refining the goals into sub-goals one after another, with the initial goals as

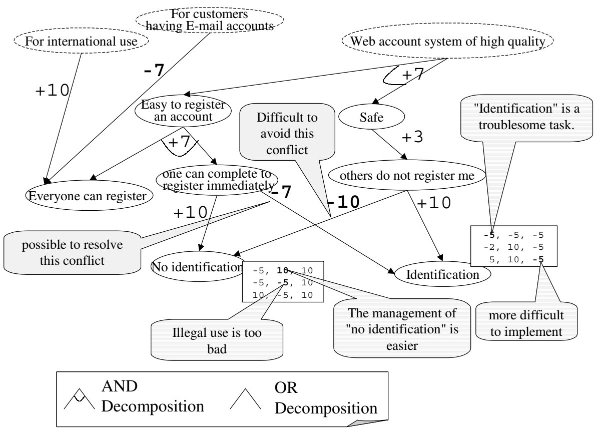

Figure 2. First Snapshot of the AGORA Graph

starting points. The sub-goals are connected to their parent goals with directed edges. The analyst can have more than one sub-goal of a parent goal, and can also use two types of decomposition corresponding to the logical combination of the sub-goals – one is AND-decomposition and the other is OR-decomposition. In AND-decomposition, unless all of the sub-goals are achieved, their parent goal cannot be achieved or satisfied. On the other hand, in OR-decomposition, when at least one sub-goal is achieved, its parent goal can be achieved. The analyst can attach attribute values – preference matrices and contribution values to the goals and the edges respectively. In the example of Figure 2, the analyst attached the preference matrices to the sub-goals "No identification" and "Identification", and the contribution value $+10$ to the edge from "For international use" to "Everyone can register". As shown in the figure, he can also associate with this attribute value the rationale why he had attached that value. For example, the analyst gave the value -5 in the preference matrix of the goal "No identification" because illegal use of accounts is bad. Associating these rationales is very helpful to decompose the goals further and to choose the goals out of the alternatives of OR-decomposed sub-goals. The details of the attributes are as follows.

- Contribution Value: This attribute value is attached to an edge between a parent goal and its sub-goal and it can be an integer from -10 to 10. The value expresses how many degrees the sub-goal contributes to the achievement of its parent goal, and the higher the value is, more contribution the sub-goal provides. The negative value means that the sub-goal blocks the achievement of the parent goal. In Figure 2, the edge from the parent goal "others do not register me" to "No identification" has the negative value -10. If the account system has no functions of user identification, someone can impersonate a correct user and use the system illegally. Thus the sub-goal "no identification" prevents the achievement of protecting impersonation and the analyst gave the negative score. The value 0 on an edge represents that there is no contribution of the sub-goal and we usually do not draw this edge. The analyst can give the different score for each edge in OR-decomposition, while he should attach the same value to all of the edges in AND-decomposition because nothing but the complete set of the edges can contribute to the achievement of the parent goal. In the example of Figure 2, the initial goal "Web account system of high quality" has been refined into two subgoals with AND-decomposition and the contribution value 7 has been attached to them. Unless both of the sub-goals "Easy to register an account" and "Safe" are achieved, their parent goal cannot be done. If both of them are achieved, the parent goal is achieved with the degree 7. In this sense, we can consider that these two goals have the same contribution value. On the other hand, the goal "one can complete to register immediately" is refined into two sub-goals with OR-decomposition, and they have the values $+10$ and -7 respectively.
- Preference matrix: It expresses what degree a stakeholder prefers to a goal or is satisfied, and is attached to a goal. Each of the value also takes an integer from -10 to 10. Each stakeholder does not only attach the preference value of his own, but also estimates the preference values of other stakeholders. As a result, the preference of a goal is represented in the form of a matrix. Figure 3 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 they estimate their preference values. Each value at the diagonal elements of the matrix, 8, 10 and 0, shows a preference value of a stakeholder estimated by himself. The values in the first line of the matrix are attached by the customer, and the customer estimates the preference values of himself, the administrator and the developer at 8, -7 and 0 respectively.

Evaluatee

Evaluator	C	A	D
C	8	-7	0
A	10	10	-10
D	5	-10	0

C = Customer
A = Administrator
D = Developer

Figure 3. An Example of a Preference Matrix

We can use the values in the matrix to find the gap of understanding a goal among the stakeholders, and this kind of stakeholders' misunderstanding or misleading goal frequently results in inappropriate decomposition of the goal. More concretely, to recognize the gap of understanding from a preference matrix, we may use the variance of each column in the matrix. If the variance is zero or sufficiently low, the analyst may decide that the stakeholders mutually understand the goal and have a consensus to it. If the variance is higher to a certain extent, the analyst should analyze the rationales why the stakeholders had provided that varied scores and he should explore the causes of misunderstanding. In the example in Figure 3, which is drawn from the goal "Identification by return E-mail" appearing in Figure 4, an administrator himself gave the value 10 to a goal. However, the developer estimated that the preference of the goal to the administrator was -10. This difference suggests the possibility of the gap of understandings the goal between the two stakeholders. The variance of the vertical elements at the column of the administrator in this matrix, i.e. the variance of -7, 10 and -10, is 116.3, which seems to be too high. On the other hand, the variance of the "customer" column is 6.3, which seems relatively low. According to this variance, the analyst could find that the administrator and the developer had different interpretations, and he could doubt whether the stakeholders had a proper understanding of the goal from the administrator view, and check their rationales. As shown in Figure 4, the attached rationales inform that the administrator considered automated sending an E-mail for user identification, on the other hand the developer considered manual sending the E-mail. The analyst should start making the goal description more concrete or decomposing it into more concrete sub-goals, in order to fill the gap of the stakeholders' understanding. The combination of the preference matrices and the rationales are very helpful to recognize and improve the misleading goals.

The stakeholders attach the value subjectively. However they can use some systematic techniques such as AHP method to decide the more objective values.

Since the preference matrix includes the preference degree for each stakeholder, we can identify the conflicts of the preference or satisfiability on a goal among the stakeholders, by checking the variance of the diagonal elements of the matrix. This topic will be mentioned in the sub-section 2.5.

2.4. Choosing and Adopting Goals from the Alternatives

At least one sub-goal is chosen for achieving its parent goal if the parent goal has been OR-decomposed. The contribution values and preference matrices help the analyst to choose suitable sub-goals.

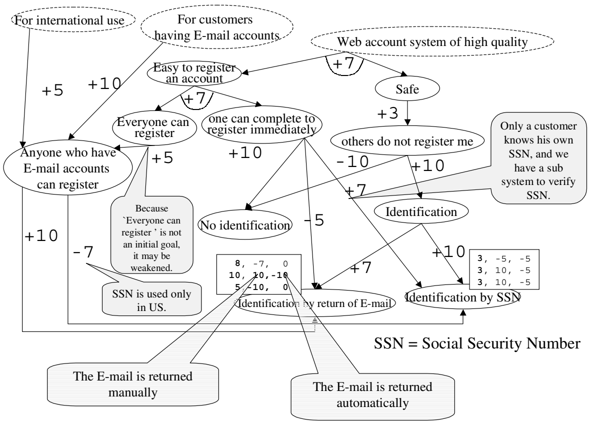

Figure 4. Second Snapshot of the AGORA Graph

Basically, if a sub-goal is connected to an edge having a high contribution value, it can be a candidate for being chosen and adopted as a successor of its parent goal. This is because the contribution value of an edge indicates how the sub-goal connected to the edge is useful to achieve its parent goal.

For example in Figure 4, a goal "others do not register me" is OR-decomposed into two sub-goals "No-identification" and "Identification". The analyst adopts the "Identification" because its edge from the parent goal has the positive and higher contribution value $+10$. Thus the system to be developed includes the function of identifying a user when the user registers to it. This example is very simple and let's see more complicated example in the figure. The goal "Identification" is OR-decomposed into two sub-goals; "Identification by return of E-mail" and "Identification by SSN". The analyst would choose and adopt either of them, and the contribution values $+7$ and $+10$ are attached to their edges respectively. Therefore, the analyst might have chosen the sub-goal "Identification by SSN". However, in the above case, he could not simply decide that because the edge from a goal "Anyone who have E-mail accounts can register" has the negative contribution value -7. If the analyst adopted the goal "Identification by SSN", the system could be used for U.S.A. only and the conflict to the initial goal "For international use" would occur. Consequently, in this case he selected another goal as shown in Figure 5. It means that the analyst cannot decide the adopted goals locally and he should consider all of the edges incoming to the goal that he would like to adopt. We will discuss the details on how to handle with such conflicts in the next sub-section.

Preference matrices also help an analyst to choose goals. Since the system described in a requirements specification is basically for the customers, the preference values of the customers are a good indicator to choose suitable goals. In AGORA, it can be recommended that an analyst may choose the goals having higher preference values of the customers.

In the figure of the final snapshot of Figure 5, a goal "Identification" is successively OR-decomposed into two sub-goals; "By return of E-mail automatically and immediately" and "Identification by SSN". A customer gave the preference value 8 to the former sub-goal, and 3 to the latter sub-goal by himself. Therefore, an analyst may choose and adopt the former sub-goal "By return of Email automatically and immediately" because it has the higher contribution value. As a result, the analyst continues to decompose it and finally gets the three sub-goals, which are enclosed with a shaded rectangle box, as shown in the figure.

2.5. Detecting and Resolving Conflicts on Goals

There are two types of conflicts on goals; one is the conflict between goals and the other one is the conflict on a goal between stakeholders. As mentioned in the last subsection, the example of the first type of conflicts appears in Figure 2 between the goals "others do not register me" and "No identification", whose edge has a negative contribution value. The second type also appears in the figure on the goal "No identification". See the diagonal elements of the preference matrix on this goal. The customer gave the preference value -5 by himself, while the developer's preference is 10 given by the developer himself. It means that if this goal is adopted, the customer is not so preferable while the developer is happy. The analyst can detect these two types of conflicts by investigating the contribution values and preference matrices. We can summarize which goals have the possibility of conflicts as follows;

- two goals that are connected with the edge whose contribution value is negative.
- the goal where the diagonal elements of its preference matrix have a large variance or are much deviated from their average value.

If the graph has a pair of the sub-goals which are connected with the edge having a negative value and the analyst cannot but adopt it, he has to consider the further decomposition into sub-goals so that he can get the sub-goals whose contribution values are increasing. For example, the first snapshot of the AGORA graph shown in Figure 2 has a pair of the goals "one can complete to register immediately" and "Identification", between which edge has the contribution value -7. The analyst decomposed the goal "Identification" further and created the two new sub-goals "Identification by return of E-mail" and "Identification by SSN" in Figure 4. These sub-goals, in particular "Identification by SSN", increase the contribution values to the achievement of the ancestor goal "one can complete to register immediately" because the usage of SSN or E-mail makes the task of user identification less troublesome. In the figure, the contribution value of the edge to "Identification by SSN" is 10 and it results in greater contribution to the achievement of the ancestor goal. This example shows that the contribution values provide some guidelines in what course the analyst progresses the goal decomposition.

When the analyst finds the large variance of the diagonal elements of the preference matrix of a goal, there is a possibility of the conflict among the stakeholders for the goal. In this case, the relevant stakeholders would be forced to negotiate for the conflict resolution of the goal. For example, suppose that the analyst adopts the three goals shown in the bottom of Figure 5. Their parent goal is "By return of E-mail automatically and immediately" and they are related to the usage of One-Time Password (OTP) for users' identification. All of the developer's preference values in these goals, i.e. -5, -8, -5, are quite lower, and it means that the developer does not consider that he is happy to implement these three goals. To resolve the developer's unhappiness, the analyst should negotiate with the stakeholders including the developer, say they will discuss the extension of the deadline of the development and/or a raise in development fee so that the developer will have no problems on taking OTP. Although it is out of scope of this paper how to negotiate with the stakeholders for this kind of conflict resolution, our AGORA method can be seamlessly combined to the supporting methods such as WinWin[3] and DDP[6].

2.6. Terminating Decomposition of Goals

When the decomposed goals grow concrete enough to design and implement an intended software system, the analyst may finish his analysis. One of the guidelines for "concreteness" of the goals is that the goal consists of operational descriptions. See the three sub-goals of the goal "By return E-mail automatically and immediately" in Figure 5. Since they are sufficiently made more concrete so that they include operational statements, e.g. "Input ...", "Issue ..." and "Register...", i.e. the statements have action verbs as main verbs, the analyst stops the further decomposition. We call such goals as *final goals*, which are the leaves of an AGORA graph.

In Figure 5, although we find three groups of final goals, each of which is an option to achieve the initial goals, the analyst adopted one of the three groups which is enclosed with a shaded box in the figure, i.e. the sub-goals of "By return of Email automatically and immediately".

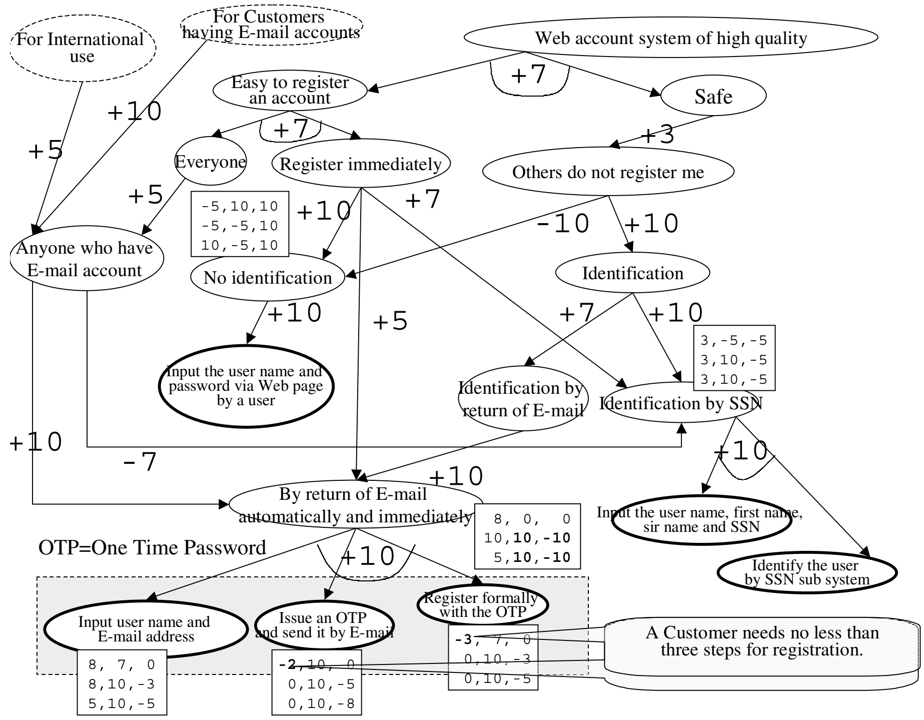

Figure 5. Third Snapshot of the AGORA Graph

3. Quality Measurement

3.1. Quality Metrics of Requirements Specification

McCall et al. defines quality factors of a final product as software quality, such as reliability, efficiency, maintainability, consistency, and usability[4]. It is so impossible to directly measure such a quality of software itself that it is indirectly measured by means of measurable quality metrics such as a precision of computation, a size of program, and a rate of using standard interfaces and data types. This type of quality factors and quality metrics is suitable to measure a source code, but we do not think that it is applicable to measure a requirements specification.

Davis[9] and IEEE Standard 830[1] mention Correctness, Unambiguity, Completeness, Consistency, Verifiability, Modifiability, Traceability and Ranked for Importance & Stability as quality factors of a requirements specification. In this paper, we adopt them to express the quality of a requirements specification. Degrees of measurability and importance in these quality factors vary with the methods that the analyst uses to develop a requirements specification. For instance, it is almost meaningless to measure an unambiguity factor in a formal method because a specification is described in a completely unambiguous formal language.

McCall suggests a measurement method in which a quality factor is calculated by values of comparably measurable quality metrics and a weighting factor matrix. In this paper, we apply this measurement method to quality factors of Davis and IEEE. For each requirements analysis method, we propose specific quality metrics of requirements specifications, which is directly and easily measurable. We also propose a calculation of each quality factors from a set of values of the quality metrics by means of a weighting factor matrix. Quality metrics depend on a requirements analysis method because they are calculated from the final or the intermediate artifacts that are specific to the method. AGORA adopts an attributed AND-OR graph of goals and the quality metrics that originate in attributes and topology of the graph.

The definitions of the quality metrics and a calculation of the quality factors are shown in the next sub-sections.

3.2. Quality Factors and Quality Metrics

In AGORA, customers' needs are described in the initial goals of an AND-OR graph, and a set of the adopted final goals can be considered as a requirement specification. In the example of Figure 5, the customer's needs are the three initial goals "For international use", "For customers having E-mail accounts" and "Web account system of high quality", while the three final goals enclosed with the shaded box are parts of the requirements specification. By evaluating the quality of the adopted goals in the AGORA graph, the analyst can estimate the quality of the final artifact of requirements analysis, i.e. the requirements specification.

Furthermore the results of the evaluation allow the analyst to improve and decompose some goals during the requirements elicitation process. That is to say, the analyst can start the activity of quality improvement before completing the requirements specification. Providing this improvement technique based on quality measurement is one of the key features of AGORA.

It is out of scope of this paper to show the complete set of all measurable quality factors and quality metrics of AGORA. This paper aims that we propose some of them, demonstrate calculation of the quality factors, and show their measurability in AGORA.

The formal definition of the quality metrics in this paper is based on the definition of the AND-OR graph shown in Figure 1. The name of each class and each relation denote a set and a predicate respectively. For instance, $FinalGoal$ and $f \in FinalGoal$ denote a set of the adopted final goals and a adopted final goal f respectively, and $incoming(g, e)$ denotes that edge e comes in goal g. We use 'has' as the name of object aggregation relationship. For instance, $has(g, m)$ denotes that m is the preference matrix of goal g.

Correctness: Correctness is a quality factor which means how many requirements in a requirements specification meet customer's needs. Correctness in AGORA is strongly related to contribution values on the path to the adopted goals. Furthermore, correctness in AGORA is related to the customer's preference value of the adopted final goals.

In this paper, we introduce the following quality metrics for correctness:

- Average of the minimums of contribution values on all of the paths from the initial goals to the adopted final goals (Sat),
- Ratio of edges whose contribution values are positive (Pos),
- Average of customer's preference value of the adopted final goals (Cup)

The formal definitions of these quality metrics are as follows:

$$Sat = (AVE_{f \in FinalGoal}\{Sat(f)\})/10$$
$$Sat(f) = AVE_{g \in \{g \in InitialGoal | \exists p \cdot p \in path(g,f)\}}\{Ctrb(f, g)\}$$
$$Ctrb(f, g) = MIN_{\{e \in path(g,f)\}}\{c \in Contribution | has(e, c)\}$$

$$Pos = \frac{\# \cup_{g \in InitialGoal, f \in FinalGoal} PositivePath(g,f)}{\#\{p \in path(g,f) | g \in InitialGoal \wedge f \in FinalGoal\}}$$
$$PositivePath(g, f) = \{p \in path(g, f) | \forall e \in p, \forall c \in Contribution \cdot (has(e, c) \rightarrow (c > 0)))\}$$

$$Cup = AVE(\cup_{f \in FinalGoal, s \in Stakeholder, m \in Preference} \{m_{s,customer} | has(f, m)\})$$

where $AVE_{p(x)}\{s(x)\}$ and $MIN_{p(x)}\{s(x)\}$ means the average and the minimum of a set of number $s(x)$ constructed from x such that $p(x)$ respectively, $path(g, f)$ means the set of paths from goal g to goal f (a path is a sequence of edges), and $m_{s,customer}$ means customer's preference value evaluated by Stakeholder s in preference matrix m.

The results of the calculation for all of the adopted three goals in Figure 5 are as follows:
$Sat = (5 + 10 + 5 + 3) + (5 + 10 + 5 + 3) + (5 + 10 + 5 + 3)/12/10 = 0.58$,
$Pos = 12/12 = 1.0$,
$Cup = ((8 + 8 + 5) + (-2 + 0 + 0) + (-3 + 0 + 0))/(3 + 3 + 3)/10 = 0.18$.

Unambiguity: Unambiguity is a quality factor which means how many requirements in a requirements specification cannot have more than one interpretation. Unambiguity in AGORA results from the unambiguity of goal descriptions. Since each goal is described in natural language, one of the quality metrics for unambiguity in AGORA can be defined with the number of ambiguous expressions, e.g. "as soon as possible", "immediately", "easy" and so on, in the goal descriptions. In the case of a restricted software domain, it is possible to estimate the unambiguity based on a collective set of common domain specific lexicons which all stakeholders can interpret uniquely. More concretely, we construct a kind of dictionary that includes the standard words in the domain in advance, and count how many words in the dictionary are included in the goal descriptions. Furthermore, it might be possible to estimate the unambiguity from semantic viewpoints of the goal descriptions when natural language processing technique will make progress in the future.

Unambiguity in AGORA is also indirectly measurable from values in a preference matrix. Ambiguity of a goal description leads to a fault of stakeholders' common recognition, i.e. the gap of the stakeholders' understanding of the goal. It appears in the deviation of vertical elements in each column of a preference matrix because these values are the evaluation of a stakeholder's preference by all stakeholders. This quality metrics is defined as follows:

$$Vdv = 1 - \frac{AVE_{s1 \in Stakeholder, f \in FinalGoal}\{Vd(s1, m)) | has(f, m)\}}{10}$$
$$Vd(s, m) = AveDev\{\cup_{i \in Stakeholder} m_{i,s}\}$$

where $AveDev(S)$ means the average of absolute deviations of a set of numbers S.

The result of the calculation for all adopted three final goals in Figure 5 is $Vdv = 1 - (1.33 + 1.33 + 1.78 + 0.89 + 0 + 2.89 + 1.33 + 1.33 + 1.78)/9/10) = 0.14$. The quality metrics of the selected final goals is the average in them. Furthermore, an average of deviation values of each final goal $AveDev$ might suggest whether the goal is ambiguous or not. For instance, the deviation value of the goal "Issue an OTP and send it by E-mail" is the highest (2.89) in the selected final goals, so that this goal can be more unambiguous.

Completeness: In general, completeness is a quality factor which means necessary requirements are not lacking in the requirements specification. In this paper, we make a focus on how many customer's needs could be specified in the adopted final goals.

Completeness in AGORA is strongly related to how exhaustively all initial goals are decomposed to the final goals through the only edges with positive contribution values. This quality metrics is defined as follows:

$$Cov = \frac{\#\{i \in InitialGoal | \exists f \in FinalGoal \cdot AllPositive(i,f)\}}{\#InitialGoal}$$

$$AllPositive(i,f) = (PositivePath(i,f) = path(i,f))$$

In Figure 5, the initial goal and all three adopted final goals are connected by only edges with positive contribution values, i.e. $Cov = 1$.

The number of unambiguous expressions and that of domain-specific lexicons, mentioned in the section of Unambiguity, are also quality metrics for completeness because they are related to the number of undefined words and/or undocumented words.

Consistency: Consistency is a quality factor which means how few inconsistency among requirements in a requirements specification.

The quality metrics for consistency in AGORA include a ratio of edges whose contribution values are positive (Pos), mentioned in Correctness. When an adopted goal has an incoming edge whose contribution value is negative, the goal prevents an achievement of the parent goal. It means that the goal contradicts to the parent goal and might lead to an inconsistency.

The deviation of horizontal elements in each row of a preference matrix is also related to consistency. When the deviation of the values in a row of a preference matrix with a goal is large, there are conflicts among the preference values of he stakeholders and the goal might lead to consistency in the stakeholders. This quality metrics is defined as follows:

$$Hdv = 1 - \frac{AVE_{s1 \in Stakeholder, f \in FinalGoal}\{Hd(s1,m)) | has(f,m)\}}{10}$$

$$Hd(s,m) = AveDev\{\cup_{i \in Stakeholder} m_{s,i}\}$$

The result of the calculation for Figure 5 is $1 - (3.33 + 5.33 + 5.56 + 4.89 + 5.56 + 6.22 + 3.78 + 5.11 + 5.56)/9/10 = 0.5$.

Verifiability: Testing is the only verification method in AGORA because all goals are described in natural language, being different from formal languages that have mathematical verification techniques. The quality metrics for verifiability include how easily appropriate test cases are generated from final goals. It is related to a ratio of ambiguous expressions and a ratio of operational descriptions. A ratio of operational descriptions is measurable by means of the number of the verbs that denote actions.

Modifiability: The quality metrics for modifiability include how an AND-OR graph is closed to a tree structure. When there are many incoming edges to a goal, the goal contributes to an achievement of many goals. In consequence, these many goals should be under consideration in case of changing the goal. The quality metrics is defined as follows:

$$Tre = \frac{\#\{g \in RefinedGoals | \#\{e | incoming(g,e)\} = 1\}\}}{\#RefinedGoals}$$

$$RefinedGoals = Goal - InitialGoal$$

The result of the calculation for Figure 5 is $(12-2)/12 = 0.83$. In the figure there are two goals whose incoming edges are more than one, out of 12 refined goals.

Traceability: There are two types of traceability: one is the traceability among objects in AGORA ('intra-traceability'), and the other one is the traceability between objects in AGORA and objects in the next step of software development process ('inter-traceability'). Inter-traceability depends on a usage of an AGORA final product in the next step, and it depends on which methods the analyst uses in the next step. Since inter-traceability is not measurable without specifying how the analyst will use the final goals of the AGORA graph, we put it aside of this paper.

The quality metrics for intra-traceability in AGORA include a connectivity between final goals and customer's needs (Con) and a degree of attached rationales (Rat).

$$Con = \frac{\#\{f \in FinalGoal | \exists g \in InitialGoal, \exists p \cdot p \in path(g,f)\}}{\#FinalGoal}$$

$$Rat = \frac{\#\{g \in GraphElement | \exists e \in Rationale \cdot has(g,r)\}}{\#GraphElement}$$

In Figure 5, all three adopted final goals are the results of several decompositions of the initial goal, i.e. $Con = 1$.

Ranked for Importance & Stability: This quality factor means how clearly the importance and stability of requirements is described when they are necessary to be specified. Importance of a goal in AGORA can be indirectly specified by means of a preference matrix with the goal. More concretely, the metrics can be the value on how many percentages a preference matrix is attached to the goal where the importance ranking should be really specified. However, just from the AGORA graph, we cannot decide whether a preference matrix should be essentially and really attached to the goal, it is difficult to define formally quality metrics for this quality factor in AGORA.

Table 1. Example of Quality Matrix

	Sat	Pos	Cup	Vdv	Cov	Hdv	Tre	Con	Rat
Correctness	0.5	0.3	0.2						
Unambiguity				1.0					
Completeness					1.0				
Inconsistency		0.6			0.4				
Modifiability							1.0		
Traceability								0.7	0.3

3.3. Calculating Quality Factors

Each quality factor is an average of the relevant quality metrics mentioned above with weighting. Quality factors QF_i ($i = 1, \ldots, n$) is defined by means of values of quality metrics QM_j ($j = 1, \ldots, m$) and a weight matrix $w_{i,j}$ ($i = 1, \ldots, n$, $j = 1, \ldots, m$) as follows:

$QF_i = \Sigma_{j=1,m} w_{i,j} \times QM_j$ where $\Sigma_{j=1,m} w_{i,j} = 1$.

We call the weight matrix $(w_{i,j})$ ($i = 1, \ldots, n$, $j = 1, \ldots, m$) as a quality matrix. We assume that a quality matrix for each requirements analysis method can be given from experience.

An example of a quality matrix of AGORA is shown in Table 1. A blank cell in the quality matrix denotes zero. Each value in Table 1 is just an example for explanation in this paper. Since Table 1 is restricted to the quality metrics defined formally in this paper, verifiability and rank of importance & stability are omitted. The calculation of correctness by means of Table 1 and the values of the relevant quality metrics in the previous subsection are as follows:

$$0.5 \times Sat + 0.3 \times Pos + 0.2 \times Cup =$$
$$0.5 \times 0.58 + 0.3 \times 1.0 + 0.2 \times 0.18 = 0.63.$$

4 Discussion

Connection among the goals through edges allows the analyst to remind missing goals or missing decompositions. Especially when decomposing the goal having more than one parent goal, the parent goals provide multiple views of the goal and an analyst can consider the decomposition from these multiple view. The parent goals play a complementary role in the decomposition of the goal. In fact, in the example of Figure5, at the first step to decompose goal "By return of E-mail automatically and immediately", which has more than one parent goal, the analyst missed the sub-goal "Input user name and E-mail address" because he focused just on the parent goal "Identification by E-mail" and on OTP (One Time Password). However he could find it from another parent goal "Register immediately" because names and E-mail addresses are necessary information for "Register". This is the example to show that the multiple parent goals help an analyst to decompose goals further.

Although a support to decide attribute values was out of scope of this paper, it is one of the most significant techniques in AGORA. In the example of this paper, it was difficult to decide the values of preference matrices. Decision support methods such as AHP[13, 14] and combining AGORA with communication support methods might be one of the promising solutions to solve the problem.

Current AGORA is a top-down requirements analysis from customer's needs as initial goals. There might be a case that customer's needs are not the most abstract initial goals. For instance, when a concrete goal such as "For customers having E-mail accounts" is given as customer's needs, it might not be suitable as an initial goal to be decomposed. On the other hand, more general or more abstract goals rather than customer's needs such as organization constraints and objectives, physical and social law, common sense, standards and so on can be initial goals, and these goals are sometimes missing. To avoid this problem, before setting up initial goals, we should identify wide range of the stakeholders such as lawyers and organizational managers as well as domain experts and they may have to participate in goal decomposition processes. Identifying stakeholders and their involvement in AGORA process is one of the future works.

The metrics that we proposed in AGORA is product metrics in the sense that we calculate them just on the final AGORA graph, neither considering the parts of the graph that have been abandoned, nor measuring the process how and in what order it was constructed. There are some of the quality factors that are difficult to be measured in our approach, and process metrics[7] is one of the promising approaches not only to make our product metrics more precise but also to enlarge the range of the measurable quality factors. For example, whether the order of goal decomposition is in top-down direction or not may be considered as the quality factor of stability. Developing this kind of process metrics is also one of the future works.

One of the most serious problems of AGORA is the management of the goal graph's complexity. For example, Figure5 seems to be beginning to become a bit unmanageable. When we use normal goal-oriented method, we sometimes decompose goals uselessly because we have no established ways to select goals to be decomposed. We can select such goals by means of contribution and preference values in AGORA. However, we still have to spend much time and effort for attaching each of these values. Therefore, we have to develop ways to narrow down the candidates of edges and goals to which such values should be attached, and ways to attach each value efficiently.

5 Conclusion

In this paper, we have proposed AGORA, a method for requirements elicitation which is an extended version of a goal-oriented method by attaching preference and contribution attributes to an AND-OR graph. Furthermore, we have proposed a method to estimate the quality of a requirements specification by means of the structural characteristics and attribute values of an AND-OR graph. In the first section, we pointed out the activities that are not supported in current goal-oriented methods. Here we summarize how AGORA contributes to support these activities.

1. Selecting the goals to be decomposed: In AGORA, We decompose only goals each of which has a relatively large contribution value to its parent, and avoid useless decomposition. Diagonal values in a preference matrix also tell us which goals may be decomposed. If diagonal values in a preference matrix are relatively large, we may select a goal having the matrix to be decomposed. Especially a diagonal value attached by a customer is important.

The variance of a preference matrix denotes that each stakeholder interprets a goal having the matrix differently, and an ambiguity of the goal causes such differences. Therefore, a goal with large variance should be selected to be decomposed.

2. Prioritizing and solving the conflict: As mentioned in sub-section2.5, there are two types of conflicts; one is among the goals and another is among the stakeholders on a goal. In the same way for selecting the goals to be decomposed, we can prioritize the goals and stakeholders by using contribution and preference values, and we can detect the conflicts during the prioritization in both cases. The way for solving the conflicts remains to future works. DDP[6] and Cost Analysis[13] are candidates of it.

3. Choosing and adopting a goal as a requirements specification: This activity is also supported in the same way of selecting the goals to be decomposed. In addition, We stop decomposing a goal when its content is operational and make it to be a part of requirements specification.

Since each final operational goal can be regarded as a use case, methods based on Scenario Analysis[16] and Use Case Modeling might be suitable for the next step of AGORA. The details of the next suitable process of AGORA still remain in the future works. The discussion about AGORA and the next process will lead to a precise measurement method for traceability.

4. Analyzing the impacts when requirements change: We can not examine the impact analysis in AGORA so much. At least, when a requirement is changed, we may find a correspond node and reconsider nodes connected to it with a large contribution value.

5. Improving the quality of the requirements based on a quality of AGORA graph: In AGORA, quality metrics of requirements specification to be described are derived from the structure of a goal graph, contribution values and preference values. We will explore how to improve the requirements quality by the metrics in the future.

Acknowledgments

The authors would like to thank Prof. Kenji Kaijiri at Shinshu University for discussions and insightful comments about the drafts of this paper.

References

[1] IEEE Recommended Practice for Software Requirements Specifications, 1998. IEEE Std. 830-1998.

[2] 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.

[3] B. Boehm, P. Bose, E. Horowitz, and M. Lee. Software Requirements Negotiation Aids: A Theory-W Based Spiral Approach. In *Proc. of 17th International Conference on Software Engineering*, pages 243–253, 1995.

[4] J. P. Cavano and J. A. McCall. A Framework for the Measurement of Software Quality. In *Proc. of ACM Software Quality Assurance Workshop*, pages 133–139, 1978.

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

[6] S. L. Cornford, M. S. Feather, J. C. Kelly, T. W. Larson, B. Sigal, and J. D. Kiper. Design and Development Assessment. In *Proceedings of the Tenth International Workshop on Software Specification and Design (IWSSD'00)*, 2000.

[7] R. J. Costello and D.-B. Liu. Metrics for Requirements Engineering. *System Software*, 29:39–63, 1995.

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

[9] A. Davis. *Software Requirements*. Prentice-Hall, 1990.

[10] GRL homepage. `http://www.cs.toronto.edu/km/GRL/`.

[11] I* homepage. `http://www.cs.toronto.edu/km/istar/`.

[12] KAOS homepage. `http://www.info.ucl.ac.be/research/projects/AVL/ReqEng.html`.

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

[14] F. Moisiadis. A Framework for Prioritizing Use Cases. In *Proceedings of the Fifth Doctoral Consortium at the Conference of Advanced Information Systems Engineering*, Jun. 1998.

[15] J. Mylopoulos, L. Chung, and E. Yu. From Object-Oriented to Goal-Oriented Requirements Analysis. *Communications of the ACM*, 42(1):31–37, Jan. 1999.

[16] C. Rolland, G. Grosz, and R. Kla. Experience with Goal-Scenario Coupling in Requirements Engineering. In *Proc. of RE'99*, pages 7–11, Jun. 1999.

[17] 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.