

非機能要求獲得のためのゴール分解方法の提案 -ゴールカテゴリを用いた場合-

岡野 道太郎^{1,a)} 中谷 多哉子^{2,b)}

概要：システムの要求獲得にゴール指向要求分析を用いる場合、ゴール分解方法の指針がないため、要求獲得が不完全なままゴール分解を終了してしまう可能性がある。本研究では、特に非機能要求におけるゴール分解方法を提案する。この方法は、Lamsweerde のゴールカテゴリにおける非機能ゴールを参考に作られている。この提案に基づき、ソフトウェア工学の共通問題である「酒屋倉庫問題」を分析したところ、提案した方法で非機能要求の課題のいくつかを解決できた。

1. はじめに

要求獲得を行う際、KAOS[1]などのゴール指向分析手法を適用することがある。KAOSでは、ゴールモデルは、木状のモデルとして表され、システムの目標を示すトップゴールから、AND分解またはOR分解によって詳細化を繰り返し、下位ゴールを定義していく。ここでAND分解とは詳細化されたサブゴールのすべてが達成されないとゴールが達成されない分解方法であり、OR分解とはサブゴールのうち1つでも達成されればゴールが達成される分解方法である。

このような手続きを経るとき、現状では、どのようにAND分解やOR分解を行っていくかの明確な指針がない。そのため、要求分析者がどのようにゴールを分析していけばよいのかわからず、要求を十分抽出できていないのにゴール分析を終了してしまうことがある。そこで我々は、ゴール分解をどのように行えばよいかの指針を開発している[2]。本稿は、非機能要求におけるゴール分解方法の提案を行う。

非機能要求のゴール分析を行う場合には、機能要求のゴール分析とは違った点がある。非機能要求には、機能的な面が含まれないので、多くのシステムで考慮すべき点が共通している点も多い。例えば、性能やレスポンスタイム、信頼性等は、多くのシステムで考慮すべきこととして挙げられる。そのため、共通項目を抜き出して、チェックリス

トを作りやすい。

しかし、機能要求と違って難しい課題もある。たとえば利用場面によって、要求されるグレードや重要度が異なる場合があることである。レスポンスタイムは、バッチ処理とオンライン処理では、要求されるグレードは異なる。この点を考慮すると、利用場面に応じて要求を挙げなければならなくなるが、利用場面は機能仕様が決められないと確定しないこともある。

そして、非機能要求はステークホルダーが数多く存在する場合がある。このとき、各ステークホルダーの関心事は大きく異なるかもしれないことも、課題となるかもしれない。例えば、開発者や開発費用を負担する経営者は、システムの移植性に関心を持つと思われるが、システムの利用者は、移植性に関心が無いかもしれない。また運用時の騒音は、運用しているシステムの近隣住民には関心があるかもしれないが、開発後にシステムに関わらない開発者にとっては、あまり関心の無いことかもしれない。

その上、システムが完成しないと要求を満たすかどうか分からない要求も有ることである。例えばレスポンスタイムの正確な数字は、全システムを完成させてからでないと確定しないため、全システムができて稼働させて初めて、要求を満たしていないことに気づくことも有り得るかもしれない。使いやすさはレスポンスタイムの影響を受けるので、使いやすさも出来上がって見ないと分からない部分がある。

このように非機能項目には、いろいろな課題があるため、単純にチェックリストを作成し確認すればよいというものではない。本稿では、これら課題を考慮しながら非機能要求におけるゴール分解を行う方法の提案を行う。

¹ 筑波大学
University of Tsukuba

² 放送大学
The Open University of Japan

a) okano@gssm.otsuka.tsukuba.ac.jp

b) tinakatani@ouj.ac.jp

本稿は以下の構成となっている。次の2章では関連研究を挙げ、3章で提案する方法を説明する。4章で酒屋倉庫問題 [3][4] にその提案を適用し、5章で適用結果を考察し、6章でまとめる。

2. 関連研究

ゴール指向分析はi*[5], KAOS[1] 等、様々な手法が提案されている。特に非機能要求に関しては、Lawrence ChungらのNFRフレームワーク [6] が知られている。NFRフレームワークは、非機能要求を階層構造にまとめたもので、このフレームワークを利用することによって、非機能要求の分析を網羅的に進めることができる。

ゴール指向分析において、同様に非機能要求を網羅的に分析する枠組みとして、Lamsweerdeのゴールカテゴリ [1] がある。ゴールカテゴリはゴールを階層構造に分類したものであり、機能ゴールと非機能ゴールに分かれる。この非機能ゴールのゴールカテゴリを利用することにより、NFRフレームワークとは異なるが、非機能要求の分析を網羅的に進めることができる。

ゴール指向分析に限らなければ、非機能要求を網羅的に扱うものは他にもあり、例えば非機能要求グレード [7] やISO9126 (JIS X 0133-1:1999) の品質特性モデルなどを使っても、非機能要求の分析を網羅的に進めることができる。

ただしこれらは、非機能要求の項目が挙がっているのであって、これらをもとにチェックするだけでは、上述の課題は解決できない。課題を解決するには、各非機能要求に対して、ステークホルダーがどのような価値観をもって要求を主張しているのかが可視化できなくてはならない。

ステークホルダーの価値観を可視化した方法としては、ゴール指向分析において、ステークホルダーを中心として分析したものとしては、所有者別ゴールモデル [8] がある。所有者別ゴールモデルは、パーティションによって区切られた空間に、ステークホルダ毎のゴールモデルを構成し、ゴール間の競合や協調関係を定義するモデルである。たとえば、複数の所有者のゴール木のノード間にある種の依存関係がある場合、両者のゴールノードをつないで、正負の貢献度を記述する。ただし、これに非機能要素の分類などは含んでいない。

3. 提案する手法

3.1 課題へのアプローチ

ここで、非機能要求の課題を整理すると、以下のとおりである。

- 個人的なこだわりや価値観の相違、立場の違いがある
- 利用場面によって要求されるグレードや重要度が異なる場合がある
- ステークホルダーが多く、関心事が異なる

- システムが完成しないと要求を満たしているか分からない指標がある

主観や価値観の対立を扱うには、所有者別ゴールモデルのように、各ステークホルダーを並べて、価値観とその対立点を図示するのがよい。この点で、ゴールモデルは向いている。しかし、利用場面やステークホルダーのように、多数のものを扱うのは表のほうがよい。ゴールモデルでは1枚に書ききれず、複数ページにまたがることになるが、この場合、一覧性に欠けると思われるからである。また、数値化が難しい場合、文章で表現することになるが、これも表のほうが望ましい。ゴールグラフでは文が書ききれないからである。

システムが完成しないと要求を満たすかどうか分からない指標に関しては、このような指標が開発に重大な影響を及ぼす場合、完成間近になってから問題が発覚してしまうと開発に大きな影響を与えてしまうので、そのような指標は重点的に管理したい。重点管理する項目だけを抜き出す等の操作は表計算ソフトが向いている。

このことから、図と表の両方を使ったほうがよいことが分かる。

3.2 提案で使用する図表

本提案では、「全体表」、「個人評価表」、「要求一覧表」、「ゴール分析図」の図表を用いて分析を行う。

全体図は、各桁にステークホルダー、各行に評価基準を配置し、交叉したセルに、ステークホルダーの評価基準における重要度を数値で記述する。評価基準は、非機能要求グレードの項目やLamsweerdeのゴールカテゴリ中の非機能ゴール等が挙げられる。評価基準のうち、重点的に管理すべき項目は網掛けないしは色つきにする。重要度については、個人評価表のところで後述する。この表により、ステークホルダーの全体と、各ステークホルダーが、どのようなことに関心を持っているかが分かる。システム全体で1枚作成される。

個人評価表は、各ステークホルダー毎に作成される。大きく3つの部分からなる。第一の部分は1行目で、ステークホルダー名と、そのステークホルダーの価値観や世界観があれば記述される。第二の部分は2行目以降の表で、各桁は業務が並び、最後に重要度が配置される。各行に評価基準を配置する。交叉したセルに、ステークホルダーが、業務に対して評価基準の観点から要求がある場合、セル番号を記述する。実際の要求は第三の部分に記述される。要求がない場合は空欄となる。桁の最後の重要度には、各ステークホルダーにとって、評価基準がどれほど重要であるか、5を最も重要、3をどちらでもない、1を全く重要でないとし、数値化して記述する。ただし、各評価基準に対して、1つも要求がない場合は記述しなくてよい。この重要度が、「全体表」の重要度に転記される。第三の部分

は、第二の部分の下に記述され、各行に要求が記述される。各行は、第二の部分で記述した要求のセル番号、要望者番号、視点、非機能要件が記述される。要望者番号とは、各ステークホルダーに割り当てられた番号である。システムの要求を決める上で、重視すべき人から昇順にステークホルダー番号を割り当てると、後の作業が行いやすい。非機能要件のところに要求を記述する。視点とは、該当する要求（非機能要件）を要求として挙げた根拠となる価値観や世界観で、第一の部分に書かれた価値観や世界観の中から、該当するものを選んで記述する。後述する事例では、価値観や世界観に番号を振り、番号で記入している。この表により、各ステークホルダーがどのような要求を持っているか具体的に分かる。

要求一覧表は、各ステークホルダーの個人評価表の第三の部分、すなわち要求が記述された部分を転記して1つの表にまとめたものである。

ゴール分析図は、上述の表を元に要求をまとめる際、ステークホルダー間で合意形成が出来ない等の問題が起こったときに作成される。問題となる評価基準Hが1つの場合は、「Hが問題となっている」というトップゴールを置く。2つ以上の場合「H1, H2・・・Hn が問題となっている」というトップゴールを置き、そのサブゴールとして「H1 が問題となっている」、「H2 が問題となっている」とする。いずれの場合も「ある評価基準が問題となっている」という形が、ここでの最終的なゴールとなる。「ある評価基準が問題となっている」のサブゴールとして、ステークホルダーSに対して、「SがHに関心をもっている」というゴールを評価基準Hを評価したステークホルダー分、記述する。このとき、重要度が3であれば、「関心を持っている」になるが、重要度が5であれば「Sがとても関心を持っている」、1であれば「Sは全く関心を持っていない」となる。ただし、すべて重要度が5のように重要度の信頼性が低い場合は、関心を持っている度合を変えて表現する。「SがHに関心を持っている」等のサブゴールとして、要求に対する価値観や世界観Wを挙げ、「SはWになるべきだと思っている」というゴールを挙げる。このサブゴールとして各要求を挙げる。これにより、どのような価値観や世界観から要求が対立し、合意形成出来ていないか、信念対立の構図が見える。

3.3 手順

各ステークホルダーに個人評価表を記入してもらい、その情報から全体表を作成する。また、個人評価表の第三の部分、まとめて1シートに転記し、要求一覧表を作成する。要求一覧表が作成されたら、セル番号、要望者番号でソートする。これによりセル番号、すなわち各業務における各評価項目の要求が、重要なステークホルダー順にならぶ。このシートをみて、合意形成上、問題があると思われる

受付係 1. ストレスなく操作 2. 問題を起こさない 3. 依頼者に迷惑をかけない

		在庫なし連絡	出庫指示書	在庫不足リスト	重要度	
サービスの質	安全性					
	セキュリティ	機密性				
		完全性	D5	E5	F5	5
		可用性	D6	E6	F6	5
	信頼性		D7	E7	F7	5
			D8	E8	F8	5
	パフォーマンス	時間				
		空間				
		価格				
インターフェース	UI	D11	E11	F11	5	
	デバイス					
	ソフト相互運用性					
	正確さ	D14	E14	F14	5	
コンプライアンス						
アーキテクチャ	インストール					
	配布					
開発	コスト					
	納期					
	可変性					
	保守性					

セル位置	要望者No	観点	非機能要件
D5	20	2, 3	障害時にデータがなくなることをないようにしてほしい
E5	20	2, 3	障害時にデータがなくなることをないようにしてほしい
F5	20	2, 3	障害時にデータがなくなることをないようにしてほしい
D6	20	1	セキュリティをあまり意識しないでふつうに操作してほしい
E6	20	1	セキュリティをあまり意識しないでふつうに操作してほしい
F6	20	1	セキュリティをあまり意識しないでふつうに操作してほしい
D7	20	2, 3	24時間365日いつでも使えるようにしてほしい
E7	20	2, 3	24時間365日いつでも使えるようにしてほしい
F7	20	2, 3	24時間365日いつでも使えるようにしてほしい
D8	20	1, 2, 3	レスポンスタイムは3秒以内
E8	20	1, 2, 3	レスポンスタイムは3秒以内
F8	20	1, 2, 3	レスポンスタイムは3秒以内
D11	20	1	使いやすいユーザーインターフェース
E11	20	1	使いやすいユーザーインターフェース
F11	20	1	使いやすいユーザーインターフェース
D14	20	2, 3	正しく処理してほしい
E14	20	2, 3	正しく処理してほしい
F14	20	2, 3	正しく処理してほしい
E14	20	2	エラー通知&入力値があまりありえない値なら警告

図 1 受付係の個人評価表

Fig. 1 The table of evaluation by the receptionist

るところに関してゴール分析図を作成する。

4. 事例

本提案を、「酒屋倉庫問題」の非機能仕様に適用した。「酒屋倉庫問題」とは、情報処理 25 巻 9 号「共通問題によるプログラム設計技法解説」[3]で紹介された問題で、酒屋の倉庫の受付係のための計算機プログラムを作成する問題である。その後 25 巻 11 号 [4]で、「在庫不足の商品が入庫次第発送する」と、問題の一部が仕様変更されている。本提案では、25 巻 11 号、すなわち、修正後の問題を元に、提案した手法を適用する。

4.1 個人評価表の記入

酒屋倉庫問題では、文章中に「倉庫係」「受付係」「依頼者」が登場する。そこでまず、この三者をステークホルダーとし、「倉庫係」「受付係」「依頼者」の「個人評価表」を記入する。本来であれば、それぞれの役割の者が記入すべきだが、本事例では、筆者が三者に成り代わって記入した。評価項目は、Lamsweerdeのゴールカタログ中、非機能ゴールの各ゴールを表形式に編集して作成している。業務は、「酒屋倉庫問題」の文中に、「在庫なし連絡」「出庫指示書作成」「在庫不足リスト作成」のプログラムを作成することが課題として明記されているので、これらを業務内容とする。「受付係」の「個人評価表」は図1であり、「倉庫係」の「個人評価表」は図2であり、「依頼者」の「個人評価表」は図3である。

倉庫係 1. システム化による新たな問題にかかりたくない

		在庫なし連絡	出庫指示書	在庫不足リスト	重要度
サービスの質	安全性				
	セキュリティ	機密性			
	ディ	完全性			
		可用性			
	信頼性		D7	E7	F7
パフォーマンス	時間				
	空間				
	価格				
インターフェース	UI				
	デバイス				
正確さ	ソフト相互運用性				
コンプライアンス		D14	E14	F14	5
アーキテクチャ	インストール				
開発	配布				
	コスト				
	納期				
	可変性				
	保守性				

セル位置	要望者No	観点	非機能要件
D7	2,1		平日はいつも使えるようにしてほしい(ダウンタイムMAX1時間)
E7	2,1		平日はいつも使えるようにしてほしい(ダウンタイムMAX1時間)
F7	2,1		平日はいつも使えるようにしてほしい(ダウンタイムMAX1時間)
D14	2,1		正しく処理してほしい
E14	2,1		正しく処理してほしい
F14	2,1		正しく処理してほしい

図 2 倉庫係の個人評価表

Fig. 2 The table of evaluation by the warehouse clerk

依頼者 1. いままでどおり、仕事を恙なく行ってほしい 2. 迷惑をかけないでほしい

		在庫なし連絡	出庫指示書	在庫不足リスト	重要度
サービスの質	安全性				
	セキュリティ	機密性	D4		4
	ディ	完全性	D5		5
		可用性			
	信頼性		D7		3
パフォーマンス	時間				
	空間				
	価格				
インターフェース	UI				
	デバイス				
正確さ	ソフト相互運用性				
コンプライアンス		D14			5
アーキテクチャ	インストール				
開発	配布				
	コスト				
	納期				
	可変性				
	保守性				

セル位置	要望者No	観点	非機能要件
D4	10,2		電話番号を、むやみやたらに他業者に教えないでほしい
D5	10,1,2		住所、電話番号の変更は、すぐに反映してほしい
D7	10,1,2		当社営業日には、連絡がつくようにしてほしい
D14	10,1,2		正しく処理してほしい

図 3 依頼者の個人評価表

Fig. 3 The table of evaluation by the requester

4.2 重要な評価項目の確認とステークホルダーの追加

ここでシステムが完成しないと要求を満たしているか分からないが、システムが完成してから問題があることに気づいたのでは遅い評価項目で、要求が出ていない評価項目がないか確認する。本事例においてシステムが完成しないとわからない評価項目としては、信頼性、開発コスト、納期が挙げられる。しかし開発コストや納期については、要求が出ていない。そこでそれらの要求を持っているであろう酒屋経営者もステークホルダーに加え個人評価表を作ることとした。これも筆者が記入している。

4.3 全体表の作成

ステークホルダーの個人評価表ができたので、その記述内容をもとに全体表を作成する。評価項目は Lamsweerde のゴールカタログ中、非機能ゴールの各ゴールを表形式に編集して作成したものであり、重要度は各ステークホルダーの個人評価表から転記している。図 5 が本事例で作成した全体表である。

酒屋経営者 1 低コスト、2 顧客に選ばれるように、3 不祥事を起こさない

		在庫なし連絡	出庫指示書	在庫不足リスト	重要度	
サービスの質	安全性					
	セキュリティ	機密性	D4	E4	F4	5
	ディ	完全性	D5	E5	F5	5
		可用性				
	信頼性		D7	E7	F7	3
パフォーマンス	時間					
	空間					
	価格					
インターフェース	UI					
	デバイス					
正確さ	ソフト相互運用性					
コンプライアンス		D10	E10	F10	5	
アーキテクチャ	インストール					
開発	配布					
	コスト					
	納期					
	可変性	D18	E18	F18	5	
	保守性	D19	E19	F19	5	

セル位置	要望者No	観点	非機能要件
D4	1,3		必要最小限のししか、アクセスできないようにする
E4	1,3		必要最小限のししか、アクセスできないようにする
F4	1,3		必要最小限のししか、アクセスできないようにする
D5	1,2,3		最新の情報であるようにすること
E5	1,2,3		最新の情報であるようにすること
F5	1,2,3		最新の情報であるようにすること
D7	1,2,3		ダウンタイムを短くする
E7	1,2,3		ダウンタイムを短くする
F7	1,2,3		ダウンタイムを短くする
D10	1,1,2		コストパフォーマンスをよくなる
E10	1,1,2		コストパフォーマンスをよくなる
F10	1,1,2		コストパフォーマンスをよくなる
D18	1,1		ランニングコスト月100万
E18	1,1		ランニングコスト月100万
F18	1,1		ランニングコスト月100万
D19	1,2		納期3か月後
E19	1,2		納期3か月後
F19	1,2		納期3か月後

図 4 「酒屋経営者の個人評価表

Fig. 4 The table of evaluation by the CEO

非機能要件		全体表			
		受付係	倉庫係	依頼者	酒屋経営者
サービスの質	安全性				
	セキュリティ	機密性		4	5
	ディ	完全性	5		5
		可用性	5		
	信頼性		5	4	3
パフォーマンス	時間	5			
	空間				
	価格				5
インターフェース	UI	5			
	デバイス				
正確さ	ソフト相互運用性				
コンプライアンス		5	5	5	5
アーキテクチャ	インストール				
開発	配布				
	コスト				5
	納期				5
	可変性				
	保守性				

図 5 全体表

Fig. 5 The whole table

4.4 要求一覧表の作成

ここで、要求中に対立しているところや矛盾しているところがないか、要求一覧表を作成して確認する。図 6 が本事例で作成した要求一覧表の一部である。各ステークホルダーの要求が記述されている部分を一覧表に転記し、セル番号と要求者番号をもとにソートする。すると、各業務における各評価項目の要求が重要なステークホルダー順に並ぶ。要求分析者はこの要求内容を見て、対立点はないか確認し、問題がある場合には、その要求のゴール分解図を作成する。

本事例で作成した要求一覧表を見て、システムの予算 1000 万と開発費用の上限が決まっていることと、24 時間

セル位置	要望者No	観点	非機能要件
D4	1	3	必要最小限の人しか、アクセスできないようにする
D4	10	2	電話番号を、むやみやたらに他業者に教えないでほしい
D5	1	2, 3	最新の情報があるようにすること
D5	10	1, 2	住所、電話番号の更新は、すぐに反映してほしい
D5	20	2, 3	障害発生率がなくなることはないようにしてほしい
D6	20	1	セキュリティをあまり意識しないでふつうに操作できてほしい
D7	1	2, 3	ダウンタイムを短くする
D7	2	1	平日はいつでも使えるようにしてほしい(ダウンタイムMAX1時間)
D7	10	1, 2	当社営業日には、連絡がつくようにしてほしい
D7	20	2, 3	24時間365日いつでも使えるようにしてほしい
D8	20	1, 2, 3	レスポンスタイムは3秒以内
D10	1	1, 2	コストパフォーマンスを良くする
D10	1	1	オンラインコスト月100万
D11	20	1	使いやすいユーザーインターフェース
D14	1	2, 3	正しく処理してほしい
D14	2	1	正しく処理してほしい
D14	10	1, 2	正しく処理してほしい
D14	20	2, 3	正しく処理してほしい
D18	1	1	システム全体予算1000万
Page	1	1	確定申告日録

図 6 要求一覧表 (一部のみ)

Fig. 6 Requests list (partial)

365 日稼働させるシステムを作成することが、対立を起すかもしれないと思われた。そこで、その部分についてゴール分析図を作成することとした。24 時間 365 日稼働させる場合、高価なシステムになる可能性があるが、その要求はどれほど強いものか見たいからである。

4.5 ゴール分析図の作成

対立するところや重要なところなど、要求分析者が詳しく分析したい部分について、要求を生み出した価値観を含めて分析するためにゴール分析図を作成する。今回はシステムの予算 1000 万と開発費用の上限が決まっていることと、24 時間 365 日稼働させるシステムを作成することが、対立を起すかもしれないので、コストと信頼性についてゴール分析図を作成することとした。それが図 7 である。受付係は、重要度をすべて 5 にしている。これは本当に重要と感じているかどうか分からないので、信頼性の重要度を「非常に重要」ではなく、「他の評価基準と同じ重要度」とした。この図を見ると、24 時間 365 日という要望は、顧客に迷惑をかけたくないことや、問題を起したくないことからの発言と読み取れるが、依頼者の要望は、営業日に届けばよいと思っている。依頼者が 24 時間 365 日依頼を出していて、即座に出庫することを求めているのであれば、たしかに受付係の要求どおり 24 時間 365 日稼働するシステムは必要であるが、そのようなシステムを稼働させても、倉庫係が 24 時間 365 日対応してくれなければ、依頼者に即座には届かない。その点から考えると、過剰な要求である可能性が高いことが読み取れる。

5. 考察

5.1 本提案の有効性

本稿は、非機能要求におけるゴール分解方法の提案を行うことを目標としていた。その際、非機能要求のゴール分析を行う場合、機能要求のゴール分析とは違った課題があることを指摘した。今回の手法で、この課題に対応できていれば、有効性が示せたことになる。

非機能要求のゴール分析の課題は、以下のものがあげら

れた。

- 個人的なこだわりや価値観の相違、立場の違いがある
- 利用場面によって要求されるグレードや重要度が異なる場合がある
- ステークホルダーが多く、関心事が異なる
- システムが完成しないと要求を満たしているか分からない指標がある

本提案では、個人の価値観を大事にするため、要求を文書化し、その要求を生み出す価値観や世界観を、ゴール分解図に表現した。

利用場面に関しては、個人評価表で、業務ごとに要求を抽出するようにした。重要度は、個人によってばらつきがあるので数値化は難しいため、数値を直接使うのではなく、要求分析者が見て、場合によっては独自の判断を加えて、ゴール分解図に表現した。

また、ステークホルダーが多いため、ゴールグラフで全てを記入すると複雑になってしまうので、表で表現した。関心事が異なるため、関心のあるところのみ個人評価表に記入し、全体表にまとめた。システムが完成しないと要求を満たしているか分からない指標は、厳重に管理する必要があるため、全体表で指標に関する評価基準を目立たせ、また十分な要求が出ていない場合はステークホルダーを追加し、要求を抽出した。

重要な指標や、要求が対立している箇所を見つけたら、ゴール分解図に要求のほかに価値観や重要度を記入し、対立構造が分かるようにした。

このように課題に対応している。

5.2 今後の課題

本提案では、個人評価表で業務に対して要求を抽出しているが、業務は本来、機能要求を分析して分かるものである。よって、機能要求の抽出方法を研究する必要がある。それが出来ると、機能要求および本提案の非機能要求の抽出方法が出来るので、要求の抽出方法全体が提案できることになる。

6. まとめ

システムの要求獲得にゴール指向要求分析を用いる場合、ゴール分解方法の指針がないため、要求獲得が不完全なままゴール分解を終了してしまう可能性がある。このような事態を避けるため、我々はゴール分解の方法について研究している。本研究では、その中で特に非機能要求におけるゴール分解方法を提案した。非機能要求の課題に対応するため、本提案は、Lamsweerde のゴールカテゴリにおける非機能ゴールを参考に要求を書き出し、対立点があった場合、ステークホルダーの価値観も含めてゴールグラフを作成し、対立点を可視化している。この提案に基づき、ソフトウェア工学の共通問題である「酒屋倉庫問題」を分

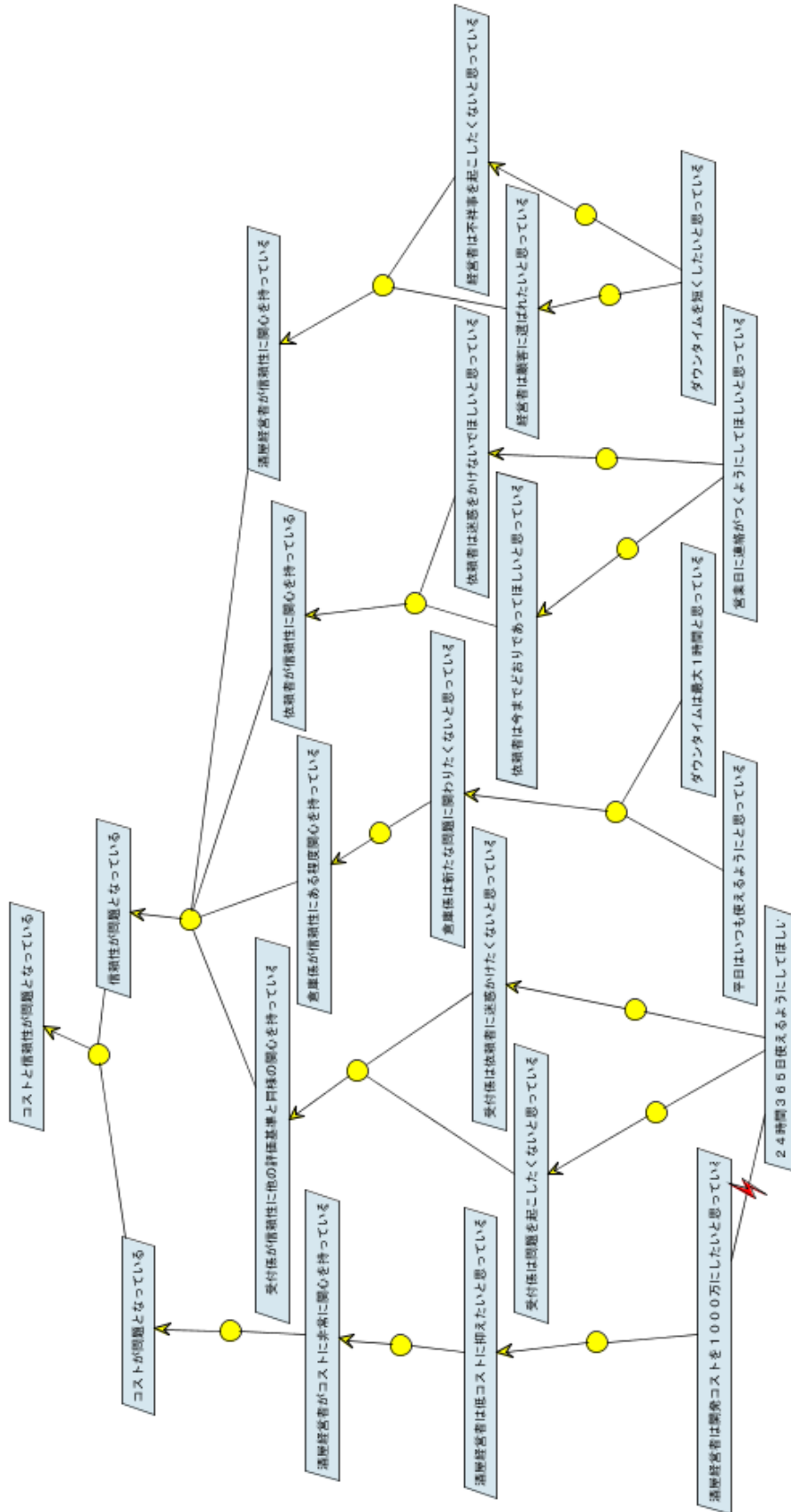


図 7 ゴール分解図

Fig. 7 The diagram of goal tree

析したところ、提案した方法で非機能要求の課題のいくつかを解決できた。

参考文献

- [1] Axel van Lamsweerde: *Requirements Engineering: From System Goals to UML Models to Software Specifications*, Wiley(2009).
- [2] 岡野道太郎, 中谷多哉子: 要求のヌケ・モレを防ぐためのゴール分解方法の提案と実験-ソフトウェア・シンポジウム 2014 WG3 の事例-, ソフトウェア・シンポジウム 2015(発表予定)
- [3] 山崎利治: 共通問題によるプログラム設計技法解説, 情報処理 Vol25 No9(1984).
- [4] 山崎利治: 共通問題によるプログラム設計技法解説 (その 2), 情報処理 Vol25 No11(1984).
- [5] *i** homepage, 入手先 <http://www.cs.toronto.edu/km/istar/>(2015/5/1)
- [6] Lawrence Chung, Brian A. Nixon, Eric Yu, John Mylopoulos: *Non-Functional Requirements in Software Engineering*, Springer(1999)
- [7] 非機能要求の見える化と確認の手段を実現する「非機能要求グレード」の公開, 入手先 <http://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html>(2015/5/8)
- [8] Takako NAKATANI, Terunobu FUJINO: Role and Owner based Business Domain Analysis, 12th Asia? Pacific Software Engineering Conference (APSEC' 05) (2005)