Visualizing Stakeholder Concerns with Anchored Map

Takanori Ugai

Fujitsu Laboratories Ltd.,
4-1-1 Kamikodanaka Nakaharaku Kawasaki 211-8588 Japan
ugai@jp.fujitsu.com

Abstract. Software development is a cooperative work by stakeholders. It is important for project managers and analysts to understand stakeholder concerns and to identify potential problems such as imbalance of stakeholders or lack of stakeholders. This paper presents a tool which visualizes the strength of stakeholders' interest of concern on two dimensional screens. The proposed tool generates an anchored map from an attributed goal graph by AGORA, which is an extended version of goal-oriented analysis methods. It has information on stakeholders' interest to concerns and its degree as the attributes of goals. Results from the case study are that (1) some concerns are not connected to any stakeholders and (2) a type of stakeholders is interested in different concerns each other. The results suggest that lack of stakeholders for the unconnected concerns and need that a type of stakeholders had better to unify their requirements. And a preliminary evaluation suggests that the tool enables users to identify imbalance of stakeholders or lack of stakeholders faster and more correctly than a matrix of stakeholders and concerns.

Keywords: Stakeholder management, Anchored map, visualization.

1 Introduction

Software development is a cooperative work by stakeholders. Because software is developed to be successful, it is extremely important to accurately understand user requirements. It is essential that the mutual agreement be handled efficiently and appropriately. However, if the stakeholders are missing, high-quality requirement specifications cannot be made, because knowledge is stakeholder-dependent. The case of the London ambulance service [7] is a typical case of missing stakeholders. It is important for project managers and analysts to understand stakeholder concerns and to identify potential problems such as imbalance of stakeholders or lack of stakeholders. Sharp at el [19] propose a technique to discovering all relevant stakeholders of a specific system.

This paper presents a tool which visualizes the relationships between stakeholders and their interest of concerns using an anchored map [14, 20] to detect the problems.

Anchored map is a technique to draw a bipartite graph whose nodes have two types of attributes. Anchored map fixes the nodes with one of the attributes on a concentric circle as anchor nodes and locates other nodes with a spring layout [6] as free nodes. The proposed tool generates an anchored map from an attributed goal graph by AGORA

M.J. Smith, G. Salvendy (Eds.): Human Interface, Part I, HCII 2011, LNCS 6771, pp. 268–277, 2011.

(Attributed Goal-Oriented Requirements Analysis) [10, 11]. AGORA is a family of goal-oriented requirements analysis [12] and Goal-oriented requirement analysis (GORA) is supposed to be one of the most promising methods for supporting the elicitation of requirements. GORA methods such as i* [3] and KAOS [5] are top-down methods for decomposing and refining customer needs into concrete goals to be achieved in order to satisfy customer needs. As a result, an AND/OR graph is generated whose nodes represent identified goals. There are a number of case studies and assessments of this method [2].

An AGORA goal graph is attributed AND-OR goal graph. A semantic tag is one of the attributes and abstracts of the meaning of the goal, such as a stereotype of UML. Stakeholder concerns are represented by semantic tags on the goals.

This tool will enable to support the Sharp's technique if the model of the technique is applied instead of the semantic tags of AGORA goals.

This paper is organized as follows. At first, AGORA and anchored map is described as basic technologies. And the way how to generate an anchored map from a goal graph is described. Then an integrated implementation of anchored map and AGORA editor is shown with a case study and a preliminary evaluation. At last, after discussion of related works, summery and future works are described.

2 Preliminaries

2.1 AGORA

AGORA [10, 11] is a part of the GORA family of analysis methods. To make up for the support functions that the existing GORA method does not have, AGORA uses an extended version of AND-OR goal graph called *attributed goal graph*. In AGORA, requirements analysts and stakeholders can attach *attributes* to nodes and edges of a graph. Fig.1 shows an excerpt example of the attributed goal graph. And Fig.2 shows

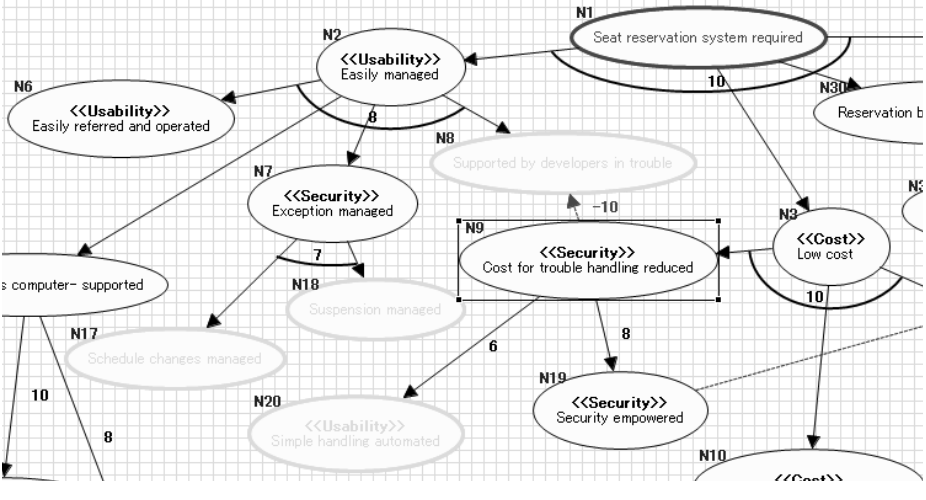
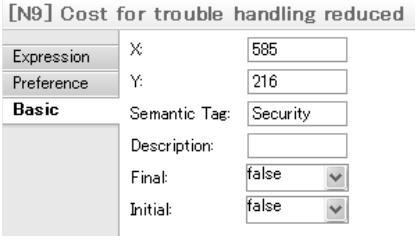


Fig. 1. An Attributed goal graph (excerpt)

snapshots of AGORA tool, including In these examples, the requirements of a seat reservation system for trains are analyzed. Although AGORA can deal with many types of attributes, we only use and explain the following two types of attributes in this paper:

- *Semantic tags*: Abstractions of the meaning of the goals, like stereotypes of UML. In the example above, the facilitator puts a semantic tag ``Security'' to a goal N_9 "Cost for trouble handling reduced" on a sub-window shown in Fig.2(a), representing that N_9 is related to security. We have extracted the possible candidates of semantic tags from, e.g., quality attributes enumerated in IEC/ISO 9126-1 [9] or NFR catalogs [4]. Requirements analysts can use additional semantic tags, e.g., domain-specific ones, if necessary.
- *Preference matrix*: Attached to a node, i.e. a goal, and represents the degree of preference or satisfaction of the goal for each stakeholder. Each preference is presented as a number from -10 to 10, and the higher value denotes that the goal is more preferable for a stakeholder. When the number is zero, he/she is not interested in the goal. Fig.2(b) is an example of the preference matrix, attached with the goal N_9: "Cost for trouble handling reduced". The columns, each of which is for the user (User), the administrator (Administrator), and the developer (Developer) are respectively to grade their own preference and preferences of other stakeholders. In this example, the values on the diagonal are -5, 9, and 0, and the analyst can understand that the user and the administrator have a conflict of interest at the goal N_{20}. The large variance of these preference values in a goal can suggest the conflicts of stakeholders' interests and the discordance of its understanding [11]. In this paper, we only use diagonal elements of the matrix, i.e., stakeholders' preferences by themselves.



Evaluator¥-ee	User	Administrator	Developer
User	-5	8	0
Administrator	8	9	0
Developer	8	8	

Fig. 2. Screenshots of AGORA tool: Left(a): Putting semantic tags for N_9, Right(b): Preference matrix for N_9

2.2 Anchored Map

This section presents the method to calculate the position of nodes in an anchored map for a bipartite graph. A bipartite graph consists of two sets of nodes and there are edges between nodes of different sets but there is no edge between nodes within a set. Anchored map is a layout technique for a bipartite graph.

Suppose set A consists of 9 nodes representing stakeholders which are User1, .., User3, Admin1, .., Admin3 and Dev1, .., Dev3 and set B consists of 8 nodes

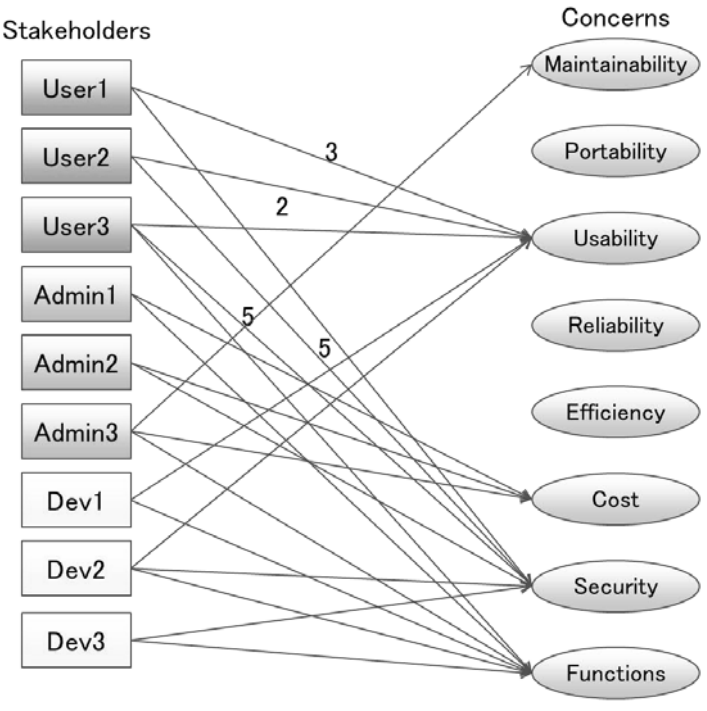


Fig. 3. Two-way layout for a biparate graph

representing concerns which are Maintainability, Portability, Usability, Reliability, Efficiency, Cost, Security and Functionality. Fig.3 shows a two-layer graphical representation of the same bipartite graph. Fig.4 shows the same bipartite graph shown in Fig.3 as an anchored map. On an anchored map of the bipartite graph, the nodes in A, called "anchor nodes", are arranged on the circumference and the nodes in B, called "free nodes", are arranged at suitable positions in relations to the adjacent anchor nodes. The strength of edge as a spring is in proportional to the value attached the edge. This means an anchor node and a free node are located closer to each other if the stakeholder is strongly interested in the concern. The distinction of the nodes is clearer than in Fig.3, and it is easy to see the relationships between free nodes.

In the case that supposing the elements in B as anchor nodes and the elements in A as free nodes, a anchored map is generated with the same way. This means two type of anchored maps are generated from a bipartite graph.

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm, if possible.

3 Generating Anchored Map from a Goal Graph

Selected concerns are based on ISO/IEC9126-1 [9]. ISO/IEC9126-1 represents the latest research into characterizing software for the purposes of software quality

control, software quality assurance and software process improvement. The ISO 9126-1 software quality model identifies 6 main quality characteristics, namely: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability.

Cost is not one of the software quality factors in ISO/IEC9126-1 but is identified on many goals.

All goals are aggregated by the concerns represented by the semantics tags. The strength of the edge between a stakeholder and a concern is the average of the absolute value of the stakeholder's preferences of the aggregated goals by the concern. This means a stakeholder and a concern are located closer to each other if the stakeholder is strongly interested in the concern.

Because of this data representation, this tool enables the users to notice lacks of stakeholders to concerns. Anchor nodes are located on the circumference at equal intervals, and the positions of anchor nodes are decided as the number of crosses of edges is reduced as far as possible. Using this positioning algorithm, the free nodes that have a similar relationship with anchor nodes are located closer to each other. Because of this positioning algorithm, this tool enables the users to notice imbalance of stakeholders.

4 Implementation and Preliminary Experiments

4.1 Implementation

The tool is implemented as a plug-in of eclipse The input is produced through AGORA tool, and the tool extracts goals with attached preference matrix and semantic tags. The layout of an anchored map is implemented based on the algorithm shown in [14].

The tool is implemented as an integrated tool of anchored map and AGORA but the idea of visualizing relationships between stakeholders and their concerns does not depend on AGORA. User Requirements Notation (URN) [1] is a possible notation. Relationships between a URL model and actors can be an input of the tool.

4.2 Preliminary Experiments: A Case Study

Nine stakeholders put their preferences and the tags. These stakeholders consist of three end-users (User1, User2, User3), three developers (Dev1, Dev2, Dev3) and three system administrators (Admin1, Admin2, Admin3). In Fig.4, squares represent stakeholders and small circles represent concerns. Squares are color-coded according to stakeholder roles such as end-user, developer and administrator.

In Fig.5, stakeholders are located on a circle as anchors and concerns are arranged as free nodes. From the figure, we can see that,

- Most of stakeholders are strongly interested in security and functions because they are located in the center of the circle.
- Developers, Admin1 and Admin3 are interested in cost but users are not.
- Users and Admin2 are interested in usability but developers are not.
- Only Admin2 is interested in maintainability.

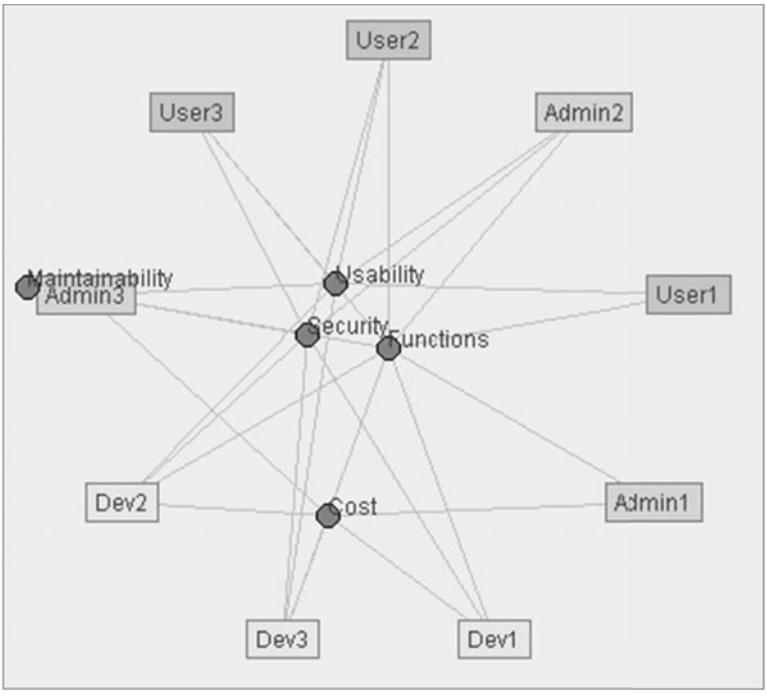


Fig. 4. Example of Anchored Map (Anchor nodes: stakeholders, Free nodes: concerns)

In Fig.5, concerns are located on a circle as anchors and stakeholders are arranged as free nodes. From the figure, we can see that

- All users are interested in similar concerns, and Admin2 is also the same as the users.
- Dev2 and Dev3 are interested in similar concerns but Dev1 is slightly different from Dev2 and Dev3.
- All administrators are interested in different concerns.
- Nobody is interested in portability, reliability and efficiency. This is because the goal graph describes business domains and does not describe the system infrastructure.

The findings suggest that (1) stakeholders who know much about the system infrastructure should be involved in the project and (2) differences of administrators should be consolidated.

The case study shows the system visualizes imbalance of stakeholders, lack of stakeholders, and some other potential useful information to manage stakeholders.

4.3 Preliminary Evaluation

We conducted experimentation whether the tool enables users to identify imbalance of stakeholders or lack of stakeholders faster and more correctly than an existing technique.

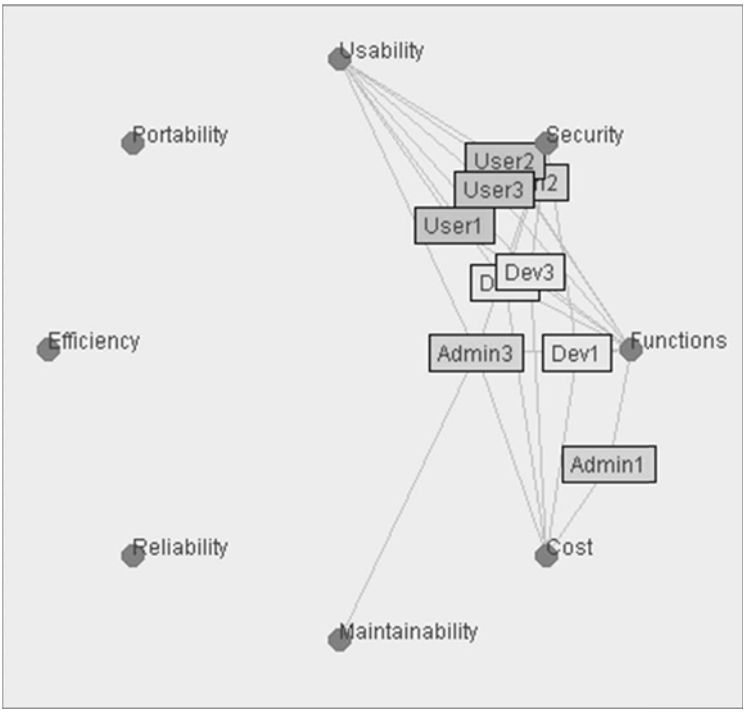


Fig. 5. Example of Anchored Map (Anchor nodes: concerns, Free nodes: stakeholders)

Table 1. Goal Graphs

System	RSS Reader	Reception
# of nodes	29	36
# of concerns	4	5
# of stakeholders	**9**	9

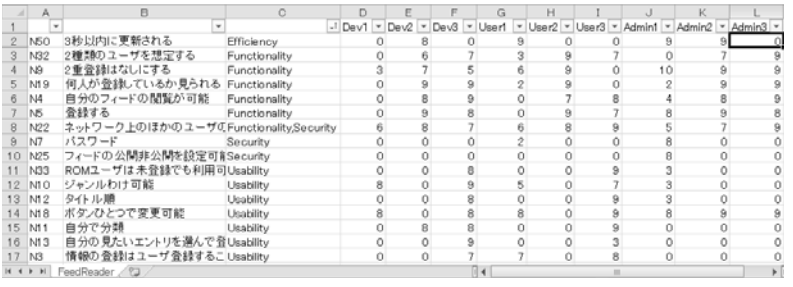
Preparation

- *Goal graph*: for a system of RSS feed reader and a system of reception supporting system for a hospital. The sizes of graphs are shown in Table1. RSS Reader has tags labeled Efficiency, Functionality, Security and Usability, and Reception system has tags labeled Functionality, Maintainability, Reliability, Security and Usability.
- *Target for comparison:* matrix of stakeholders and concerns. Fig.6 is a matrix for the RSS Reader. They are provided as Excel sheets. The columns are ID for goal, goal description, semantic tags, and preferences of stakeholders.
- *Task:* Subjects are asked to answer 17 questions. They can spend two minutes for each and twenty five minutes in total. Twelve questions are expected to be answered in the worst case. Questions are such as follows:
  - Which concerns is nobody interested in?
  - Which concerns is only one stakeholder interested in?

- Which concerns is everybody interested in?
- Which concerns are all developers interested in?

The questions are designed to lead the findings shown in section 4.2.

- *Subjects:* two students, one researcher, and one software engineer. Each subject answers for two goal graphs. The subject does the task with the tool for one graph and with Excel sheet for another graph. They are assigned tasks as Table2.



	A	B	C	D	E	F	G	H	I	J	K	L
1				Dev1	Dev2	Dev3	User1	User2	User3	Admin1	Admin2	Admin3
2	N50	3秒以内に更新される	Efficiency	0	8	0	9	0	0	9	9	0
3	N32	2種類のユーザを想定する	Functionality	0	6	7	3	9	7	0	7	9
4	N9	2重登録はなしにする	Functionality	3	7	5	6	9	0	10	9	9
5	N19	何人が登録しているか見られる	Functionality	0	9	9	2	9	0	2	9	9
6	N4	自分のフィードの閲覧が可能	Functionality	0	8	9	0	7	8	4	8	9
7	N5	登録する	Functionality	0	9	8	0	9	7	8	9	8
8	N22	ネットワーク上のほかのユーザの	Functionality,Security	6	8	7	6	8	9	5	7	9
9	N7	パスワード	Security	0	0	0	2	0	0	8	0	0
10	N25	フィードの公開非公開を設定可	Security	0	0	0	0	0	0	8	0	0
11	N33	ROMユーザは未登録でも利用可	Usability	0	0	8	0	0	9	3	0	0
12	N10	ジャンルわけ可能	Usability	8	0	9	5	0	7	3	0	0
13	N12	タイトル順	Usability	0	0	8	0	0	9	3	0	0
14	N18	ボタンひとつで変更可能	Usability	8	0	8	8	0	9	8	9	9
15	N11	自分で分類	Usability	0	8	8	0	0	9	0	0	0
16	N13	自分の見たいエントリを選んで登	Usability	0	0	9	0	0	3	0	0	0
17	N3	情報の登録はユーザ登録するこ	Usability	0	0	7	7	0	8	0	0	0

FeedReader

Fig. 6. Matrix of stakeholders and concerns

Results. The fourth column in Table2 shows the number of answered questions, the fifth column shows the number of correct answers and the last column shows the time period to finish the task.

Table 2. Task assignment and Results

Subject	Graph	Tool or Table	Completed	Correct	Time
Student1	FeedReader	Tool	17	17	7m10s
	Reception	Table	17	17	8m36s
Student2	FeedReader	Table	17	15	13m37s
	Reception	Tool	17	17	5m58s
Researcher	FeedReader	Tool	17	17	6m19s
	Reception	Table	17	17	17m53s
Engineer	FeedReader	Table	17	17	13m11s
	Reception	Tool	17	17	10m35s

One subject got two wrong answers with table and the others got correct answers for all questions. All subjects have completed the task in short period with the tool than with table. The total time for FeedReader with tool is 13 minutes and 29 seconds and the total time with table is 26m47s. The total time for Reception with tool is 16m33s and the total time with table is 26m29s.

The number of subjects is very small, but the result suggest the tool enables users to identify imbalance of stakeholders or lack of stakeholders faster and more correctly than a matrix of stakeholders and concerns.

5 Related Works

Schneider [18] proposes a technique used to visualize informal communications in a project. Heim [8] proposes a technique how the relationships between requirements are visualized using graph structures. Rohleder [17] proposes a graphical representation on non-functional requirements impact by eight composite rules and seven types of links.

Stakeholder management has a long history [16] but there is no graph representation of relationships between stakeholders and their concerns. Pouloudi and Whitley [15] suggest four principles of stakeholder identification, and describe and approach which is based on these principles, and which they used to identify stakeholders in the drug use management domain. Lyytinen and Hirschheim [13] also provides some guidance for stakeholder identifications for IS, while acknowledge in that the identification of the set of all stakeholders is far from trivial. Sharp at el [19] propose a technique to discovering all relevant stakeholders of a specific system. Our proposing tool will be able to support the Sharp's technique if the model of the technique is applied instead of the semantic tags of AGORA goals. However the Sharp's technique must be done separately in requirements analysis. Our tool is combined with goal graph editor and missing stakeholders will be identified in the middle of requirements analysis.

6 Summary and Future Works

This study presented a technique for visualizing stakeholders concerns in a project using an anchored map. The tool produces a graph which represents the relationship between stakeholder and system concerns. The tool can switch anchors and free nodes, and stakeholder preferences for goals are aggregated with semantic tags. Requirements analysts can easily obtain an overview of the analysis, to find which concern has high priority for end users or system administrators, or to identify the concerns which the smallest number of stakeholders are interested.

References

1. Amyot, D.: Introduction to the user requirements notation: Learning by example. Computer Networks 42(3), 285–301 (2003)
2. Anton, A.I., Potts, C.: The use of goals to surface requirements for evolving systems. In: Proc. International Conference on Software Engineering, pp. 157–166 (1998)
3. Cares, C., Franch, X.: Toward a framework for improving goal-oriented requirement model quality. In: Workshop on Requirements Engineering (2009)
4. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Springer, Heidelberg (1999)
5. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. In: Science of Computer Programming, pp. 3–50 (1993)
6. Eades, P.: A heuristic for graph drawing. Congressus Numerantium 42, 149–160 (1984)
7. Finkelstein, A., Dowell, J.: A comedy of errors: the london ambulance service case study. In: Proc. 8th International Workshop on Software Specificaiton and Design, pp. 2–4 (1996)

8. Heim, P., Lohmann, S., Lauenroth, K., Ziegler, J.: Graph-based visualization of requirements relationships. In: Proc. Requirements Engineering Visualization, pp. 51–55. IEEE Computer Society, Los Alamitos (2008)
9. ISO/IEC: ISO/IEC 9126-1:2001. Software engineering – Product quality – Part1:Quality model. ISO/IEC (2001)
10. Kaiya, H., Horai, H., Saeki, M.: AGORA: Attributed goal-oriented requirements analysis method. In: Proc. Requirements Engineering Conference, pp. 13–22 (2002)
11. Kaiya, H., Shinbara, D., Kawano, J., Saeki, M.: Improving the detection of requirements discordances among stakeholders. Requirements Engineering 10(4), 289–303 (2005)
12. van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: Proc. Requirements Engineering Conference, pp. 249–263 (2001)
13. Lyytinen, K., Hirschheim, R.: Information systems failures–a survey and classification of the empirical literature, pp. 257–309. Oxford University Press, Inc., New York (1987)
14. Misue, K.: Anchored maps: Visualization techniques for drawing bipartite graphs. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4551, pp. 106–114. Springer, Heidelberg (2007)
15. Pouloudi, A., Whitley, E.A.: Stakeholder identification in inter-organizational systems: gaining insights for drug use management systems. European Journal of Information Systems, 1–14 (March 1997)
16. Preble, J.F.: Toward a comprehensive model of stakeholder management. Business and Society Review 110(4), 407–431 (2005)
17. Rohleder, C.: Visualizing the impact of non-functional requirements on variants: A case study. In: Proc. Requirements Engineering Visualization, pp. 11–20 (2008)
18. Schneider, K., Stapel, K., Knauss, E.: Beyond documents: Visualizing informal communication. In: Proc. Requirements Engineering Visualization, pp. 31–40. IEEE Computer Society, Los Alamitos (2008)
19. Sharp, H., Finkelstein, A., Galal, G.: Stakeholder identification in the requirements engineering process. In: Proc. Tenth Int. Database and Expert Systems Applications Workshop, pp. 387–391 (1999)
20. Ugai, T., Aoyama, K.: Organization diagnosis tools based on social network analysis. In: Smith, M.J., Salvendy, G. (eds.) HCI International 2009. LNCS, vol. 5617, pp. 181–189. Springer, Heidelberg (2009)