

推薦論文

要求獲得におけるステークホルダの偏りと不足を検出する 可視化ツール

鵜飼 孝典^{1,2,a)} 林 晋平² 佐伯 元司²

受付日 2011年9月12日, 採録日 2012年1月13日

概要: ソフトウェアの要求獲得は、ステークホルダによる協調作業である。プロジェクトマネージャや分析者にとって、ステークホルダの関心事を理解し、ステークホルダの偏りや不足などの潜在的な問題を知っておくことは重要である。本稿では、ゴール指向分析手法の1つである AGORA の成果物である属性つきゴールグラフを対象に、要求分析作業中にステークホルダと、ステークホルダのシステム品質に関する関心事の関係をアンカーマップを使って可視化した。また、この手法を利用して、システムの信頼性、効率性、使用性などのシステム品質に関する重要な関心事をもれなく獲得するためにステークホルダの不足や偏りを発見することを支援するツールを開発した。このツールは、属性つきゴールグラフから自動的に、ステークホルダと品質に関する関心事の関係を抽出し、可視化する。さらに、実装したツールを用いた評価実験により、ステークホルダの偏りや不足を同定するのに、既存によく使われているステークホルダと要求の対応表よりも短時間で、正しくできることを示した。

キーワード: 要求獲得, ゴール指向分析, ゴールグラフ, ステークホルダ, アンカーマップ

A Supporting Tool to Identify Stakeholders' Imbalance and Lack in Requirements Analysis

TAKANORI UGAI^{1,2,a)} SHINPEI HAYASHI² MOTOSHI SAEKI²

Received: September 12, 2011, Accepted: January 13, 2012

Abstract: Software requirements elicitation 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 or lack of stakeholders. This paper presents a technique and a tool which visualize the strength of stakeholders' interest of concerns on two dimensional screens. The tool generates anchored maps from an attributed goal graph by AGORA, which is an extended version of goal-oriented analysis methods. It has stakeholders' interest to concerns and its degree as the attributes of goals. Additionally an experimental evaluation is described, whose results show the user of the tool could identify imbalance and lack of stakeholders more accurately in shorter time than the case with a table of stakeholders and requirements.

Keywords: requirements elicitation, goal-oriented analysis, goal graph, stakeholder, anchored map

¹ 富士通研究所
Fujitsu Laboratories Ltd., Kawasaki, Kanagawa 211-8588,
Japan

² 東京工業大学
Tokyo Institute of Technology, Meguro, Tokyo 152-8552,
Japan

^{a)} ugai@jp.fujitsu.com
本稿の内容は 2011 年 3 月のグループウェアとネットワークサー
ビス研究会にて報告され、同研究会主査により情報処理学会論文
誌ジャーナルへの掲載が推薦された論文である。

1. はじめに

ソフトウェアの要求獲得では、最初にステークホルダの同定を行い、ステークホルダに対するインタビューや要求獲得会議を行い、要求仕様書の作成を行う。ソフトウェアの要求獲得は、ステークホルダによる協調作業であるため、できるだけ早い時期に適切なステークホルダを発見するこ

とが重要である [1]。しかしながら、最初にすべての必要なステークホルダの同定を行うことは難しく、要求仕様書を作成する段階やレビューの段階でステークホルダの不足が原因でシステムのある側面（たとえばセキュリティ）に関する要求獲得が不十分になることがある。また、集められたステークホルダの興味が偏っていることが原因で、ある側面（たとえば性能）に関して詳細な要求がとれているが、別の重要な側面（たとえば使いやすさ）に関する十分な要求が獲得できていないということがある。開発するソフトウェアに関わるステークホルダの不足や偏りがインタビューの途中段階や要求獲得会議の段階で容易に分かるようになれば、少ない手戻りで対応することができ、最終的にできあがるソフトウェアが高品質であることも期待できる。

文献 [2] によれば、ステークホルダとは“プロジェクトに積極的に関与しているか、またはプロジェクトの実施あるいはプロジェクトの完了の結果、自らの利益にプラスまたはマイナスの影響を受ける個人や組織”と定義されている。開発しようとしているソフトウェアに関わるすべての人がステークホルダとなりうるが、それぞれ、エンドユーザ、システム管理者、経営者、ソフトウェアの開発者などの役割を持っている。ソフトウェア開発においては、適切な人数のステークホルダを集め、集められたステークホルダの合意によって開発が行われる。ソフトウェア開発を成功させるためには、ユーザの要求を正確に理解することが不可欠で、そのためには情報提供者である必要なステークホルダを欠けることなく同定することが重要である。Sharp らは、システムに関わるすべてのステークホルダを発見する技術を提案している [1] が、この技術は、開発に入る前に行うことを想定しており、それに基づいてステークホルダを集めたとしても、実際に集められたステークホルダで十分であったかどうか知ることは難しい。

我々はこの課題に対し、現在人手で行っている情報抽出をできる限り自動化し、抽出した情報を可視化することによって、ステークホルダの不足と偏りをこれまでの対応表と人手によるレビューよりも直感的に短時間でしかも要求分析作業中に行えるようにすることを目的とする。

2次元平面上に情報を配置し、その位置関係から2種類の情報の対応関係を、定性的かつ直感的に認識できるようにするために開発されたレイアウトアルゴリズムとしてアンカーマップ [3] がある。我々は、このアンカーマップを用いて、要求分析作業中にステークホルダとステークホルダの品質に関する関心事の関係を可視化する手法を開発した。ここでは、ステークホルダの不足を発見するということは、関心を持つステークホルダがいない関心事を同定することであり、特定の関心事に対して関心を持つステークホルダが不足していることを意味する。ここでステークホルダは、個人の場合もあるし、ある役割に属するステークホルダ全体の場合もある。

また、我々は、ゴール指向分析手法の1つである AGORA (Attributed Goal-Oriented Requirements Analysis Method: 属性つきゴール指向要求分析法) [4], [5] を対象として、提案手法を実現するツール SViewer を開発した。このツールは、属性つきゴールグラフから自動的に、ステークホルダと品質に関する関心事の関係を抽出し、可視化する。さらに、実装したツールを用いた評価実験により、ステークホルダの偏りや不足を同定するのに、すでによく使われているステークホルダと要求の対応表よりも短時間で、正しくできた。

なお、属性つきゴールグラフをいかに作るか、特に属性値をどのように振るかは AGORA 手法自身の問題であるので、本稿の範囲外であり、言及しない。我々の手法は、AGORA 手法で得られた属性つきゴールグラフを活用する手法である。

以下本稿では、2章で AGORA とアンカーマップ、ステークホルダの関心事の概要を述べた後、3章でゴールグラフからアンカーマップを作成する方法を述べる。さらに4章で実装、5章で提案ツールの評価実験について述べる。最後に6章で他の研究との関連を述べ、7章でまとめと今後の課題を述べる。

2. 準備

2.1 AGORA

ゴール指向要求分析法 [6] では、顧客やユーザの要求を達成すべきゴールととらえ、それを AND-OR 分解し、より具体的なサブゴールの集合へと詳細化していく。分析結果は、ゴールグラフで表現される。AGORA [7] では、ゴールグラフに属性や属性値をつけることができ、それを属性つきゴールグラフと呼ぶ。属性、属性値として、ゴール達成へサブゴールがどれだけ貢献するかの度合いを示す貢献度、UML のステレオタイプのようなゴールの内容の意味的なカテゴリを示すセマンティックタグ、各ステークホルダがそのゴールが達成されるのがどれくらい好ましいかの度合いを示す満足度行列などを扱うことができる。さらに、たとえば、ゴールを達成するためのコストなど、ゴール特有の属性とその計算式を定義できる。本研究では、それらのうち、セマンティックタグと満足度行列の2つを用いる。

図 1 は、属性つきゴールグラフの例である。この例では、ある小規模な病院の受付で行われている外来受付、保険点数計算、診察料の支払いなどの業務の効率化と省力化を目的として、業務をシステム化するために病院受付業務の要求を分析している。楕円がゴール、矢印がゴール間の貢献関係を示す。たとえば、N12 の“問診表データの管理”というゴールは、N8 の“問診表を書く”というゴールのサブゴールである。緑のゴールは、ステークホルダ間で採用するという合意がとられた最終ゴールを示す。また、N29 と N30 は、N10 に対して OR 関係であり、N29 または N30

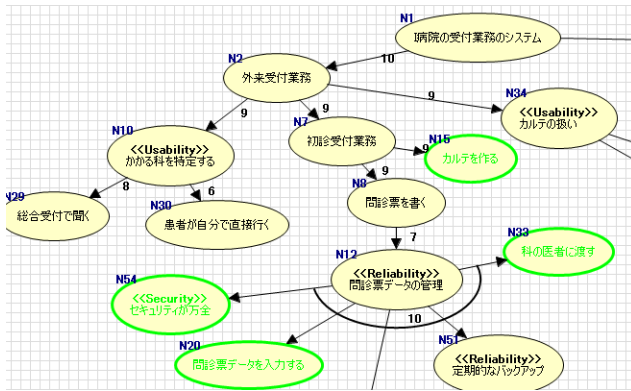


図 1 属性つきゴールグラフの例

Fig. 1 Example of an attributed goal graph.

表 1 ゴール N54 の満足度行列

Table 1 Satisfaction matrix for Goal N54.

	ユーザ 1	管理者 1	開発者 1
ユーザ 1	-5	8	0
管理者 1	8	9	0
開発者 1	8	8	0

のどちらかが達成されると N10 が達成されることを示す。N54, N20, N51, N33, (画面外の) N52 は N12 に対して AND 関係であり, N12 を達成するには, N54, N20, N51, N33, N52 のすべてのゴールが達成されなければならないことを示す。さらにこの例では, N54 の“セキュリティが万全”というゴールに対し, N54 がセキュリティに関するものであることを示すように <<Security>> というセマンティックタグをつけている。

表 1 は, N54 のゴールに付与された満足度行列であり, N54 に対するステークホルダ全員の値をまとめた結果を表している。満足度行列には, 各ステークホルダにとってそのゴールが達成されるのがどれくらい好ましいか, 他のステークホルダにとってどれくらい好ましいと思われるかの度合いを 10 点満点で入力する。逆にそのゴールが達成されることが好ましくない場合, その不満の度合いを 10 点満点で評価し負の値を入力する。またそのゴールの達成に無関心の場合は, 0 を入力し, 満足度行列には, -10~10 の値が入力される。

表 1 のそれぞれの行, ユーザ 1, 管理者 1, 開発者 1 はそれぞれのステークホルダがつけた点数を示す。満足度行列では, 縦軸が評価者, 横軸が予想する対象となるステークホルダを示す。たとえば, 表 1 の 1 行目は, N54 のゴールに対するユーザ 1 の入力を示し, 1 行 1 列目は, 入力者自身にとっては, このゴールが達成されることは望ましくないことで, その度合いは, 10 点満点中 -5 点であることを示す。1 行 2 列目は, ユーザ 1 がこのゴールの達成に対する管理者 1 の満足度を予想した値で, このゴールの達成は, 管理者 1 にとって望ましいものであり, その度合いは 10 点満点中 8 点であると予想したことを示す。1 行 3 列目

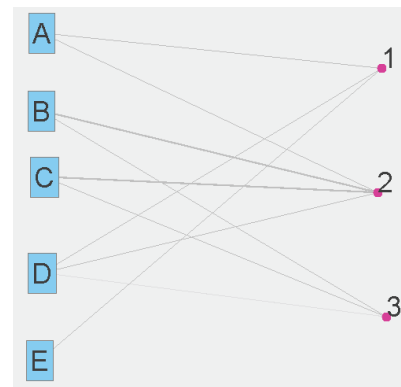


図 2 2部グラフの2階層レイアウト例

Fig. 2 Two-layered layout for a bipartite graph.

は, 同様にユーザ 1 がこのゴールの達成に対する開発者 1 の満足度を予想したものを示し, 開発者にとっては関心のないものであろうと予想して 0 点をつけたことを示す。同様に他の 2 人のステークホルダも採点を行う。このように作られた満足度行列では, 対角要素が自分自身の関心の度合いを示す。本稿では, 自分にとっての満足度である対角要素のみを扱う。

好む, 無関心, 好まないの 3 点法などのような簡単な点数づけにすることも考えられるが, 文献 [4] において, “AGORA ではゴール間で満足度に差をつけたいことが多く, その際にスケールが小さいと後から全体に対して値の振り直しを行わなければならない。たとえば, A, B, C の順で満足度をつけたとき, D が B よりは好ましいが, A ほどではないというようなことを想定している。また, 3 点法などには 10 点法からすぐに変換することができることから 10 点法を用いている”と書かれている。AGORA の採点法で今回の手法には不都合がないため, そのまま用いている。

我々の手法は AGORA のゴールグラフを利用し, AGORA の満足度行列をそのまま利用した。ただし, これによって作られたアンカーマップからステークホルダの不足や偏りを発見する目的に限定した場合, 10 点法が最適であるかどうか, スケーリングの差が結果に与える影響を分析, 評価するのは今後の課題とする。

2.2 アンカーマップ

たとえば, “1”, “2”, “3” の 3 つの整数からなる集合と “A”, “B”, “C”, “D” の 4 つのアルファベットからなる 2 種類の集合があったとする。整数の集合の要素とアルファベットの集合の要素間に関係があるとする。整数の集合とアルファベットの集合を 2 種類のノード集合, それぞれの集合の要素間の関係を枝と定義すると, たとえば, 図 2 のようなグラフが定義できる。このように 2 種類のノードから構成されるグラフで, 異なる種類のノードの間に枝が存在し, 同じ種類のノードには枝が存在しないものを 2 部グ

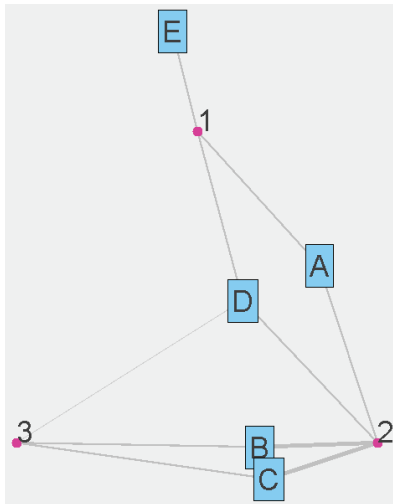


図 3 アンカーマップによるレイアウト例 (1)
Fig. 3 Layout with Anchored Map (1).

ラフと呼び、図 2 のように描画したものを 2 階層レイアウトと呼ぶ。

アンカーマップとは 2 部グラフの 2 種類のノードのうち 1 種類に位置の制約を課して関係構造の把握を助ける描画手法である。アンカーマップは、1 つの観点に制約を課して表示することによって、複数の観点を持つノードの関係の理解を容易にすることができる。制約を課すノードをアンカーノード、それ以外をフリーノードと呼ぶ。

アンカーマップでは、アンカーノードを 1 つの円の円周上に配置する。円周上に配置した以外のノードはフリーノードとして、隣接するアンカーノードに関連して、スプリングアルゴリズム [8] を用いて適切な位置に配置される。スプリングアルゴリズムでは、枝で結ばれたノード間には、距離が離れれば離れるほど、また枝に付与された値が大きければ大きいほど、大きな引力が働き、それ以外のノード間には、距離が近ければ近いほど大きな斥力が働くと考えて、反復的にノードの位置を計算する。アンカーノードとフリーノードを入れ換えて、同じようにアンカーマップを作成することもできる。すなわち、1 つの 2 部グラフから 2 種類のアンカーマップを作成することができる。

図 3 と図 4 は、2 階層レイアウトで描画された図 2 と同じ 2 部グラフを 2 通りのアンカーマップで描画したものである。図 3 では整数が、図 4 ではアルファベットがアンカーノードとして固定されている。たとえば図 3 では“1”だけが“E”と関係があることが分かる。また、“1”と“A”または“D”、“2”と“A”または“D”は、それぞれ点数が 10、“3”と“D”は点数が 2 であり、“A”は“1”と“2”の間点に配置され、“D”は“3”に近づくが、“1”、“2”により近く配置され、結果的に“A”と“D”が近い場所に配置され、近い関係にあることが分かる。“B”と“2”は点数が 6、“B”と“3”は点数が 3 であり、“B”は“2”と“3”から、1 対 2 の距離に配置されている。“C”も“B”とまったく同

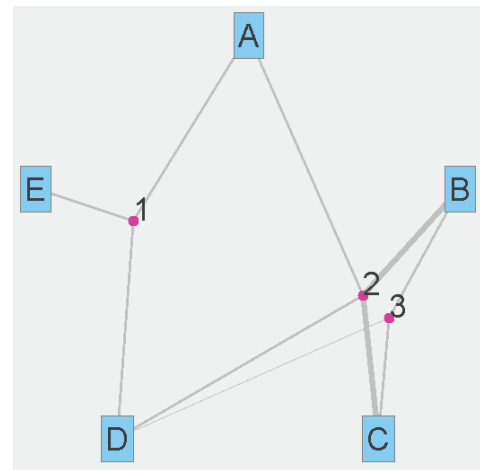


図 4 アンカーマップによるレイアウト例 (2)
Fig. 4 Layout with Anchored Map (2).

じ点数であるが、見やすさのために完全に同じではなく少しずれた位置に配置されている。また、図 4 においては、“2”と“3”が近くに配置され、近い関係にあることを意味する。

また、アンカーノードは同心円上に等距離に配置され、その位置は、1) エッジの交差ができるだけ少なくなる、2) 共通につながっているフリーノードが多いほど近くなるなどの手法が提案されている [3]。

3. ゴールグラフからのアンカーマップの生成

3.1 ゴールグラフにおける関心事とステークホルダの関係

ステークホルダと関心事をそれぞれアンカーマップの 2 種類のノードの集合の要素とし、アンカーマップにおける値の強さは、属性つきゴールグラフの満足度行列から抽出する。ステークホルダを示すノードは、ステークホルダの役割ごとに色分けされる。前述したとおり、AGORA では各ステークホルダがそのゴールが達成されるのがどれくらい好ましいかの度合いを $-10 \sim 10$ で入力する。好ましい場合に正の値、好ましくない場合は負の値、無関心の場合は 0 とする。したがって、正負にかかわらず、絶対値が大きい場合にそのゴールに対して強い関心を示していると考えることができる。そこで、そのゴールに付与されているセマンティックタグによって表現されている関心事に対するステークホルダの関心の強さは、そのステークホルダの満足度の絶対値とする。

すべてのゴールは、関心事ごとに集計される。ステークホルダと関心事の間の枝に付与する値は、関心事ごとに集計されたゴールに付与されたステークホルダの満足度の絶対値の平均とする。たとえば、図 1 に示したゴールグラフには、信頼性のタグがつけられたゴールが 2 つ (N12 と N51) ある。このグラフに対して、ユーザ 1 というステークホルダが、8 と 4 の満足度をつけたとすると、ユーザ 1 と信頼性の間の枝に付与する値は、 $6 (= (8 + 4) / 2)$ とな

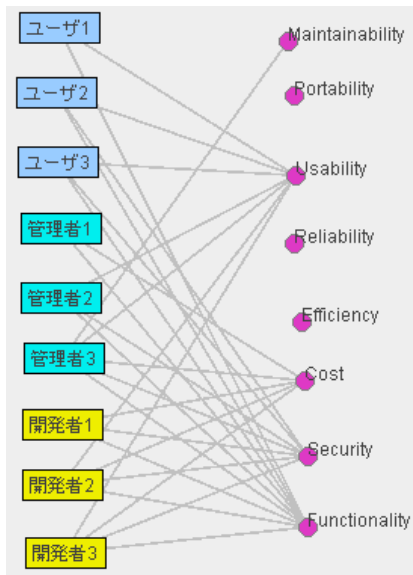


図 5 ゴールグラフの2階層レイアウト例

Fig. 5 Two-layered layout of a goal graph for a bipartate graph.

る。同様にユーザ1と他の関心事、他のステークホルダとそれぞれの関心事の間の値も計算する。

同じステークホルダが、同じタグが振られてはいるが、異なるゴールに大きく異なる点をつけた場合を考える。たとえば、A というステークホルダが信頼性が振られた2つのゴールに各々10と0をつけてしまった場合、アンカーマップの枝の点はこれらの絶対値の平均をとっているの、大きな差があるという情報は消え、他のステークホルダがこれらのゴールに5と5を振った場合と同じになる。このような場合に、差の情報をマップ上に可視化し、ステークホルダの評価の安定性や本当にその関心事に関心があるかどうかを提示する手法も考えられる。しかしながら、本研究では、上記の極端な例を区別することではなく、大局的に見てステークホルダが不足しているかどうかを人間が判断するための情報に人間が使うことが目的である。そのため上記の例を区別し、異なる対策を示唆するということは本稿では対象外とする。

以上のように定義した、ステークホルダと関心事の関係を2部グラフで表現し、図5のように図示した場合、誰も興味を持たない関心事を見つけることや、その関心事に対するステークホルダが不足していることは容易に分かる。しかしながら、ある特定の関心事に、ある役割のステークホルダが誰も興味を持っていないことは容易には分からない。たとえば、この図では、管理者や開発者にはコストに関心を持っているものがあるが、ユーザは誰もコストに興味を持っていない。このことは、開発をこのまま続けるとユーザがコストを考慮することなく要求を出し続け、その結果開発費が予算を大きく上回り、後で機能を削減することが困難になるリスクがあることを示唆している。したがって、要求獲得の早い時期からユーザにもある程度コストに

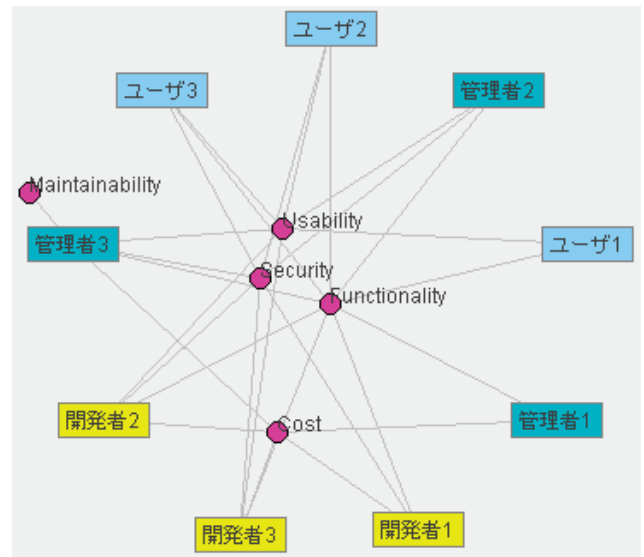


図 6 ゴールグラフのアンカーマップによるレイアウト例 (1)

Fig. 6 Layouts of a goal graph with Anchored Map (1).

ついて興味を持たせることが望ましい。

本研究は、ステークホルダが多少の誤差はあるものの真実の点をつけてくれることを前提としている。たとえば、本当はどちらも絶対値が10であるにもかかわらず片方は本当に10を、もう片方は0を振るというようなことはないという前提に立っている。これはステークホルダが少なくとも自分は関心があるかどうか、強いが弱いかは判断できるという前提であり、自然なものであると考えている。

3.2 ゴールグラフに対するアンカーマップ

我々の手法では、ステークホルダと関心事の関係を可視化するためにアンカーマップを用いている。枝で結ばれたノード間には、距離が離れば離れるほど、また枝に付与された値が大きければ大きいほど大きな引力が働くので、ステークホルダが、その関心事に強い関心があるとき、お互いにより近く配置される。図6のノードの識別は図5よりも明解であり、フリーノード間の関係を容易に読み取ることができる。このように表現することによって、強い関心を持つステークホルダがいない関心事の近くにはステークホルダが配置されず、また誰も関心を持っていない場合にはどのステークホルダもその関心事につながらない。これにより、プロジェクトマネージャや分析者が関心事に対するステークホルダの不足を知ることができる。

アンカーノードの配置は、エッジの交差ができるだけ少なくなるようにしている。この配置アルゴリズムによって、同じアンカーノードに対して枝を持つフリーノードが、近くに配置され、ユーザは、関心事に対するステークホルダの偏りを知ることができる。

同様のアンカーマップは、ステークホルダとそのステークホルダのそれぞれの関心事への興味の大きさが得られれば、AGORAの属性つきゴールグラフ以外からの構築も可

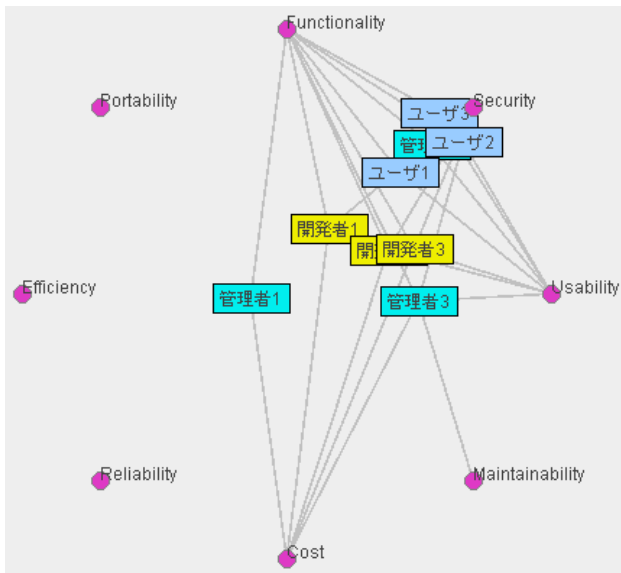


図 7 ゴールグラフのアンカーマップによるレイアウト例 (2)
 Fig. 7 Layouts of a goal graph with Anchored Map (2).

能である．たとえば，User Requirements Notation [9] のゴール指向要求記述言語 (GRN) の部分から同様のデータを作成して，適用することが可能と考える．

3.3 グラフのパターン

本ツールを利用することにより以下のことが読み取れる．

まず，図 6 のようにステークホルダをアンカーノード，関心事をフリーノードにした図からは次のようなことが読み取れる．

- 関心事が円周に近いところに配置された場合，その関心事は，少数のステークホルダにしか関心を持たれておらず，十分な検討がなされていない場合がある．
- 特に多くの関心事が円周に近いところに配置され，真ん中周辺に関心事が配置されない場合，集まったステークホルダがばらばらに関心を示しており，全体として合意がとられていないことがある．
- 描画される関心事が少ない場合，検討されていない関心事があり，検討洩れの場合がある．

また，図 7 のように関心事をアンカーノード，ステークホルダをフリーノードにした図からは次のようなことが読み取れる．

- ステークホルダが円周に近いところに配置される場合，そのステークホルダは，少数の関心事にのみ関心を示しており，他に関心を示すべき関心事に注意を払っていない場合がある．
- 特に多くのステークホルダが円周に近い位置に配置される場合，全体を把握するステークホルダがおらず，大局を見ずに細かいところの議論ばかりされている場合がある．
- 同じ色で表現される同じ役割のステークホルダがそれ

ぞれ異なる位置に配置される場合，それぞれの関心が異なり，意見が一致しない場合がある．

- 複数人の同じ役割のステークホルダが関心事全体をカバーするために集められたにもかかわらず，同じ場所に配置される場合には，全体を十分カバーできていない場合がある．

ステークホルダをアンカーノード，関心事をフリーノードにした場合は，ステークホルダの点数のつけ方によって，関心事が配置される位置が変わる．たとえば，一番興味のある関心事のみに 10 をつけ，それ以外は 1 をつけるステークホルダと，少しでも興味がある関心事にすべて 10 をつけるステークホルダがいた場合，多くの関心事が後者のステークホルダ付近に配置される．この場合でも，関心事は中央には配置されず，関心の偏りが表れるため，ステークホルダの偏りが見逃されることにはならない．一方，関心事をアンカーノード，ステークホルダをフリーノードにした場合は，ステークホルダは各関心事への関心の大きさに対して相対的な位置に配置される．そのため，関心事への関心の大きさに偏りがあれば，それに従って，円周に近い場所に配置される．

4. 実装と利用事例

4.1 ツールに対する要件

AGORA のゴールグラフから，アンカーマップを用いて，ステークホルダと関心事の関係を可視化するツール SViewer を開発した．SViewer では，ある関心事に対して必要な知識を持っている人が含まれていないことに気づけば，該当する人をステークホルダとして加え，ある関心事ばかり詳細化されるようであれば，他の関心事に詳しい人をステークホルダに加えるという使い方が想定される．そこで，ツールとしては，次のような機能が必要になる．

要件 1 ソフトウェアの要求獲得会議において，複数人のステークホルダによってゴールグラフを作成できる．

要件 2 ステークホルダを任意の時点で追加でき，追加されたステークホルダも各ゴールに満足度を振ることができる．

要件 3 作成途中のゴールグラフに対するステークホルダと関心事のアンカーマップが自動的に作成でき，各ステークホルダがそれを参照できる．

4.2 実装

要件 1 と要件 2 を満たすためには，AGORA エディタとの密接な連携が必要である．そのため，SViewer は Eclipse のプラグインとして実装された AGORA エディタから呼び出す外部ツール (Java Applet) として実装された．要件 3 を満たすために，アンカーマップとして表示するデータは，ゴールグラフのエディタに表示されるゴールグラフのゴールに付与された満足度行列とセマンティックタグから

作られる。アンカーマップは、3章で示した配置アルゴリズムを用いて描画される。

ステークホルダをアンカーノードとする場合(図6)、関心事をアンカーノードとする場合(図7)のいずれにおいても、四角がステークホルダを示し、小さな円が関心事を示す。四角は、ユーザ、開発者、システム管理者など、それぞれの役割ごとに色分けされる。各ノードは、マウスによって移動させることができ、複数のノードが重なった場合に、上になったノードを移動させて、下になったノードの内容を確認することができる。フリーノードは、移動させた後にマウスを離すと、スプリングの描画原理に従ってゆっくり元の位置に戻る。ただし、アンカーノードは、最初に位置が計算されて配置されるため、描画後に移動しても元には戻らない。

4.3 利用事例

図1に示した病院の受付業務のシステムのゴールグラフに対して、9人のステークホルダに満足度を振らせて得たアンカーマップが、これまでに説明してきた図6と図7である。ステークホルダは、ユーザ1からユーザ3の3人のユーザ、開発者1から開発者3の3人の開発者、管理者1から管理者3の3人のシステム管理者の全9人で構成されている。

図6では、ステークホルダはアンカーとして1つの円の円周上に等間隔で配置され、関心事はフリーノードとして配置されている。この図では、セキュリティと機能性、使用性の3つの関心事が円の中心付近に配置されていることから、多くのステークホルダがこの3つの関心事にほぼ等しく興味を持っていることが分かる。ほかにも次のようなことが分かる。

- 3人の開発者と管理者1、管理者3はコストに興味を持っているが、ユーザは誰もコストに興味を示していない。
- 管理者3だけが保守性に興味を示している。

ユーザが誰もコストに興味を示していないこのプロジェクトでは、ユーザが思いつくままに要求を出して、要求の実現の優先順位をつけることが困難になることが懸念される。

一方、図7では、関心事がアンカーノードとして1つの円の円周上に配置され、ステークホルダがフリーノードとして配置される。この図では、管理者2は、3人のユーザが重なって隠れた位置に描画されている。この図からは、次のようなことが分かる。

- すべてのユーザは同じ関心事を持っている。管理者2もユーザと同様である。
- すべての開発者も同じ関心事を持っている。しかし、開発者は、ユーザよりもコストに近いところに配置されており、ユーザとは異なり、コストにも関心を持つ

ていることが分かる。

- すべての管理者がそれぞれ異なる関心事に興味を示している。
- 誰も移植性、信頼性、効率性に興味を示していない。これは、抽出元のゴールグラフでは、アプリケーションのビジネス領域が主に記述されており、インフラストラクチャについては記述されていないことによる。これらの事実は、(1)システムのインフラストラクチャについて詳しい人をプロジェクトに加えるべきである、(2)管理者の関心を統一すべきである、ということを示唆している。

この事例によって、SViewerがステークホルダの偏りと不足を可視化できたといえる。また、その他のステークホルダ管理のための有益な情報が得られるともいえる。

5. 評価実験

5.1 目的と手段

目的: SViewerを用いた作業と既存の技術を用いた作業を比較し、前者が効率的にステークホルダの偏りと不足を発見できることを示す。

手順: 前述の両者の作業の正確さとかかった時間の測定、作業結果に関する作業員への質問の回答により判断する。作業には、ステークホルダの偏りと不足に関する複数の質問に答えるという方法を採用し、次のように情報を収集した。

- (1) 作業の正確さを測るために規定時間内に回答できた質問数、回答できた質問に対する正解率を計測した。
- (2) 作業にかかった時間として、各質問への回答に要した時間を測定した。
- (3) 作業結果に関して、被験者に正解数を予想させた。
- (4) プロジェクトが持つリスクを自由記述で記述させた。

自由記述によって得たプロジェクトが持つリスクについては、記述量と内容を分析した。

またツールに対する改良すべき点などを得るために、自由記述によるSViewerに対する意見も収集した。

5.2 実験概要

セマンティックタグ ソフトウェア品質の国際標準であるISO/IEC9126-1[10]を参考に、開発者の意見から、現場でふだんから特に意識する性質を取り入れ、次のように8種類を独自に定義した。

- **機能性 (Functionality)**: 目的から求められる必要な機能を表すゴール
- **信頼性 (Reliability)**: 機能が正常動作続けるために必要なゴール
- **使用性 (Usability)**: 分かりやすさ、使いやすさを表すゴール
- **効率性 (Efficiency)**: 目的達成のために使用する資

表 2 実験に用いたゴールグラフ

Table 2 Goal graphs for the experiment.

システム	ノード数	使われた関心事数	ステークホルダ数
フィードリーダー (F)	29	4	9
病院の受付業務 (H)	36	5	9

	A	B	C	D	E	F	G	H	I	J	K	L
				開発者1	開発者2	開発者3	ユーザ1	ユーザ2	ユーザ3	管理者1	管理者2	管理者3
2	N5	登録する	Functionality	0	9	8	0	9	7	8	9	8
3	N14	登録したエントリをまとめて開く		8	0	0	10	0	0	0	0	0
4	N34	人気のある情報をフラッシュする		0	0	0	0	0	0	0	0	0
5	N3	情報の登録はユーザ登録すること	Usability	0	0	7	7	0	8	0	0	0
6	N13	自分の見たいエントリを選んで登録	Usability	0	0	9	0	0	3	0	0	0
7	N21	自分のフィードの追加が可能		0	0	0	2	0	0	3	0	0
8	N4	自分のフィードの閲覧が可能	Functionality	0	8	9	0	7	8	4	8	9
9	N23	自分のフィードのページを持つ		0	0	0	3	0	0	3	0	0
10	N11	自分で分類	Usability	0	8	8	0	0	9	0	0	0
11	N15	個々のエントリを選択できる		0	0	0	0	0	0	0	0	0
12	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
13	N6	メールアドレスを登録する		0	0	0	8	0	0	0	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	N24	ページには自社のロゴをつける		10	0	0	10	0	0	8	0	0
16	N25	フィードの公開非公開を設定可能	Security	0	0	0	0	0	0	8	0	0
17	N16	フィードのランク付けを設定する		5	0	0	6	0	0	3	0	0

図 8 比較実験用ステークホルダと関心事の一覧表

Fig. 8 Table of the stakeholders and their concerns.

源の多少を表すゴール

- 保守性 (Maintainability) : 保守作業の容易性を表すゴール
- 移植性 (Portability) : 別環境へ移した際そのまま動作するために必要なゴール
- セキュリティ (Security) : システムで扱うデータの安全性を表すゴール
- コスト (Cost) : システムで実現するためにかかる費用や手間を表すゴール

ゴールグラフ 病院の受付業務 (H) とフィードリーダー (F) の支援システムのゴールグラフを用いた。病院の受付業務は、図 1 にその一部を示したものである。また、フィードリーダーは、ブログやニュースサイトなどが配信するフィードを取得、購読するためのアプリケーションソフトウェアを開発することを目的としたものである。

それぞれのゴールグラフは、もともと 3 人 (管理者, エンドユーザ, 開発者) のステークホルダの役割を持った人によって作られたものであり、表 2 に示した大きさである。満足度行列は、今回の実験のために、ゴールグラフの作成、今回の実験のどちらにも関わらないソフトウェア開発者 9 人をそれぞれの役割として 3 人ずつに割り当て、満足度を振らせて作成した。満足度を振らせる際には、実験の目的は知らせず、ゴールグラフに示されたソフトウェアを開発するつもりで、その役割になったという想定で作業させた。

病院の受付業務には機能性, 保守性, 信頼性, 使用性, セキュリティの 5 種類のタグが、フィードリーダーには効率性, 機能性, 使用性, セキュリティの 4 種類のタグが割り振られた。

比較対象 比較対象とする既存の技術として、各ゴールに

対して割り当てられた関心事と、そのゴールに対するステークホルダの満足度の一覧表を用いた。表の例を図 8 に示す。この表は、第 1 列からゴールの ID, ゴール記述, セマンティックタグと続き、最後に各ステークホルダの満足度で構成される。満足度は、無関心の場合は 0 で、絶対値が大きいほど強い関心を示していることを意味する。表は、Excel 形式のスプレッドシートとして被験者に提供することにより、各列をキーにした並べ換えを容易に行え、並べ換えや Excel のその他の機能は自由に使ってよいことにした。

作業 (質問) 被験者に作業として回答させるために 17 の質問を次のように用意した。各質問は、回答の曖昧性を排除するために選択式とした。これらの質問は、被験者が 4.3 節で示したようなステークホルダの不足や偏りを正しく導き出せるかどうかを調べるためのものである。ステークホルダが不足しているとは、ステークホルダによって関心が示されていない、あるいは少数のステークホルダによってのみ関心が示されている関心事がある場合である。したがって、ステークホルダの不足を導き出せたかどうか調べる質問として、関心を示しているステークホルダがいないことを問えばよい。ステークホルダの不足には、個人単位で少数である場合と、ある役割の人が少ないという場合があるので、少数のステークホルダが関心を示す場合については、個人と役割の両方の場合が必要となる。ステークホルダの偏りを導き出せたかどうか調べるにはステークホルダの関心が集中しているか分散しているかを問えばよい。そのために、同じ関心事と異なる関心事に関心を示すステークホルダを答えさせる質問を作成した。また、ステークホルダの偏りを導き出せたかどうか調べる別の種類の質問として、ステークホルダ

の関心の分散を答えさせる質問を作成した。これらに基づき、以下の17の質問を得た。

- (1) ゴールグラフに含まれない関心事はどれか？
- (2) 役割 R のステークホルダが共通に関心を示している関心事はどれか？
- (3) 役割 S のステークホルダが誰も関心を示していない関心事はどれか？
- (4) 役割 T と同じ関心事に関心を示している役割はどれか？
- (5) ステークホルダが全体として強く興味を示している関心事はどれか？
- (6) ステークホルダが誰も興味を示していない関心事はどれか？
- (7) 誰も興味を持っていない関心事はどれか？
- (8) 1人のステークホルダだけが興味を示している関心事はどれか？
- (9) すべてのステークホルダが興味を示している関心事はどれか？
- (10) すべての開発者が興味を示している関心事はどれか？
- (11) ゴール G に割り振られている関心事と同じ関心事に関心を示しているのは誰か？
- (12) 役割 U と同じ関心事に関心を示しているのは誰か？
- (13) 役割 V の役割の中で他のステークホルダと違う関心事に関心を示しているのは誰か？
- (14) 関心事 X に8以上の関心を示しているのは誰か？
- (15) 2つ以上同じ関心事に関心を示しているのは誰と誰か？
- (16) 3つ以上同じ関心事に関心を示している役割はどれか？
- (17) すべての関心事に関心を示している役割はどれか？

(1) から (10) は、関心事を答える質問、(11) から (15) はステークホルダを答える質問、(16) と (17) は、役割を答える質問になっており、いずれも選択肢から答える。役割 R, S, T, U, V, 関心事 X, ゴール G は適当な回答が得られるように問題の対象になるゴールグラフにあわせて割り当てられる。

これらの質問には、答えを組み合わせると特定の関心事へのステークホルダの関心の偏りやステークホルダの不足が導き出せたかどうか分かるようになっているものもある。たとえば、(8) の質問と (9) の質問への回答を組み合わせると特定の関心事へのステークホルダの関心の偏りに気づくことができる。また、(16) の質問と (17) の質問への回答を組み合わせることで、役割ごとの関心の偏りを導き出せたかどうか調べることができる。

被験者 10人の被験者で実験を行った。各被験者のプロフィールは以下のとおりである。

大学院生：ソフトウェア工学を専攻している大学院学生
 大学生：ソフトウェア工学の基礎的な教育を受けている
 情報工学科の大学4年生

研究者 1：大学に所属するソフトウェア工学の研究者

研究者 2：企業に勤めるソフトウェア工学の研究者

研究者 3：企業に勤めるプロジェクトマネジメントの研究者

エンジニア 1：企業に勤める経験5年のソフトウェア開発者

エンジニア 2：企業に勤める経験3年のソフトウェア開発者

エンジニア 3：企業に勤める経験7年のソフトウェア開発者

プロマネ 1：企業に勤める経験3年のプロジェクトマネージャ

プロマネ 2：企業に勤める経験15年のプロジェクトマネージャ

手順 各被験者には、2つのゴールグラフのうち、一方に対しては Excel で作成した一覧表とゴールグラフを用いて各質問に回答させる。もう一方に対しては、SViewer を用いて回答させる。ゴールグラフに対する SViewer の利用と Excel 表の利用は、表 3 のように割り当てた。各質問には2分の制限時間を設け、2分以内に質問に答えられなければ次の問題に答えるように指示した。全体の作業時間は、25分とした。被験者は、各質問に制限時間一杯かかれば12問に回答することになり、時間が余れば13問以降にも回答し、最大で用意した17問すべてに答える。制限時間内に17問すべてに回答し終えた場合は、作業をそこで打ち切った。実験の前に、Excel による一覧表と SViewer の使い方に慣れるために練習を行った。練習には、5つのゴールと3人のステークホルダ、4種類の関心事からなる小さなゴールグラフを用意した。被験者には、このゴールグラフを対象に本番と同様の問題に、一覧表と SViewer を使って、それぞれ10問ずつ回答させた。

5.3 結果と考察

表 3 の4列目は、制限時間内(25分)に回答できた質問の数を示し、5列目は、正解数を示す。最後の列は、全問の回答にかかった時間を示す。

- 完了数：すべての被験者が、両方のゴールグラフに対して、すべての問題に対して、制限時間である2分以内に回答した。そのため、質問への回答数に対する SViewer の使用の有無による差はなかった。
- 正解数：1人の被験者が表を用いた場合に、2つの質問に対して間違いがあったが、その他はすべての質問

表 3 作業の割当てと結果
Table 3 Task assignment and results.

被験者	作業対象	SViewer/一覧表	完了数	正解数	予想	時間
大学院生	F	SViewer	17	17	16	7 m10 s
	H	一覧表	17	17	14	8 m36 s
大学生	H	SViewer	17	17	15	5 m58 s
	F	一覧表	17	15	12	13 m37 s
研究者 1	F	SViewer	17	17	17	6 m19 s
	H	一覧表	17	17	15	17 m53 s
研究者 2	H	SViewer	17	17	16	8 m50 s
	F	一覧表	17	17	15	9 m29 s
研究者 3	F	SViewer	17	17	17	7 m12 s
	H	一覧表	17	17	16	13 m33 s
エンジニア 1	H	SViewer	17	17	13	10 m35 s
	F	一覧表	17	17	10	13 m11 s
エンジニア 2	H	SViewer	17	17	14	10 m15 s
	F	一覧表	17	17	10	14 m11 s
エンジニア 3	F	SViewer	17	17	16	7 m29 s
	H	一覧表	17	17	15	9 m25 s
プロマネ 1	F	SViewer	17	17	16	15 m11 s
	H	一覧表	17	17	15	20 m25 s
プロマネ 2	H	SViewer	17	17	17	8 m15 s
	F	一覧表	17	17	15	10 m21 s

に正しく回答した。

- **時間**：病院の受付業務については、SViewer を使った場合には 8 分 47 秒、表を使った場合には 13 分 58 秒かかった。フィードリーダでは SViewer を使った場合は、平均で 8 分 40 秒、表を使った場合には 12 分 10 秒かかった。病院受付業務、フィードリーダとも、ウェルチの t 検定で完了時間に有意な差があることが確認された。プロジェクトマネージャとエンジニアは SViewer を使ったとき、大学院生、大学生、研究者よりも回答にかかった時間が長い。これに対し、プロジェクトマネージャとエンジニアは、慎重に間違いのないように回答したと実験後のヒアリングで答えたが、SViewer の優位性に影響はなかった。
- **自己予想**：すべての被験者が、Excel による一覧表よりも SViewer を使った方が予想正解数が多かった。
- **自由記述形式によるプロジェクトのリスク**：次のようなリスクがあげられた。
 - 保守性に対して興味を持っている人がいないため、システム稼働後の運用に関することが十分に検討されていないと考えられる。
 - 信頼性に対して、ユーザと管理者が興味を持っていないため、システム稼働後のトラブル発生に対する対応方法が十分に検討されていないと考えられる。
 - 管理者 1 が他の管理者と違う関心事に興味を持っているので、優先順位づけのときに、意見が割れると考えられる。

大学院生が、一覧表を使った方が多くのリスクをあげたが、他の 9 人は、いずれも SViewer を使った方が多くのリスクをあげた。記述内容については、SViewer 利用の有無による顕著な違いは見られなかった。

- **SViewer に対する感想**：次のような感想が得られた。
 - 興味を持っていない特性を見つけやすい。
 - 特性ごとにまとめて傾向を見るのによい。
 - 特性に共通して関心を持っている人を見つけやすい。
 - ステークホルダ別々ではなく、役割ごとにまとめて表示されるとよい。
 - 満足度の値を確認できない。
 - ステークホルダをアンカーノードにした図と、関心事をアンカーノードにした図の両方を同時に見たい。
 - ノードが重なって見にくい。
 - 同じ場所に配置されたノードが強い関心でステークホルダにつながっているのか、弱い関心でステークホルダにつながっているのか分からない。
 - SViewer だけでは不安で、使い慣れた表形式も同時に見たい。

以上の結果は、以下のことを示唆している。すべての被験者が、一覧表よりも SViewer を使った方が予想正解数が多かったということは、自信を持って回答したと考えられる。プロジェクトが持つリスクに関する記述は、作業の後に行われたため、作業を行っている中で気づいたことが記述されたと考えている。SViewer の使用の有無により内容に差が出なかったのは、同じ作業を実施したためである。

と考えている。SViewer を使った場合に記述量が多かったのは、質問に対する回答が短時間で少ない負担で実施できたからだと考えている。被験者の数は少ないが、上記結果は、SViewer が、ステークホルダの偏りや不足を同定するのに、すでによく使われているステークホルダと要求の対応表よりも短時間で、正しくできるために役立つことが示されたといえる。

一方で、各リンクの満足度の値を容易に分かるようにすることが必要であることが分かった。同じ場所にあるノードに対する関心の強さは、ノードの色の濃淡を用いて、強い関心でつながっているノードほど濃い色で表現することが考えられる。また、役割ごとにまとめて表示するのは、ステークホルダが多数の場合に有効だと考えられる。これの解決方法として、役割ごとに、その役割のステークホルダの関心の大きさの和や平均を用いることが考えられる。

5.4 結果に対する脅威

前節で述べた結果に対して影響を与える可能性のある4つの要因について検討する。

- **プロジェクト (F, H) の均質性**：表2に示すように病院の受付業務 (H) の方がフィードリーダー (F) よりもノード数が多く複雑なゴールグラフである。また使われた関心事数も1つ多いが、ステークホルダの数は同じである。ステークホルダと関心事の組合せも病院の受付業務の方が多いため、同じ問題を解く場合、病院の受付業務の方が長くなると考えられるが、SViewer を使った場合で平均1秒、表を使った場合で、2分8秒長くなった。この結果は、SViewer の優位性に影響を与えるものではなく、逆に組合せが多い場合ほど、有効であることを示唆するものと考えられる。
- **作業順 (慣れ)**：被験者に異なる題材に対して、時間をあけずに同じような問題を続けて解かせているため、問題に対して慣れが発生していて、後から解く方が短い時間で解ける可能性があると考えられる。しかしながら、今回の実験では、SViewer を用いた作業を先に実行させたため SViewer を用いた方が短い時間で終わり、SViewer の有効性を脅かす影響はないと考える。
- **被験者の均質性**：被験者が均質である場合、同じような回答の傾向が出ることが考えられるが、今回の実験では、学生2人、研究者3人、実務で開発やマネジメントをしているエンジニア3人を被験者にして、結果が偏らないように配慮した。エンジニア3人の作業時間がやや学生や研究者よりも長く慎重に答えを出してはいるが、SViewer の有効性に影響を与えるものではない。
- **ステークホルダの均質性**：ゴールグラフにして満足度を振ったステークホルダと被験者の関心の傾向が同じ場合、回答が容易になることが考えられる。実験に用

いたゴールグラフは、大学において作られたもので、学生と研究者によって作成された。それに今回の実験とは無関係なエンジニアを6人割り当て、それぞれの役割を演じる形で満足度を割り当てさせた。2つのゴールグラフは、ステークホルダの興味が異なるものとなり、自由記述形式によるプロジェクトのリスクも異なるものが得られた。以上の結果から、ステークホルダの割り振り方も SViewer の有効性に影響を与えるものではないと考える。

6. 関連研究

要求分析に関連して、グラフ構造を用いた研究として Heim らは、要求間の関係を可視化する技術を提案している [11]。本稿で扱う関心事は、ソフトウェアの非機能要求を定義するのにも用いられる。Rohleder は、8つの組合せルールと7つのリンクタイプによって、非機能要求の影響をグラフィカルに表現する手法を提案している [12] が、これは、非機能要求の機能要求に与える影響を表現する点が、本稿で述べた SViewer とは異なる。

ステークホルダの同定に関して、Pouloudi らは、4つの原則を示唆し、それらの原則に基づいたマネジメントの領域に効き目があるアプローチを提案した [13]。Lyytinen らは、情報システムの開発におけるステークホルダの同定のためのガイドラインを示した [14]。これらの原則を SViewer からプロジェクトの潜在的なリスクを読み取るためのガイドラインとして使うことが考えられる。Sharp らは、システムに関わるすべてのステークホルダを同定する技術を提案している [1]。SViewer は、AGORA のゴールのセマンティックタグの代わりに Sharp らの提案するモデルを用いることで、統合することも考えられる。しかしながら、Sharp らの技術は、要求分析とは別の作業として行う必要がある。SViewer は、ゴールグラフのエディタと統合されており、不足しているステークホルダは、分析作業の途中で見つけることができる。

7. まとめ

我々は、システムの信頼性、コスト、使用性などのシステム品質に関する重要な関心事をみれなく獲得するためにステークホルダの不足や偏りを発見することを支援するツール SViewer を開発した。SViewer では、ゴール指向分析手法の1つである AGORA の成果物であるゴールグラフを対象に、要求分析作業中にステークホルダとステークホルダの品質に関する関心事の関係を自動的に抽出し、アンカーマップを使って可視化する。

本稿では、事例によってステークホルダの偏りや不足を見つめることができることを示した。さらに得られた気づきから、プロジェクトの潜在的なリスクが得られる可能性があることも示した。また、SViewer の実装とその評価実

験によって、少ないサンプルではあるが、ステークホルダの偏りや不足を同定するのに、これまでによく利用されているステークホルダと要求の対応表では12分から14分かかっていたタスクを、SVIEWERでは同じタスクを9分弱で、より短時間で、正しくできることを示した。最初にあげたロンドンの救急車のサービスの失敗事例においてこのようなツールがあれば、情報入力の使用性への関心を持つステークホルダが不足していることに気づき、救急隊員から要求を抽出できたであろうと考えられる。

評価実験を通じて、各リンクの満足度の値を容易に分かるようにすることが必要であることが分かり、同じ場所にあるノードに対する関心の強さはノードの色の濃淡を用いて、強い関心でつながっているノードほど濃い色で表現することを考えた。また、役割ごとにまとめて表示することがステークホルダが多数の場合に有効だと考えられる。これの解決方法として、役割ごとにその役割のステークホルダの関心の大きさの和や平均を用いることが考えられる。これらの機能を実装して、効果検証を行うことが今後の課題である。

AGORAの満足度行列をそのまま利用して作られたアンカーマップからステークホルダの不足や偏りを発見する場合に、10点法が最適であるかどうか、あるいはスケールングの差が結果に与える影響を分析、評価するのも今後の課題である。

また、ステークホルダをアンカーノード、関心事をフリーノードにした場合は、ステークホルダの点数のつけ方によって、関心事が配置される位置が変わる。今回行った実験で用いた2つのゴールグラフに関しては、アンカーマップによる表現が既存の表を使った方法に比較して、誤解や曖昧さを増やすことはなかった。しかし、この実験に使ったアンカーマップがたまたま被験者の解釈のぶれが小さくなるようなものであった可能性は残っている。このようなぶれを防ぎ、安定した分析を可能にするには、満足度行列の点のつけ方に基準を設け、できるだけ人によって点数のつけ方にぶれが出ないようにすることが必要となる。点数をより具体的な例に沿ってつけられるようにするなどが考えられるが、実例への適用や実験を通じて、プロジェクトにおけるリスクが明確にできるような適切な基準を作成することも今後の課題である。

同じステークホルダが、同じタグが振られてはいるが、異なるゴールに大きく異なる点をつけた場合、そのステークホルダがその関心事に関心があるのかどうかあいまいになる。たとえば、Aというステークホルダが信頼性が振られた2つのゴールに各々10と0をつけてしまったような場合、そのステークホルダは信頼性について、関心があるのか無関心なのか分からない。このような場合には、つけられた点の偏差などの情報をマップ上に可視化し、ステークホルダの評価の安定性や本当にその関心事に関心があるの

かどうかを提示することが今後の課題として考えられる。

さらに、事例を増やし、さらにSVIEWERの有効性、実効性を示すことが必要であるとも考えている。事例を増やすことにより、3.3節に示したような読み取りパターンがさらに得られ、グラフの読解を容易にすると考えている。

謝辞 本稿執筆にあたり、下書き時から有益なコメントをくださった東京工業大学大学院情報理工学専攻の井上渉氏に感謝します。アンカーマップのライブラリを本システムのために改修下さった筑波大学の三末和男准教授に感謝します。

参考文献

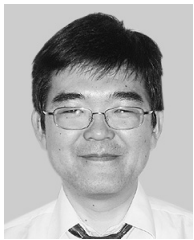
- [1] Sharp, H., Finkelstein, A. and Galal, G.: Stakeholder identification in the requirements engineering process, *Proc. 10th International Database and Expert Systems Applications Workshop*, pp.387-391 (1999).
- [2] Project Management Institute: *A Guide To The Project Management Body of Knowledge* (2005).
- [3] Misue, K.: Anchored Maps: Visualization Techniques for Drawing Bipartite Graphs, *HCI (2)*, Jacko, J.A. (Ed.), *Lecture Notes in Computer Science*, Vol.4551, pp.106-114, Springer (2007).
- [4] Kaiya, H., Horai, H. and Saeki, M.: AGORA: Attributed Goal-Oriented Requirements Analysis Method, *Proc. 10th Requirements Engineering Conference*, pp.13-22 (2002).
- [5] Kaiya, H., Shinbara, D., Kawano, J. and Saeki, M.: Improving the detection of requirements discordances among stakeholders, *Requirements Engineering*, Vol.10, No.4, pp.289-303 (2005).
- [6] van Lamswerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour, *Proc. 9th Requirements Engineering Conference*, pp.249-263 (2001).
- [7] Saeki, M., Hayashi, S. and Kaiya, H.: A Tool for Attributed Goal-Oriented Requirements Analysis, *Proc. 24th IEEE/ACM International Conference on Automated Software Engineering*, pp.670-672 (2009).
- [8] Eades, P.: A Heuristic for Graph Drawing, *Congressus Numerantium*, Vol.42, pp.149-160 (1984).
- [9] Amyot, D.: Introduction to the User Requirements Notation: Learning by Example, *Computer Networks*, Vol.42, No.3, pp.285-301 (2003).
- [10] ISO/IEC 9126-1:2001. Software engineering - Product quality - Part 1: Quality model, ISO/IEC (2001).
- [11] Heim, P., Lohmann, S., Lauenroth, K. and Ziegler, J.: Graph-based Visualization of Requirements Relationships, *Proc. 3rd Requirements Engineering Visualization*, pp.51-55 (2008).
- [12] Rohleder, C.: Visualizing the Impact of non-Functional Requirements on Variants: A Case Study, *Proc. 3rd Requirements Engineering Visualization*, pp.11-20 (2008).
- [13] Pouloudi, A. and Whitley, E.A.: Stakeholder identification in inter-organizational systems: Gaining insights for drug use management systems, *European Journal of Information Systems*, Vol.6, No.1, pp.1-14 (1997).
- [14] Lyytinen, K. and Hirschheim, R.: Information systems failures - A survey and classification of the empirical literature, *Oxford Surveys in Information Technology*, pp.257-309, Oxford University Press, Inc. (1987).

推薦文

本研究は、ソフトウェア開発会議への参加者の関心事に対する強さを開発会議の中間成果物であるゴールグラフから可視化する技術である。可視化によって意見をいってもらうべき人の不足や特定の関心事に対する参加者の偏りなどが容易に分かりソフトウェア開発の失敗を早い段階で防ぐ。グループ作業支援の研究に貢献するものとして、本発表を推薦します。

(グループウェアとネットワークサービス研究会主査

小林 稔)



鵜飼 孝典 (学生会員)

1990年東京工業大学工学部情報工学科卒業。1992年同大学大学院理工学研究科電気電子工学専攻修士課程修了。同年(株)富士通研究所入社。2012年東京工業大学大学院情報理工学研究科計算工学専攻博士後期課程修了予定。

要求工学, ナレッジマネジメントの研究に従事。ACM, ソフトウェア科学会, 人工知能学会各会員。



林 晋平 (正会員)

2004年北海道大学工学部情報工学科卒業。2006年東京工業大学大学院情報理工学研究科計算工学専攻修士課程修了。2008年同専攻博士後期課程修了。2009年より同専攻助教。博士(工学)。ソフトウェア変更やソフトウェア開発環境の研究に従事。IEEE-CS, ACM各会員。

ア開発環境の研究に従事。IEEE-CS, ACM各会員。



佐伯 元司 (正会員)

1978年東京工業大学工学部電気電子工学科卒業。1980年同大学大学院理工学研究科情報工学専攻修士課程修了。1983年同専攻博士後期課程修了。同大学助手, 助教授を経て, 2000年より同大学大学院情報理工学研究科計算工

学専攻教授。工学博士。要求工学やソフトウェア開発技法等の研究に従事。IEEE-CS, ACM, ソフトウェア科学会, 人工知能学会, 電子情報通信学会各会員。