

論文 / 著書情報
Article / Book Information

題目(和文)	ゴール指向要求分析における形式的ゴール選択方法の構築
Title(English)	
著者(和文)	佐藤慎一
Author(English)	Shin-Ichi Sato
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第9056号, 授与年月日:2013年3月26日, 学位の種別:課程博士, 審査員:猪原 健弘
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第9056号, Conferred date:2013/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

博士論文

ゴール指向要求分析における
形式的ゴール選択方法の構築

平成 24 年度



指導教員 猪原健弘 教授

東京工業大学大学院社会理工学研究科
価値システム専攻

佐藤慎一

目次

概要	1
第1章 緒論	2
1.0 本章の概要	2
1.1 研究の背景	2
1.2 要求工学の経緯とゴール指向要求分析	4
1.3 本研究の目的	5
1.4 本論文の目標	6
1.5 本論文の主たる成果	7
1.6 本論文の構成	8
1.7 用語の定義	10
1.8 本章のまとめ	13
後注	13
第2章 既存研究	14
2.0 本章の概要	14
2.1 要求獲得	14
2.2 ゴール指向要求分析の概要	20
2.3 既存のゴール指向要求分析法	23
2.3.1 AGORA	23
2.3.1.1 貢献度の形式的定義	27
2.3.1.2 ゴール選択上のコンフリクト	27
2.3.1.3 ステークホルダ間の認識不一致	28
2.3.2 属性値に基づくゴール選択手法	29
2.3.2.1 表現定義	32
2.3.2.2 基本法則	33

2.3.2.3	ゴールグラフの簡約化	34
2.3.2.4	属性値に基づくゴール選択手法におけるゴール選択手順.....	34
2.4	本章のまとめ	45
第3章	ゴール選択基準間のコンフリクトの形式的検出方法	47
3.0	本章の概要.....	47
3.1	貢献度と顧客のニーズに関する妥当性間のコンフリクト	47
3.1.1	顧客のニーズに関する妥当性.....	47
3.1.2	コンフリクト検出指標	49
3.2	コンフリクト検出アルゴリズム.....	50
3.3	属性値の伝搬を伴うゴールグラフの簡約化.....	54
3.3.1	ゴールグラフの簡約化に伴う属性値の処理	63
3.3.2	適用	64
3.4	本章のまとめ	66
	後注.....	67
第4章	ゴール指向要求分析における形式的ゴール選択方法	68
4.0	本章の概要.....	68
4.1	ゴール選択アルゴリズム	68
4.1.1	評価	72
4.1.2	考察	74
4.2	階層分析法に基づく形式的ゴール選択方法.....	82
4.2.1	階層分析法	82
4.2.2	階層分析法に基づく形式的ゴール選択方法	84
4.2.2.1	考察	88
4.3	ゴール選択基準の拡張.....	90
4.3.1	顧客のニーズに関する妥当性の拡張	90
4.3.1.1	拡張された顧客のニーズに関する妥当性の計算手順	91
4.3.1.2	評価	94
4.3.1.3	考察	97
4.4	本章のまとめ	98
	後注.....	98
第5章	ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法	100
5.0	本章の概要.....	100
5.1	ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法.....	100
5.2	適用	103
5.3	補遺	104
5.3.1	不確実性推論の形式的ゴール選択への応用	104

5.3.1.1	エキスパートシステムにおけるゴール指向とゴール指向要求分析の差異	109
5.3.1.2	確信度に基づくゴール選択方法	110
5.3.1.3	AND/OR グラフの形式的定義	114
5.3.1.4	ゴールグラフの形式的定義	116
5.3.1.5	貢献度の形式的定義	117
5.3.1.6	確信度に基づく形式的ゴール選択方法	117
5.3.2	閉路があるゴールグラフを仮定した場合の扱い	119
5.3.2.1	閉路があるゴールグラフに対するモデル検査	120
5.3.2.2	閉路があるゴールグラフに対する <i>ccc-dfs</i> および <i>gsl-dfs</i> の構築	121
5.3.3	数値計算やシミュレーションによるゴール選択の傾向や一般的性質の導出可能性	121
5.3.3.1	ゴールグラフの深さに関連する指標の形式的定義	121
5.3.3.2	ゴールグラフの深さと増加率	122
5.4	本章のまとめ	124
	後注	124
第6章	結論	125
6.0	本章の概要	125
6.1	結果の要約	125
6.2	今後の課題	127
6.2.1	ゴール選択ゲームの構築	127
6.2.2	ゴール選択基準の精緻化	129
6.2.3	ネットワーク分析の諸概念に基づくゴールの品質特性評価	131
6.2.4	形式概念分析に基づくゴール選択基準の優先順位付け	131
6.3	本章のまとめ	134
	謝辞	135
	参考文献	136

目次

1.1	要求工学プロセス [要求工学ワーキンググループ 07, van Lamsweerde 09].....	4
1.2	非機能要求の分類 [van Lamsweerde 09].....	11
2.1	落水型ライフサイクルモデル [Royce 70].....	15
2.2	反復型ライフサイクルモデル	15
2.3	要求工学技術マップ [Tsumaki 06, Nakatani 10, 妻木 12]	18
2.4	ゴールグラフの例 [Hayashi 12]	21
2.5	ゴール指向要求分析の発展段階 [Yamamoto 06, 山本 07].....	21
2.6	満足度行列の例	26
2.7	ゴール間のコンフリクトの例 [Hayashi 12]	26
2.8	属性「コスト」およびその計算方法の定義の例 [Yamamoto 08].....	26
2.9	ゴール選択上のコンフリクトの例 [Kaiya 02]. (a), (b)は各々, コンフリクト解決 前と後を表す	28
2.10	3階層ゴールグラフの例 [斎藤 06, 山本 07]	30
2.11	簡約化後のゴールグラフ [斎藤 06, 山本 07].....	36
2.12	属性設定後のゴールグラフ [斎藤 06, 山本 07].....	38
3.1	貢献度-妥当性-コンフリクトを持つ2階層-OR分解-部分ゴールグラフの例 [佐藤 11, 佐藤 12c]	48
3.2	深さ優先探索 [Aho 83]	51
3.3	<i>ccc-dfs</i> [佐藤 12a]	51
3.4	<i>ccc</i> (図 3.3 の 6 行目の処理) [佐藤 12a]	52
3.5	<i>dcc</i> (図 3.4 の 13 行目の処理) [佐藤 12a].....	53
3.6	図 3.1 に対する貢献度-妥当性-コンフリクト検出結果 [佐藤 12a].....	54
3.7	図 2.10 に貢献度および満足度行列を振ったゴールグラフ.....	55
3.8	図 3.7 に対する貢献度-妥当性-コンフリクト検出結果	56

3.9	図 3.7 において, 2 つ以上の子ゴールを持つ 2 階層-OR 分解-部分ゴールグラフ	57
3.10	貢献度と満足度が振られた会議の設定のゴールグラフ	59
3.11	図 3.10 に対する貢献度-妥当性-コンフリクト検出結果	60
3.12	図 3.10 において, 2 つ以上の子ゴールを持つ 2 階層-OR 分解-部分ゴールグラフ	61
3.13	属性値の伝搬を伴うゴールグラフの簡約化前の多階層-部分ゴールグラフ	65
3.14	図 3.13 に対して, 属性値の伝搬を伴うゴールグラフの簡約化を適用した後の 2 階層-部分ゴールグラフ [佐藤 12c]	66
4.1	<i>gsl-dfs</i>	70
4.2	<i>gsl</i> (図 4.1 の 18 行目の処理)	71
4.3	<i>pth</i>	72
4.4	図 3.1 に対する <i>gsl-dfs</i> の適用結果	72
4.5	図 2.10 に貢献度を振ったゴールグラフ	73
4.6	図 4.5 に対する <i>gsl-dfs</i> の適用結果	74
4.7	図 4.5 の初期ゴールから要求までの経路を構成する 2 階層-部分ゴールグラフ	75
4.8	貢献度が振られた会議の設定のゴールグラフ [Hayashi 12]	77
4.9	図 4.8 に対する <i>gsl-dfs</i> の適用結果	79
4.10	図 4.8 の初期ゴールから要求までの経路を構成する 2 階層-部分ゴールグラフ	80
4.11	階層図の概念図. G は総合目的, C_j は評価基準, A_i は代替案. c_j は, 総合目的に対する評価基準の重み, a_{ji} は, 評価基準に対する代替案の評価値	83
4.12	ゴール選択の階層図	85
4.13	総合目的に対する評価基準の一对比較行列. G は, 総合目的「海外収益向上」, C_1, C_2 は各々, 評価基準「貢献度」, 「顧客のニーズに関する妥当性」を表す	86
4.14	各評価基準に対する代替案の一对比較行列. (a), (b) は各々, 評価基準「貢献度」, 「顧客のニーズに関する妥当性」に対する代替案の一对比較行列である. C_1, C_2 は各々, 評価基準「貢献度」, 「顧客のニーズに関する妥当性」を表す. A_1, A_2 は各々, 代替案「英語公用化」, 「現地通訳雇用」を表す	87
4.15	意味タグをゴール選択基準に加えた場合のゴール選択の階層図	88
4.16	総合目的に対する評価基準の一对比較行列. (a) は, 図 4.13 に基づいて最初に作成されたもの. (b) は, $C \cdot I \leq 0.1$ となるように (a) を修正したもの. G は, 総合目的「海外収益向上」, C_1, C_2, C_3 は各々, 評価基準「貢献度」, 「顧客のニーズに関する妥当性」, 「セキュリティ」を表す	89
4.17	評価基準「セキュリティ」に対する代替案の一对比較行列. C_3 は, 評価基準「セキュリティ」を表す. A_1, A_2 は各々, 代替案「英語公用化」, 「現地通訳雇用」を表す	90
4.18	適用対象の 2 階層-OR 分解-部分ゴールグラフ [新原 04, 佐藤 13]	91
4.19	提案方法の階層図 [佐藤 13]	92
4.20	満足度行列から作成される, 総合目的に対する評価基準の一对比較行列 [佐藤 13]. G は, 総合目的「すぐに登録可能」, C_1, C_2 は各々, 評価基準「利用者」, 「経営者」を表す	93

4.21	満足度行列から作成される, 各評価基準に対する代替案の一对比較行列 [佐藤 13]. (a), (b) は各々, 評価基準「利用者」, 「経営者」に対する代替案の一对比較行列である. C_1, C_2 は各々, 評価基準「利用者」, 「経営者」を表す. A_1, A_2 は各々, 代替案「本人確認しない」, 「本人確認する」を表す	94
4.22	C_2 に対する C_1 の一对比較値に対する $Cup^E(A_1)$ と $Cup^E(A_2)$ のグラフ [佐藤 13]. 実線は $Cup^E(A_1)$. 一点鎖線は $Cup^E(A_2)$	95
4.23	各評価基準に対する代替案の一对比較行列 [佐藤 13]. (b) は図 4.21 (b) と同じ行列. (a)' は, 評価基準に対する代替案の優先順位が (b) と異なる仮想的な行列	96
4.24	C_2 に対する C_1 の一对比較値に対する $Cup^E(A_1)$ と $Cup^E(A_2)$ のグラフ [佐藤 13]. 実線は $Cup^E(A_1)$. 一点鎖線は $Cup^E(A_2)$	96
5.1	ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法	101
5.2	形式的ゴール選択の準備	101
5.3	形式的ゴール選択	102
5.4	ゴール選択基準間のコンフリクト検出	102
5.5	図 3.7 中のステークホルダ間の認識不一致を持つゴールに対して, これを解決したゴールグラフ	105
5.6	図 5.5 に対して属性値の伝搬を伴うゴールグラフの簡約化を行った後のゴールグラフ ..	106
5.7	図 5.6 に対する貢献度-妥当性-コンフリクト検出結果	107
5.8	エキスパートシステムの基本構造 [飯島 93]	110
5.9	エキスパートシステムの推論に用いられる AND/OR グラフの例 [戸内 97]	110
5.10	AND/OR/COMB グラフの例 [Ishizuka 82a, Ishizuka 82b, Ishizuka 83, 石塚 83b, 石塚 83c, 石塚 85, 寺野 87]	114
5.11	AND/OR グラフの例 [大須賀 91]	115
5.12	ゴールグラフ [Hayashi 12]	118
5.13	閉路があるゴールグラフの例	121
5.14	ゴール数が 3 のゴールグラフ	123
6.1	三次元満足度行列の例	130
6.2	ゴール選択基準の優先順位付け方法の概念図	134

表目次

1.1	ソフトウェアの品質特性 (ISO/IEC 9126) [ISO 91].....	12
2.1	工程別相対欠陥修正コスト [Boehm 81]	16
2.2	図 2.12 のタスクゴールの KPI [斎藤 06, 山本 07].....	37
3.1	図 3.1 における貢献度と顧客のニーズに関する妥当性の値 [佐藤 12c]	49
3.2	図 3.9(a)~(f) の $Cov(g)$ と $Cup(g)$ の比較	58
3.3	図 3.12(a)~(i) の $Cov(g)$ と $Cup(g)$ の比較	62
3.4	「製造業務の効率化」の貢献度-妥当性-コンフリクト検出結果 [佐藤 12c].....	66
4.1	図 4.7(a)~(g) において選択される子ゴール	76
4.2	図 4.10(a)~(m) において選択される子ゴール	78
4.3	基準尺度 [Saaty 80]	84
4.4	C_2 に対する C_1 の一対比較値に対する $Cup^E(A_1)$ と $Cup^E(A_2)$ [佐藤 13].....	95
4.5	C_2 に対する C_1 の一対比較値に対する $Cup^E(A_1)$ と $Cup^E(A_2)$ [佐藤 13].....	96
5.1	図 5.6 中のすべての 2 階層-部分ゴールグラフにおいて選択される子ゴールの一覧	108
5.2	確信度の値の解釈 [薦田 97]	111

概要

本論文は、ソフトウェア工学の一分野である要求工学において精力的に研究が行われている「ゴール指向要求分析」に関する研究成果をまとめたものである。本研究の目的は、「ゴール指向要求分析に基づいて要求獲得を行う際に生じる種々のコンフリクトを考慮に入れた上で、形式的に要求を獲得するための方法論の確立」である。本論文では、この目的を達成するための具体的な目標として主目標1つと、それを達成するための副目標2つを定めている。主目標は、(1)「ゴール指向要求分析におけるコンフリクトを考慮に入れた形式的ゴール選択方法の構築」である。副目標は、(1a)「ゴール指向要求分析における形式的コンフリクト検出・解決方法の構築」と(1b)「ゴール指向要求分析における形式的ゴール選択方法の構築」である。ゴール指向要求分析に基づいて要求獲得を行う上で生じるコンフリクトは多種多様であり、生じ得るコンフリクトの全容は未だ不明である。そこで本論文では、コンフリクトの内、著者等が発見した「ゴール選択基準間のコンフリクト」に着目する。そして、ゴール選択基準間のコンフリクトに関して、副目標(1a)を達成する方法を提案する。また、本論文では、(1b)を達成するために、任意のゴールグラフに対して、特定のゴール選択基準に基づいて形式的にゴール選択が可能なアルゴリズムを提案する。そして、副目標(1a)と(1b)を達成するために構築した上記の方法を効果的に組み合わせることで、主目標(1)を達成するために、「ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法」を提案する。提案方法は、ゴール選択基準間のコンフリクトという特定のコンフリクトに関して、主目標(1)を部分的に達成するものである。

1.0 本章の概要

本章では、本論文の緒論として、次の項目について述べる。1.1 節では、本研究の背景について述べる。1.2 節では、ゴール指向要求分析を要求工学の経緯の中で概観する。1.3 節では、本研究の目的について述べる。1.4 節では、本研究の目的を達成するための具体的な目標について述べる。1.5 節では、本論文の主たる成果について述べる。1.6 節では、本論文の構成について述べる。1.7 節では、本論文で用いる用語を定義する。

1.1 研究の背景

「ゴール (Goal)」に関する研究は、古くは紀元前 4 世紀の Aristotle の時代から行われてきた。Aristotle は、「目的論」を標榜し、万物は目的を持っていると説いた。目的論は、17 世紀の Descartes に端を発する「機械論」に対置される [廣松 98]。機械論は古典力学の発展と共に膾炙し、いわゆる「Laplace の悪魔」なる概念が登場するに至るまで隆盛を極めた。さらに、19 世紀には、Darwin の進化論によって、生命は目的を持つわけではなく、結果としてそう解釈できるだけであるという考えが社会に浸透すると共に、天秤は完全に機械論に傾くかに思われた。しかし、20 世紀に入ると、Luhmann は、Maturana によって提示されたオートポイエーシスの論理をシステム論の中心的な機構に据えた第三世代システム [河本 95]、すなわち「社会システム」の構想を打ち立て、社会システム（人間活動システム）の中で目的概念が決定的に重要であることを喝破した [Luhmann 73]。また、科学哲学においても、Popper の反証主義、Hacking の新実験主義などを考慮した上で、やはり Aristotle の目的論は、認識論的に疑わしい個所は存在しないという主張もなされている [Chalmers 99]。

前世紀までの目的論の歴史を鑑みるに、事物それ自体が本質的に目的を持つか否かは未だもって定かではない。しかし、社会は自然と異なり人工的なものであるから、多様な解釈が可能である。そして、社会をシステムとして捉えた場合においては、事物を目的を持つものとして解釈することはできそうである。事実、資本主義の社会システムの中では、その構成要素である企業は、一様に営利をその目的とするという側面がある。そして、企業の構成要素である社員は、企業が営利を上げるための何らかの目的を達成するために行動している。今日では、従来人手で行われてきた作業の多くが情報シ

システムに代行されるようになった。その意味では、情報システムもまた、何らかの目的を持つ社会システムの構成要素であるといえよう。人と物を区別せず、何らかの目的を持つ主体を意味する「エージェント (agent) ^{注1)}」なる術語も、種々の学問領域で自然に用いられている。エージェントの集合もまたエージェントであるとするならば、なるほど目的を持つエージェントが成す階層構造として社会システムを捉えることができる。今日では、エージェントベースの社会シミュレーションによって、社会システムを分析する研究分野も確立されている [出口 09]。

このように、目的論は現代においても色褪せていない。むしろ、20世紀の末に上梓された、現代の企業経営における目的概念の重要性を指摘した小説『The Goal』 [Goldratt 12] が、全米で250万部という驚異的な売れ行きを記録したことから、目的概念の重要性は目的論と機械論が混淆する現代の複雑な価値基準の下で運営される社会システムにおいて、一層強く認識されるようになってきている。今後も人類は社会を形成し、その営みの中で生活していくことは確かである。そして、社会システムとゴールが互いに不可分な関係にある以上、目的論は、今後も変容・複雑化の一途をたどるであろう社会システムに付随して研究され続けられる宿命にある、人類にとって万古不易の命題であるように思われる。

昨今、社会システムの一部である情報システムの開発においても、目的概念が重要であることが認識されるようになってきた。本論文は、ソフトウェア工学の一分野である要求工学において盛んに研究が行われている「ゴール指向要求分析 (GORA: Goal-Oriented Requirements Analysis)」を対象とした諸研究の成果をまとめたものである。

要求工学は、情報システム開発における要求分析プロセスを研究の対象とする分野である。要求分析プロセスは、まず最初に、これから開発する情報システムに対する要求を顧客から獲得する「要求獲得」の段階があり、次に、獲得した要求を要求仕様書としてまとめる「要求仕様記述」の段階がある。その後、要求仕様書が顧客のニーズを満たしているかどうかを確認する「要求検証」の段階がある。要求獲得の段階で顧客の真のニーズを正確に獲得することは一般に困難であり、獲得した時点での抽象性の高い要求を要求仕様書というより具体的なモデルに変換する上で、獲得されていたはずの顧客のニーズが漏れてしまう場合もある。したがって、この要求検証の段階では、顧客の真のニーズと要求仕様書に記述された内容の間に差異が見つかることが常である。この差異に基づいて要求の変更を行うことになるため、要求の変更に対処するプロセスが恒常的に必要となる。このプロセスは「要求管理」段階と呼ばれる。どのような要求に変更すればよいかを明らかにするために、要求獲得段階から、以上のプロセスを繰り返すこともある。つまり、図1.1 [要求工学ワーキンググループ 07, van Lamsweerde 09] に示す通り、要求分析は、要求獲得、要求仕様記述、要求検証、要求管理の4つの段階を繰り返しつつ、徐々に顧客の要求をより完全なものへと発展させていく、反復型のプロセスである [佐伯 01]。

ゴール指向要求分析は、要求分析プロセスの最初の段階である要求獲得を行うための要求獲得方法の1つである。ゴール指向要求分析は、既存の要求獲得方法の内、形式的に要求獲得を行うことが可能な方法である。しかしながら、その研究は未だ途上にあり、ゴール指向要求分析を用いた形式的な要求獲得方法は確立されていない。このような背景から、著者は、ゴール指向要求分析における形式的な要求獲得方法の構築に取り組んできた。本論文は、その一定の成果をまとめたものである。具体的には、ゴール指向要求分析における形式的ゴール選択方法を構築した結果について述べる。

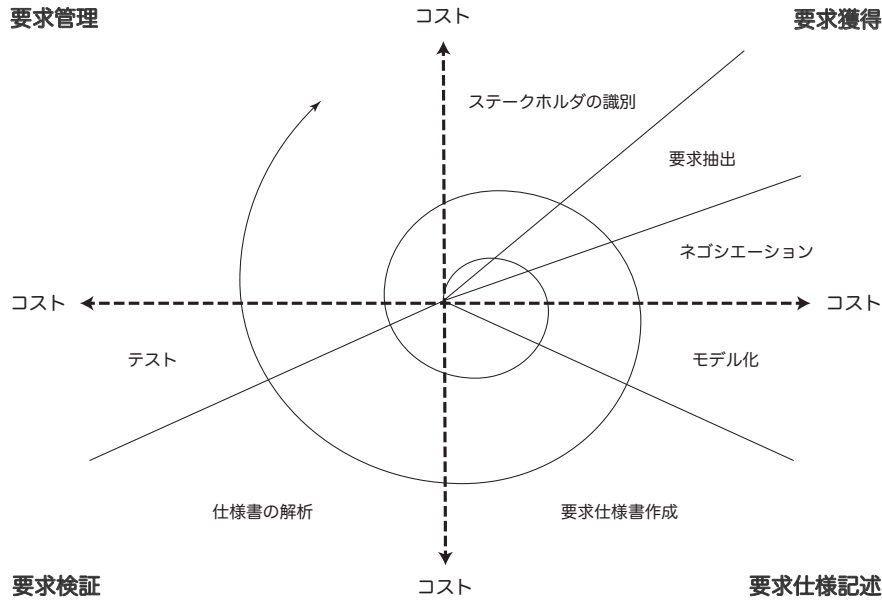


図 1.1: 要求工学プロセス [要求工学ワーキンググループ 07, van Lamswerde 09]

1.2 要求工学の経緯とゴール指向要求分析

要求工学 (RE: Requirements Engineering) は, 1970 年代から研究が始まったとされる. これは, 1968 年に提唱された「ソフトウェア危機」において, 情報システム開発プロジェクトの主な失敗の原因が, 要求の欠陥であるということが判明したことに起因する [Theyer 94]. ソフトウェア危機とは, 情報システム開発特有の困難さのために, 供給者 (1.7 節 (3) 参照) が顧客のニーズを満たすだけのソフトウェアを, 種々のコスト制約のために供給できない状況を示す言葉である. 要求工学が始まった 1970 年は, 落水 (ウォーターフォール) 型ライフサイクルモデル [Royce 70] (図 2.1) が提案された年でもある. その後, 落水型ライフサイクルモデルは, 情報システム開発の標準的な開発プロセスとなった. したがって, 要求工学は, 落水型ライフサイクルモデルの要求分析工程において, 後工程からの手戻りのない要求を定義する方法論の確立を目的とする研究分野として出発した.

その後, 1980 年代には, 落水型ライフサイクルモデルが要求変更に対応できないという問題が明らかになり, 落水型に代わり, 反復型ライフサイクルモデル (図 2.2) を奨励する動きが出てきた. 1985 年には, 当時, 落水型プロジェクトを最も行っていたとされる米国防総省 (Department of Defense: DoD) が, 情報システム開発および調達のための開発標準として「DOD-STD-2167」 [Department of Defense 85] を策定した. そこでは, 落水型が奨励されていた. 一方, 同年, 要求変更に対応できないという落水型の問題点を克服することを目的として, Boehm によって, 反復型の 1 つであるスパイラルモデル [Boehm 85] が提案された. この翌年の 1986 年には, Brooks による高名な論考『No Silver Bullet』 [Brooks 86] が発表され, 落水型に代わる新たなパラダイムとしての反復型が人口に膾炙することになった. このような情勢にあって, さらにその翌年の 1987 年には, 米国防総省は DOD-STD-2167 の内容に反して, 反復型を奨励する勧告を出した [Larman 03]. 要求工学において, ゴール指向要求分析の概念が登場した年は, ちょうどこの 1987 年にあたる [Feather 87]. 反復型は, 要求分析において誤りが含まれることを前提とするモデルであることから, ともしれば,

誤りのない要求分析を行うことを目指す要求工学が軽視されるのではないかという懸念があったものと考えられる。もしそうであるならば、ゴール指向要求分析は、反復型へのアンチテーゼとして、要求工学の重要性を改めて知らしめるために登場した方法であるという見方もできよう。

1990年代に入ると、要求工学に関する国際会議が開催されるようになった。現行の要求工学に関する最高峰の国際会議である「International Requirements Engineering Conference (RE)」は、1993年に設立された「International Symposium on Requirements Engineering (RE)」と、1994年に設立された「International Conference on Requirements Engineering (ICRE)」という、要求工学の国際会議が、2002年に統合されて誕生したという経緯がある。REに次ぐ重要な要求工学の国際会議としては、1994年に設立された「Requirements Engineering: Foundation for Software Quality (REFSQ)」が挙げられる。これら2つが、現行の要求工学の主要な国際会議である。これらの設立と共に、1990年代を端緒として、種々のゴール指向要求分析法が提案された。詳細は2.2節に譲るが、現行の代表的なゴール指向要求分析法であるKAOS[van Lamsweerde 91]と*i** framework[Yu 97]も1990年代に提案されている。この他、要求工学に関連の深い国際会議としては、年次開催される「International Conference on Software Engineering (ICSE)」, 「Conference on Advanced Information Systems Engineering (CAiSE)」, 「Asia-Pacific Software Engineering Conference (APSEC)」や、隔年開催される「Joint Meeting of The European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering (ESEC/FSE)」が挙げられる[Glinz 07]。特に近年では、2003年に要求工学において提案されてきた多様な方法を比較評価して、俯瞰的にその有用性について議論する国際会議「Comparative Evaluation in Requirements Engineering (CERE)」が設立され、要求工学の体系化への試みが行われている。2011年には、要求工学知識体系 (REBOK: Requirements Engineering Body Of Knowledge) の第1版[REBOK 11]が登場した。また、2008年に要求工学と法との関連について研究する国際会議「Requirements Engineering and Law (RELAW)」が発足するなど、要求工学の社会科学への応用を目指した研究も行われ始めている。

現在、要求工学に関する論文誌としては、「Requirements Engineering Journal (REJ)」が最も有名である。ただし、この論文誌には、研究論文のみが掲載される。実務家向けの経験論文が掲載される要求工学に関する論文誌には、「IEEE Software Magazine」がある。この他、要求工学関連の論文が掲載される論文誌には、「IEEE Transactions on Software Engineering (TSE)」, 「Information and Software Technology (IST)」, 「Software and Systems Modeling (SoSym)」, 「Information Systems」が挙げられる[Glinz 07]。

以上の国際会議・論文誌において、現在、国内外の要求工学の研究者によるゴール指向要求分析に関する研究成果が、精力的に発表されている。ゴール指向要求分析に関する既存研究に関しては、2.2節で詳細に述べる。

1.3 本研究の目的

要求工学プロセス(図1.1)における最初の段階である要求獲得は、現実と情報システムの差異やステークホルダ間の知識や文化的背景の差異、そして要求の不安定さを考慮に入れた上で行わな

ければならない複雑な作業である (2.1 節 (1)~(3) 参照). それ故, 要求獲得を行う際には, 一般に, 種々のコンフリクトが生じる. ゴール指向要求分析を用いた要求獲得は, 要求仕様記述前の要求獲得の段階で, 要求間にコンフリクトを生じさせる原因となる種々のコンフリクトを分析する上で有用な方法であり, このコンフリクト分析は, ゴール指向要求分析の主要な関心事の1つである [Yu 98, Lapouchnian 05]. しかしながら, ゴール指向要求分析において, 種々のコンフリクトを考慮に入れた上で要求獲得するための形式的な方法は, 著者の知る限り未だ確立していない. したがって, ゴール指向要求分析を用いた要求獲得を行う上で, 種々のコンフリクトを分析する指標が確立されるべきである.

そこで, 本研究は, 次を目的とする.

本研究の目的

ゴール指向要求分析に基づいて要求獲得を行う際に生じる種々のコンフリクトを考慮に入れた上で, 形式的に要求を獲得するための方法論を確立すること.

1.4 本論文の目標

1.3 節で述べた目的を達成するための具体的な目標が, 本論文の目標である. 本論文では, 次に示す主目標1つと, それを達成するための副目標2つの計3つの目標を掲げる.

本論文の主目標

(1) ゴール指向要求分析におけるコンフリクトを考慮に入れた形式的ゴール選択方法の構築.

本論文の副目標

(1a) ゴール指向要求分析における形式的コンフリクト検出・解決方法の構築.

(1b) ゴール指向要求分析における形式的ゴール選択方法の構築.

副目標 (1a) に関する内容は, 本論文の3章に対応している. 3章では, ゴール選択基準間のコンフリクト (2.4 節参照) の1つである, 貢献度-妥当性-コンフリクト (3.1 節参照) を取り上げる. さらに, ゴールグラフ上でこのコンフリクトを持つすべてのゴールを検出可能なアルゴリズムを構築した結果について述べる (3.2 節参照). このアルゴリズムは, 貢献度-妥当性-コンフリクト以外にも, コンフリクトが生じる任意のゴール選択基準の対を検出可能な汎用性を持つ.

副目標 (1b) に関する内容は, 本論文の4章に対応している. 本論文では, ゴール指向要求分析における広義のゴール選択 (1.7 節 (10) 参照) の形式的な定義を与える (4.1 節参照). そして, この定義に基づいて, 任意のゴール選択基準を用いて広義のゴール選択が可能なアルゴリズムを提案する (4.1 節参照).

最後に, 副目標 (1a) と (1b) に関する内容を効果的に組み合わせることで, ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法を提案する. この方法は, 本論文の主目標 (1) を部分的に達成するものである. この方法は5章に対応している.

1.5 本論文の主たる成果

本論文の主たる成果は、次の4つである。

本論文の主たる成果

- (1) ゴール選択基準間のコンフリクトを検出するアルゴリズムを構築 (3.2 節参照)。
- (2) 属性値の伝搬を伴うゴールグラフの簡約化方法を構築 (3.3 節参照)。
- (3) ゴール指向要求分析におけるゴール選択アルゴリズムを構築 (4.1 節参照)。
- (4) ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法を構築 (5.1 節参照)。

(1)は、ゴールグラフ上のすべての2階層-OR分解-部分ゴールグラフ (1.7 節 (9b) 参照) を対象として、ゴール選択基準間のコンフリクト (2.4 節参照) を持つすべてのゴールを検出するアルゴリズムである。実用規模のゴールグラフを想定した場合、ゴール選択基準間のコンフリクトの有無を手作業で確認することは現実的ではない。さらに、ゴールグラフは一度作成したら二度と変更されない類のものではなく、要求獲得の定められた期間内でなるべく改善されるべきである [Hayashi 12]。また、要求変更は、情報システム開発のライフサイクルの任意の工程で起こり得るため、その都度ゴールグラフの修正を行う可能性がある。このように、頻繁な変更が想定されるゴールグラフの作成作業においては、手作業でこのコンフリクトを検出することは、やはり現実的ではない。こうした人手コスト過多の問題に加えて、より重要な問題として、人手による検証は、人的エラーが混入する可能性を排除できないことが挙げられる。(1)のアルゴリズムを用いた自動検証を行うことで、これらのリスクを軽減することが可能である。

(2)は、ゴール g のゴール選択基準間のコンフリクトの有無を検出する指標 $Ccc(g)$ (式 (3.3)) を、任意の2階層-部分ゴールグラフ (1.7 節 (9) 参照) に対して適用可能にするものである。 $Ccc(g)$ は、ゴール選択基準間のコンフリクトの1つである、貢献度-妥当性-コンフリクト (3.1 節参照) を検出するために定義された指標である [佐藤 11]。しかし、 $Ccc(g)$ は、2階層-OR分解-部分ゴールグラフに対して定義された指標であり、2階層-部分ゴールグラフに対しては、このコンフリクトを検出することができない。ゴールグラフ上の2階層-部分ゴールグラフは、2階層-AND分解-部分ゴールグラフ (1.7 節 (9a) 参照)、2階層-OR分解-部分ゴールグラフ (1.7 節 (9b) 参照)、そして2階層-AND/OR分解-部分ゴールグラフ (1.7 節 (9c) 参照) の3通りの部分ゴールグラフの総称である (1.7 節 (9) 参照)。ゴールグラフ上の2階層の部分ゴールグラフは、この3通り以外存在しない。(2)は、任意の2階層-部分ゴールグラフを2階層-OR分解-部分ゴールグラフに変型することが可能な方法である。それ故、(2)を適用した後に $Ccc(g)$ を適用することによって、任意の2階層-部分ゴールグラフの親ゴールに対して、 $Ccc(g)$ を用いたゴール選択基準間のコンフリクト検出が可能となる。

(3)は、任意のゴールグラフに対して、形式的に広義のゴール選択 (1.7 節 (10) 参照) を行うことが可能なアルゴリズムである。著者が知る限り、ゴール指向要求分析における広義のゴール選択の形式的な定義は存在しない。そこで、本論文では、ゴール指向要求分析における広義のゴール選択の

形式的な定義を与える (4.1 節参照). その上で, この定義に基づいて形式的に広義のゴール選択を行うためのアルゴリズムを提案する. このアルゴリズムは, 特定のゴール選択基準において最適な要求を獲得するものであり, 任意のゴール選択基準が設定可能である.

(4) は, ゴール選択基準間のコンフリクトを考慮に入れた上で形式的に広義のゴール選択を行うことが可能な方法である. この方法は, 以上の成果 ((1), (2), (3)) を効果的に組み合わせることで実現される.

1.6 本論文の構成

本論文の構成は以下の通りである.

2 章では, 3, 4, 5 章で述べる内容の前提知識となる既存研究について述べる. 本論文では一貫して, ゴール指向要求分析に基づく要求獲得上の問題を扱っている. そのため, 2.1 節では, 「要求獲得」の全体像を述べ, 続く 2.2 節では, 「ゴール指向要求分析」の概要について述べる. 2.3 節では, 3, 4, 5 章で述べる内容に関連するゴール指向要求分析法として, 既存の 2 つの属性付きゴール指向要求分析法「AGORA」[Kaiya 02] と「属性値に基づくゴール選択手法」[斎藤 06, 山本 07] について詳述する. ただし, 本論文では, 既存研究にはない次の点についても述べている. 2.3.1.1 節において, グラフ理論に基づいて貢献度 [Kaiya 02] (2.3.1 節 (属性-1) 参照) の形式的定義を行った. 貢献度は, 既存の多くのゴール指向要求分析法に備わっている, ゴール指向要求分析における代表的な属性である. しかしながら, 既存の貢献度は, 当該ゴールの初期ゴール (2.2 節参照) に対する貢献度と親ゴール (2.2 節参照) に対する貢献度の関連が明確ではなかった. 本論文では, この関連をグラフ理論に基づいて厳密に定義した (式 (2.1)). 2.3.2 節において, 属性値に基づくゴール選択手法を形式的に再構築した. 属性値に基づくゴール選択手法は, ビジネスプロセスの動的な変化に伴って, 広義のゴール選択 (1.7 節 (10) 参照) 結果を変更する枠組みを持つ画期的なゴール指向要求分析法である. しかしながら, 属性値に基づくゴール選択手法が提案された文献 [斎藤 06, 山本 07] では, 集合論の記法上の誤りや曖昧な点が散見されるため, 形式的な分析が困難であった. 本論文では, 主に集合論に基づいてなるべく形式的な分析が行えるようにいくつかの新たな定義を設け, 手法全体を再構築した. これにより, 属性値に基づくゴール選択手法は, 4.1 節で提案されるゴール選択アルゴリズム *gsl-dfs* (図 4.1) とは異なるアプローチに基づく広義のゴール選択方法であることが明確になる.

3 章では, 異なる 2 つのゴール選択基準に対して, ゴール選択基準間のコンフリクトを定義する. そして, ゴール選択基準間のコンフリクトの具体例として, 3.1 節で貢献度と顧客のニーズに関する妥当性 (3.1.1 節参照) という 2 つのゴール選択基準の間にコンフリクトがあることを例示する. 本論文では, ゴール g がゴール選択基準間のコンフリクトを持つか否かを形式的に検出する指標 $Dip(g)$ (式 (3.2)) を新たに定義する. また, ゴール g の貢献度を見積もる指標 $Cov(g)$ (式 (2.2)) とゴール g の顧客のニーズに関する妥当性を見積もる指標 $Cup(g)$ (式 (3.1)) を各々定義し, 各々のゴール選択基準値を定量的に比較できるようにする. そして, ゴール選択基準として貢献度と顧客のニーズに関する妥当性を設定した場合の $Dip(g)$ を特に, $Ccc(g)$ (式 (3.3)) と定義する. $Ccc(g)$ を用いることで, ゴールグラフ上の任意の 2 階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) の親ゴール g が, 「貢献度と顧客のニーズに関する妥当性の間のコンフリクト」(「貢献度-妥当性-コンフリクト」)

を持つか否かを形式的に検出することが可能になる。3.2節では、ゴールグラフ上の貢献度-妥当性-コンフリクトを持つ2階層-OR分解-部分ゴールグラフのすべての親ゴールを検出するアルゴリズム *ccc-dfs* (図3.3) を提案する。3.3節では、ゴールグラフ上の任意の2階層-部分ゴールグラフ (1.7節(9)参照) に対して、貢献度-妥当性-コンフリクトを検出する方法「属性値の伝搬を伴うゴールグラフの簡約化」を提案する。属性値の伝搬を伴うゴールグラフの簡約化は、AND分解で接続された子ゴールとOR分解で接続された子ゴールを共に持つような親ゴール g に対しても、 $Ccc(g)$ を用いて、貢献度-妥当性-コンフリクトを形式的に検出することを可能にする。

4章では、ゴール指向要求分析における「ゴール選択アルゴリズム」を提案する。4.1節では、ゴール指向要求分析における広義のゴール選択の形式的定義 (式(4.1)) を行い、その定義に基づいてゴール選択アルゴリズム *gsl-dfs* を提案する。このアルゴリズムは、子ゴールの親ゴールに対するゴール選択基準値を定める必要がある。しかし、ゴール選択基準間のコンフリクトが存在する場合は、単一のゴール選択基準では形式的に狭義のゴール選択 (1.7節(10)参照) ができない。そのため、ゴール選択基準間のコンフリクトが存在するゴールに対して、形式的にゴール選択するための方法が確立されなければならない。4.2節では、階層分析法 [Saaty 80] に基づいてこの方法を構築した結果について述べる。この方法は、「階層分析法に基づく形式的ゴール選択方法」と呼ばれる。しかし、この方法は階層分析法の評価基準としてゴール選択基準を設定する。そのため、ゴール選択基準それぞれ自体に妥当性があることを前提としている。そこで、ゴール選択基準の妥当性を保証するために、ゴール選択基準はなるべく拡張されるべきである。4.3節では、ゴール選択基準のうち、特に、顧客のニーズに関する妥当性に関して、4.3.1節でこれを拡張する方法について提案する。具体的には、従来の顧客のニーズに関する妥当性を見積もる指標 $Cup(g)$ が、当該ゴールに対して振られた「満足度」 [Kaiya 02] (2.3.1節(属性-2)参照) のみから定義されるのに対して、当該ゴールの満足度だけでなく、当該ゴールとその親ゴールとの関係の中で顧客のニーズに関する妥当性を拡張するものである。拡張された $Cup(g)$ は $Cup^E(g)$ と呼ばれる。

5章では、ゴール選択基準間のコンフリクトを考慮に入れたゴール選択を行う上で問題となる種々のコンフリクトを考慮し、なるべく解決した上で形式的に広義のゴール選択を行う方法を提案する。具体的には、ゴール選択基準間のコンフリクトとしては、貢献度-妥当性-コンフリクトを取り上げる。また、ゴール選択基準間のコンフリクトを行う上で問題となるコンフリクトとしては、「ゴール選択上のコンフリクト」 (2.3.1.2節参照) と「ステークホルダ間の認識不一致」 (2.3.1.3節参照) の2つを考慮する。前者は、初期ゴールの達成を阻害するゴールが選択されることを予防するための措置である。後者は、顧客のニーズに関する妥当性が満足度行列から算出される (3.1.1節参照) ことから、満足度行列の品質を保証するための措置である。そして、これらのコンフリクトを解決した上で、ゴールグラフ上のAND分解とOR分解を共に持つゴールに対しては、属性値の伝搬を伴うゴールグラフの簡約化を用いて、 $Ccc(g)$ のみで、ゴールグラフ上のすべてのゴールに対して、貢献度-妥当性-コンフリクトを検出可能にする。その上で、*ccc-dfs* を用いて、ゴールグラフ上の貢献度-妥当性-コンフリクトを持つすべてのゴールを検出する。検出された貢献度-妥当性-コンフリクトを持つゴールに対しては、階層分析法に基づく形式的ゴール選択方法を用いてゴール選択を行う。貢献度-妥当性-コンフリクトを持たないゴールに対しては、すべてのゴール選択基準で最大のゴール選択基準値を

持つ子ゴールを選択する。広義のゴール選択の形式的定義から、すべてのゴールに対してゴール選択が完了した時点で広義のゴール選択が完了する。5.1節では、この方法について、フローチャートを用いて詳細に述べる。5.2節では、既存の例題に対して提案方法を適用し、適切に機能することを確認する。5.3節では、本論文の補遺を述べる。5.3.1節では、不確実性推論の形式的ゴール選択への応用について述べる。不確実性推論は、主にエキスパートシステムにおける推論エンジンにおいて用いられる。そして、不確実性推論は、ゴール指向要求分析と同様に、AND/OR グラフに基づくゴール指向の分析がなされる。そこで、5.3.1.1節では、エキスパートシステムのゴール指向とゴール指向要求分析の差異について述べる。ところで、不確実性推論は、なんらかの確実性測度に基づいて行われる。5.3.1.2節では、代表的なエキスパートシステムである MYCIN の推論エンジンで採用されている確実性測度である「確信度」について詳述する。また、提案方法を既存の例題に対して適用し、適切に機能することを確認する。5.3.1.3節では、ハイパーグラフとして定義される AND/OR グラフの形式的定義について述べる。5.3.1.4節では、5.3.1.3節の内容に立脚して、ハイパーグラフとしてゴールグラフを形式的に定義する。5.3.1.5節では、5.3.1.4節の内容に立脚して、貢献度の形式的定義を与える。5.3.1.6節では、5.3.1.1節～5.3.1.5節の内容に基づいて、確信度に基づく形式的ゴール選択方法を構築した結果について述べる。また、提案方法を既存の例題に適用し、提案方法が適切に機能することを確認する。5.3.2節では、閉路があるゴールグラフを仮定した場合の取り扱いについて述べる。5.3.2.1節では、5.3.1.4節の内容に基づいて、ゴールグラフに対するモデル検査の概要を述べる。5.3.2.2節では、*ccc-dfs* と *gsl-dfs* を閉路があるゴールグラフに対しても適用可能なアルゴリズムとして拡張することの展望について述べる。5.3.3節では、数値計算やシミュレーションによるゴール選択の傾向や一般的性質の導出可能性について述べる。5.3.3.1節では、ゴールグラフの深さに関連する指標を4つ定義する。その上で、5.3.3.2節では、シミュレーションを行うことによって明らかにされるべきゴール選択上重要となるゴールグラフの一般的性質として、ゴールグラフの深さと増加率の間に関連性があることを指摘し、これについて論じる。

6章では、まとめと今後の課題を述べる。6.1節では、本論文の総括として、各章節の内容のまとめを述べる。6.2節では、本論文の補遺として、今後の課題4点について述べる。6.2.1節では、協力ゲームの枠組みにおける特性関数形ゲームに基づいて「ゴール選択ゲーム」を構築することの展望について述べる。6.2.2節では、統計学の知見に基づいて、ゴール選択基準値を見積もる指標を精緻化することの展望について述べる。6.2.3節では、ネットワーク分析の中心性などの諸概念に基づいて「ゴールの品質特性」[Kaiya 02, 鵜飼 10]の評価方法を構築することの展望について述べる。6.2.4節では、形式概念分析[Ganter 99, 鈴木 07]に基づいてゴール選択基準の優先順位付けを形式的に行う方法の展望について述べる。

1.7 用語の定義

- (1) **利用者**: 開発された情報システムを使う人々 [大西 02].
- (2) **顧客**: 情報システムを発注する際に、お金を出す立場の人々。利用者と顧客は同じ人々である場合もあるし、異なる人々である場合もある [大西 02].

- (3) **供給者**: 顧客のために情報システムを開発する人々 [大西 02].
- (4) **ステークホルダ**: 要求分析に関わるすべての人. 利用者, 顧客および供給者はすべてステークホルダに含まれる [大西 02]. 要求分析には, 通常, 利用者, 経営者, 供給者が参加する [玉井 04]. 経営者は顧客に該当するステークホルダである.
- (5) **ニーズ**: 要求という形で具体化される前の期待や要望 [大西 09].
- (6) **要求**: 開発する情報システムに対して, そのステークホルダが明示的あるいは潜在的に持つニーズ [大西 02, 玉井 04]. 要求分析は, そのような要求を獲得し, 最終的には要求仕様書としてまとめる工程である. 情報システム開発においては, ステークホルダの要求は最初は抽象的であるが, 次第に詳細化されて要求仕様書となる [Sommerville 97]. 要求には, 「機能要求」と「非機能要求」の2種類がある. 前者は情報システムを構成する要素のうち, 特に, ソフトウェアが何をするかについて表している要求である [大西 02]. 後者は, 機能要求以外の要求である [大西 02]. 換言すれば, 情報システムの動作により実現すべき要求が機能要求であり, 機能要求を実現する上での制約条件が非機能要求である [田中 08]. 非機能要求には, 性能, 使いやすさ, 安全性, 保守性, 可搬性などがあり, 組織文化や法律から要請されるものも, 非機能要求に含まれる [玉井 04]. 要求分析段階では, 機能要求と非機能要求を区別して考慮し, 要求間のトレードオフの関係 (コンフリクト) を明確にしておくことが重要である [玉井 04]. ただし, 機能要求と非機能要求のいずれに属するのか明確に区別することが困難な要求も存在する [Jackson 00]. また, 非機能要求はしばしば機能要求に分解され得る [Loucopoulos 95, Sommerville 97]. 非機能要求の一覧 [van Lamsweerde 09] を図 1.2 に示す. ソフトウェアの品質に関する国際標準規格 ISO/IEC 9126 [ISO 91] (表 1.1) で定義されている6つの品質特性と副特性を非機能要求として定める場合もある.

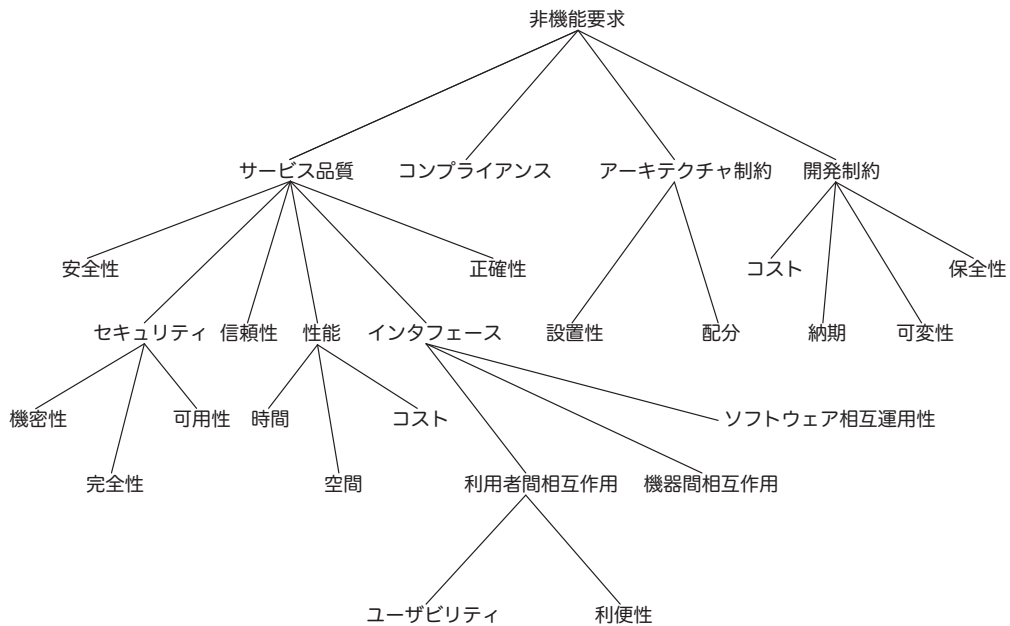


図 1.2: 非機能要求の分類 [van Lamsweerde 09]

表 1.1: ソフトウェアの品質特性 (ISO/IEC 9126) [ISO 91]

品質特性	説明	副特性
機能性	必要な機能が実装されているか	合目的性, 正確性, 相互運用性, セキュリティ, 標準適合性
信頼性	機能が正常に動作し続けるか	成熟性, 障害許容性, 回復性, 標準適合性
使用性	利用者にとって使いやすいか	理解性, 習得性, 操作性, 魅力性, 標準適合性
効率性	目的達成のために使用する資源は適切か	時間的効率性, 資源効率性, 標準適合性
保守性	改訂作業に必要な労力はないか	解析性, 変更性, 安定性, 試験性, 標準適合性
可搬性	他の環境へ移しやすいか	順応性, 設置性, 共存性, 置換性, 規格準拠性

(7) **妥当性:** 情報システム開発においては, V&V が重要であるとされる [Boehm 81]. 前者の V は「検証 (Verification)」であり, 後者の V は「妥当性確認 (Validation)」である. ISO 9001 規格 [ISO 08] によれば, 検証とは, 生成された成果物とその前段で確立された規定事項を満たしているかどうかを確認することであり, 妥当性確認とは, 最終成果物が実際に顧客や利用者のニーズを満足させる能力を備えているかどうかを確認することとされる. ある要求の必要度合いは, 顧客や利用者しか判断できないことが多く, 妥当性は顧客や利用者から直接確認してもらうのが一般的である [大西 02]. こうした背景から, 「要求の妥当性確認」は, 各要求事項がステークホルダのニーズに合致しているかどうかを確認すること [大西 09] とされる. したがって, 本論文では「妥当性」という用語を, 「ステークホルダのニーズに合致しているかどうか」という意味で用いる.

(8) **ダミーゴール:** 存在しない架空のゴール. 任意のゴールの子ゴールである.

(9) **部分ゴールグラフ:** ゴールグラフの一部を構成するゴールグラフ. 本論文では, ゴールグラフ上の1つの親ゴールとそのすべての子ゴールからなる部分ゴールグラフを「2階層-部分ゴールグラフ」と呼ぶ. ゴールグラフ上の任意のゴールは, ダミーゴールを子ゴールに持つため, 2階層-部分ゴールグラフの親ゴールである. 2階層-部分ゴールグラフは, 次の3種類 ((9a)~(9c)) に分割される.

(9a) **2階層-AND分解-部分ゴールグラフ**

親ゴールがすべての子ゴールに対して AND 分解で接続している 2階層-部分ゴールグラフ.

(9b) **2階層-OR分解-部分ゴールグラフ**

親ゴールがすべての子ゴールに対して OR 分解で接続している 2階層-部分ゴールグラフ. 親ゴールが子ゴールを1つしか持たない 2階層-部分ゴールグラフは, その子ゴールとダミーゴールとの 2階層-OR分解-部分ゴールグラフとみなしてよい.

(9c) 2階層-AND/OR 分解-部分ゴールグラフ

(9a), (9b)以外の2階層-部分ゴールグラフ. すなわち, 親ゴールが子ゴールに対して, AND分解で接続している場合もあれば, ダミーゴール以外の子ゴールとOR分解で接続している場合もある2階層-部分ゴールグラフ.

2階層-部分ゴールグラフ以外の任意の部分ゴールグラフを「多階層-部分ゴールグラフ」と呼ぶ.

- (10) **ゴール選択**: 「狭義のゴール選択」[佐藤 11]と, 「広義のゴール選択」[佐藤 11]の総称. 狭義か広義かを特に断らない場合, いずれか一方, または両方を指す. 狭義のゴール選択とは, 2階層-OR分解-部分ゴールグラフにおいて, 親ゴールの達成のために選択される少なくとも1つの子ゴールを決定すること. 広義のゴール選択とは, ゴールグラフにおいて, 初期ゴールを達成するために選択される少なくとも1つの最終ゴールを決定すること.

1.8 本章のまとめ

- 1.1 節では, 本研究の背景について述べた.
- 1.2 節では, ゴール指向要求分析が現れた経緯を, 要求工学の経緯の中で概観した.
- 1.3 節では, 本研究の目的について述べた.
- 1.4 節では, 本研究の目的を達成するための具体的な目標について述べた.
- 1.5 節では, 本論文の主たる成果について述べた.
- 1.6 節では, 本論文の構成について述べた.
- 1.7 節では, 本論文で用いる用語を定義した.

後注

- ^{注1)} エージェントを単に「認識し行動する主体」と定義し, 目的を持つエージェントを特に, 「合理的) エージェント (Rational Agent)」と呼んで区別する立場もある [Russell 10].

既存研究

2.0 本章の概要

本章では、3, 4, 5章で述べる内容の前提知識となる既存研究について述べる。本論文では一貫して、ゴール指向要求分析に基づく要求獲得上の問題を扱っている。そのため、2.1節では、「要求獲得」の全体像を述べ、続く2.2節では、「ゴール指向要求分析」の概要について述べる。2.3節では、3, 4, 5章で述べる内容に関連するゴール指向要求分析法として既存の2つのゴール指向要求分析法について詳述する。

2.1 要求獲得

情報システムの要求分析から仕様設計、実装、テスト、そして納品後の運用保守に至るまでの一連の過程は「ライフサイクル」と呼ばれる。システム思考[von Bertalanffy 69, Checkland 81, Checkland 90, Klir 91, Checkland 97]によって、ライフサイクルを1つのシステムとして捉えた場合の全体像は、「ライフサイクルモデル」と呼ばれる。情報システム開発は何らかのライフサイクルモデルに基づいて行われる[玉井 04]。既存のライフサイクルモデルは、「落水型（ウォーターフォール）ライフサイクルモデル」[Royce 70]（図2.1）と「反復型ライフサイクルモデル」（図2.2）に分割される。反復型はさらに、「進化型」と「漸増型（インクリメンタル）」の2つに分割される。進化型は、反復ごとにプロトタイプを1から作り直す。漸増型は、反復ごとにプロトタイプに機能を追加していく。進化型と漸増型のいずれの場合も、最終的なプロトタイプが納品される情報システムとなる。一般に反復型という場合、反復型かつ漸増型の「反復漸増型」を指す。そのため、本論文でもこの慣例を踏襲する。落水型は、情報システム開発のライフサイクルを明確にいくつかの「工程」に区切り、各工程を順番に開発していくモデルである。落水型は、前工程への「手戻り」を許さないため、プロジェクト管理がしやすいという利点がある。しかしその一方で、各工程を一度しか行わないため、前工程へのフィードバックがまったくなく、「要求変更」に柔軟に対応することができず、顧客のニーズを満たす情報システムの開発を行うことが困難であるという欠点がある。それ故、各々の工程において、V&V（1.7節（7）参照）が十全に行われなければならない[Boehm 81]。

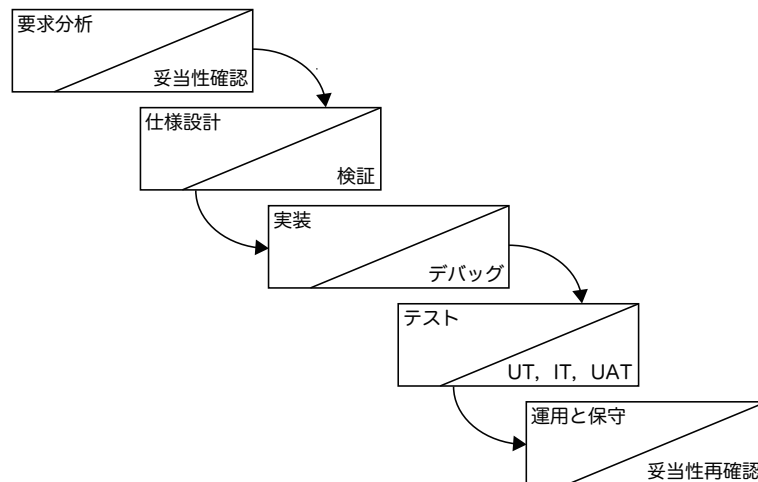


図 2.1: 落水型ライフサイクルモデル [Royce 70]

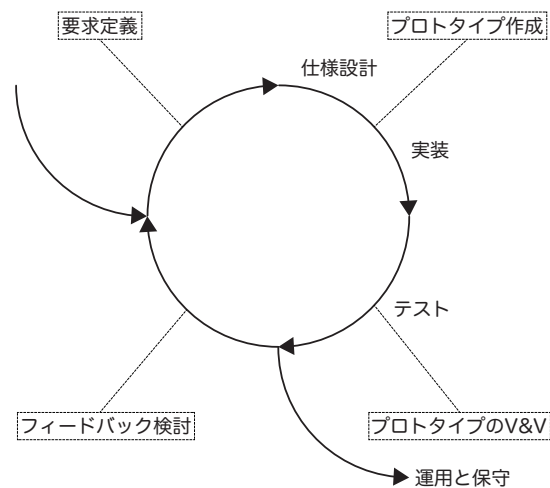


図 2.2: 反復型ライフサイクルモデル

落水型で開発が行われる場合、その定義に反して、ほぼ確実に手戻りが発生することが大きな問題となっている。手戻りは、後工程で発生した欠陥が、それ以前の工程における欠陥に起因することが判明した場合に発生する。ある工程で発生した欠陥が、当該工程で修正されずに後工程に移行した場合の欠陥修正コストは、後工程になるほど急激に増加することが知られている [Boehm 81] (表 2.1)。それ故、手戻りが発生した際の欠陥修正コストをなるべく最小限に抑えるために、なるべく早い工程で、欠陥を発見し修正しておくことが重要となる。

しかし、落水型の最初の工程である要求分析工程において獲得された要求が、顧客のニーズを満たしているかどうかの妥当性確認 (1.7 節 (7) 参照) は、システム開発後の受け入れテスト (UAT: User Acceptance Test) の段階で行われる事項であり、この時点で要求分析工程への手戻りが発生した場合、欠陥修正コストが膨大となるため、プロジェクトの失敗を意味する [Boehm 81]。そのため、落水型では、要求分析工程において、あらかじめ要求変更が発生しないように、顧客のニーズを的確に反映した要求を定めることが重要となる。

要求変更に対応するために、落水型に代わる新たなライフサイクルモデルとして提唱され

表 2.1: 工程別相対欠陥修正コスト [Boehm 81]

工程	相対欠陥修正コスト
要求分析	0.1~0.2
仕様設計	0.5
実装	1
テスト	2~5
運用と保守	20

たのが、「反復型ライフサイクルモデル」である。反復型に分類される具体的なライフサイクルモデルとしては、スパイラルモデル [Boehm 85] や種々のアジャイル (Agile) 開発モデル [Gilb 77, Gilb 88, Jacobson 99, Schwaber 01, Kruchten 03, Beck 04] が挙げられる [Larman 03]。反復型は、要求分析からテストまでのライフサイクルを繰り返すことで、フィードバックを反映させながら開発を進める。しかし、要求分析が収束するという前提があり、この前提と要求変更に対応することの間にはトレードオフの関係がある。要求分析が収束せず発散する場合、開発期間や開発コストの超過を招き、結果としてプロジェクトの失敗を招くことになる。そのため、反復型であってもやはり、要求分析を収束させるために、なるべく要求変更が発生しないよう顧客のニーズを的確に反映した要求を定めることが重要となる。

このように、落水型、反復型のいずれのライフサイクルモデルにおいても要求分析は存在し、定義された要求の品質が、プロジェクトの成功を左右する。The Standish Group によれば、2000 年から 2008 年までの情報システム開発プロジェクトにおいて、失敗または問題のあるプロジェクトは、全体の $\frac{2}{3}$ 以上であることが報告されている [The Standish Group 09]。その原因としては、全体の $\frac{1}{3}$ 程度が要求分析の問題であるとされている [The Standish Group 95]。また、要求通りに情報システムが構築された場合であっても、実際に使われている機能は全体のわずか $\frac{1}{3}$ 程度であることが報告されている [The Standish Group 95]。これらの結果は、情報システム開発プロジェクトにおける要求分析の重要性を裏づけるものであり、かつ、現状の多くのプロジェクトでは、顧客のニーズを十分に反映した要求分析ができていないことを示すものである。

要求分析は、主に、(1) 顧客から要求を獲得する「要求獲得」と、(2) 獲得した要求を基に要求仕様書を記述する「要求仕様記述」からなり、前者を行った後に後者を行う [玉井 04] (図 1.1)。したがって、要求分析の成果物とは、この要求仕様書に他ならない。それ故に、要求の品質とは要求仕様書の品質ということになる。前述の通り、落水型ライフサイクルモデルにおける欠陥修正コストは、各工程を経るごとに急激に増加することが知られており、いわゆるデスマーチプロジェクトを軽減するためには、なるべく要求分析工程において、その成果物である要求仕様書の品質を高める必要がある。要求仕様書は、要求獲得作業において獲得された要求を基に記述されるため、要求獲得を成功させることが、品質の高い要求仕様書を作成する上で必須の条件となる。

要求獲得は次のように定義される [Chastek 91]。

要求獲得の定義 [Chastek 91]

要求獲得とは、共同体の制約に合致する要求を定義するために、ニーズを発見し、共同体間の

相違を埋めるプロセスである。

より詳細には、「個々のステークホルダのニーズを掘り起こし、それらを、彼らが属している共同体のニーズへと昇華させ、もし、異なった共同体同士の間にはニーズの違いがあれば、それを調整するためのプロセス」[大西 09]が要求獲得である。

要求獲得を行う際には、次のことを考慮する必要がある。

要求獲得において考慮すべきこと [大西 09]

通常、人は、たとえ同じ組織に属していても、それぞれ独自の考え方を生きている。受けた教育や文化的背景の違いによって、価値観や行動基準が異なるだろう。属する組織での立場の違いによっても、利害関係が異なることがある。同じ1つの共同体であっても、その中には、異なったニーズを持った複数のステークホルダが存在している。しかし、要求は、個々のステークホルダの個人的な要求ではないから、彼らが属している共同体の要求へとまとめ上げなければならない。1つの共同体としてのニーズが集約できても、複数の共同体がその問題に関与している場合には、異なった目標と異なった価値基準を持った共同体同士の間で、ニーズの衝突が発生するかもしれない。相反する要求を1つの情報システム上で実現することは困難であり、ニーズの違いを何らかの形で解消しなければならない。

要求獲得は、要求分析において最も難しい作業であり、その難しさは、次の3つに原因があるとされる [大西 09]。

要求獲得を困難にする問題 [大西 09]

(1) 現実と情報システムの差異

現実における要求を、情報システムの要求として獲得しなければならない。獲得された情報システムの要求は、なるべく現実の要求と合致している必要がある。そのため、現実と情報システムモデルを架橋する要求獲得方法を用いる必要がある。

(2) ステークホルダ間の知識や文化的背景の差異

要求獲得は、顧客に対して、供給者が行う。しかし、顧客と供給者では、使用する用語、知識、思考法、行動様式などが異なっていることが多い。同じ用語が異なった意味で用いられている場合もある。また、立場上の差異に基づく利害関係も考慮されなければならない。さらに、同じ立場であっても地位や役割が異なればニーズが異なり得るため、ステークホルダ間のニーズを適切に反映することで、ステークホルダ間の合意を形成する機構を備えた要求獲得方法を用いる必要がある。

(3) 要求の不安定さ

落水型ライフサイクルモデル (図 2.1) において、要求分析工程への手戻りはどの工程においても起こり得る。この原因のほとんどは、顧客の意図の変更に直接または間接に起因する。このことを要求の不安定さ (Volatility) といい、情報システム開発のプロジェクト管理上の大きな問題であるとされている [玉井 04]。このように、要求獲得は、一度行えば二

度と行わなくてよい類のものではない。ある時点では最適な要求であっても、変更後の要求は、他の要求とコンフリクトを持つものであるかもしれない。したがって、要求変更に対応できるような要求獲得方法を用いる必要がある。

要求獲得には、以上で述べた通り、考慮されるべき問題が多岐に渡るため、要求獲得の方法も工学的な方法に限らず、多様な方法が必要となる。具体的には、ソフトウェア工学はもちろん、システム工学をはじめとする工学的な方法だけではなく、民族誌学、認知心理学、人類学、社会学、言語学などの社会科学における方法も重要であり、これらを融合させた学際的な方法が必要となる [Nuseibeh 00]。

既存の要求獲得方法には、図 2.3 のようなものがある [Tsumaki 06, Nakatani 10, 妻木 12]。この図は、要求獲得方法を、「操作特性」と「領域特性」の2つの観点から分類したものである。横

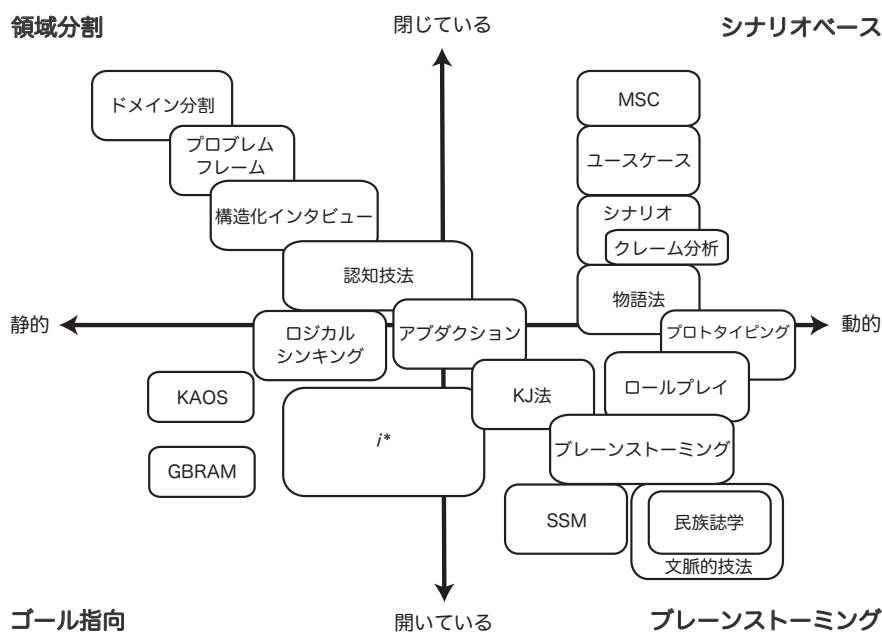


図 2.3: 要求工学技術マップ [Tsumaki 06, Nakatani 10, 妻木 12]

軸は、操作特性に対応しており、静的なほど形式的な方法であり、動的なほど、非形式的（適用する上で高度な発想を必要とする）な方法である。縦軸は、領域特性に対応しており、閉じているほど既知の安定した領域であり、開いているほど、未知で不安定な領域である。

第1象限は「シナリオベース」と呼ばれる [大西 09]。「シナリオ」とは、ある状況に限定した情報システムの具体的な利用例 [大西 02] である。現在、情報システム開発の実プロジェクトにおいては、オブジェクト指向開発方法論が浸透しており、UML [Rumbaugh 98] をその標準記法として採用する方向に進みつつある [玉井 04]。UML において定義されている記法のひとつである「ユースケース」は、この象限に位置づけられる。ユースケースとは、利用者が情報システムを使うときの典型的なシナリオを、情報システムへの要求を明確化する目的で記述したもの [玉井 04] である。ユースケースは、記法が簡潔なモデルであるという利点がある。しかし、少数のモデル構成要素を使って、現実の要求をユースケースとしてモデル化することが難しいという欠点がある [大西 09]。また、この象限に位置づけられる方法は、概して、シナリオの作成に多くの時間を要するという欠点がある

[大西 09].

第2象限は、「領域分割」と呼ばれる[大西 09]. 対象とする領域が明確ならば、要求を漏れなく獲得することができる。顧客自身が領域内の問題を正確に把握していない場合、この方法を用いることで、問題の発見が可能であり、顧客の潜在的な要求を獲得することができる[大西 09]. この象限に位置づけられる方法は、適用領域が安定しているため、要求変更が少なく、かつ、明確であることから、要求変更への対応が容易であるという利点がある[玉井 04]. しかし、開発する情報システムの具体的なイメージが明確にならず、情報システムのシステム境界がはっきりしないという欠点がある[玉井 04].

第3象限は、「ゴール指向」と呼ばれる[大西 09]. この象限に位置づけられる方法は、「ゴール指向要求分析」と呼ばれ、まず、顧客のニーズを基に達成すべき目標を定めなければならないという特徴がある。したがって、顧客のニーズが明確であることが適用の前提となる。定めた目標に対して、全体的に整合のとれた要求が得られるという利点があるけれども、定めた目標に特化した構造となりがちであるという欠点がある。すなわち、定めた目標の解決には極めて有効な方法であるけれども、要求獲得状況の変化や要求変更に対応するための柔軟性に乏しい[玉井 04].

第4象限は、「ブレインストーミング」と呼ばれる[大西 09]. この象限に位置づけられる方法には、「発想法」として有名なKJ法[川喜田 67, 川喜田 70]や、「合意形成法」として有名なSSM (Soft Systems Methodology: ソフトシステム方法論)[Checkland 81, Checkland 90, Wilson 90]がある。この象限に位置づけられる方法は、概して社会科学の領域に属し、現実問題をマクロな視点で捉え、マクロにモデル化する。そのため、現実の多様な問題をなるべくそのままモデル化することが可能であるという利点がある。しかしその反面、これらの方法で獲得された要求は、多様な問題が考慮されている抽象的な要求であり、それを情報システムモデルの具体的な要求に変換する際の詳細を規定することが困難であるという欠点がある。どのような規定を採用するかの詳細は分析者に委ねられるため、最終的な要求の品質が分析者の属人性や恣意性に依存してしまう。

このように、要求獲得方法の4つの類型には、各々一長一短があり、要求獲得状況に応じて、適切な類型に属する要求獲得方法が選択されることが重要である[大西 09]. この点に付言すると、図2.3において、唯一「アブダクション[Hartshorne 32]」のみ、第1象限から第4象限までのすべてに通じる要求獲得方法であることがわかる。これは、アブダクションが、いかなる要求獲得状況においても有用な性質を持つことを示している。また、アブダクションは、通常、ステークホルダから獲得することが難しい「想定外」の事象に対する要求を獲得する上で重要である[妻木 11]. それ故、アブダクションはなんらかの形で要求獲得方法に取り入れられるべきである。現在、アブダクションを形式的に行うための研究は盛んに行われている[井上 92, Sato 09, 古川 09, Inoue 10a, Inoue 10b, 井上 10, 佐藤 10, Inoue 11]. 本研究の目的(1.3節参照)である形式的な要求獲得方法論の構築の過程において、形式的なアブダクションを導入することは重要になると考えられる。

情報システム開発の失敗原因を要求獲得方法の4つの類型から考察すると、次のことがわかる。現状、情報システム開発の実プロジェクトでは、第1象限のユースケースや第2象限の構造化インタビューによる要求獲得が主流である。つまり、現状の要求獲得は、閉じている領域に限定した要求獲得を行っているといえる。したがって、開かれている領域まで考慮に入れた要求獲得を行うことが、

情報システム開発を成功させる鍵となる。そこで、開かれている領域に対応可能な、第3象限または第4象限の要求獲得方法を用いた要求獲得を積極的に行うべきである。しかしながら、第3象限と第4象限に位置づけられる要求獲得方法を用いて要求獲得を行うための方法論は未だ確立されていない。ユースケースや、構造化インタビューのように、第3象限と第4象限から、いわゆる枯れた方法が登場するまでは、今後、基礎研究と産業上の応用が一層推進される必要がある。第3象限は、要求獲得の工学的側面に重きを置く方法であり、第4象限は、ブリコラージュ [Lévi-Strauss 85] 的側面に重きを置く方法である。これらは互いに素な方向性であり、いずれか一方に立脚したアプローチをとる必要がある。本論文では、次章以降、開かれている領域に対してなるべく形式的な要求獲得を行う方法を提案するために、第3象限のゴール指向に立脚して行ってきた研究成果について述べる。

2.2 ゴール指向要求分析の概要

ゴール指向要求分析(GORA: Goal-Oriented Requirements Analysis) [van Lamsweerde 01, van Lamsweerde 04, Lapouchnian 05] は、「ゴール」と呼ばれる達成すべき目標をより具体的なゴールに分解していくことによって、最終的に情報システムで実現可能なゴールまで詳細化されたら、そのゴールを要求と見なす要求獲得方法の1つである。ゴールは、(1) 達成の真偽が定量的に評価できるか否かによって、(1a) 達成の真偽が定量的に評価できるゴールである「ハードゴール」と、(1b) 達成の真偽が定性的にしか評価できないゴールである「ソフトゴール」に分割される。機能要求 (1.7 節 (6) 参照) はハードゴールでなければならない。非機能要求 (1.7 節 (6) 参照) はソフトゴールでなければならない。また、(2) その時点での達成の真偽によって、(2a) 現在達成されていないが、達成すべきゴールである「達成ゴール」と、(2b) 既に達成されており、今後も達成を維持し続けるべきゴールである「維持ゴール」に分割される [REBOK 11]。

分解元のゴールは「親ゴール」、分解先のゴールは「子ゴール」と呼ばれる。分解の関係は、「AND 分解」または「OR 分解」のいずれかである。ここで、AND 分解とは、すべての子ゴールが達成された場合に、親ゴールが達成される分解関係である。OR 分解とは、子ゴールのうち、いずれか1つが達成された場合に、親ゴールが達成される分解関係である。親ゴールと子ゴールは矢線で結ばれ、この矢線は「枝」と呼ばれる。枝は、分解元の親ゴールから、分解先の子ゴールへ引かれる。すなわち、分解元の親ゴールを第1成分、分解先の子ゴールを第2成分とする対(親ゴール, 子ゴール)が枝である。ゴールを分解していくことによって作成されるグラフは、「ゴールグラフ」と呼ばれる。ゴールグラフは、無閉路かつ有向な AND/OR グラフ [Chang 71, Nilsson 71, Nilsson 80] である。ゴールグラフの例 [Hayashi 12] を図 2.4 に示す。ゴール指向要求分析では、まず最初に、顧客のニーズに基づいて「初期ゴール」を定める。ゴール指向要求分析では、一般に、複数の初期ゴールを定めることが可能である。初期ゴールは親ゴールを持たない。逆に、ダミーゴール (1.7 節 (8) 参照) 以外の子ゴールを持たないゴールは「最終ゴール」と呼ばれる。最終ゴールは、要求の候補である。

ゴール指向要求分析は、ソフトウェア工学の一分野である「要求工学 (RE: Requirements Engineering)」 [Zave 97] と呼ばれる研究分野において研究されている、「要求獲得」方法の1つである。要求工学は、任意のライフサイクルモデルにおける最初の作業である、「要求分析」を研究の対象とする。要求工学の重要性は、1970年代の後半から認識されるようになった [Yamamoto 06, 山本 07]。

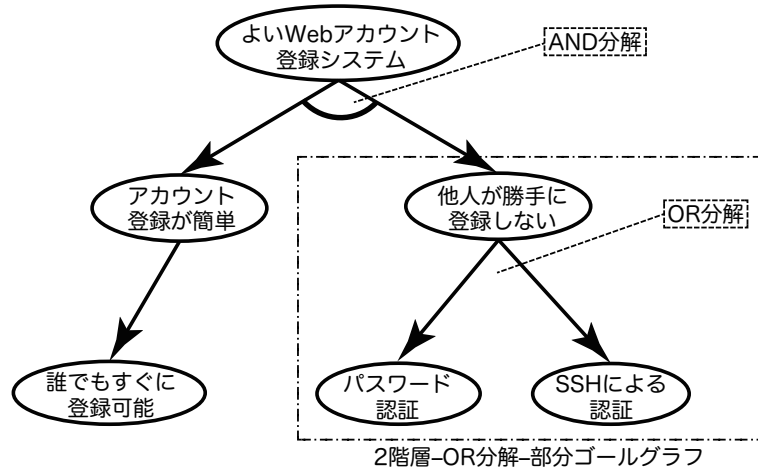


図 2.4: ゴールグラフの例 [Hayashi 12]

当初は、構造化設計 [Demarco 79] やオブジェクト指向設計 [Rumbaugh 90] の上流としての要求分析工程でいかにして要求仕様書を定義し、設計の妥当性を保証するかということが主要な研究課題であった [Yamamoto 06, 山本 07]. しかし、次第に要求仕様書に照らして正しく動作する情報システムの機能を実現する (検証 (1.7 節 (7) 参照) に対応) だけでなく、情報システムによって解決される顧客要求の品質をゴールによって保証すること (妥当性確認 (1.7 節 (7) 参照) に対応) もまた、重要であるという動きが出てきた [Yamamoto 06, 山本 07]. このような雰囲気の中、ゴール指向要求分析の概念は、1987 年に Feather [Feather 87] によって初めて提唱されたとされている [Yu 11]. その後、現在に至るまで多くのゴール指向要求分析法が提唱されてきた. 図 2.5 [Yamamoto 06, 山本 07] に、代表的なゴール指向要求分析法をその発展段階と共に示す.

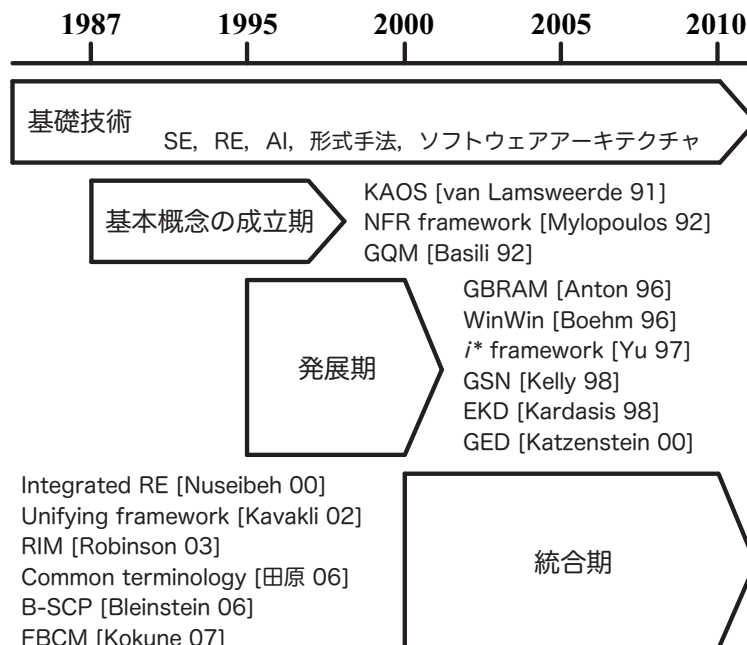


図 2.5: ゴール指向要求分析の発展段階 [Yamamoto 06, 山本 07]

1990年代前半までの第1段階では、van LamsweerdeらのKAOS[van Lamsweerde 91]、MylopoulosらのNFR framework [Mylopoulos 92] および Basili らのGQM[Basili 92]が提案された。この段階では、ゴール指向要求分析の基本的な概念が形成された[Yamamoto 06, 山本 07]。特に、ハードゴールとソフトゴールの概念が確立していた点は特筆に値する。ソフトゴールの起源は、Simonが提唱した「満足化 (Satisficing)」[Simon 57]の概念であるとされる[山本 07]。ハードゴールは達成の真偽が定量的に評価できるため、「最適化 (Optimizing)」が可能である。これに対して、ソフトゴールは、最適化は定性的にしか評価できないため、最適化はできないけれども、満足化はできるようなゴールとして登場したのである。非機能要求は次第に機能要求になるという経験則に基づく定説[Loucopoulos 95]を、ソフトゴールからハードゴールへの分解という形で、ゴールグラフ上で自然に表現されるようになったのがこの頃である。

1990年代後半から2000年前後までの第2段階では、AntonによるGBRAM[Anton 96]やYuらの^{アイスター} framework [Yu 97]をはじめとする様々なゴール指向要求分析法[Boehm 96, Kardasis 98, Kelly 98, Katzenstein 00]が提案された[Yamamoto 06, 山本 07]。ビジネスプロセスとの連携も視野に入れたゴール指向要求分析法が登場したところに、この段階の特徴がある。2000年前後から現在に至るまでの第3段階では、これまでに提案されたゴール指向要求分析法を統合するためのフレームワークの構築や、適用分野に応じて最適なゴール指向要求分析法を組み合わせるための方法論に関する研究が行われている[Nuseibeh 00, Kavakli 02, Robinson 03, Bleistein 06, 田原 06, Kokune 07]。

図2.5から、ゴール指向要求分析は、その時々ソフトウェア工学における問題意識および社会の情勢を反映しながら発展してきたことがわかる。ゴール指向要求分析の概念が初めて提唱された1980年代は、落水型ライフサイクルモデル(図2.1)が要求変更柔軟に対応できないという問題が明らかになり、落水型に代わり、反復型ライフサイクルモデル(図2.2)を奨励する動きが出てきた頃にあたる。1985年には、落水型の問題点である要求変更柔軟に対応することを目的として、Boehmによって、反復型の1つであるスパイラルモデル[Boehm 85]が提案された。この翌年の1986年には、Brooksによる論考『No Silver Bullet』[Brooks 86]が発表され、落水型に代わる新たなパラダイムとしての反復型の存在が人口に膾炙することになった。前述の通り、ゴール指向要求分析の概念が登場した年は、ちょうどこの翌年の1987年にあたる。反復型は、要求分析において誤りが含まれることを前提とするモデルであることから、ともすれば、誤りのない要求分析を行うことを目指す要求工学が軽視されるのではないかと懸念が、当時の要求工学の研究者の間で蔓延していたのではないかと推測される。もしそうであったならば、ゴール指向要求分析は、反復型へのアンチテーゼとして、要求工学の重要性を改めて知らしめるために登場した方法であるという見方もできよう。

ゴール指向要求分析の基本概念的成立期にあたる第1段階は、要求獲得が、システム科学における「ハードアプローチ」と「ソフトアプローチ」の双方にまたがる問題であるということが認識されるようになった時期にあたる[Jirotko 92, Petrie 92]。そのため、システム科学における諸成果を応用することで、要求獲得の問題を、なるべく形式的、かつ、定量的に分析するための方法論が追求され、ゴール指向要求分析の基本概念的成立した。ゴール指向要求分析の発展期にあたる第2段階は、インターネットの普及に伴い、社会が急速にIT化しはじめた時期にあたる。ITが経営にとって大き

な影響を与えるようになったことから、ビジネスプロセスと連携するゴール指向要求分析法が登場したといえる。その後、携帯電話や情報家電、自動車などへIT機器が埋め込まれる社会になり、高度なICT（ユビキタス）社会が到来して現在に至っている。このような社会において、今後、人間系も含めた社会システム（人間活動システム）の中における情報システムの位置付けや役割を明確にするための方法として、ゴール指向要求分析の重要性は一層高まるとされている[山本 07]。

2.3 既存のゴール指向要求分析法

本節では、既存のゴール指向要求分析法のうち、本論文の内容と関連する「AGORA」[Kaiya 02]と「属性値に基づくゴール選択手法」[斎藤 06, 山本 07]の2つのゴール指向要求分析法について述べる。これらは共に、ゴールに種々の「属性」を付加することによって、ゴール指向による要求獲得の問題（2.1節参照）を解決することを指向するゴール指向要求分析法である。2.3.1節では、AGORAについて述べる。続く2.3.2節では、属性値に基づくゴール選択手法について述べる。2.3.1節および2.3.2節の内容から、要求獲得に関して、これら2つは相補的な関係にあることが明確になる。

なお、属性値に基づくゴール選択手法に関する既存の文献[斎藤 06, 山本 07]には、集合記法上の誤りや内容が不明瞭な点が散見されるため、主な内容はそのままに、本論文の目的（1.3節参照）に沿うよう、なるべく形式的に記述する。

2.3.1 AGORA

AGORA（Attributed Goal-Oriented Requirements Analysis Method: 属性付きゴール指向要求分析法）[Kaiya 02]は、ゴールに属性を付与することで、ゴール指向要求分析における次の問題を支援するために開発された。（1）ゴール間のコンフリクトの解決および、特定のゴールに対するステークホルダ間の認識不一致の解決、（2）要求仕様書の品質とゴール指向要求分析の関連付け、（3）要求変更の影響分析。2002年に最初の論文[Kaiya 02]が発表されて以来、現在まで発展を続けており、最新のAGORAでは、これらの問題にすべて対応可能となっている。

（1）の「ゴール間のコンフリクト」および「ステークホルダ間の認識不一致」は、本節（属性-3）と2.3.1.3節で、各々の詳細を述べる。（2）は、他のゴール指向要求分析法にはないAGORA特有の利点であり、要求仕様書作成前の要求獲得段階で、要求仕様書の品質を見積もることができる種々の指標が提案されている[Kaiya 02]。（3）の「要求変更」は、要求獲得を困難にする問題の1つ（2.1節（3）参照）でありながら、ゴール指向要求分析の欠点（2.1節参照）とされている。この欠点を克服することは、ゴール指向要求分析の主要な課題の1つである。この項目に関連したAGORAの文献としては、[Tanabe 08]や[Hayashi 12]がある。

AGORAでは、あらかじめ次の4つの属性が定義されている。

AGORAが持つ属性とその定義[Kaiya 02, Tanabe 08, Yamamoto 08]

（属性-1）貢献度

親ゴールと子ゴールを結ぶ枝に対して振られる値であり、-10から10までの整数値

をとる。貢献度は、子ゴールが達成されると親ゴールの達成にどの程度貢献するかの度合いを表し、値が大きいほど貢献の度合いが大きく、値が負の場合は親ゴールの達成を阻害することを意味する。値が0の場合は、親ゴールの達成に子ゴールの達成が関係ないことを意味し、枝は描かれない。OR分解では、親ゴールと子ゴールを結ぶすべての枝に対して個々に貢献度を振るが、AND分解では、すべての子ゴールが達成されて初めて親ゴールが達成されることから、親ゴールと子ゴールを結ぶ枝全体に対して1つの貢献度を振る。

ゴール指向要求分析では、親ゴールを達成するために必要とされる、より具体的なゴールが子ゴールである。そのため、ゴール分解は必ず正の貢献度でなされる。したがって、負の貢献度はゴール分解時には関係がなく、ゴール間でゴールの内容に関する意味上のコンフリクト（ゴール間のコンフリクト（本節（属性-3）参照））があることを明示するために用いられる。負の貢献度で結ばれた子ゴールは、親ゴールをゴール分解したものであると見なされる。

（属性-2）満足度

ゴールが達成された場合に、ステークホルダが各々の立場から判断してどの程度満足するかの度合いを表す。この度合いは、貢献度同様 -10 から 10 までの整数値をとる。ただし、ステークホルダ間の満足度一致の度合いを測定するため、自身の立場から評価するだけでなく、他のステークホルダならばどのような満足度を持つかを推測し、他のステークホルダの立場に立った評価も行う。そのため、満足度は行列として個々のゴールに振られ、この行列は「満足度行列」と呼ばれる。AGORAにおいて、特定のゴールに満足度行列を振るかどうかは分析者の判断にゆだねられる。

満足度行列の例を図2.6に示す。AGORAの満足度行列は、ステークホルダとして少なくとも顧客を含む[鵜飼10]。ここでは、ステークホルダとして利用者、経営者、供給者の3者を考えている。この内、利用者および経営者は顧客(1.7節(2)参照)であるとする。満足度にステークホルダ間の影響度の差異を反映させないために、各々のステークホルダは、満足度を振る際、他のステークホルダが振った満足度を見ることはできない。たとえば、図2.6において、利用者は経営者が振った2行目の各成分の満足度(10, 10, -10)および、供給者が振った3行目の各成分の満足度(5, -10, 0)を見ることはできない。満足度行列のすべての成分を見ることができるのは、すべてのステークホルダに対して独立した立場から分析を行う権限が与えられた「(要求)分析者」だけである。

（属性-3）意味タグ

あるゴールの内容と、ゴールグラフ上の別のゴールの内容が、その意味において相容れない場合がある。これは「ゴール間のコンフリクト」と呼ばれる。ゴールグラフ上に、この意味上のコンフリクトを持つゴールの対が存在する場合、互いに矛盾した要求が獲得されることになる。そのため、この矛盾はなるべく取り除かれなければな

らない。この問題に対して、AGORAでは、任意のゴールに対して、「ゴール特性」と呼ばれる、ゴールの意味情報を付与ことができる。ゴール特性は、「意味タグ」として、個々のゴールに付けられる。意味タグを用いる際には、あらかじめゴール特性を定義しておく必要がある。たとえば、非機能要求に関するゴール特性を定義する場合は、ISO/IEC 9126 [ISO 91] (表 1.1) で定義されている「セキュリティ」や「ユーザビリティ」といったゴール特性を定義することが考えられる [Tanabe 08, Hayashi 12]。意味タグが付けられたゴールグラフの例を図 2.7 に示す。意味タグは、ゴール間のコンフリクトの存在を明確にする。また、意味タグを利用することで、ゴール分解を行う際に、ゴールグラフ上の既存のゴールと矛盾しないゴールを追加することが可能になる。

意味タグを利用せずにゴール間のコンフリクトを形式的に検出するためには、何らかの「自然言語処理」を導入する必要がある。しかし、既存の自然言語処理は、構文や文脈など、自然言語の表層的な側面を形式的に分析することに留まっており、自然言語の持つ意味を理解するための形式的な方法は未だ確立されていない。今後、自然言語処理が、自然言語の意味を扱える段階にまで発展すれば、ゴール間のコンフリクトを形式的に検出する方法として利用することが可能になるだろう。また、貢献度 (2.3.1 節 (属性-1) 参照) を形式的に定めるためには、自然言語処理が必須である。貢献度は、親ゴールと子ゴールの対に対して、親ゴールと子ゴールの内容から定められるゴール選択基準である。したがって、ゴールの内容が自然言語で記述されている場合には、自然言語処理に基づいて「(意味に関する) ゴール間の距離」を測定し、それに基づいて貢献度を定める必要がある。本論文は、属性 (ゴール選択基準) 設定後のゴール選択を形式的に行う方法を研究の対象としており、ゴール選択前の、属性を形式的に定める方法については対象としていない。貢献度のように、ゴール間で定められるゴール選択基準値を形式的に定める上で、意味を理解する自然言語処理は必須であり、自然言語処理の今後の発展が期待される。

(属性-4) 記述理由

貢献度または満足度に対して、なぜその値を振ったのかという理由を記述することが可能である。これは、「記述理由」と呼ばれる。記述理由は、吹き出しの中に自然言語で記述される。記述理由の例を図 2.9 (a) に示す。

これら以外に、任意の属性を追加定義することもできる [Yamamoto 08]。ただし、その際には、属性の持つ値 (「属性値」) の計算方法も含めて定義される。特定のゴールが持つ属性値は「ゴールの添字・属性名」で表される。追加する属性は、ゴールグラフ上のすべてのゴールに対して定義され、属性値の計算方法は個々の枝に対して定義される。図 2.8 に、属性として「コスト」を定義し、その計算方法として、すべての枝に一括して $\sum_{g' \in Sel(g)} Cst(g')$ という計算方法を定義した場合の例 [Yamamoto 08] を示す。ただし、 $Sel(g)$ は、ゴール g に選択されたすべての子ゴールを要素とする集合。ゴールグラフ上のすべてのゴールを要素とする集合を G で表すと、

Cst は、定義域を G 、値域を整数全体の集合とする関数。

		被評価者		
		U	O	S
評価者	U	8	-7	0
	O	10	10	-10
	S	5	-10	0

U:利用者
O:経営者
S:供給者

図 2.6: 満足度行列の例

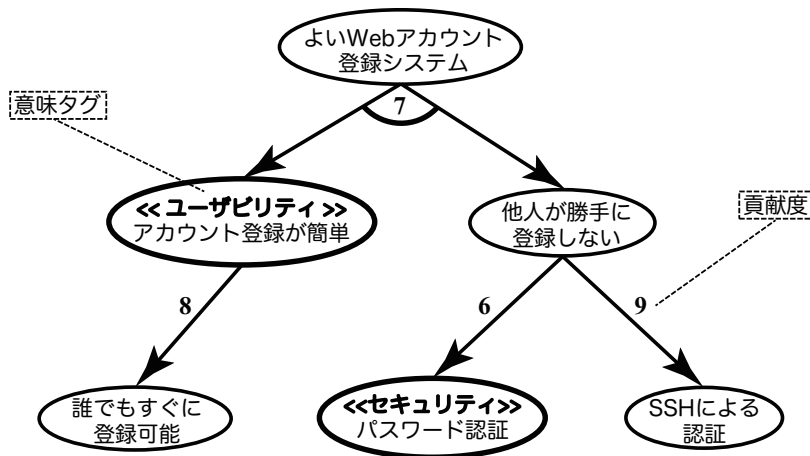


図 2.7: ゴール間のコンフリクトの例 [Hayashi 12]

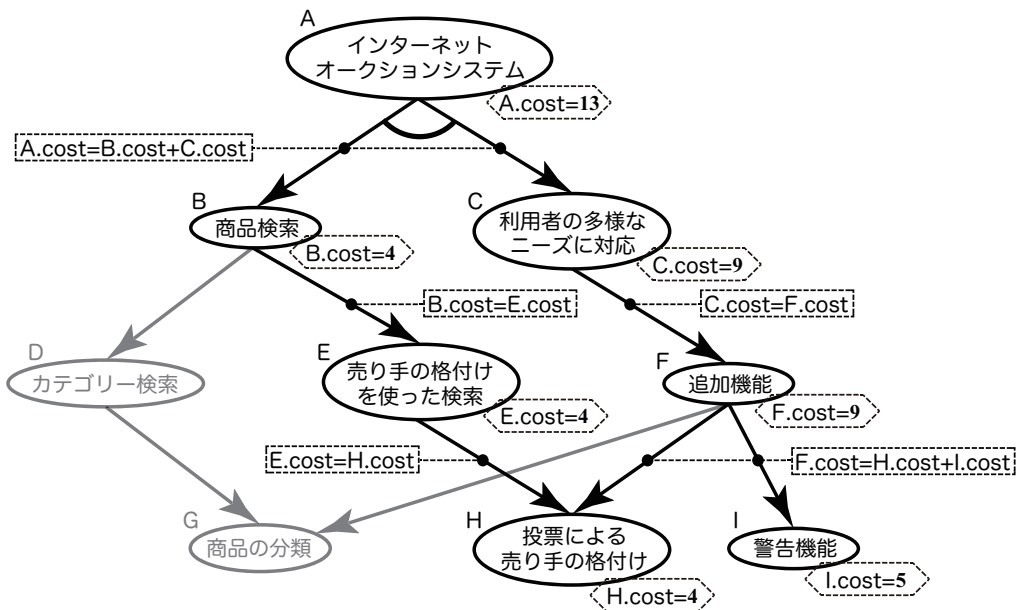


図 2.8: 属性「コスト」およびその計算方法の定義の例 [Yamamoto 08]

2.3.1.1 貢献度の形式的定義

G をゴールグラフ上のすべてのゴールを要素に持つ集合であるとする。分解関係を考慮しない場合、ゴールグラフは、 G と、 G と G の直積の部分集合 $E \subset G \times G$ の対 (G, E) で定義される [鶴飼 10]。 G の要素 $g \in G$ を「ゴール」、 E の要素 $(g, g') = e \in E$ を「枝」と呼ぶ。グラフ理論では、有限の列 $W = (g_0, e_1, g_1, e_2, \dots, e_n, g_n)(g_i \in G, e_i \in E, e_i = (g_{i-1}, g_i), i \in [1, n])$ を「有向歩道」という。有向歩道は、しばしば枝 e_n を省略して、単に (g_0, g_1, \dots, g_n) と書かれる。ある有向歩道において、すべての g_i が異なる場合 $((\forall i, i' \in [1, n], i \neq i')(g_i \neq g_{i'}))$ 、その有向歩道は、「有向道」と呼ばれる。本論文では、ゴール g からゴール g' までの有向道を「 (g, g') -道」と呼ぶ。本論文では、少なくとも1つの (g, g') -道が存在するとき、またそのときに限り、ゴール g からゴール g' へ「到達可能」と定義する $(\exists (g, g')$ -道 $\Leftrightarrow R(g, g')$)。また、 g から g' へ到達可能なすべての (g, g') -道を要素とする集合を、 (g, g') によって添字付けられた集合 $Path_{(g, g')}$ で表す $(Path_{(g, g')} := \{ (g, g')$ -道 $\mid R(g, g') \})$ 。

本論文では、枝 e の貢献度を $Cov(e)$ (Contribution (value)) で表す。また、枝 (g, g') の貢献度 (ゴール g' のゴール g への貢献度) を $Cov(g, g')$ で表し、次式で定義する。

$$Cov(g, g') := \begin{cases} Cov(e) & \text{if } \exists e \in E \text{ s.t. } e = (g, g'), \\ \frac{\sum_{(g, g')\text{-道} \in Path_{(g, g')}} Cov(e)}{|e|} & \text{if } \neg(\exists e \in E \text{ s.t. } e = (g, g')) \wedge R(e), \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

式(2.1)は、次のような定義である。ゴール g とゴール g' を結ぶ枝が存在する場合には、その枝に振られた貢献度を $Cov(g, g')$ の貢献度とする。 g と g' を結ぶ枝が存在せず、かつ、 g から g' へ到達可能である場合には、 g から g' までのすべての (g, g') -道に属する枝の貢献度の平均を、 $Cov(g, g')$ の貢献度とする。その他の場合、すなわち、 g から g' へ到達可能でない場合は、 $Cov(g, g')$ の貢献度は0とする。

ゴール g' に対する親ゴール g がただ1つしか存在しない場合、簡単のため、 $Cov(g, g')$ を $Cov(g')$ と略記する。

$$\exists! e \in E \text{ s.t. } e = (g, g') \Rightarrow Cov(g') := Cov(e) \quad (2.2)$$

2.3.1.2 ゴール選択上のコンフリクト

複数の親ゴールを持ち、そのうち少なくとも1つの親ゴールに対しては、負の貢献度 (2.3.1節 (属性-1) 参照) で接続しているゴールを考える。ゴール分解は必ず正の貢献度でなされる (2.3.1節 (属性-1) 参照) ため、ゴールグラフ上の任意のゴールは少なくとも1つの親ゴールと正の貢献度で接続する。したがって、このゴールが選択される場合、少なくとも1つの親ゴールの達成には貢献するけれども、少なくとも1つの親ゴールの達成を阻害することになる。本論文では、この状況を「ゴー

ル選択上のコンフリクト」と定義する。ゴール選択上のコンフリクトがあるゴールが選択されることは、初期ゴール（2.2節参照）の達成に必要なゴールの達成を阻害するゴールを達成することになるためなるべく避けるべきである。

ゴール選択上のコンフリクトが生じ得るのは、少なくとも1つの親ゴールに対して、負の貢献度で接続しているゴールである。したがって、貢献度を参照することによって、ゴール選択上のコンフリクトが生じる可能性のあるゴールを検出することができる。ゴール選択上のコンフリクトの例 [Kaiya 02] を図 2.9(a) に示す。図 2.9(a) では、ゴール「本人確認しない」と「本人確認する」は、

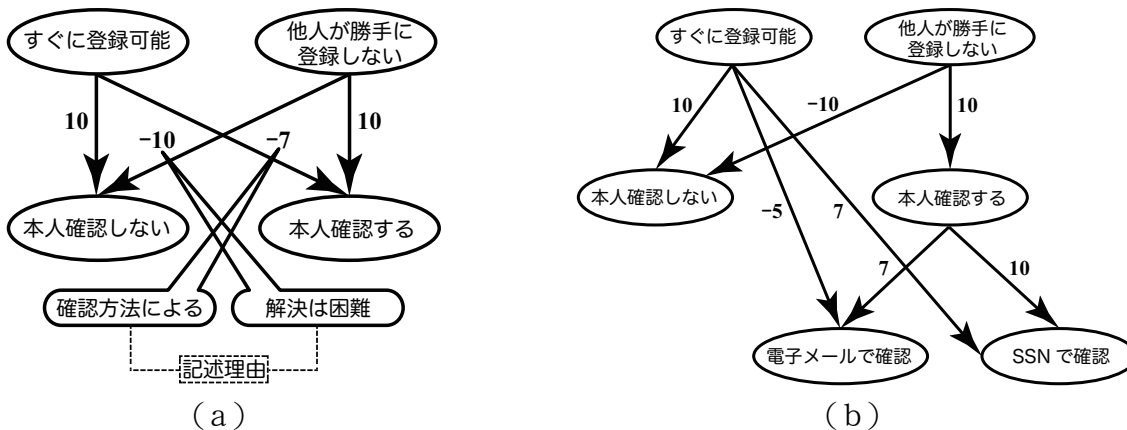


図 2.9: ゴール選択上のコンフリクトの例 [Kaiya 02]. (a), (b) は各々、コンフリクト解決前と後を表す

いずれも「すぐに登録可能」と「他人が勝手に登録しない」を親ゴールに持ち、そのうち一方の親ゴールに対して負の貢献度で接続しているため、ゴール選択上のコンフリクトを生じ得るゴールである。

ゴール選択上のコンフリクトを解決する方法としては、子ゴールをさらにゴール分解し、子ゴールが負の貢献度で接続している親ゴールに対して、正の貢献度を持つようなゴールを作成することが考えられる [Kaiya 02]. ただし、常にそのようなゴールが存在するとは限らない。仮にそのようなゴールが存在し、作成された場合、親ゴールから子ゴールへ接続していた枝は、親ゴールから作成された新たなゴールへと接続先が変更されるものとする。

たとえば、図 2.9(a) の「本人確認する」は、記述理由 (2.3.1 節 (属性-4) 参照) の内容から、「すぐに登録可能」に対して正の貢献度で接続する確認方法を内容に持つ子ゴールに分解できそうである。そのような子ゴールとして、たとえば、「電子メールで確認」や「SSN (Social Security Number: 社会保障番号) で確認」が考えられる。このうち、「電子メールで確認」は人手による確認が必要になるため、「本人確認する」が負の貢献度で接続している「すぐに登録可能」に対して、やはり負の貢献度で接続する。一方、「SSN で確認」は、確認を自動化できるため、「すぐに登録可能」に対して、正の貢献度で接続する [Kaiya 02]. 結果を図 2.9(b) に示す。

2.3.1.3 ステークホルダ間の認識不一致

要求獲得では、ステークホルダ間の知識や文化的背景の差異が問題となる (2.1 節 (2) 参照)。ゴール指向要求分析を用いた要求獲得においては、この問題は、ゴールグラフ上の任意のゴールが、任意のステークホルダにとって妥当性 (1.7 節 (7) 参照) を満たすものであるかどうかという問題に対応

する。ゴールグラフ上の少なくとも1つのゴールが、少なくとも1人のステークホルダの妥当性を満たさない場合、現実と情報システムの差異(2.1節(1)参照)が生じてしまう。

ゴールの妥当性は、「ステークホルダ間の『認識不一致(Discordance)』」[Kaiya 05]を調べることで評価することが可能である。ステークホルダ間の認識不一致とは、(1)ゴールの内容に対する解釈が異なっている場合と、(2)解釈は同じであるが、ゴールの内容に対する評価が異なっている場合を指す。このうち(2)は、(2a)評価基準の違いと、(2b)関心の違いに分割される。(1)または(2a)は満足度行列(2.3.1節(属性-2)参照)の「縦数値対立」に対応しており、(2b)は満足度行列の「対角数値対立」に対応している。ここで縦数値対立とは、満足度行列の特定の列で正負の値が混在していることであり、対角数値対立とは、自身の立場からの選好を表す主対角成分において、正負の値が混在していることである。図2.6の満足度行列の例では、第2列が(-7, 10, -10)であり、正負の値が混在するため、縦数値対立があることがわかる。一方、主対角成分は、(8, 10, 0)であり、対角数値対立はないことがわかる。

ステークホルダ間の認識不一致を解決することは、縦数値対立と対角数値対立を解決することに対応する。縦数値対立は、被評価者の解釈または評価基準を評価者が誤解していることにより生じる。したがって、被評価者の記述理由(2.3.1節(属性-4)参照)を参照することで、それらを正しく理解し、被評価者の立場から満足度を振り直すことで、縦数値対立を解決することができる。一方、対角数値対立は、ステークホルダの立場の違いに基づくステークホルダ間の本質的な満足度の差異を表しており、これを解決することは一般に難しい。

2.3.2 属性値に基づくゴール選択手法

情報システム開発プロジェクトの過程において、企業を取り巻くビジネスプロセスは変化し続けている。したがって、プロジェクト開始時点で獲得された要求は、たとえそれがその時点で最適な要求であったとしても、ビジネスプロセスの変化に伴って、後に要求変更を迫られる場合がある[南波 09]。特に、ビジネスプロセスの変化に伴う要求変更は、要求分析を根本的にやり直す必要があるため、要求獲得から再度行う必要がある。したがって、ビジネスプロセスの変化に動的に対応可能な要求獲得方法が確立される必要がある。

「属性値に基づくゴール選択手法」[斎藤 06, 山本 07]は、ビジネスプロセスの変更に動的に対応するためのゴール指向要求分析法として提案された。属性値に基づくゴール選択手法では、「3階層ゴールグラフ」と呼ばれる、独自のゴールグラフを採用している。3階層ゴールグラフの例[斎藤 06, 山本 07]を図2.10に示す。3階層ゴールグラフとは、次の3つのゴールから構成され、各々のゴールが階層を成すゴールグラフである。図2.10は、製造企業A社の生産管理プロセスにおける「生産コストの削減」を戦略ゴールとして、実際に作成された3階層ゴールグラフである。

3階層ゴールグラフを構成するゴールとその定義[斎藤 06, 山本 07]

(1) 戦略ゴール

顧客のニーズに基づいて設定される、ビジネス戦略上達成すべきゴール。初期ゴール(2.2

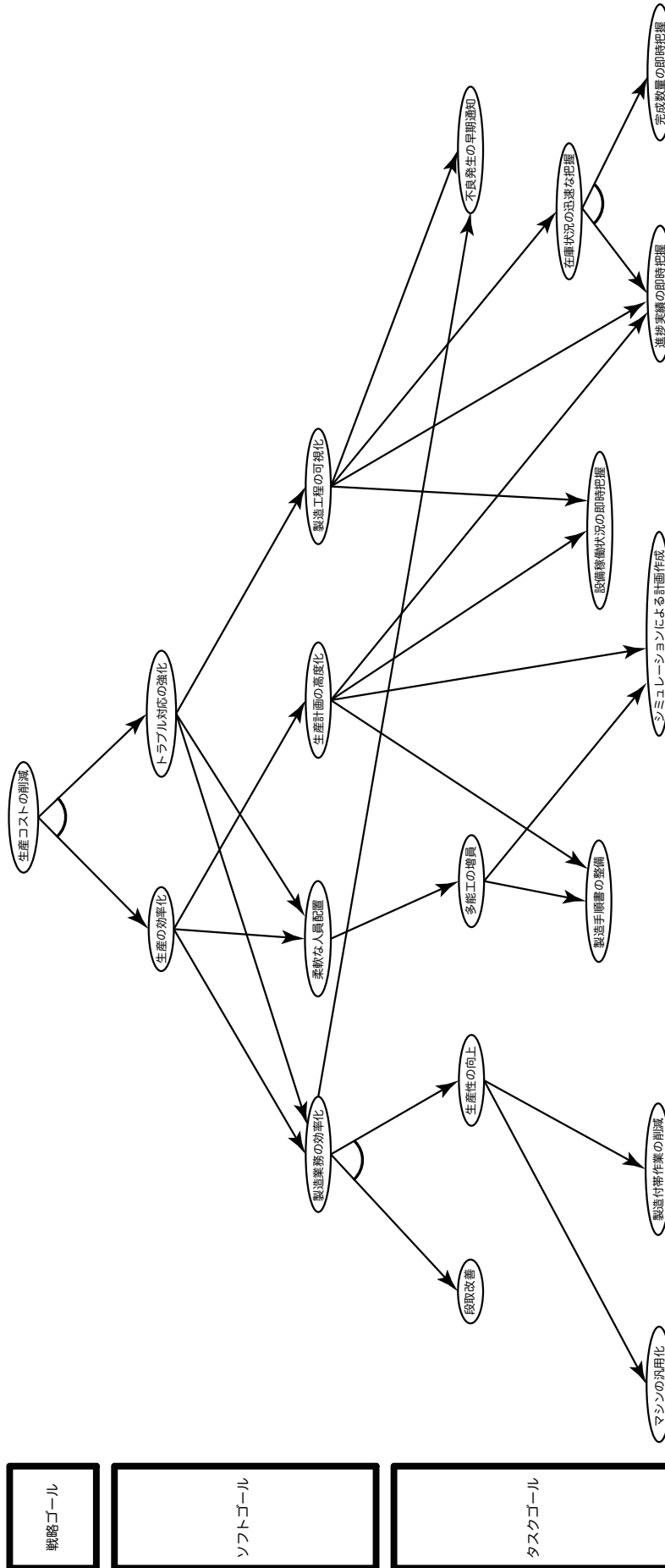


図 2.10: 3 階層ゴールグラフの例 [斎藤 06, 山本 07]

節参照) でなければならない。

(2) ソフトゴール

戦略ゴールを達成するために実現すべきより具体的なゴール。達成されたかどうかが明確に判定できない程度の曖昧な内容を持つ。非機能要求 (1.7 節 (6) 参照) に対応する。

(3) タスクゴール

ソフトゴールを達成するために実現すべきより具体的なゴール。達成されたかどうかは明確に判定できる具体的な内容を持つ。機能要求 (1.7 節 (6) 参照) に対応する。タスクゴールは、ハードゴール (2.2 節 (1a) 参照) である。

3階層ゴールグラフにおいて、ソフトゴールおよびタスクゴールは、少なくとも1つのゴールの子ゴールである。しかし、ソフトゴールおよびタスクゴールは、ダミーゴール (1.7 節 (8) 参照) ではないものとする。

属性値に基づくゴール選択手法では、次の3つの属性が定義されている。

属性値に基づくゴール選択手法が持つ属性とその定義 [斎藤 06, 山本 07]

(属性-1) 寄与度

AGORA における貢献度 (2.3.1 節 (属性-1) 参照) に同じ。

(属性-2) 有効度

戦略ゴールの達成にどの程度貢献するかの度合い。タスクゴールを子ゴールを持つソフトゴールに対して設定される。有効度は $Vld(s)$ で表され、次式で定義される。

$$Vld(s) := \frac{\sum_{\substack{(i,s)\text{-道} \in Path_{(i,s)} \\ i \in STG}} \min_{e \in (i,s)\text{-道}} Cov(e)}{|Path_{(i,s)}|} \quad (2.3)$$

ただし、 SG をゴールグラフ上のすべてのソフトゴールを要素とする集合とした場合、 $s \in SG$ 。 STG は、戦略ゴールを要素とする集合。 (i, s) -道 は、ゴール i からゴール s までの有向道 (2.3.1.1 節参照)。 $Cov(e)$ は、枝 $e \in E$ の貢献度 (2.3.1.1 節参照)。 $Path_{(i,s)}$ は、 i から s へ到達可能なすべての (i, s) -道を要素とする集合 (2.3.1.1 節参照)。

(属性-3) 充足度

タスクゴールの達成の有無を表すブール値。タスクゴールに対して設定される。タスクゴールが達成されていれば (タスクゴールが維持ゴール (2.2 節 (2b) 参照) ならば) 1, そうでなければ (タスクゴールが達成ゴール (2.2 節 (2a) 参照) ならば) 0 の値をとる。充足度は、 $Acv(t, k_t)$ で表される2変数のブール値関数として定義される。ただし、 TG をゴールグラフ上のすべてのタスクゴールを要素とする集合とした場合、

$t \in TG$. k_t は、タスクゴール t に対応する KPI (Key Performance Indicator: 主要業績評価指標) [Parmenter 07] である. k_t は人手により主観的に定められる定数である. タスクゴールは機能要求 (1.7 節 (6) 参照) に対応しているため, 定量的な評価が可能である. $Acv(t, k_t)$ は, 次式で評価される.

$$Acv(t, k_t) \text{ が } 1 \text{ である} \Leftrightarrow k_{t,n} \geq k_{t,o} \quad (2.4)$$

ただし, $k_{t,n}$ は, 「現状」の k_t . $k_{t,o}$ は「目標値」の k_t である. $Acv(t, k_t)$ の値を求めることに先だって, これら 2 つの値を定める必要がある.

また, ゴール選択 (1.7 節 (10) 参照) のために, 次の 4 つの補助集合が定義されている.

属性値に基づくゴール選択手法が持つ補助集合とその定義 [斎藤 06, 山本 07]

(補助集合-1) PSG

タスクゴールを選択しておらず, かつ, 少なくとも 1 つのタスクゴールを子ゴールに持つようなソフトゴールを要素とする集合 (Pending Soft Goals). 分析開始時点での PSG は特に, PSG^1 で表す.

(補助集合-2) PTG

ソフトゴールから選択されるか否かが決定していないタスクゴールを要素とする集合 (Pending Task Goals).

(補助集合-3) SUB_s

ソフトゴール $s \in PSG$ を親ゴールとするすべてのタスクゴールを要素とする集合. SUB_s の部分集合であり, s に対して正の寄与度 (2.3.2 節 (属性-1) 参照) を持つタスクゴールのみを要素とする集合は, SUB_s^+ で表す.

(補助集合-4) SEL_s

ソフトゴール $s \in PSG$ が選択したタスクゴールの集合. ただし, s から選択されるタスクゴールは 1 つのみ ($|SEL_s| = 1$) であるとする.

2.3.2.1 表現定義

ゴールグラフにおいて親ゴール G に対して子ゴール x が接続している場合, 次のように表す.

$$(G \leftarrow x) \quad (2.5)$$

ただし, 項 G と項 x はそれぞれ親ゴール G , 子ゴール x に対応する. 記号 \leftarrow は, 記号の左にある項には記号の右にある項が必要であることを意味する. また, 親ゴール G に子ゴール x が接続し, 子ゴール x にその子ゴール y が接続している ($x \leftarrow y$) 場合には, 次に示す入れ子表現で階層関係を表

す.

$$(G \leftarrow (x \leftarrow y)) \quad (2.6)$$

親ゴール G に対して, 子ゴール x, y が OR 分解の関係である場合, 論理演算子 $+$ を用いて次のように表す.

$$(G \leftarrow x + y) \quad (2.7)$$

親ゴール G に対して子ゴール x, y が AND 分解の関係である場合, 論理演算子 $*$ を用いて次のように表す.

$$(G \leftarrow x * y) \quad (2.8)$$

2.3.2.2 基本法則

表現定義で記述された論理式には, 次に示す法則が成立する.

分配則 [齋藤 06, 山本 07]

$$(G \leftarrow (w \leftarrow x + y) * z) \Leftrightarrow (G \leftarrow x * z + y * z) \quad (2.9)$$

$$(G \leftarrow (w \leftarrow x * y) + z) \Leftrightarrow (G \leftarrow (x + z) * (y + z)) \quad (2.10)$$

吸収則 [齋藤 06, 山本 07]

$$(G \leftarrow (w \leftarrow x + y) * x) \Leftrightarrow (G \leftarrow x) \quad (2.11)$$

$$(G \leftarrow (w \leftarrow x * y) + x) \Rightarrow (G \leftarrow x) \quad (2.12)$$

結合則 [齋藤 06, 山本 07]

$$(G \leftarrow (w \leftarrow x + y) + z) \Leftrightarrow (G \leftarrow x + y + z) \quad (2.13)$$

$$(G \leftarrow (w \leftarrow x * y) * z) \Leftrightarrow (G \leftarrow x * y * z) \quad (2.14)$$

式 (2.9)~(2.14) に加えて, 「交換則」も成立する.

式 (2.12) のみ同値関係が成立しない理由は, 次の通りである. 式 (2.11) を右から左へ適用した場合, ゴールグラフ上に任意のゴール y が追加される. y は, (1) 親ゴール w の達成に関係のないゴールである場合と, (2) 親ゴール w の達成に関係のあるゴールである場合の 2通りが存在する. (1) の場合, 貢献度が 0 であるため, 枝は描かれぬ (2.3.1 節 (属性-1) 参照). すなわち, y は式 (2.11) の適用対象となるゴールグラフ上のゴールとして加わらないため, 問題がない. (2) の場合, 貢献度は正または負のいずれかであるため, 枝は描かれ, y はゴールグラフ上のゴールとして追加される.

しかし、親ゴールの達成に関係のあるゴールが追加されているため、問題がない。したがって、式(2.11)を右から左へ適用することは可能である。

式(2.11)と同様に、式(2.12)を右から左へ適用した場合、ゴールグラフ上に任意のゴール y が追加される。(1)の場合、式(2.11)と同様の理由から問題がない。しかし、(2)の場合、 y の貢献度が0であっても、必ずしも $x*y$ の貢献度が0であるとは限らないため、親ゴール w の達成に関係のない y が追加されてしまう可能性がある。このような y がゴールグラフ上に追加されることは、ゴール分解の定義に矛盾するため許されない。したがって、式(2.12)を右から左へ適用することはできない。

2.3.2.3 ゴールグラフの簡約化

ゴールグラフの簡約化とは、基本法則(2.3.2.2節参照)を用いて、ゴールグラフ上の多階層-部分ゴールグラフ(1.7節(9)参照)が2階層-部分ゴールグラフ(1.7節(9)参照)になるまで階層縮約することである。

基本法則のうち、交換法則はゴールグラフの簡約化に利用できない。残りの法則のうち、式(2.10)、式(2.14)を左から右へ適用した場合、2階層-AND分解-部分ゴールグラフ(1.7節(9a)参照)が得られる。式(2.9)、式(2.11)、式(2.12)、式(2.13)を左から右へ適用した場合、2階層-OR分解-部分ゴールグラフ(1.7節(9b)参照)が得られる。したがって、ゴールグラフの簡約化を行った結果得られる2階層-部分ゴールグラフは、必ず、2階層-AND分解-部分ゴールグラフか2階層-OR分解-部分ゴールグラフのいずれかになる。

ゴールグラフの簡約化は、次の手順を繰り返すことで行われる。

ゴールグラフの簡約化の手順

- (1) ゴールグラフの簡約化を行う対象となる多階層-部分ゴールグラフの部分ゴールグラフのうち、基本法則の式(2.9)~(2.14)の左辺の部分ゴールグラフと同型なものが存在するかどうかを調べる。そのような部分ゴールグラフが少なくとも1つ存在する場合は、部分ゴールグラフを1つ選択し、(2)へ進む。存在しない場合は、ゴールグラフの簡約化は終了となる。
- (2) 選択した部分ゴールグラフに対して、該当する基本法則を左から右へ適用し、(1)へ戻る。

(1)において選択可能な複数の部分ゴールグラフが存在する場合、どの部分ゴールグラフから選択していくかによって、最終的な結果が異なる。すなわち、多階層のゴールグラフに対してゴールグラフの簡約化を行った結果得られる2階層-部分ゴールグラフは、一般に複数存在する。

2.3.2.4 属性値に基づくゴール選択手法におけるゴール選択手順

属性値に基づくゴール選択手法は、次の4つの手順から広義のゴール選択(1.7節(10)参照)を行う。(1)ゴールグラフの作成、(2)ゴールグラフの簡約化、(3)ゴールグラフの属性設定、(4)タスクゴールの選択。以下、各々の手順の詳細を述べる。

(1) ゴールグラフの作成

図2.10が作成されたものとする。

(2) ゴールグラフの簡約化

タスクゴールを子ゴールに持つソフトゴールが最上位のゴールであるような多階層-部分ゴールグラフ (1.7節(9)参照) は、一般に複数存在する。属性値に基づくゴール選択手法では、このうち、構成要素(ゴール)の数が最大のものに対して、ゴールグラフの簡約化が行われる。図2.10では、そのような多階層-部分ゴールグラフは3つ存在する。以下、それらの各々に対して、ゴールグラフの簡約化を適用した結果を示す。簡約化後のゴールグラフ[斎藤 06, 山本 07]を図2.11に示す。

(2a)

$$\left(\begin{array}{l} \text{製造業務の効率化} \leftarrow \text{段取改善} * (\text{生産性の向上} \leftarrow \text{マシンの汎用化} \\ + \text{製造付帯作業の削減}) + \text{不良発生}の\text{早期通知} \end{array} \right)$$

式(2.9)

$$\left(\begin{array}{l} \text{製造業務の効率化} \leftarrow \text{段取改善} * \text{マシンの汎用化} + \text{段取改善} * \text{製造付帯作業の削減} \\ + \text{不良発生}の\text{早期通知} \end{array} \right)$$

(2b)

$$\left(\begin{array}{l} \text{柔軟な人員配置} \leftarrow \text{多能工の増員} \leftarrow \text{製造手順書の整備} \\ + \text{シミュレーションによる計画作成} \end{array} \right)$$

⇔

$$\left(\begin{array}{l} \text{柔軟な人員配置} \leftarrow (\text{多能工の増員} \leftarrow \text{製造手順書の整備} \\ + \text{シミュレーションによる計画作成}) + \text{ダミーゴール} \end{array} \right)$$

式(2.13)

$$\left(\begin{array}{l} \text{柔軟な人員配置} \leftarrow \text{製造手順書の整備} + \text{シミュレーションによる計画作成} \\ + \text{ダミーゴール} \end{array} \right)$$

$$\left(\begin{array}{l} \text{柔軟な人員配置} \leftarrow \text{多能工の増員} \leftarrow \text{製造手順書の整備} \\ + \text{シミュレーションによる計画作成} \end{array} \right)$$

(2c)

$$\left(\begin{array}{l} \text{製造工程の可視化} \leftarrow \text{設備稼働状況の即時把握} + \text{進捗実績の即時把握} \\ + (\text{在庫状況の迅速な把握} \leftarrow \text{在庫状況の迅速な把握} * \text{完成数量の即時把握}) \\ + \text{不良発生}の\text{早期通知} \end{array} \right)$$

式(2.12)

$$\left(\begin{array}{l} \text{製造工程の可視化} \leftarrow \text{設備稼働状況の即時把握} + \text{進捗実績の即時把握} \\ + \text{不良発生}の\text{早期通知} \end{array} \right)$$

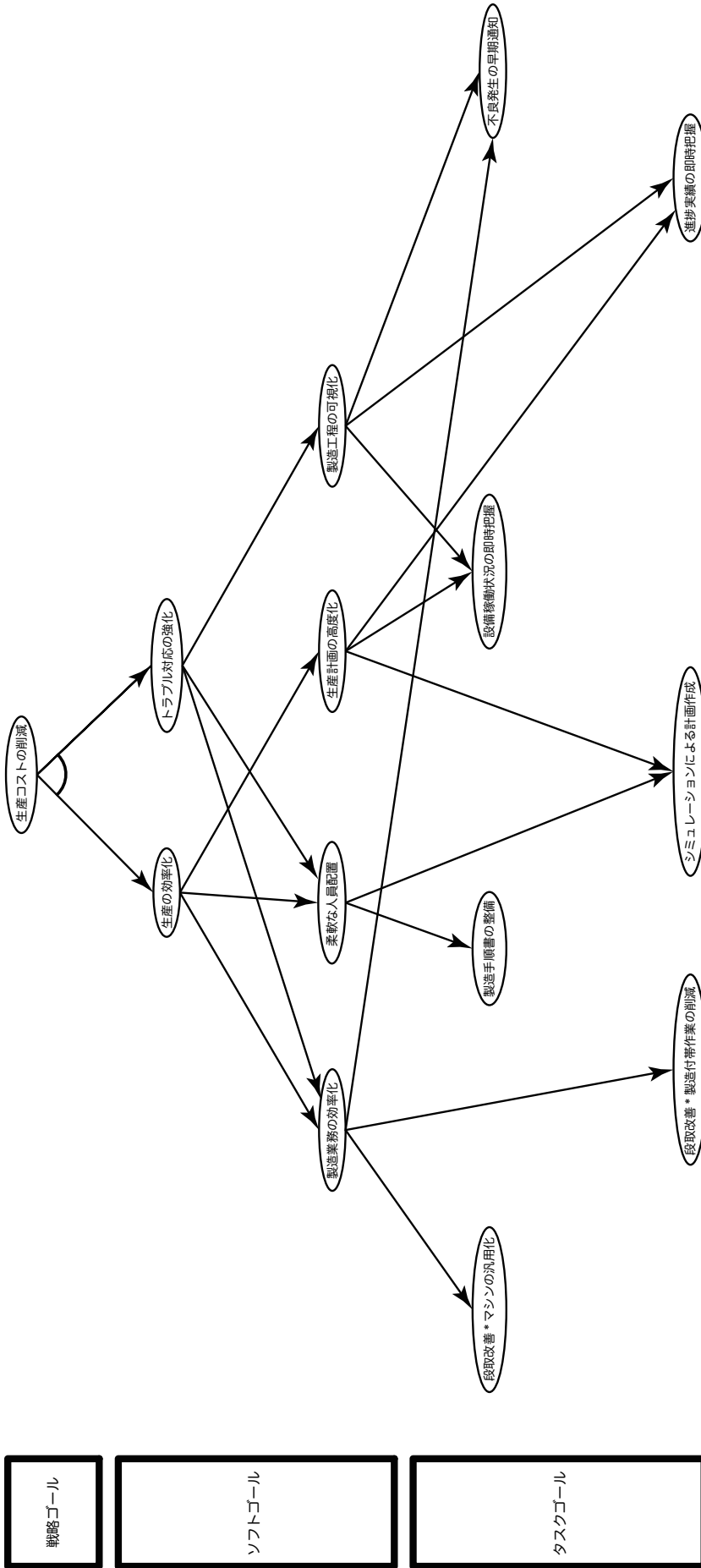


図 2.11: 簡約化後のゴールグラフ [斎藤 06, 山本 07]

(3) ゴールグラフの属性設定

属性値に基づくゴール選択手法では、「寄与度」、「有効度」、「充足度」という3つの属性が設定される。以下、それらの各々について設定方法を示す。簡約化後のゴールグラフ (図 2.11) にこれらの属性を設定したゴールグラフ [斎藤 06, 山本 07] を図 2.12 に示す。

(3a) 寄与度の設定

ステークホルダ (1.7 節 (4) 参照) 間で話し合いを行い、合意のもとで、ゴールグラフ上のすべての枝に対して寄与度 (2.3.2 節 (属性-1) 参照) を振る。

(3b) 有効度の設定

(3a) で設定された寄与度と式 (2.3) から、少なくとも1つのタスクゴールを子ゴールに持つゴールグラフ上のすべてのソフトゴールに対して、有効度 (2.3.2 節 (属性-2) 参照) を計算する。図 2.11 において、そのようなソフトゴールは、「製造業務の効率化」、「柔軟な人員配置」、「生産計画の高度化」、「製造工程の可視化」の4つ存在する。図 2.12 の通りに寄与度が設定されたため、各々の有効度は、 $Vld(\text{製造業務の効率化}) = \frac{8+7}{2} = 7.5$, $Vld(\text{柔軟な人員配置}) = \frac{7+6}{2} = 6.5$, $Vld(\text{生産計画の高度化}) = \frac{5}{1} = 5$, $Vld(\text{製造工程の可視化}) = \frac{8}{1} = 8$ となる。

(3c) 充足度の設定

属性値に基づくゴール選択手法では、簡約化後のすべてのタスクゴールに対して充足度を設定する。簡約化後のゴールグラフ (図 2.11) では、7つのタスクゴールが存在する。その各々に対して、表 2.2 に示す通りに KPI (2.3.2 節 (属性-3) 参照) が設定された [斎藤 06, 山本 07]。表 2.2 において、任意のタスクゴールの目標値が現状よりも高く設定されているため、式 (2.4) から、任意のタスクゴールの充足度は 0 となる。

表 2.2: 図 2.12 のタスクゴールの KPI [斎藤 06, 山本 07]

タスクゴール (t)	KPI (k_t)	X	Y
段取改善 * マシンの汎用化	単位時間あたりの生産量 [個/時]	45	35
段取改善 * 製造付帯作業の削減	設備稼働率 [%]	65	57
製造手順書の整備	整備済みの工程の割合 [%]	25	0
シミュレーションによる計画作成	計画用 PC の設置率 [%]	20	0
設備稼働状況の即時把握	状況把握までの時間 [任意の時間単位]	5 [分]	1 [時]
進捗実績の即時把握	実績把握までの時間 [任意の時間単位]	5 [分]	0.5 [日]
不良発生の早期通知	通知までの時間 [任意の時間単位]	5 [分]	1 [日]

X: 目標値 ($k_{t,o}$)

Y: 現状 ($k_{t,n}$)

(4) タスクゴールの選択

(3b) で有効度が設定されたソフトゴールを親ゴールとする 2 階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) に対して、狭義のゴール選択 (1.7 節 (10) 参照) を行う。ただし、選択されるタスクゴールは、1つのみとする。タスクゴールの選択は、次の手順で行う。(4a) 選

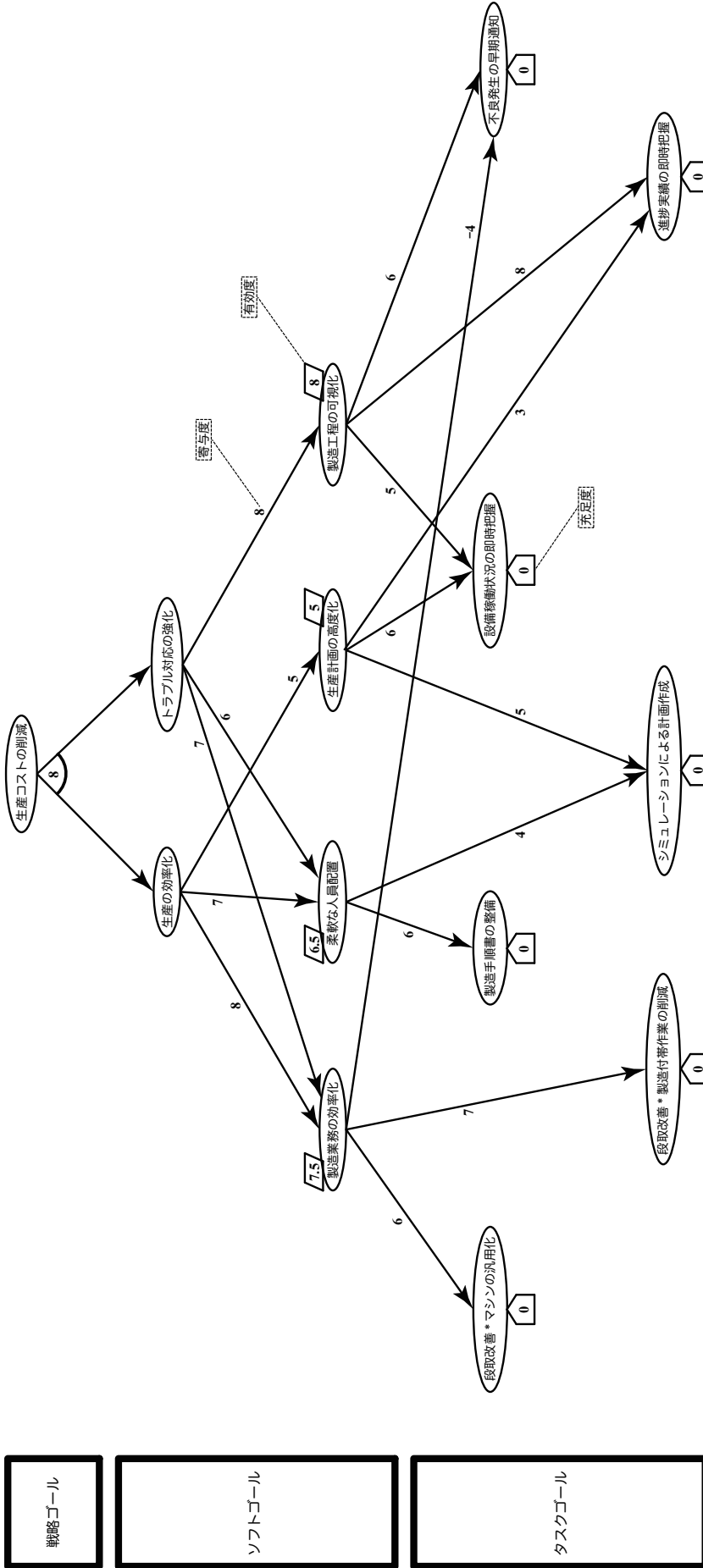


図 2.12: 属性設定後のゴールグラフ [斎藤 06, 山本 07]

択不要ソフトゴールの抽出, (4b) ソフトゴールの選択, (4c) タスクゴールの選択, (4d) 被影響ソフトゴールの抽出, (4e) 終了判定. 以下, 各々の手順の詳細を述べる.

(4a) 選択不要ソフトゴールの抽出

選択不要ソフトゴール s^* は次式で定義される.

$$s^* \text{ は選択不要ソフトゴールである} : \Leftrightarrow s^* \in PSG \wedge |SUB_{s^*}^+| = 1 \quad (2.15)$$

s^* に相当するソフトゴールが少なくとも1つ存在するならば, そのうちの1つを s^* とし, 以下の手順を行った後, (4d) へ進む. そうでなければ, 以下の手順を行わず, (4b) へ進む.

s^* を親ゴールとする2階層-OR分解-部分ゴールグラフでは, ゴール選択候補となるタスクゴールが1つのみであるため, そのタスクゴールのみを選択する.

$$SEL_{s^*} := SUB_{s^*}^+ \quad (2.16)$$

s^* と $\exists! t^* \in SEL_{s^*}$ は, 選択済みとなるため, これらは PSG と PTG から各々除かれる.

$$PSG := PSG \setminus \{s^*\} \quad (2.17)$$

$$PTG := PTG \setminus \{t^*\} \quad (2.18)$$

(4b) ソフトゴールの選択

選択されるソフトゴール $s^\#$ は, 次式で定義される.

$$\begin{aligned} s^\# \text{ は選択されるソフトゴールである} \\ : \Leftrightarrow s^\# \in PSG \text{ s.t. } Vld(s^\#) = \max_{s \in PSG} \{Vld(s)\} \end{aligned} \quad (2.19)$$

$s^\#$ に相当するソフトゴールが複数存在する場合, そのうちの1つを $s^\#$ とする.

$s^\#$ は選択済みとなるため, PSG から除かれる.

$$PSG := PSG \setminus \{s^\#\} \quad (2.20)$$

(4c) へ進む.

(4c) タスクゴールの選択

$s^\#$ を親ゴールとする2階層-OR分解-部分ゴールグラフにおいて選択されるタスクゴー

ル $t^\#$ は, 次式で定義される.

$$t^\# \text{ は選択されるタスクゴールである}$$

$$:\Leftrightarrow \begin{cases} t^\# \in SUB_{s^\#}^+ \text{ s.t. } Cov(t^\#, s^\#) = \max_{t \in SUB_{s^\#}^+} \{Cov(t, s^\#)\} \\ \quad \text{if } (\forall t^\# \in PTG)(Acv(t^\#, k_t) = 1), \\ t^\# \in SUB_{s^\#}^+ \text{ s.t. } Cov(t^\#, s^\#) = \max_{t \in SUB_{s^\#}^+ \text{ s.t. } Acv(t, s^\#)=0} \{Cov(t, s^\#)\} \\ \quad \text{otherwise} \end{cases} \quad (2.21)$$

$t^\#$ に相当するタスクゴールが複数存在する場合, そのうちの1つを $t^\#$ とする.

式(2.21)のうち, 上式が成立する場合, 任意の $t \in PTG$ に対して, $Acv(t, k_t) := 0$ かつ $k_{t,n} := k_{t,o}$ とし, その後, $k_{t,o}$ を再定義する.

$s^\#$ の子ゴールのうち, $t^\#$ が選択されるため, 次式が成立する.

$$SEL_{s^\#} := \{t^\#\} \quad (2.22)$$

$s^\#$ を親ゴールとする2階層-OR分解-部分ゴールグラフにおいて, $t^\#$ が選択されるとき, またそのときに限り, $t^\#$ 以外は選択されない. そのため, $s^\#$ に接続する任意のタスクゴールは選択されるか否かが決定するため, PTG から除かれる.

$$PTG := PTG \setminus SUB_{s^\#} \quad (2.23)$$

(4d)へ進む.

(4d) 被影響ソフトゴールの抽出

被影響ソフトゴール s^\natural は, 次式で定義される.

$$s^\natural \text{ は被影響ソフトゴールである}$$

$$:\Leftrightarrow \begin{cases} s^\natural \in PSG \wedge s^\natural \neq s^* \wedge t^* \in SUB_{s^\natural}^+ & \text{if } SEL_{s^*} \neq \emptyset, \\ s^\natural \in PSG \wedge s^\natural \neq s^\# \wedge t^\# \in SUB_{s^\natural}^+ & \text{otherwise} \end{cases} \quad (2.24)$$

被影響ソフトゴールを親ゴールとする2階層-OR分解-部分ゴールグラフは, t^* または $t^\#$ を選択する.

$$(\forall s^\natural \in PSG) \left(SEL_{s^\natural} := \begin{cases} \{t^*\} & \text{if } SEL_{s^*} \neq \emptyset, \\ \{t^\#\} & \text{otherwise} \end{cases} \right) \quad (2.25)$$

s^\natural は選択済みとなるため, PSG から除かれる.

$$(\forall s^\natural \in PSG) (PSG := PSG \setminus \{s^\natural\}) \quad (2.26)$$

(4c)と同様に, s^d に接続する任意のタスクゴールは選択されるか否かが決定するため, PTG から除かれる.

$$(\forall s^d \in PSG)(PTG := PTG \setminus SUB_{s^d}) \quad (2.27)$$

(4e)へ進む.

(4e) 終了判定

終了条件は, 次式で定義される.

$$\text{終了条件を満たす} :\Leftrightarrow (\forall s \in PSG^I)(SEL_s \neq \emptyset) \quad (2.28)$$

この終了条件が成立するとき, またそのときに限り, $PSG = \emptyset$ かつ $PTG = \emptyset$ が成立する. 終了条件を満たさない場合は, (4a)へ戻る. 終了条件を満たす場合, 最終的に獲得される要求 r は, 次式で定義される.

$$r \text{ は要求である} :\Leftrightarrow r \in \bigcup_{s \in PSG^I} SEL_s \quad (2.29)$$

以上の手順(4a)~(4e)を, 図2.12に対して適用する. ただし, 簡単のため, 有効度が設定されているソフトゴールと, タスクゴールの各々に対して, 次のように大文字アルファベットの添字が対応しているものとする. $A \mapsto$ 「製造業務の効率化」, $B \mapsto$ 「柔軟な人員配置」, $C \mapsto$ 「生産計画の高度化」, $D \mapsto$ 「製造工程の可視化」, $E \mapsto$ 「段取改善*マシンの汎用化」, $F \mapsto$ 「段取改善*製造付帯作業の削減」, $G \mapsto$ 「製造手順書の整備」, $H \mapsto$ 「シミュレーションによる計画作成」, $I \mapsto$ 「設備稼働状況の即時把握」, $J \mapsto$ 「進捗実績の即時把握」, $K \mapsto$ 「不良発生 of 早期通知」.

1回目

(4a₁) 選択不要ソフトゴールの抽出

分析開始時点では,

$$PSG^I := PSG = \{A, B, C, D\} \text{ および } PTG = \{E, F, G, H, I, J, K\}$$

である. 式(2.15)から, 選択不要ソフトゴールを要素とする集合は,

$$\left\{ s^* \mid s^* \in PSG \wedge |SUB_{s^*}^+| = 1 \right\} = \emptyset$$

であり, 選択不要ソフトゴールは存在しないことがわかる. そのため, (4b)へ進む.

(4b₁) ソフトゴールの選択

式(2.19)から、選択されるソフトゴールの候補を要素とする集合は、

$$\left\{ s \in PSG \mid Vld(s) = \max_{s' \in PSG} \{Vld(s')\} \right\} = \{D\}$$

であることがわかる。この集合はDのみを要素とする一点集合であるため、 $s^\# = D$ となる。Dは選択済みとなるため、 PSG から除かれる。

$$PSG := PSG \setminus \{D\} = \{A, B, C\}$$

(4c₁) タスクゴールの選択

式(2.21)から、選択されるタスクゴールの候補を要素とする集合は、

$$\left\{ t \in SUB_D^+ \mid Cov(t, D) = \max_{t' \in SUB_D^+ \text{ s.t. } Acv(t', D)=0} \{Cov(t', D)\} \right\} = \{J\}$$

であることがわかる。この集合はJのみを要素とする一点集合であるため、 $t^\# = J$ ($SEL_D = \{J\}$)となる。Dに接続する任意のタスクゴールは、選択済みか否かが決定するため、 PTG から除かれる。

$$PTG := PTG \setminus SUB_D = \{E, F, G, H\}$$

(4d₁) 被影響ソフトゴールの抽出

式(2.24)から、被影響ソフトゴールを要素とする集合は、

$$\left\{ s^\sharp \in PSG \mid s^\sharp \neq D \wedge J \in SUB_{s^\sharp}^+ \right\} = \{C\}$$

であることがわかる。この集合はCのみを要素とする一点集合であるため、 $s^\sharp = C$ となる。Cは選択済みとなるため、 PSG から除かれる。

$$PSG := PSG \setminus \{C\} = \{A, B\}$$

(4e₁) 終了判定

式(2.28)が成立しない ($\because SEL_A = \emptyset, SEL_B = \emptyset$) ため、(4a)へ戻る。

2回目

(4a₂) 選択不要ソフトゴールの抽出

分析開始時点では、

$$PSG = \{A, B\} \text{ および } PTG = \{E, F, G, H\}$$

である。式 (2.15) から、選択不要ソフトゴールを要素とする集合は、

$$\left\{ s^* \mid s^* \in PSG \wedge |SUB_{s^*}^+| = 1 \right\} = \emptyset$$

であり、選択不要ソフトゴールは存在しないことがわかる。そのため、(4b)へ進む。

(4b₂) ソフトゴールの選択

式 (2.19) から、選択されるソフトゴールの候補を要素とする集合は、

$$\left\{ s \in PSG \mid Vld(s) = \max_{s' \in PSG} \{Vld(s')\} \right\} = \{A\}$$

であることがわかる。この集合は A のみを要素とする一点集合であるため、 $s^\# = A$ となる。A は選択済みとなるため、PSG から除かれる。

$$PSG := PSG \setminus \{A\} = \{B\}$$

(4c₂) タスクゴールの選択

式 (2.21) から、選択されるタスクゴールの候補を要素とする集合は、

$$\left\{ t \in SUB_A^+ \mid Cov(t, A) = \max_{t' \in SUB_A^+ \text{ s.t. } Act(t', A)=0} \{Cov(t', A)\} \right\} = \{F\}$$

であることがわかる。この集合は F のみを要素とする一点集合であるため、 $t^\# = F$ ($SEL_A = \{F\}$) となる。A に接続する任意のタスクゴールは、選択済みか否かが決定するため、PTG から除かれる。

$$PTG := PTG \setminus SUB_A = \{G, H\}$$

(4d₂) 被影響ソフトゴールの抽出

式 (2.24) から、被影響ソフトゴールを要素とする集合は、

$$\left\{ s^\sharp \in PSG \mid s^\sharp \neq A \wedge F \in SUB_{s^\sharp}^+ \right\} = \emptyset$$

であることがわかる。被影響ソフトゴールは存在しないため、(4e)へ進む。

(4e₂) 終了判定

式 (2.28) が成立しない ($\because SEL_B = \emptyset$) ため、(4a)へ戻る。

3回目

(4a₃) 選択不要ソフトゴールの抽出

分析開始時点では,

$$PSG = \{B\} \text{ および } PTG = \{G, H\}$$

である. 式 (2.15) から, 選択不要ソフトゴールを要素とする集合は,

$$\left\{ s^* \mid s^* \in PSG \wedge |SUB_{s^*}^+| = 1 \right\} = \emptyset$$

であり, 選択不要ソフトゴールは存在しないことがわかる. そのため, (4b)へ進む.

(4b₃) ソフトゴールの選択

式 (2.19) から, 選択されるソフトゴールの候補を要素とする集合は,

$$\left\{ s \in PSG \mid Vld(s) = \max_{s' \in PSG} \{Vld(s')\} \right\} = \{B\}$$

であることがわかる. この集合は B のみを要素とする一点集合であるため, $s^\# = B$ となる. B は選択済みとなるため, PSG から除かれる.

$$PSG := PSG \setminus \{B\} = \emptyset$$

(4c₃) タスクゴールの選択

式 (2.21) から, 選択されるタスクゴールの候補を要素とする集合は,

$$\left\{ t \in SUB_B^+ \mid Cov(t, B) = \max_{t' \in SUB_B^+ \text{ s.t. } Acv(t', B)=0} \{Cov(t', B)\} \right\} = \{G\}$$

であることがわかる. この集合は G のみを要素とする一点集合であるため, $t^\# = G$ ($SEL_B = \{G\}$) となる. B に接続する任意のタスクゴールは, 選択済みか否かが決定するため, PTG から除かれる.

$$PTG := PTG \setminus SUB_B = \emptyset$$

(4d₃) 被影響ソフトゴールの抽出

式 (2.24) から, 被影響ソフトゴールを要素とする集合は,

$$\left\{ s^\sharp \in PSG \mid s^\sharp \neq A \wedge F \in SUB_{s^\sharp}^+ \right\} = \emptyset$$

であることがわかる. 被影響ソフトゴールは存在しないため, (4e)へ進む.

(4e₃) 終了判定

式 (2.28) が成立するため, 終了する. 式 (2.29) から, 最終的に獲得される要求

は,

$$\bigcup_{s \in PSG^I} SEL_s = \{F, G, J\}$$

の要素である F, G, J, すなわち, 「段取改善 * 製造付帯作業の削減」, 「製造手順書の整備」, 「進捗実績の即時把握」である.

属性値に基づくゴール選択手法では, 戦略ゴールの達成に必要とされる要求として, その時点で未達成なタスクゴールのうち, 最も優先順位が高い最小限の要求を獲得することができる. しかし, 獲得された要求が1つでも達成できなかった場合, 戦略ゴールが達成できなくなってしまうことから, ロバスト性の向上のために, 優先順位が高い順に, 開発期間や予算などの制約を考慮に入れた上で, なるべく多くの要求を獲得しておくべきである.

反復型ライフサイクルモデル (図 2.2) を採用している場合, 繰り返し要求分析が行われる. 2回目以降の要求分析では, 前回の要求分析時に獲得された要求が達成されたかどうかのフィードバックが得られる. そのため, 本手法と併用することによって, ロバスト性を考慮した複数の要求を獲得可能である. これに対して, 落水型ライフサイクルモデル (図 2.1) を採用している場合, 要求分析は一度しか行われなため, 獲得された要求が達成されたかどうかのフィードバックが得られない. したがって, 落水型では, 本手法を用いて獲得される要求は常に一定であるため, ロバスト性を考慮した複数の要求を獲得することはできない.

2.4 本章のまとめ

2.1 節では, 情報システム開発における要求獲得の重要性について述べた. また, 既存の要求獲得方法について概説し, ゴール指向要求分析はその中の1つに位置付けられることを述べた. 他の要求獲得方法に対するゴール指向要求分析の特徴についても言及した.

2.2 節では, ゴール指向要求分析の概要について述べた. ゴール指向要求分析の黎明期から現在に至るまでに登場した様々なゴール指向要求分析法の特徴を, 登場した年代当時のソフトウェア工学における問題意識や社会情勢と関連させて述べ, その発展経緯について概説した.

2.3 節では, 既存のゴール指向要求分析法のうち, AGORA [Kaiya 02] と属性値に基づくゴール選択手法 [斎藤 06, 山本 07] について述べた. これらは共に, ゴールに属性を設定することによって, ゴール指向要求分析に基づく要求獲得を支援する.

AGORA は, 満足度 (2.3.1 節 (属性-2) 参照) としてステークホルダの選好を採取することで, 妥当性 (1.7 節 (7) 参照) を考慮に入れた要求獲得を支援している. しかしながら, AGORA は, 要求獲得のための具体的な手順は規定しない [斎藤 06, 山崎 06, 山本 07]. 一方, 属性値に基づくゴール選択手法は, 寄与度 (2.3.2 節 (属性-1) 参照) に基づいた狭義のゴール選択 (1.7 節 (10) 参照) が行われるけれども, 寄与度に基づくゴール選択は, 妥当性を反映しているかどうかの保証はないため, 獲得された要求に基づいて開発された機能が, 結果として使われず, 無駄になる可能性がある.

したがって、寄与度だけではなく、妥当性を考慮に入れたゴール選択指標を確立し、それに基づいたゴール選択が行われるべきである。しかし、ゴール選択基準によって選択される子ゴールが異なる場合、狭義のゴール選択を行うことができない。本論文では、このことを「ゴール選択基準間のコンフリクト」と呼ぶ。このコンフリクトを考慮に入れた上でゴール選択を行うためには、まず、ゴールグラフ上でこのコンフリクトを持つゴールを検出することが必要である。第3章では、このコンフリクトを形式的に検出する方法について述べる。

ゴール選択基準間のコンフリクトの 形式的検出方法

3.0 本章の概要

狭義のゴール選択 (1.7 節 (10) 参照) は, ゴール指向要求分析に基づく要求獲得を行う上での重要な意思決定問題である. 2 章で述べた通り, 要求獲得は考慮されるべき問題が多岐に渡るため, 複数の観点からのゴール選択基準を設定することが可能である. しかし, ゴール選択基準の増加に伴い, 複数のゴール選択基準間でコンフリクトが生じる場合がある. 本章では, その具体例として, 3.1 節で「貢献度」と「顧客のニーズに関する妥当性」という 2 つのゴール選択基準の間でコンフリクトがあることを例示する. 3.2 節では, ゴールグラフ上の貢献度と顧客のニーズに関する妥当性の間のコンフリクトを持つ 2 階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) のすべての親ゴールを検出するアルゴリズムを提案する. 3.3 節では, ゴールグラフ上の任意の 2 階層-部分ゴールグラフ (1.7 節 (9) 参照) に対して, 貢献度と顧客のニーズに関する妥当性の間のコンフリクトを検出可能にする方法を提案する.

3.1 貢献度と顧客のニーズに関する妥当性の間のコンフリクト

3.1.1 顧客のニーズに関する妥当性

「貢献度」(2.3.1 節 (属性-1) 参照) は, ゴール選択基準として利用できる (2.3.2 節参照). 貢献度に基づくゴール選択は, 親ゴールの達成に最適な子ゴールを選択する. しかし, 貢献度は高いけれども, ステークホルダが属する組織特有の制約などによって, 実際には達成が困難な子ゴールである可能性がある. また, 貢献度が低く, かつ, 現状では達成が困難な子ゴールであっても, 経営戦略上どうしても達成しておきたい子ゴールが存在する場合も考えられる. したがって, 貢献度だけでなく, 貢献度では考慮されない, 子ゴールを達成することの妥当性 (1.7 節 (7) 参照) も考慮に入れたゴール選択を行うべきである. ゴール指向要求分析は, 顧客 (1.7 節 (2) 参照) から要求を獲得するための方法である. そのため, 「顧客のニーズ」を反映していることの妥当性を考慮に入れたゴール選択

基準が必要となる。

本論文では、満足度行列 (図 2.6) から、ゴール g の「顧客のニーズに関する妥当性」を次式 $Cup(g)$ で定義する [佐藤 11].

$$Cup(g) := \frac{\sum_{\substack{s \in Stakeholder \\ c \in Customer}} m(g)_{s,c}}{|Stakeholder| \times |Customer|} \quad (3.1)$$

ただし、*Stakeholder* は、立場の異なる個々のステークホルダを要素とする集合。*Customer* は、*Stakeholder* に包まれる、顧客に該当するステークホルダの集合。 G は、ゴールグラフ上のすべてのゴールを要素として持つ集合であり、 $g \in G$ はゴールを表す。 $m(g)_{s,c}$ は、ステークホルダ $s \in Stakeholder$ から見た顧客 $c \in Customer$ にとっての、ゴール g の満足度である。たとえば、貢献度と満足度行列が振られた 2 階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) である図 3.1 [佐藤 11] では、 $Cup(\text{英語公用化}) = -7.5$, $Cup(\text{現地通訳雇用}) = -0.5$ である。したがって、

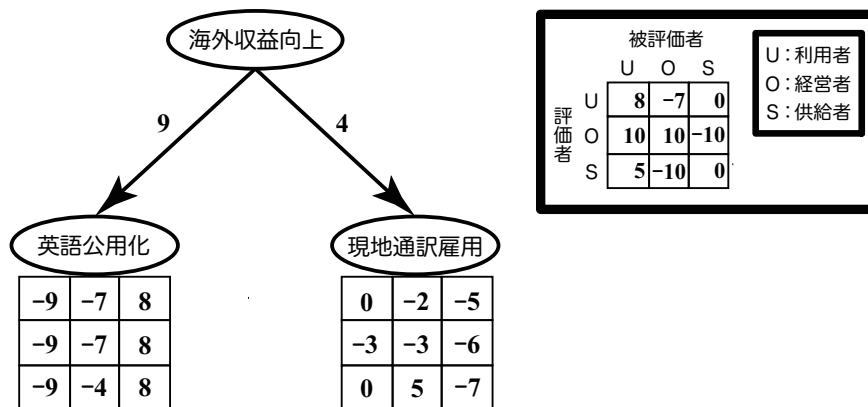


図 3.1: 貢献度-妥当性-コンフリクトを持つ 2 階層-OR 分解-部分ゴールグラフの例 [佐藤 11, 佐藤 12c]

図 3.1 では、顧客のニーズに関する妥当性をゴール選択基準とする場合、子ゴールのうち $Cup(g)$ が最大である、「現地通訳雇用」が選択される。 $Cup(g)$ は、 $Cov(g)$ (式 (2.2)) 同様、 -10 から 10 までの値をとるため、貢献度と定量的な比較が可能である。

ゴール選択上、貢献度と顧客のニーズに関する妥当性が共に最も高い子ゴールを選択すべきである。しかし、貢献度と顧客のニーズに関する妥当性の間には、一方が高くても、他方が低い場合が存在する。貢献度と顧客のニーズに関する妥当性のどちらを優先するかで、ゴール選択結果が変化する場合、「貢献度と顧客のニーズに関する妥当性間のコンフリクト」(本論文では、簡単のため、「貢献度-妥当性-コンフリクト」と呼ぶ)がある [佐藤 11]. 貢献度-妥当性-コンフリクトは、ゴール選択基準間のコンフリクト (2.4 節参照) の 1 つである。

図 3.1 は、貢献度-妥当性-コンフリクトが存在する 2 階層-OR 分解-部分ゴールグラフの仮想的な例である [佐藤 11]. 図 3.1 における貢献度 ($Cov(g)$) と顧客のニーズに関する妥当性 ($Cup(g)$) の値を表 3.1 に示す。表 3.1 から、図 3.1 において貢献度と顧客のニーズに関する妥当性の中で優先順位の相違があり、貢献度と顧客のニーズに関する妥当性のどちらを優先するかでゴール選択結果が変

表 3.1: 図 3.1 における貢献度と顧客のニーズに関する妥当性の値 [佐藤 12c]

ゴール	貢献度 ($Cov(g)$)	顧客のニーズに関する妥当性 ($Cup(g)$)	X
英語公用化	9	-7.5	有
現地通訳雇用	4	-0.5	

X: 貢献度-妥当性-コンフリクトの有無 ($Ccc(g)$)

わってしまうことがわかる。

3.1.2 コンフリクト検出指標

「ゴール選択基準間のコンフリクト」(2.4 節参照) は、次で定義される $Dip(g)$ で評価できる。 $Dip(g)$ は、ゴール選択基準によって選択されるゴールの優先順位の相違を判定する指標であり、ゴール $g \in G \setminus FG$ に対して、その子ゴールの少なくとも1つの対 (s, s') において、あるゴール選択基準 ($Gsc(s)$ と $Gsc(s')$) と別のあるゴール選択基準 ($Gsc'(s)$ と $Gsc'(s')$) の間で優先順位の相違がある場合は真、ない場合は偽の値をとる。

$$Dip(g) \text{ が真である} \Leftrightarrow \left\{ (s, s') \in Sub_g \times Sub_g \mid s \neq s', \right. \\ \left. \left((Gsc(s) < Gsc(s')) \wedge (Gsc'(s) > Gsc'(s')) \right) \vee \left((Gsc(s) > Gsc(s')) \wedge (Gsc'(s) < Gsc'(s')) \right) \right\} \neq \emptyset \quad (3.2)$$

ただし、 G は、ゴールグラフ上のすべてのゴールを要素として持つ集合であり、 $g \in G$ はゴールを表す。 FG は、最終ゴール (2.2 節参照) を要素とする集合。 Sub_g は、ダミーゴール (1.7 節 (8) 参照) 以外のゴール g に隣接するすべての子ゴールを要素とする集合。 $Gsc(g)$ は、ゴール g のあるゴール選択基準値。 $Gsc'(g)$ は、ゴール g の $Gsc(g)$ を除くあるゴール選択基準値。

「貢献度-妥当性-コンフリクト」(3.1.1 節参照) は、ゴール選択基準間のコンフリクトの1つである。したがって、貢献度-妥当性-コンフリクトを検出する指標は、式 (3.2) に基づいて定義することができる。具体的には、式 (3.2) の関数 Gsc と Gsc' を、各々「貢献度」(2.3.1 節 (属性-1) 参照) と「顧客のニーズに関する妥当性」(3.1.1 節参照) を求める関数に置換すればよい。すなわち、貢献度-妥当性-コンフリクトは、次で定義される $Ccc(g)$ で評価される。 $Ccc(g)$ は優先順位の相違を判定する指標であり、ゴール $g \in G \setminus FG$ に対して、その子ゴールの少なくとも1つの対 (s, s') において貢献度 ($Cov(s)$ と $Cov(s')$) と顧客のニーズに関する妥当性 ($Cup(s)$ と $Cup(s')$) の間で優

先順位の相違がある場合は真、ない場合は偽の値をとる [佐藤 11]注1).

$$Ccc(g) \text{ が真である} \Leftrightarrow \left\{ (s, s') \in Sub_g \times Sub_g \mid s \neq s', \right. \\ \left. \left((Cov(s) < Cov(s')) \wedge (Cup(s) > Cup(s')) \right) \vee \left((Cov(s) > Cov(s')) \wedge (Cup(s) < Cup(s')) \right) \right\} \neq \emptyset \quad (3.3)$$

ただし、 G は、ゴールグラフ上のすべてのゴールを要素として持つ集合であり、 $g \in G$ はゴールを表す。 FG は、最終ゴールを要素とする集合。 Sub_g は、ダミーゴール (1.7 節 (8) 参照) 以外のゴール g に隣接するすべての子ゴールを要素とする集合。 $Cov(g)$ は、ゴール g の親ゴールへの貢献度 (式 (2.2)). $Cup(g)$ は、ゴール g の顧客のニーズに関する妥当性 (式 (3.1)).

3.2 コンフリクト検出アルゴリズム

ある頂点と連結するすべての頂点を訪問するアルゴリズムの1つに、「深さ優先探索」がある [Aho 83]. 任意のゴールグラフ上の貢献度-妥当性-コンフリクト (3.1 節参照) を持つすべてのゴールを検出するアルゴリズムは、深さ優先探索に立脚して構築可能である [佐藤 12a]. 本論文では、このアルゴリズムを「*ccc-dfs*」と呼ぶ。以下、*ccc-dfs* について詳述する。ただし、本論文のアルゴリズムはすべて、[Aho 83] で用いられている Pascal 類似の疑似言語に基づいて記述される。

深さ優先探索は、次のようなアルゴリズムである [Aho 83].

深さ優先探索 [Aho 83]

有効グラフ D があって、最初すべての頂点に *unvisited* (未訪問) というマークがついているものとする。深さ優先探索では、 D 中の1つの頂点 v を出発点として定める。 v には *visited* (訪問済) というマークを付ける。次に深さ優先探索を再帰的に使い、 v に隣接する未訪問の頂点を1つずつ調べる。 v から行ける頂点を全部訪問し終わった時点で、 v に関する探索は終了する。まだ未訪問の頂点が残っていたら、そのうちの1つを新しい出発点として選ぶ。 D の頂点を全部訪問するまでこれを繰り返す。

深さ優先探索の手続きは図 3.2 の通りである。ここで、 v, w は頂点を表す変数。 $mark[v]$ は、 v が未訪問であるか訪問済であるかを表す、*unvisited* または *visited* のブール値を要素とするリストである。 $L[v]$ は隣接リストであり、 v に隣接する頂点を要素として持つ。

次に、*ccc-dfs* の手続き [佐藤 12a] を図 3.3 に示す。深さ優先探索と *ccc-dfs* の違いは、訪れる各頂点 v に対して、貢献度-妥当性-コンフリクト検出を行うか否かだけである。この処理は、図 3.2 において、4 行目と 6 行目の間で行えばよい。図 3.3 では、6 行目でこの処理が行われている。一連の処理は、*ccc* という手続きにまとめられている。

ccc の詳細 [佐藤 12a] を図 3.4 に示す。図 3.4 の 11 行目の制御式「 $n.and = 0 \text{ and } n.or \geq 2$ 」は、 v が、 v の任意の子ゴールと OR 分解のみで接続しているかどうかを判定している。この制御式が真である場合、 v は貢献度-妥当性-コンフリクト検出の対象となるゴールである。その場合、 v の

```

1: procedure dfs ( v: vertex );
2: var
3:   w: vertex;
4: begin
5:   mark[v] := visited;
6:   for  $L[v]$  上の各頂点 w do
7:     if mark[w] = unvisited then
8:       dfs(w)
9:   end; { dfs }

```

図 3.2: 深さ優先探索 [Aho 83]

```

1: procedure ccc-dfs ( v: vertex );
2: var
3:   w: vertex;
4: begin
5:   mark[v] := visited;
6:   ccc(v);
7:   for  $L[v]$  上の各頂点 w do
8:     if mark[w] = unvisited then
9:       ccc-dfs(w)
10:  end; { ccc-dfs }

```

図 3.3: *ccc-dfs* [佐藤 12a]

コンフリクトの有無を調べるために、関数 *dcv* が呼ばれる。 *dcv* は、引数で与えられるゴール *v* を親ゴールとする 2 階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) において、自分自身を除くすべての子ゴールの対 (*w*, *w'*) を調べる関数である。少なくとも 1 つの (*w*, *w'*) において、貢献度と顧客のニーズに関する妥当性の間で優先順位の相違がある場合、1 を返し、そうでない場合、0 を返す。1 を返す場合、*v* は貢献度-妥当性-コンフリクトを持ち、0 を返す場合、*v* はこのコンフリクトを持たないことになる。 *dcv* の詳細 [佐藤 12a] を図 3.5 に示す。

ここで、*M* は、対象とするゴールグラフ上の任意の 2 階層-部分ゴールグラフの子ゴールの最大数以上の数が設定されているものとする。 *cov*(*w*) は、*w* の *v* への貢献度を返す関数。 *cup*(*w*) は、*w* の顧客のニーズに関する妥当性を返す関数。 *num*(*w*) は、*w* の添字を返す関数である。ただし、*w* の添字は、0 を含まない自然数全体の集合の要素であり、ゴールグラフ上の任意の 2 階層-OR 分解-部分ゴールグラフにおいて、任意の子ゴールが親ゴールよりも大きな値に対応するように付番する。 *L.cv*, *L.cp*, *L.nm* は、各々、*cov*(*w*), *cup*(*w*), *num*(*w*) を要素とするリストである。

ccc-dfs の評価のために、これを実装したコンピュータプログラムを作成した。このコンピュータプログラムを図 3.1 の 2 階層-OR 分解-部分ゴールグラフに対して適用した結果 [佐藤 12a] を、図

```

1: procedure ccc ( v: vertex );
2: var
3:   w: vertex;
4:   n_and, n_or: integer; n_and := 0; n_or := 0;
5: begin
6:   for  $L[v]$  上の各頂点 w do
7:     if w が v の子ゴール and w と v は AND 分解で接続 then
8:       n_and := n_and + 1
9:     else if w が v の子ゴール and w と v は OR 分解で接続 then
10:      n_or := n_or + 1;
11:   if n_and = 0 and n_or >= 2 then begin
12:     { v は検出対象 (OR 分解のみ) };
13:     if dcv(v) = 1 then
14:       { v はコンフリクト有り }
15:     else
16:       { v はコンフリクト無し }
17:   else
18:     { v は検出対象外のゴール }
19:   end
20: end; { ccc }

```

図 3.4: *ccc* (図 3.3 の 6 行目の処理) [佐藤 12a]

3.6 (満足度行列 (図 2.6) のうち、縦数値対立 (2.3.1.3 節参照) がある列の成分または対角数値対立 (2.3.1.3 節参照) がある主対角成分は、黄色で強調されている) に示す。ただし、1 は「海外収益向上」、2 は「現地通訳雇用」、3 は「英語公用化」の添字である。出発点は 1 とした。図 3.6 では、1 のみがその任意の子ゴール (2 と 3) に対して OR 分解で接続しているため、検出対象となる。そして、2 と 3 の間に優先順位の相違があり、少なくとも 1 つの子ゴールの対の間で優先順位の相違があることから、1 は貢献度-妥当性-コンフリクトを持つ。一方、2 と 3 は、最終ゴール (2.2 節参照) であり、子ゴールを持たないため、検出対象とならない。図 3.6 から、作成したコンピュータプログラムは、以上の事実を正しく反映していることがわかる。この結果は、*ccc-dfs* の正しさを示すものである。

次に、*ccc-dfs* が、一般のゴールグラフに対して適用可能であることを確認するために、*ccc-dfs* を、図 2.10 に貢献度 (2.3.1 節 (属性-1) 参照) および満足度行列 (図 2.6) を振った図 3.7 に対して適用した結果を、図 3.8 に示す。図 3.7 には、各ゴールに対応する添字も併せて記載している。

図 3.8 では、図 3.7 中のすべてのゴールに対して、貢献度-妥当性-コンフリクトの検出対象であるか否かが判定されており、図 3.7 中で、2 つ以上の子ゴールと OR 分解のみで接続された 6 つのゴール (2 (「生産の効率化」), 3 (「トラブル対応の強化」), 6 (「生産計画の最適化」), 7 (「製造工程の可

```

1: function dcv ( v: vertex ): integer;
2: var
3:   w: vertex;
4:   k, l, m, con: integer; k := 0; con := 0;
5:   L.cv: array [0..M + 1] of integer;
6:   L.cp: array [0..M + 1] of real;
7:   L.nm: array [0..M + 1] of integer;
8: begin
9:   for L[v] 上の各頂点 w do
10:    if w が v の子ゴール then begin
11:      k := k + 1;
12:      L.cv[k] := cov(w);
13:      L.cp[k] := cup(w);
14:      L.nm[k] := num(w)
15:    end;
16:    for l := 1 to k do begin
17:      for m := 1 to k do begin
18:        if l < m then
19:          if ((L.cv[l] > L.cv[m]) and (L.cp[l] < L.cp[m])) or ((L.cv[l] < L.cv[m])
and (L.cp[l] > L.cp[m])) then begin
20:            { L.nm[l] と L.nm[m] は優先順位の相違有り };
21:            if con = 0 then con := 1
22:          end
23:        else
24:          { L.nm[l] と L.nm[m] は優先順位の相違無し };
25:          m := m + 1
26:        end;
27:      l := l + 1
28:    end;
29:    return con
30: end; { dcv }

```

図 3.5: *dcv* (図 3.4 の 13 行目の処理) [佐藤 12a]

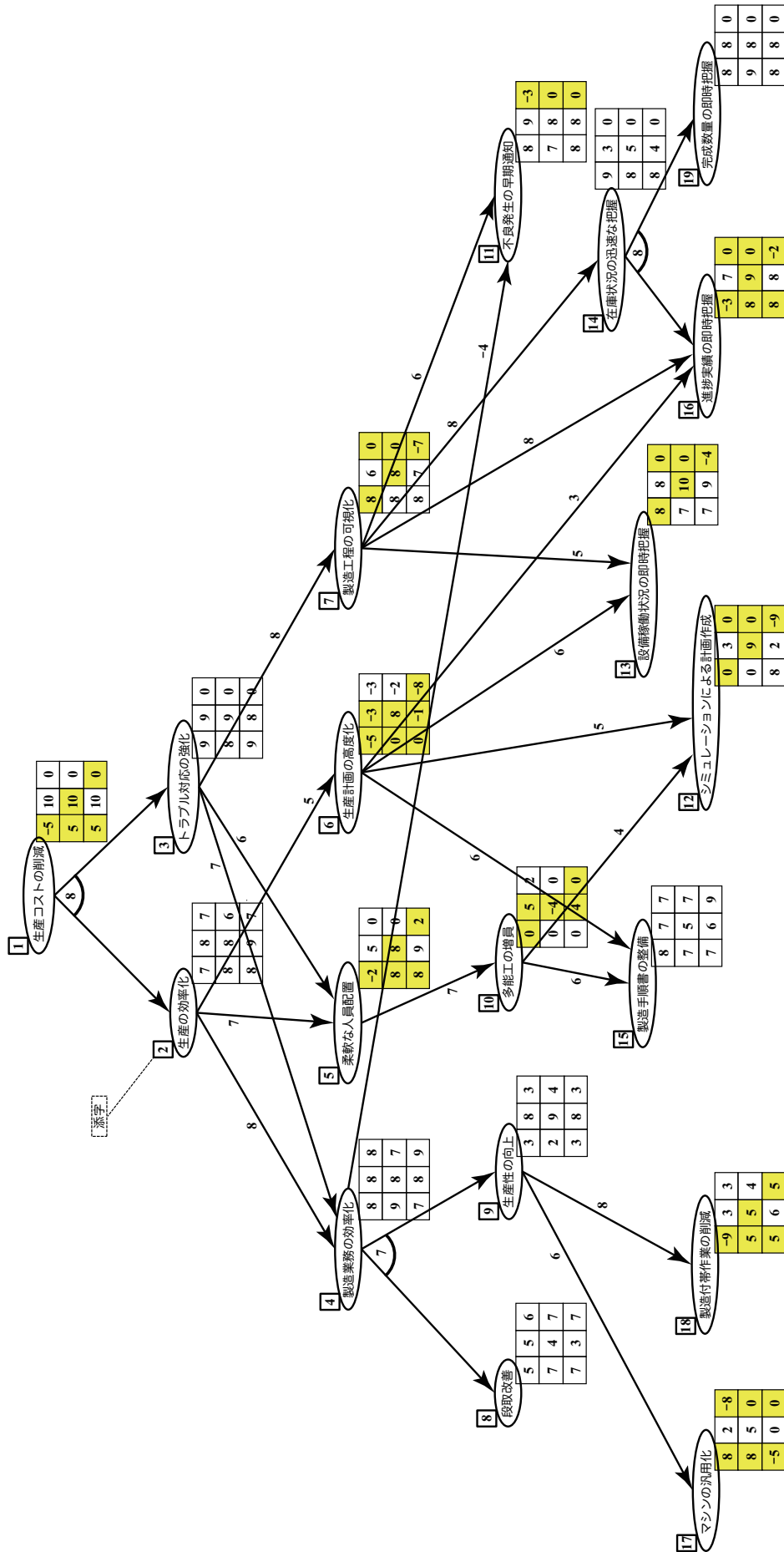


図 3.7: 図 2.10 に貢献度および満足度行列を振ったゴールグラフ

```

1: >ccc-dfs.exe
2: 1は検出対象外のゴール
3: 2は検出対象
4: 4と5は優先順位の相違無し
5: 4と6は優先順位の相違無し
6: 5と6は優先順位の相違無し
7: 2はコンフリクト無し
8: 4は検出対象外のゴール
9: 3は検出対象
10: 4と5は優先順位の相違無し
11: 4と7は優先順位の相違有り
12: 5と7は優先順位の相違無し
13: 3はコンフリクト有り
14: 5は検出対象外のゴール
15: 10は検出対象
16: 12と15は優先順位の相違無し
17: 10はコンフリクト無し
18: 12は検出対象外のゴール
19: 6は検出対象
20: 12と13は優先順位の相違無し
21: 12と15は優先順位の相違無し
22: 12と16は優先順位の相違有り
23: 13と15は優先順位の相違無し
24: 13と16は優先順位の相違無し
25: 15と16は優先順位の相違無し
26: 6はコンフリクト有り
27: 13は検出対象外のゴール
28: 7は検出対象
29: 11と13は優先順位の相違有り
30: 11と14は優先順位の相違有り
31: 11と16は優先順位の相違有り
32: 13と14は優先順位の相違有り
33: 13と16は優先順位の相違有り
34: 14と16は優先順位の相違無し
35: 7はコンフリクト有り
36: 11は検出対象外のゴール
37: 14は検出対象外のゴール
38: 16は検出対象外のゴール
39: 19は検出対象外のゴール
40: 15は検出対象外のゴール
41: 8は検出対象外のゴール
42: 9は検出対象
43: 17と18は優先順位の相違有り
44: 9はコンフリクト有り
45: 17は検出対象外のゴール
46: 18は検出対象外のゴール

```

図 3.8: 図 3.7 に対する貢献度-妥当性-コンフリクト検出結果

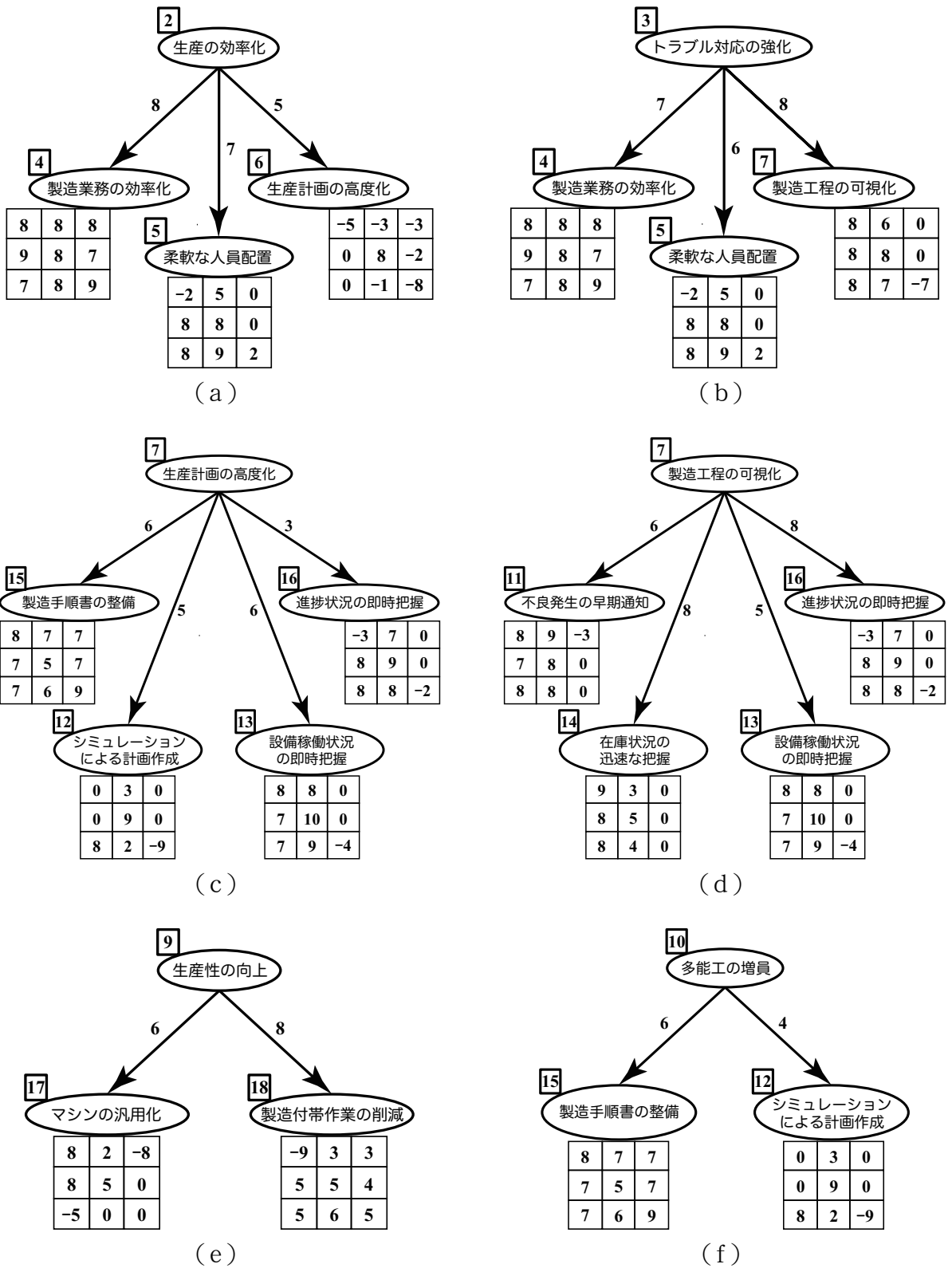


図 3.9: 図 3.7 において, 2つ以上の子ゴールを持つ 2 階層-OR 分解-部分ゴールグラフ

表 3.2: 図 3.9(a)~(f) の $Cov(g)$ と $Cup(g)$ の比較

(a)

V	W	X ^{注2)}	Y	Z
(4, 5)	$8 > 7$	$8 > 6$	無	
(4, 6)	$8 > 5$	$8 > -0.167$	無	偽
(5, 6)	$7 > 5$	$6 > -0.167$	無	

V: 子ゴールの対
W: $Cov(g)$ の比較
X: $Cup(g)$ の比較
Y: 優先順位の相違の有無
Z: $Ccc(g)$

(b)

V	W	X	Y	Z
(4, 5)	$7 > 6$	$8 > 6$	無	
(4, 7)	$7 < 8$	$8 > 7.5$	有	真
(5, 7)	$6 < 8$	$6 < 7.5$	無	

V: 子ゴールの対
W: $Cov(g)$ の比較
X: $Cup(g)$ の比較
Y: 優先順位の相違の有無
Z: $Ccc(g)$

(c)

V	W	X ^{注2)}	Y	Z
(15, 12)	$6 > 5$	$6.667 > 3.667$	無	
(15, 13)	$6 = 6$	$6.667 < 8.167$	無	
(15, 16)	$6 > 3$	$6.667 > 6.167$	無	真
(12, 13)	$5 < 6$	$3.667 < 8.167$	無	
(12, 16)	$5 > 3$	$3.667 < 6.167$	有	
(13, 16)	$6 > 3$	$8.167 > 6.167$	無	

V: 子ゴールの対
W: $Cov(g)$ の比較
X: $Cup(g)$ の比較
Y: 優先順位の相違の有無
Z: $Ccc(g)$

(d)

V	W	X ^{注2)}	Y	Z
(13, 16)	$5 < 8$	$8.167 > 6.167$	有	
(13, 14)	$5 < 8$	$8.167 > 6.167$	有	
(13, 11)	$5 < 6$	$8.167 > 8$	有	真
(16, 14)	$8 = 8$	$6.167 = 6.167$	無	
(16, 11)	$8 > 6$	$6.167 < 8$	有	
(14, 11)	$8 > 6$	$6.167 < 8$	有	

V: 子ゴールの対
W: $Cov(g)$ の比較
X: $Cup(g)$ の比較
Y: 優先順位の相違の有無
Z: $Ccc(g)$

(e)

V	W	X	Y	Z
(17, 18)	$6 < 8$	$3 > 2.5$	有	真

V: 子ゴールの対
W: $Cov(g)$ の比較
X: $Cup(g)$ の比較
Y: 優先順位の相違の有無
Z: $Ccc(g)$

(f)

V	W	X ^{注2)}	Y	Z
(15, 12)	$6 > 4$	$6.667 > 3.667$	無	偽

V: 子ゴールの対
W: $Cov(g)$ の比較
X: $Cup(g)$ の比較
Y: 優先順位の相違の有無
Z: $Ccc(g)$

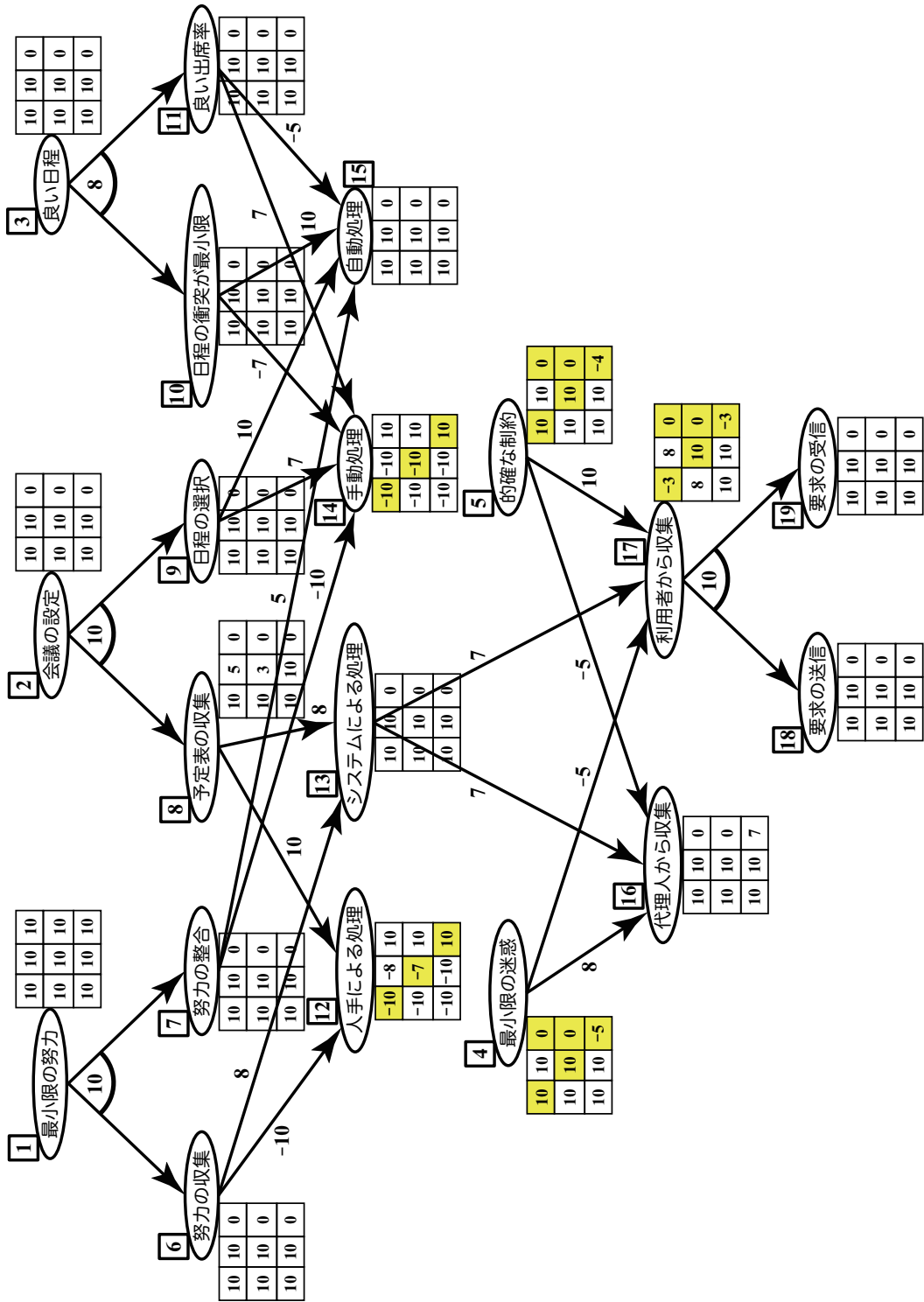


図 3.10: 貢献度と満足度が振られた会議の設定のゴールグラフ

```
1: >ccc-dfs.exe
2: 1は検出対象外
3: 6は検出対象
4: 12と13は優先順位の相違無し
5: 6はコンフリクト無し
6: 12は検出対象外
7: 8は検出対象
8: 12と13は優先順位の相違有り
9: 8はコンフリクト有り
10: 2は検出対象外
11: 9は検出対象
12: 14と15は優先順位の相違無し
13: 9はコンフリクト無し
14: 14は検出対象外
15: 7は検出対象
16: 14と15は優先順位の相違無し
17: 7はコンフリクト無し
18: 15は検出対象外
19: 10は検出対象
20: 14と15は優先順位の相違無し
21: 10はコンフリクト無し
22: 3は検出対象外
23: 11は検出対象
24: 14と15は優先順位の相違有り
25: 11はコンフリクト有り
26: 13は検出対象
27: 16と17は優先順位の相違無し
28: 13はコンフリクト無し
29: 16は検出対象外
30: 4は検出対象
31: 16と17は優先順位の相違無し
32: 4はコンフリクト無し
33: 17は検出対象外
34: 5は検出対象
35: 16と17は優先順位の相違有り
36: 5はコンフリクト有り
37: 18は検出対象外
38: 19は検出対象外
```

図 3.11: 図 3.10 に対する貢献度-妥当性-コンフリクト検出結果

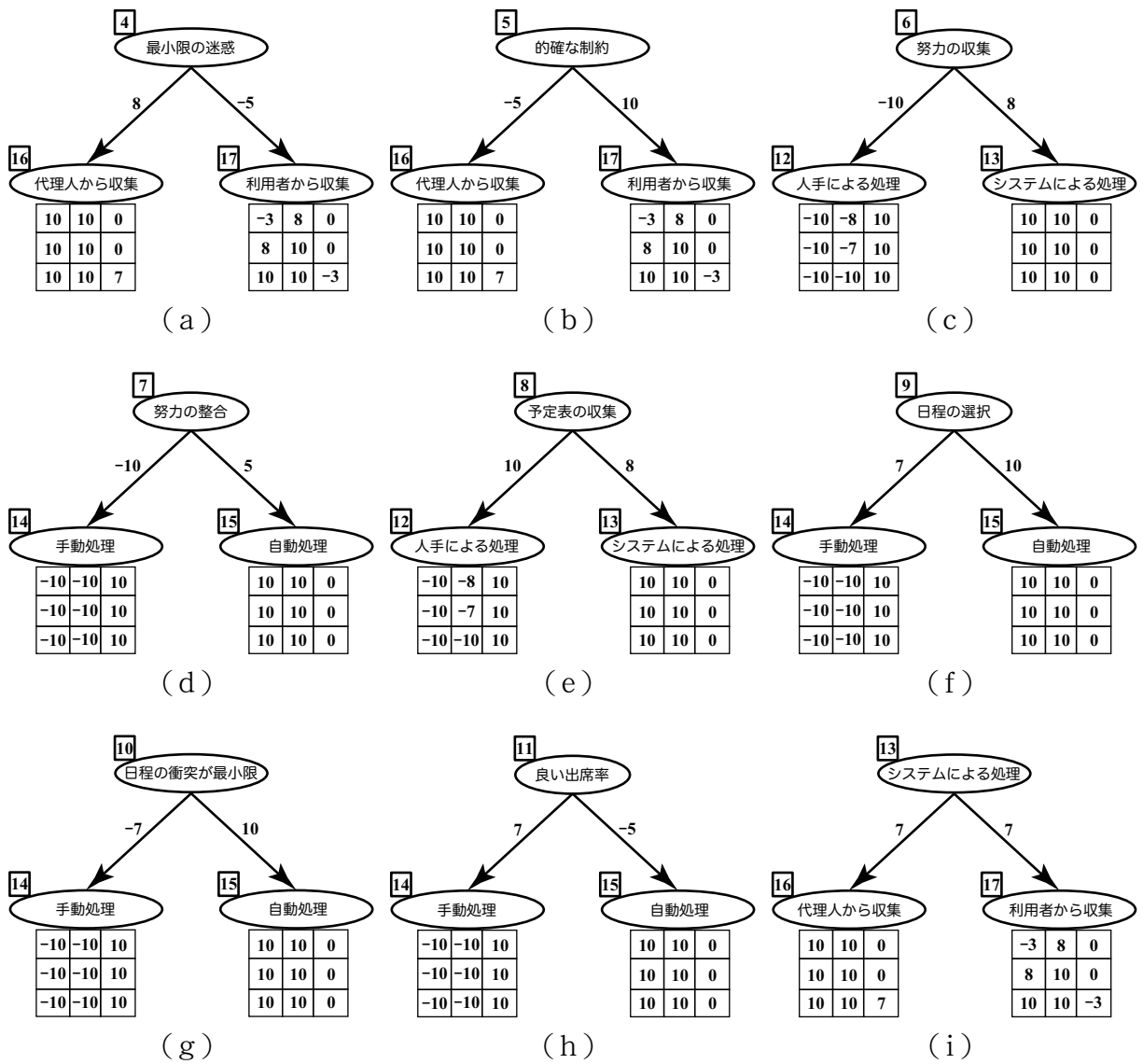


図 3.12: 図 3.10 において, 2つ以上の子ゴールを持つ2階層-OR分解-部分ゴールグラフ

表 3.3: 図 3.12(a)~(i) の $Cov(g)$ と $Cup(g)$ の比較

(a)					(b)				
V	W	X ^{注2)}	Y	Z	V	W	X ^{注2)}	Y	Z
(16, 17)	$8 > -5$	$10 > 7.167$	無	偽	(16, 17)	$-5 < 10$	$10 > 7.167$	有	真
(c)					(d)				
V	W	X ^{注2)}	Y	Z	V	W	X	Y	Z
(12, 13)	$-10 < 8$	$-9.167 < 10$	無	偽	(14, 15)	$-10 < 5$	$-10 < 10$	無	偽
(e)					(f)				
V	W	X ^{注2)}	Y	Z	V	W	X	Y	Z
(12, 13)	$10 > 8$	$-9.167 < 10$	有	真	(14, 15)	$7 < 10$	$-10 < 10$	無	偽
(g)					(h)				
V	W	X	Y	Z	V	W	X	Y	Z
(14, 15)	$-7 < 10$	$-10 < 10$	無	偽	(14, 15)	$7 > -5$	$-10 < 10$	有	真
(i)									
V	W	X ^{注2)}	Y	Z					
(16, 17)	$7 = 7$	$10 > 7.167$	無	偽					

V: 子ゴールの対
 W: $Cov(g)$ の比較
 X: $Cup(g)$ の比較
 Y: 優先順位の相違の有無
 Z: $Ccc(g)$

属性値の伝搬を伴うゴールグラフの簡約化は、[斎藤 06]で提案された「ゴールグラフの簡約化」(2.3.2.3節参照)を利用する。以下、図 2.10 において AND 分解と OR 分解が共に存在するゴール「製造業務の効率化」に対して本方法を適用し、その有効性を示す。

3.3.1 ゴールグラフの簡約化に伴う属性値の処理

属性値の伝搬を伴うゴールグラフの簡約化は、ゴールグラフの簡約化 (2.3.2.3 節参照) [斎藤 06, 山本 07]を行う上で、[斎藤 06, 山本 07]にはない以下 2 つの処理が追加される。これら 2 つの処理は、まとめて「ゴールグラフの簡約化における属性値の伝搬計算方法」と呼ばれる [佐藤 12c]。

ゴールグラフの簡約化における属性値の伝搬計算方法 [佐藤 12c]

(1) 貢献度の処理

属性値に基づくゴール選択手法 [斎藤 06, 山本 07] (2.3.2 節参照) では、ゴールグラフの簡約化 (2.3.2.3 節参照) を行う段階では、枝に対して貢献度は振られていない (2.3.2 節参照)。しかし、属性値の伝搬を伴うゴールグラフの簡約化は、 $Ccc(g)$ (式 (3.3)) で貢献度-妥当性-コンフリクト (3.1 節参照) が行えないゴール g に対して行われるものであり、あらかじめ枝に貢献度 (2.3.1 節 (属性-1) 参照) が振られているゴールグラフが適用対象となる。そのため、ゴールグラフの簡約化の過程において、基本法則 (2.3.2.2 節参照) の適用によりゴールが削除され、削除されたゴールの親ゴールと削除されたゴールの子ゴールが新たな枝で結ばれる場合と、複数のゴールが統合される場合の貢献度を新たに定める必要がある。ゴールグラフの簡約化における属性値の伝搬計算方法では、前者の場合、削除されたゴールとその親ゴールを結ぶ枝の貢献度と、削除されたゴールとその子ゴールを結ぶ枝の貢献度の平均を新たな枝の貢献度とする。後者の場合、統合される前の各々のゴールの貢献度の平均を、統合されたゴールの貢献度とする。一度のゴールグラフの簡約化において前者と後者を共に行わなければならない場合 (式 (2.9) または式 (2.10) を用いたゴールグラフの簡約化が行われる場合)、後者を行った後に前者を行う。

なお、ゴールグラフの簡約化の過程で削除されたゴールが親ゴールであった場合、その子ゴールへの枝も共に削除されるものとする。ゴールグラフの簡約化の過程で統合されたゴールが親ゴールであった場合、子ゴールへの枝は、統合後のゴールから子ゴールへの枝として継承されるものとし、貢献度もそのまま継承されるものとする。

(2) 満足度の処理

属性値に基づくゴール選択手法では、ゴールグラフの簡約化を行う上で、ゴールに対して満足度行列が振られていない (2.3.2 節参照)。しかし、属性値の伝搬を伴うゴールグラフの簡約化は、(1)と同様の理由から、ゴールに満足度行列が振られているゴールグラフが適用対象となる。そのため、ゴールグラフの簡約化の過程において、複数のゴールが 1 つのゴールに統合されることによって現れる新たなゴールの満足度を定める必要がある。ゴールグラフの簡約化における属性値の伝搬計算方法では、統合される前の個々のゴールに対して振ら

れていた満足度行列の同行同列成分の値の平均を、統合されたゴールの満足度行列の同行同列成分の値とする。

貢献度と満足度以外の属性値が設定されている場合、それらの処理については規定しない。

ゴールグラフの簡約化における属性値の伝搬計算方法を伴うゴールグラフの簡約化のことを、「属性値の伝搬を伴うゴールグラフの簡約化」と呼ぶ[佐藤 12c].

ゴールグラフの簡約化によって新たに現れるゴールに添字を付番する場合、新たに現れるゴールに統合されたゴールのうち少なくとも1つが他のゴールの親ゴールである場合には、新たに現れるゴールの添字は、*ccc-dfs*の動作を保証するために、他のゴールよりも小さくしなければならない(3.2節参照)。そのため、属性値の伝搬を伴うゴールグラフの簡約化を行った結果、新たに現れたゴールに対して添字を付番する場合は、次に定義する「ゴールグラフの簡約化における添字の処理」に従うものとする。

ゴールグラフの簡約化における添字の処理

属性値の伝搬を伴うゴールグラフの簡約化の過程で、統合または削除されたゴールの添字のうち、最も順序が小さい添字から順に、新しく現れたゴールに付番する。

ゴールグラフの簡約化によって統合または削除されたゴールに付番されていた添字は、ゴールグラフの簡約化後には欠番となる。そして、ゴールグラフの簡約化によって、ゴールの数が減ることはあっても増えることはない。そのため、統合または削除される前のゴールに対して付番されていた添字を、統合されたゴールに対して付番するという、添字の再利用が可能である。ゴールグラフの簡約化における添字の処理は、添字の総数がゴールグラフ中のゴールの総数以下であることを保証する。ゴールグラフ上の任意の2階層部分ゴールグラフ(1.7節(9)参照)の子ゴールの最大数は、ゴールの総数以下である。*ccc-dfs*(図3.3)においては、定数 M にゴールグラフ上の任意の2階層部分ゴールグラフの子ゴールの最大数以上の数を設定する必要がある(3.2節参照)が、以上の性質によって、ゴールグラフの簡約化に伴い M の値を変更する必要がないという利点がある。また、統合して新たに現れたゴールの添字は、その任意の子ゴールの添字未満であるという利点もある。よって、ゴールグラフの簡約化における添字の処理は、ゴールグラフ上の任意の2階層部分ゴールグラフに対して、親ゴールの添字は子ゴールの添字より大きいという*ccc-dfs*の動作条件を保証する。

3.3.2 適用

図2.10において、AND分解とOR分解が共に存在する「製造業務の効率化」を親ゴールとする、図3.13の多階層部分ゴールグラフ(1.7節(9)参照)に対して、属性値の伝搬を伴うゴールグラフの簡約化(3.3節参照)を行う。属性(貢献度および満足度)は図3.13の通りに振られているものとする。手順は次の通りである。

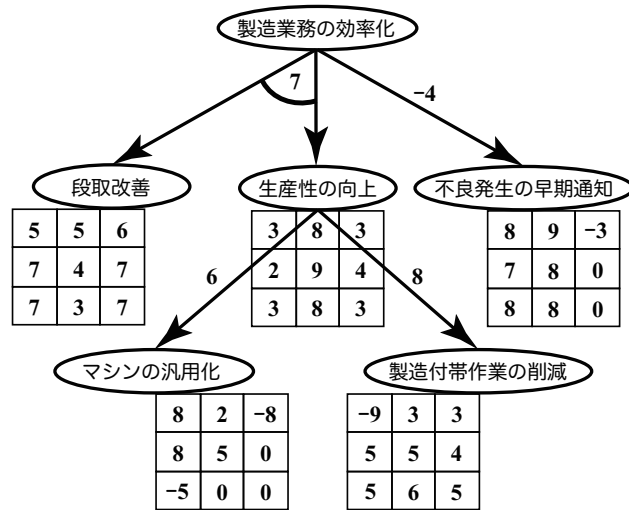


図 3.13: 属性値の伝搬を伴うゴールグラフの簡約化前の多階層-部分ゴールグラフ

(1) ゴールグラフの簡約化

$$\left(\begin{aligned} & \text{製造業務の効率化} \leftarrow \text{段取改善} * (\text{生産性の向上} \leftarrow \text{マシンの汎用化} \\ & \quad + \text{製造付帯作業の削減}) + \text{不良発生の早期通知} \end{aligned} \right)$$

式(2.9)
 \Leftrightarrow

$$\left(\begin{aligned} & \text{製造業務の効率化} \leftarrow \text{段取改善} * \text{マシンの汎用化} + \text{段取改善} * \text{製造付帯作業の削減} \\ & \quad + \text{不良発生の早期通知} \end{aligned} \right)$$

(2) 貢献度および満足度の処理

ゴールグラフの簡約化に伴い、「段取改善」と「マシンの汎用化」は「段取改善*マシンの汎用化」に統合される。またその際、「生産性の向上」が削除され、「段取改善*マシンの汎用化」は「生産性の向上」の親ゴールであった「製造業務の効率化」と直接枝で結ばれる。この場合、「段取改善*マシンの汎用化」の貢献度を求めるために2段階の計算が必要となる(3.3.1節(1)参照)。一方、「段取改善*マシンの汎用化」の満足度は、一度の計算で求めることができる。以下に具体的な計算過程を示す。

(2a) 貢献度の処理

「段取改善」と「マシンの汎用化」の貢献度を平均する。その平均は、 $Ave(Cov(\text{段取改善}), Cov(\text{マシンの汎用化})) = 6.5$ である。これと、「生産性の向上」の貢献度の平均 $Ave(Ave(Cov(\text{段取改善}), Cov(\text{マシンの汎用化})), Cov(\text{生産性の向上})) = 6.75$ が「段取改善*マシンの汎用化」の貢献度である。

(2b) 満足度の処理

「段取改善」と「マシンの汎用化」の満足度行列の各々の同行同列成分を平均する。たとえば、各々の1行1列成分の平均 $Ave(5, 8) = 6.5$ が、「段取改善*マシンの汎用化」

の満足度行列の1行1列成分の値である。他の成分の値も同様の計算で求められる。

「段取改善 * 製造付帯作業の削減」の貢献度および満足度も「段取改善 * マシンの汎用化」と同様に計算される。

図3.13の多階層-部分ゴールグラフに対して、属性値の伝搬を伴うゴールグラフの簡約化を行った後の2階層-部分ゴールグラフ[佐藤 12c] (1.7節(9)参照)を図3.14に示す。ゴール「製造業務の

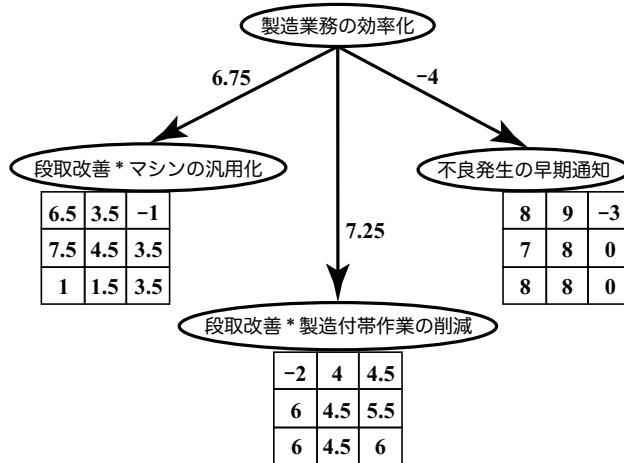


図 3.14: 図 3.13 に対して、属性値の伝搬を伴うゴールグラフの簡約化を適用した後の2階層-部分ゴールグラフ [佐藤 12c]

効率化」に対して、そのすべての子ゴールがOR分解で結ばれている(2階層-OR分解-部分ゴールグラフ(1.7節(9b)参照)である)ため、 $Ccc(g)$ (式(3.3))を用いて貢献度-妥当性-コンフリクト(3.1節参照)が検出できる。結果[佐藤 12c]を表3.4に示す。

表 3.4: 「製造業務の効率化」の貢献度-妥当性-コンフリクト検出結果 [佐藤 12c]

ゴール	X ^{注4)}	Y ^{注4)}	Z
段取改善 * マシンの汎用化	6.8	4.1	有
段取改善 * 製造付帯作業の削減	7.3	3.8	
不良発生の早期通知	-4	8	

X: 貢献度 ($Cov(g)$)

Y: 顧客のニーズに関する妥当性 ($Cup(g)$)

Z: 貢献度-妥当性-コンフリクトの有無 ($Ccc(g)$)

3.4 本章のまとめ

3.1節では、2つのゴール選択基準「貢献度」[Kaiya 02] (2.3.1節(属性-1)参照)と「顧客のニーズに関する妥当性」[佐藤 11] (3.1.1節参照)の間でゴール選択基準間のコンフリクト(2.4節参照)が生じる場合があることを述べた。このコンフリクトは、「貢献度と顧客のニーズに関する妥当性」間のコンフリクト(本論文では、簡単のため、「貢献度-妥当性-コンフリクト」と呼ぶ)と呼ばれる

[佐藤 11]. 2階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) において, 貢献度-妥当性-コンフリクトが存在する場合, 形式的に狭義のゴール選択 (1.7 節 (10) 参照) ができない. そのため, このコンフリクトを持つゴールを検出することが重要であることを述べた.

3.2 節では, ゴールグラフ上の貢献度-妥当性-コンフリクトを持つすべてのゴールを検出するアルゴリズム *ccc-dfs* [佐藤 12a] について述べた. *ccc-dfs* は, 「深さ優先探索」[Aho 83] に基づいて構築された.

3.3 節では, ゴール g が貢献度-妥当性-コンフリクトを持つか否かを検出する指標 $Ccc(g)$ (式 (3.3)) が, その定義上, AND 分解と OR 分解を共に持つゴールに対してはこのコンフリクトを検出できないことから, これを検出可能にする方法を提案した. この方法は, 「属性値の伝搬を伴うゴールグラフの簡約化」[佐藤 12c] と呼ばれる (3.3.1 節参照). 属性値の伝搬を伴うゴールグラフの簡約化は, 「ゴールグラフの簡約化」(2.3.2.3 節参照) を行う際に, 「貢献度の処理」(3.3.1 節 (1) 参照) と「満足度の処理」(3.3.1 節 (2) 参照) を追加したものである.

ccc-dfs に 3.3 節で述べた属性値の伝搬を伴うゴールグラフの簡約化を組み込むことで, 貢献度-妥当性-コンフリクトを持つ, ゴールグラフ上のすべてのゴールを検出することが可能となる. しかしながら, 検出されたゴールの各々に対して, 選択される子ゴールを人手で決定することは, 人手コストが大きいことやヒューマンエラーが混入する可能性があることから, 現実的ではない. この問題に対して, 貢献度-妥当性-コンフリクトをはじめとするゴール選択基準間のコンフリクトを考慮した上で, 形式的に狭義のゴール選択を行うことができれば, ゴール指向要求分析を用いて形式的に広義のゴール選択 (1.7 節 (10) 参照) を行うことが可能になる. 第 4 章では, 階層分析法に基づいて形式的に狭義のゴール選択を行う方法について述べる.

後注

注1) [佐藤 11] では, $Ccc(g)$ は $Dip(g)$ として定義されている. 本論文では, ゴール選択基準間のコンフリクト (2.4 節参照) のうち, 貢献度-妥当性-コンフリクト (3.1 節参照) であることの明示性, および, *ccc-dfs* (図 3.3) との名称の整合性の観点から, ゴール g の貢献度-妥当性-コンフリクトを検出する指標を $Dip(g)$ ではなく $Ccc(g)$ と表記する. 本論文における $Dip(g)$ は, 貢献度-妥当性-コンフリクトを含むゴール g のゴール選択基準間のコンフリクトを検出する指標として定義される (式 (3.2)).

注2) 小数点第 4 位で四捨五入した値を示している.

注3) [Hayashi 12] に掲載されている会議の設定のゴールグラフには, 各ゴールに満足度行列は振られていない. 各ゴールの満足度行列中のすべての満足度は, 筆者が恣意的に振ったものである.

注4) 小数点第 2 位で四捨五入した値を示している.

ゴール指向要求分析における 形式的ゴール選択方法

4.0 本章の概要

本章では、ゴール指向要求分析によって形式的に広義のゴール選択（1.7節（10）参照）を行うための方法を提案する。4.1節では、ゴール指向要求分析における広義のゴール選択に形式的な定義を与え、その定義に基づいて「ゴール選択アルゴリズム」*gsl-dfs*を提案する。*gsl-dfs*は、子ゴールの親ゴールに対するゴール選択基準値を定める必要がある。しかし、ゴール選択基準間のコンフリクト（2.4節参照）が存在する場合は、単一のゴール選択基準では形式的に狭義のゴール選択（1.7節（10）参照）ができない。そのため、ゴール選択基準間のコンフリクトが存在するゴールに対して、形式的に狭義のゴール選択を行うための方法が確立されなければならない。4.2節では、階層分析法に基づいてこの方法を構築した結果について述べる。この方法は階層分析法の評価基準としてゴール選択基準を設定する。そのため、ゴール選択基準それ自体に妥当性があることを前提としている。そこで、ゴール選択基準の妥当性を保証するために、ゴール選択基準はなるべく拡張されるべきである。4.3節では、ゴール選択基準のうち、特に、顧客のニーズに関する妥当性（3.1.1節参照）に関して、4.3.1節でこれを拡張する方法について提案する。

4.1 ゴール選択アルゴリズム

ゴール指向要求分析（2.2節参照）においては、広義のゴール選択（1.7節（10）参照）が完了したとき、またそのときに限り、要求獲得が完了する。広義のゴール選択において、初期ゴール（2.2節参照）を達成することができる最小限の最終ゴール（2.2節参照）を要求として採用すれば、ゴール指向要求分析における最小限の要求が獲得されたとみなすことができる（ただし、ロバスト性向上のためには、種々の制約を考慮した上で、なるべく多くの最終ゴールを要求として採用すべきである（2.3.2.4節参照）。そこで、本論文では、ゴール指向要求分析における広義のゴール選択を、次の通り形式的に定義する。

広義のゴール選択の形式的定義

ゴール指向要求分析において広義のゴール選択がなされる。

: \Leftrightarrow ゴールグラフ上のすべての2階層-部分ゴールグラフで狭義のゴール選択がなされる. (4.1)

\Leftrightarrow すべての初期ゴールに対して、少なくとも1つの(初期ゴール, 最終ゴール)-道が定まる。

ただし、広義のゴール選択とは、ゴールグラフにおいて、初期ゴール (2.2節参照) を達成するために選択される少なくとも1つの最終ゴール (2.2節参照) を選択すること (1.7節(10)参照)。狭義のゴール選択とは、2階層-OR分解-部分ゴールグラフ (1.7節(9b)参照) において、親ゴールの達成のために選択されるべき少なくとも1つの子ゴールを決定すること (1.7節(10)参照)。2階層-部分ゴールグラフは、ゴールグラフ上の1つの親ゴールとそのすべての子ゴールからなる部分ゴールグラフ (1.7節(9)参照)。(初期ゴール, 最終ゴール)-道は、初期ゴールから最終ゴールまでの有向道 (2.3.1.1節参照)。

2階層-部分ゴールグラフは、(1) 2階層-AND分解-部分ゴールグラフ、(2) 2階層-OR分解-部分ゴールグラフ、および(3) 2階層-AND/OR分解-部分ゴールグラフに分割できる (1.7節(9)参照)。(1)は、定義上すべての子ゴールが選択されるため、狭義のゴール選択の結果はあらかじめ一意に定まっている。(3)は、「属性値の伝搬を伴うゴールグラフの簡約化」(3.3節参照)によって、(2)に変形できる。したがって、2階層-部分ゴールグラフにおける狭義のゴール選択の問題は、(2)の2階層-OR分解-部分ゴールグラフにおける狭義のゴール選択の問題に還元可能である。

以上より、ゴールグラフ上のすべてのゴールを訪問し、各々のゴールを訪問するごとに、(1)ならば、すべての子ゴールを選択し、(2)ならば、何らかのゴール選択基準によって少なくとも1つの子ゴールを選択すれば、ゴールグラフ上のすべての2階層-部分ゴールグラフに対して狭義のゴール選択を行うことができる。式(4.1)から、この操作が完了するとき、広義のゴール選択が完了する。これはすなわち、ゴール指向要求分析における要求獲得が完了することを意味する。ゴールグラフ上のすべてのゴールを訪問することは、「深さ優先探索」[Aho 83] (図3.2)によって可能である。深さ優先探索に基づいて広義のゴール選択を行う「ゴール選択アルゴリズム」*gsl-dfs*を、図4.1に示す。ただし、 v, w はゴールを表す変数。 $mark[v]$ は、 v が未訪問であるか訪問済であるかを表す、*unvisited* または *visited* のブール値を要素とするリストである。 $L[v]$ は隣接リストであり、 v に隣接するゴールを要素として持つ。 $L_{sg}[v][w]$ は、 v が w を選択する場合1、そうでない場合0のブール値を要素とするリストである。 M には、対象とするゴールグラフ上の子ゴールの最大数以上の数(初期ゴールを除くすべてのゴール以上の数)が設定されているものとする。 v の子ゴールが1つだけ ($n_{and} = 0$ and $n_{or} = 1$) または v がすべての子ゴールとAND分解で接続している ($n_{and} \geq 2$ and $n_{or} = 0$) 場合、すべての子ゴールが選択される (図4.1, 行13~16)。 v が2つ以上の子ゴールとOR分解のみで接続している ($n_{and} = 0$ and $n_{or} \geq 2$) 場合、または、少な


```

1: procedure gsl-dfs ( v: vertex );
2: var
3:   w: vertex;
4:   n_and, n_or, i: integer; n_and := 0, n_or := 0;
5:   L_sg: array [0..M + 1, 0..M + 1] of boolean;
6: begin
7:   mark[v] := visited;
8:   for L[v] 上の各頂点 w do
9:     if w が v の子ゴール and w は v と AND 分解で接続 then
10:      n_and := n_and + 1
11:     else if w が v の子ゴール and w は v と OR 分解で接続 then
12:       n_or := n_or + 1;
13:     if (n_and = 0 and n_or = 1) or (n_and >= 2 and n_or = 0) then
14:       for L[v] 上の各頂点 w do
15:         if w が v の子ゴール then
16:           L_sg[v][w] := 1
17:         else if (n_and = 0 and n_or >= 2) or (n_and >= 1 and n_or >= 1) then
18:           gsl(v)
19:         else
20:           L_sg[v][w] := 0;
21:       for L[v] 上の各頂点 w do
22:         if mark[w] := unvisited then
23:           gsl-dfs(w)
24:     end; { gsl-dfs }

```

図 4.1: *gsl-dfs*

くとも1つの子ゴールとAND分解で接続し、かつ、少なくとも1つの子ゴールとOR分解で接続している ($n_and \geq 1$ **and** $n_or \geq 1$) 場合、 v は少なくとも1つの子ゴールを選択しなければならない。この処理は、手続き *gsl* にまとめられている (図 4.1, 行 17~18)。手続き *gsl* の詳細を、図 4.2 に示す。 v とその子ゴールがこれら以外の接続関係である場合、 v に選択される子ゴールはないものとする (図 4.1, 行 19~20)。

手続き *gsl* (図 4.1, 行 18) は、 v のすべての子ゴールに対して、そのゴール選択基準値を要素とするリスト $L_vl[w]$ を作成し、そのうちの最大値をとる子ゴールを選択する手続きである (最大値をとる子ゴールが複数存在する場合、それらすべての子ゴールが選択される)。 $L_vl[w]$ に代入される w のゴール選択基準値を定めるための何らかの関数が必要となり、この関数は、{ w のゴール選択基準値 } (図 4.2, 行 9) に置換される。

```

1: procedure gsl ( v: vertex );
2: var
3:   w: vertex;
4:   max, flg: integer; flg := 1;
5:   L.vl: array [0..M + 1] of real;
6: begin
7:   for L[v] 上の各頂点 w do
8:     if w が v の子ゴール then
9:       L.vl[w] := { w のゴール選択基準値 }
10:    else
11:      L.vl[w] := 0;
12:    for L[v] 上の各頂点 w do
13:      if flg = 1 then
14:        max := L.vl[w];
15:        flg := 0
16:      else if max < L.vl[w] then
17:        max := L.vl[w];
18:      for L[v] 上の各頂点 w do
19:        if max = L.vl[w] then
20:          L.sg[v][w] := 1
21: end; { gsl }

```

図 4.2: *gsl* (図 4.1 の 18 行目の処理)

gsl-dfs の出発点は、ゴールグラフ上のどのゴールを指定してもよい。何故ならば、任意の出発点に対して、*gsl-dfs* の実行後には、ゴールグラフ上のすべてのゴール *v* に対して、*L.sg*[*v*][*w*] が値を持つからである。つまり、ゴールグラフ上のすべての 2 階層部分ゴールグラフに対して狭義のゴール選択がなされたことになる。このように、式 (4.1) から、*gsl-dfs* は、任意の出発点に対して広義のゴール選択が可能である。

任意の初期ゴールを出発点として定めた *gsl-dfs* を実行後の *L.sg*[*v*][*w*] において、*L.sg*[*v*][*w*] = 1、かつ、*w* が最終ゴールであるような *w* が、各々要求である。しかし、要求がどのようにして獲得されたかというプロセスの明示性の基準である「追跡可能性」[IEEE Standards Board 98] の観点からは、単に獲得された要求を知るだけでなく、出発点として指定される初期ゴールから要求として採用される最終ゴールまでの経路も併せて知ることができればなおよい。この経路を得る手続き *pth* を図 4.3 に示す。ただし、*v* は初期ゴールを表す変数である。

```

1: procedure pth (v: vertex );
2:   for L[v] 上の各頂点 w do
3:     if s[v][w] = 1 then begin
4:       { v->w };
5:       pth(w)
6:     end
7: end; { pth }

```

図 4.3: *pth*

4.1.1 評価

gsl-dfs (図 4.1) の評価のために、これを実装したコンピュータプログラムを作成した。ゴール選択基準値を定める関数としては、ゴール *w* のその親ゴール *v* への貢献度を求める関数 $Cov(v, w)$ (式 (2.1)) を設定した。このコンピュータプログラムを、図 3.1 の 2 階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) に対して適用した結果を、図 4.4 に示す。ただし、1 は「海外収益向上」、2 は「現地通訳雇用」、3 は「英語公用化」の添字である。結果は、手続き *pth* (図 4.3) によって、初期ゴール (2.2 節参照) から最終ゴール (2.2 節参照) までの経路として端末上に表示されている。図

```

1: >_gsl-dfs.exe
2: 1->2

```

図 4.4: 図 3.1 に対する *gsl-dfs* の適用結果

3.1 では、初期ゴールは 1 であり、最終ゴールは 2 および 3 である。2 階層-部分ゴールグラフ (1.7 節 (9) 参照) は 1 を親ゴール、2 と 3 を子ゴールとする 2 階層-OR 分解-部分ゴールグラフだけである。したがって、1 は 2 または 3 の少なくとも 1 つをゴール選択する必要がある。*gsl-dfs* は、2 階層-OR 分解-部分ゴールグラフに対しては、親ゴールへのゴール選択基準値が最大となる子ゴールを選択する。この場合、 $Cov(1, 2) = 9 > Cov(1, 3) = 4$ であるから、2 が選択される。したがって、初期ゴールから最終ゴールまでの経路は「1->2」となる。図 4.4 から、構築したコンピュータプログラムは、この結果を正しく得ている。これは、*gsl-dfs* の正しさを示すものである。

次に、*ccc-dfs* が、一般のゴールグラフに対して適用可能であることを確認するために、*ccc-dfs* を、図 2.10 に貢献度 (2.3.1 節 (属性-1) 参照) を振った図 4.5 に対して適用した結果を、図 4.6 に示す。図 4.5 には、各ゴールに対応する添字も併せて記載している。

図 4.6 より、初期ゴール 1 (「生産コストの削減」) から最終ゴールまでの経路は、「1->2->4->8」、
「1->2->4->9->18」、
「1->3->7->14->16」、
「1->3->7->14->19」、
「1->3->7->16」の 5 つあることがわかる。この経路の要素のうち、最終ゴールに該当するゴールが、要求である。したがってこの場合、8 (「段取改善」)、16 (「進捗実績の即時把握」)、18 (「製造付帯作業の削減」)、19 (「完成数量の即時把握」) が、要求となる。1 から、要求となるこれらの最終ゴールまでの経路が正しいことを確認するために、上記 5 つの経路を構成するすべての 2 階層-部分ゴールグラフと、その各々のゴール

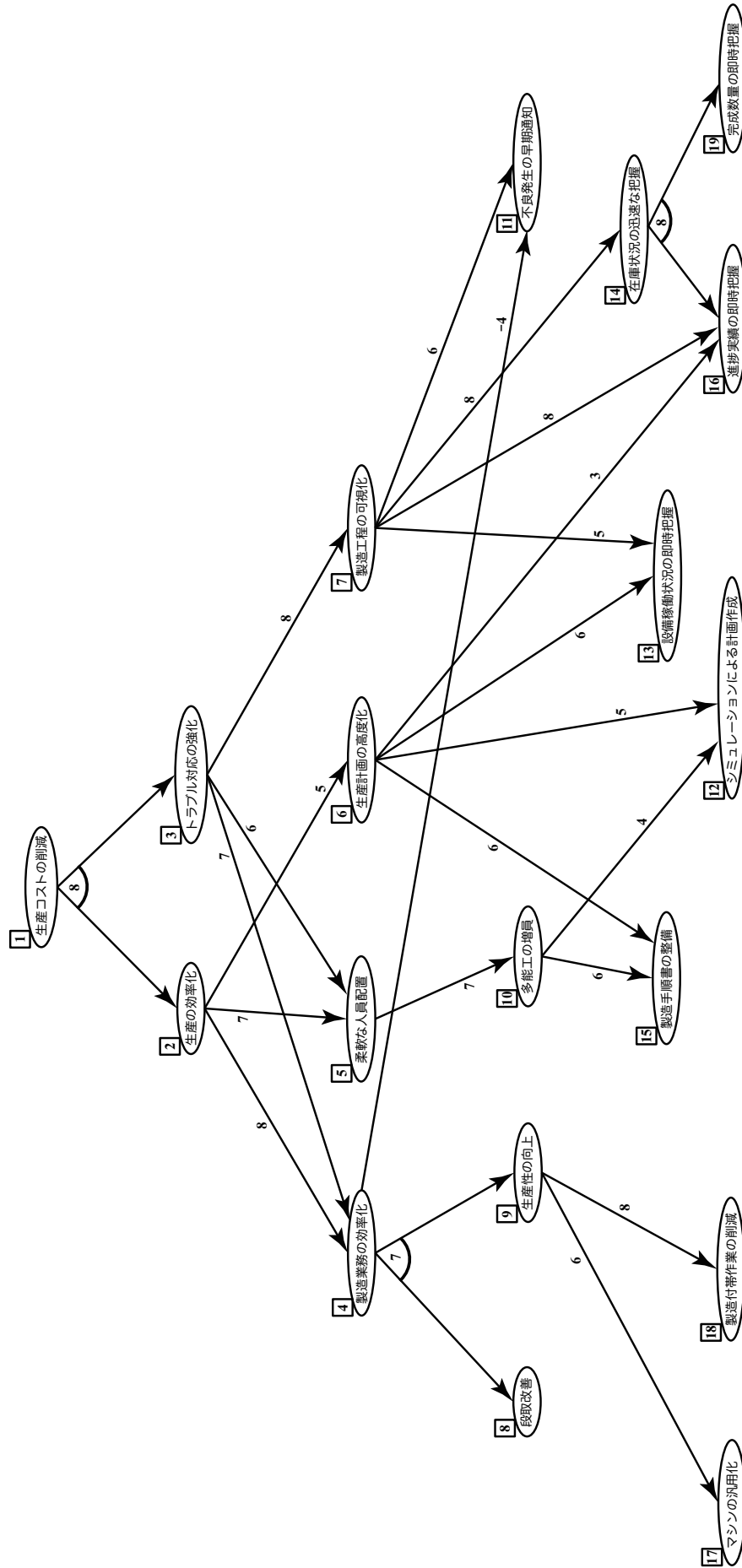


図 4.5: 図 2.10 に貢献度を振ったゴールグラフ

```

1: >_gsl-dfs.exe
2: 1->2_2->4_4->8_4->9_9->18_1->3_3->7_7->14_14->16_14->19_7->16

```

図 4.6: 図 4.5 に対する *gsl-dfs* の適用結果

選択結果を図 4.7(a)~(g)と表 4.1(a)~(g)に示す。図 4.6は、これらの結果を正しく反映している。したがって、*gsl-dfs*は、図 4.5のゴールグラフに対して正しく機能することがわかる。

ところで、ゴールグラフは一般に複数の初期ゴールを定めることが可能である(2.2節参照)。*gsl-dfs*は初期ゴールが複数ある場合にも対応している。ただし、*pth*(図 4.3)は、ある1つの初期ゴールから最終ゴールまでの経路を表示するものであるため、初期ゴールが複数ある場合には、*pth*を初期ゴールごとに繰り返して適用する必要がある。

初期ゴールが複数のゴールグラフに対して *gsl-dfs* が適用可能であることを確認するために、*gsl-dfs* を、5つの初期ゴールが存在するゴールグラフ(図 4.8[Hayashi 12])に対して適用した結果を、図 4.9に示す。図 4.8には、各ゴールに対応する添字も併せて記載している。

図 4.9より、すべての初期ゴールに対して、最終ゴールまでの経路が表示されていることがわかる。初期ゴール1(「最小限の努力」)から最終ゴールまでの経路は、「1->6->13->16」, 「1->6->13->17->18」, 「1->6->13->17->19」, 「1->7->15」の4つあることがわかる。初期ゴール2(「会議の設定」)から最終ゴールまでの経路は、「2->8->12」, 「2->9->15」の2つあることがわかる。初期ゴール3(「良い日程」)から最終ゴールまでの経路は、「3->10->15」, 「3->11->14」の2つあることがわかる。初期ゴール4(「最小限の迷惑」)から最終ゴールまでの経路は、「4->16」の1つのみであることがわかる。初期ゴール5(「利用者から収集」)から最終ゴールまでの経路は、「5->17->18」, 「5->17->19」の2つあることがわかる。これらの経路の要素のうち、最終ゴールに該当するゴールが、要求である。図 4.8のゴールグラフ上のすべての最終ゴール(12(「人手による処理」), 14(「手動処理」), 15(「自動処理」), 16(「代理人から収集」), 18(「要求の送信」), 19(「要求の受信」))が経路の要素に含まれているため、これらの最終ゴールが要求となる。各々の初期ゴール1~5から最終ゴールまでの経路が正しいことを確認するために、上記5つの経路を構成するすべての2階層部分ゴールグラフと、その各々のゴール選択結果を図 4.10(a)~(m)と表 4.2(a)~(m)に示す。図 4.9は、これらの結果を正しく反映している。したがって、*gsl-dfs*は、初期ゴールが複数存在する図 4.8のゴールグラフに対して、適切に機能することがわかる。

4.1.2 考察

重み付き有向グラフの経路を求める既存のアルゴリズムとしては、Dijkstra(ダイクストラ)法[Aho 83]や、Warshall-Floyd(ウォーシャルフロイド)法[Aho 83]が有名である。しかしながら、前者は、重みが非負である場合にのみ適用可能であるという制約がある。ゴールグラフに対する広義のゴール選択の形式的定義(式(4.1))において、この重みは、狭義のゴール選択(1.7節(10)参照)におけるゴール選択基準値に対応する。既存のゴール選択基準としては、貢献度(2.3.1節(属性-1)参照)や顧客のニーズに関する妥当性(3.1.1節参照)が提案されており、これらは共に負の値

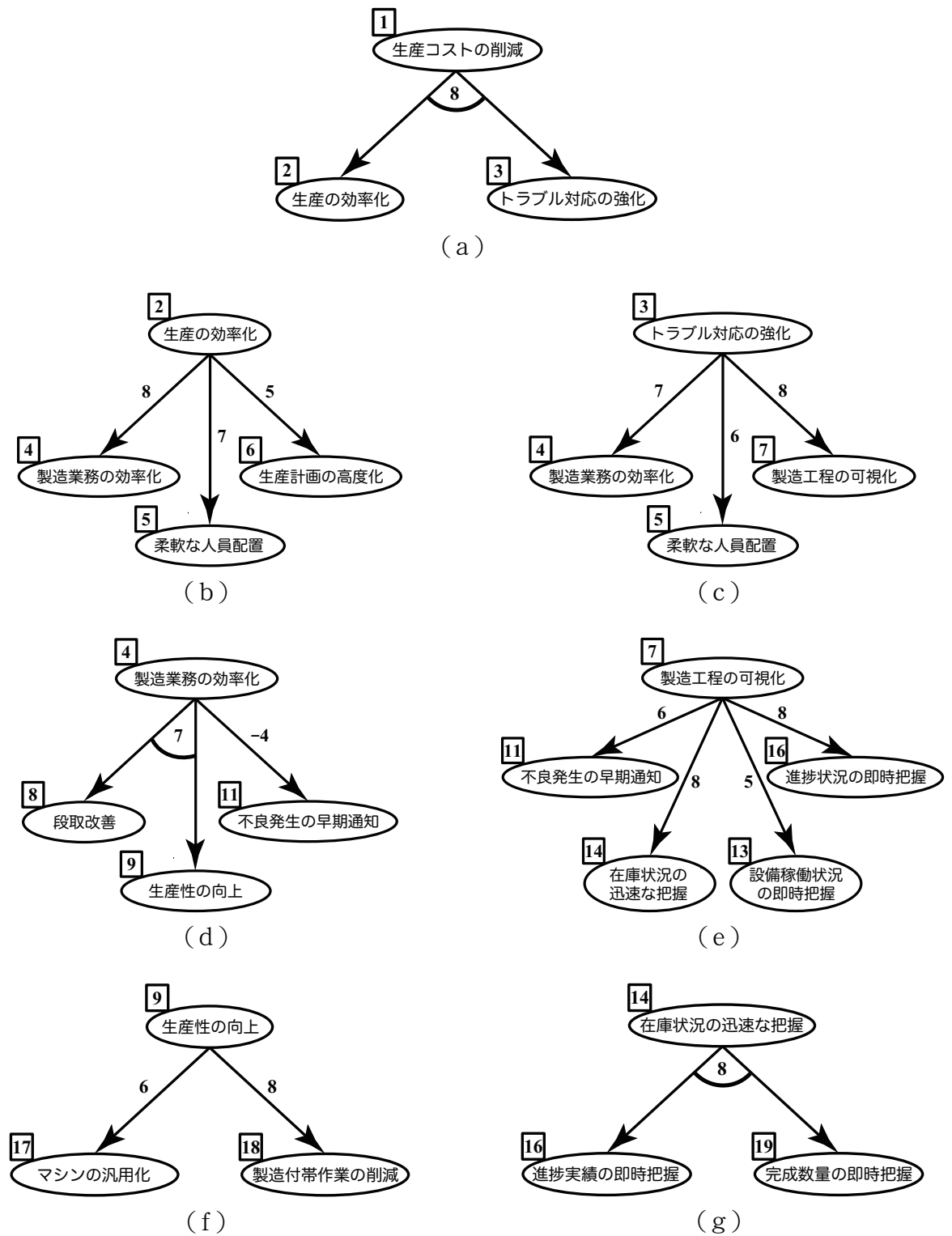


図 4.7: 図 4.5 の初期ゴールから要求までの経路を構成する 2 階層-部分ゴールグラフ

表 4.1: 図 4.7(a)~(g) において選択される子ゴール

(a)

W	X	Y	Z
1	2 3	8 8	2, 3

W: 親ゴール
 X: 子ゴール
 Y: 貢献度
 Z: 選択される子ゴール

(b)

W	X	Y	Z
2	4 5 6	8 7 5	4

W: 親ゴール
 X: 子ゴール
 Y: 貢献度
 Z: 選択される子ゴール

(c)

W	X	Y	Z
3	4 5 7	7 6 8	7

W: 親ゴール
 X: 子ゴール
 Y: 貢献度
 Z: 選択される子ゴール

(d)

W	X	Y	Z
4	8 9 11	7 7 -4	8, 9

W: 親ゴール
 X: 子ゴール
 Y: 貢献度
 Z: 選択される子ゴール

(e)

W	X	Y	Z
7	11 13 14 16	6 5 8 8	14, 16

W: 親ゴール
 X: 子ゴール
 Y: 貢献度
 Z: 選択される子ゴール

(f)

W	X	Y	Z
9	17 18	6 8	18

W: 親ゴール
 X: 子ゴール
 Y: 貢献度
 Z: 選択される子ゴール

(g)

W	X	Y	Z
14	16 19	8 8	16, 19

W: 親ゴール
 X: 子ゴール
 Y: 貢献度
 Z: 選択される子ゴール

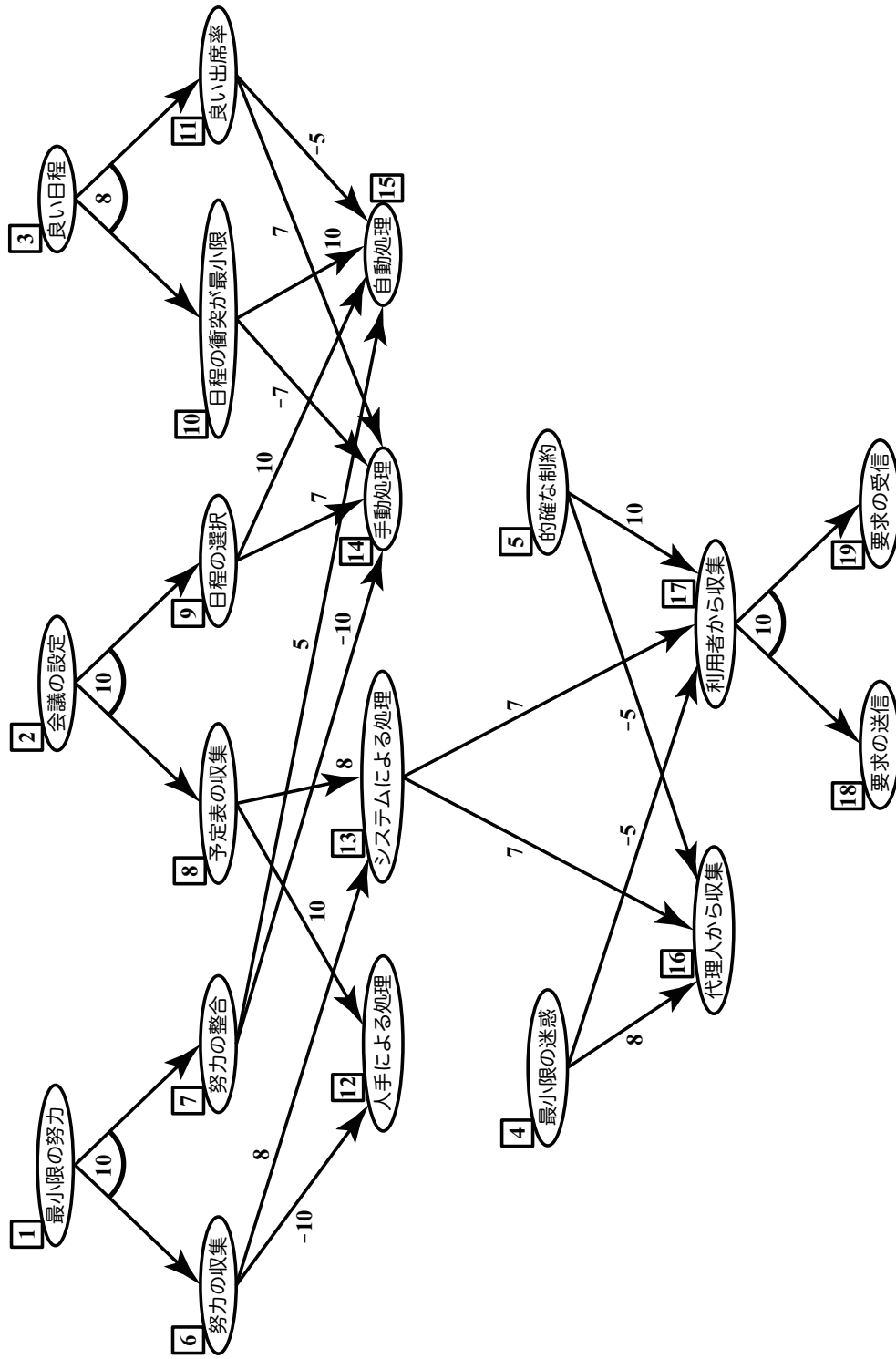


図 4.8: 貢献度が振られた会議の設定のゴールグラフ [Hayashi 12]

表 4.2: 図 4.10(a)~(m) において選択される子ゴール

(a)			
W	X	Y	Z
1	6 7	10 10	6, 7

W: 親ゴール
X: 子ゴール
Y: 貢献度
Z: 選択される子ゴール

(b)			
W	X	Y	Z
2	8 9	10 10	8, 9

(c)			
W	X	Y	Z
3	10 11	8 8	10, 11

(d)			
W	X	Y	Z
4	16 17	8 -5	16

(e)			
W	X	Y	Z
5	16 17	-5 10	17

(f)			
W	X	Y	Z
6	12 13	-10 8	13

(g)			
W	X	Y	Z
7	14 15	-10 5	14

(h)			
W	X	Y	Z
8	12 13	10 8	12

(i)			
W	X	Y	Z
9	14 15	7 10	15

(j)			
W	X	Y	Z
10	14 15	-7 10	15

(k)			
W	X	Y	Z
11	14 15	7 -5	14

(l)			
W	X	Y	Z
13	16 17	7 7	16, 17

(m)			
W	X	Y	Z
17	18 19	10 10	18, 19

```

1: >_gsl-dfs.exe
2: 1->6_6->13_13->16_13->17_17->18_17->19_1->7_7->15
3: 2->8_8->12_2->9_9->15
4: 3->10_10->15_3->11_11->14
5: 4->16
6: 5->17_17->18_17->19

```

図 4.9: 図 4.8 に対する *gsl-dfs* の適用結果

になる場合がある。そのため、一般にダイクストラ法を広義のゴール選択のためのアルゴリズムとして適用することはできない。一方、後者は、重みが非負の場合にも適用可能である。しかしながら、 n 個の頂点と e 本の枝を持つ有向グラフに対して、ウォーシャルフロイド法は、計算量が $\mathcal{O}(n^3)$ であり、同有向グラフに対する深さ優先探索の計算量 $\mathcal{O}(e)$ ($n \leq e$) に比べて増加率が大きい [Aho 83]。本論文で提案した *gsl-dfs* (図 4.1) は、重み (ゴール選択基準) が負の場合に適用可能であり、かつ、深さ優先探索に基づいて構築されているため、線形時間で広義のゴール選択結果を得ることが可能であるという利点がある。今後の課題として、ウォーシャルフロイド法に基づく広義のゴール選択アルゴリズムを構築し、*gsl-dfs* の計算量や実行時間との差異を厳密に検証し比較することで、*gsl-dfs* の実用性を評価することが挙げられる。

gsl-dfs は、広義のゴール選択において「分割統治法 (Divide-and-conquer method)」[Aho 83] が上手くいくことを示している。システム S は、それを構成する要素の集合 T とその集合上に定義される関係 R の対 $S := (T, R)$ として定義される [Klir 91]。ただし、 T はシステム境界によって定められるオブジェクトを要素とする集合であり、Thingfood と呼ばれる。 R は T 上に定義される n 項関係 (一般には 2 項関係) であり、Systemhood と呼ばれる。分解関係を考慮しないゴールグラフ (G, E) もまた、ゴールの集合 G を Thingfood、枝の集合 E を Systemfood とするシステムである。システムは、一般に「全体は部分の総和以上のものである」という性質を持つ。すなわち、システムを構成する部分システムにおける最適解の総和が、システム全体の最適解となるとは限らない。そのため一般に、システムに対して分割統治法は適用できない。しかし、広義のゴール選択の形式的定義 (式 (4.1)) においては、ゴールグラフ中の個々の 2 階層部分ゴールグラフにおける狭義のゴール選択の結果の総和が広義のゴール選択結果 (要求) となるため、ゴールグラフは、広義のゴール選択に関して要素還元可能なシステムである。それ故、分割統治法による広義のゴール選択が上手くいくのである。

ゴール選択アルゴリズム *gsl-dfs* (図 4.1) は、深さ優先探索 (図 3.2) に基づいて構築された。しかし、深さ優先探索は、いわゆる厳密解法であるため、内在する問題が NP 困難な問題クラスである場合、実用的な時間 (多項式時間) で解を見出すことができない [半田 12]。広義のゴール選択結果は、狭義のゴール選択結果の総和であることから、狭義のゴール選択結果が多項式時間で得られるならば、広義のゴール選択結果もまた多項式時間で得られる^{注1)}。しかし、狭義のゴール選択は、本質的に NP 困難な問題である。

gsl-dfs は、狭義のゴール選択として、最も好ましい最小限の子ゴールを選択するものである。しかし、狭義のゴール選択は、その定義 (1.7 節 (10) 参照) 上、「少なくとも 1 つ」の子ゴールを選択す

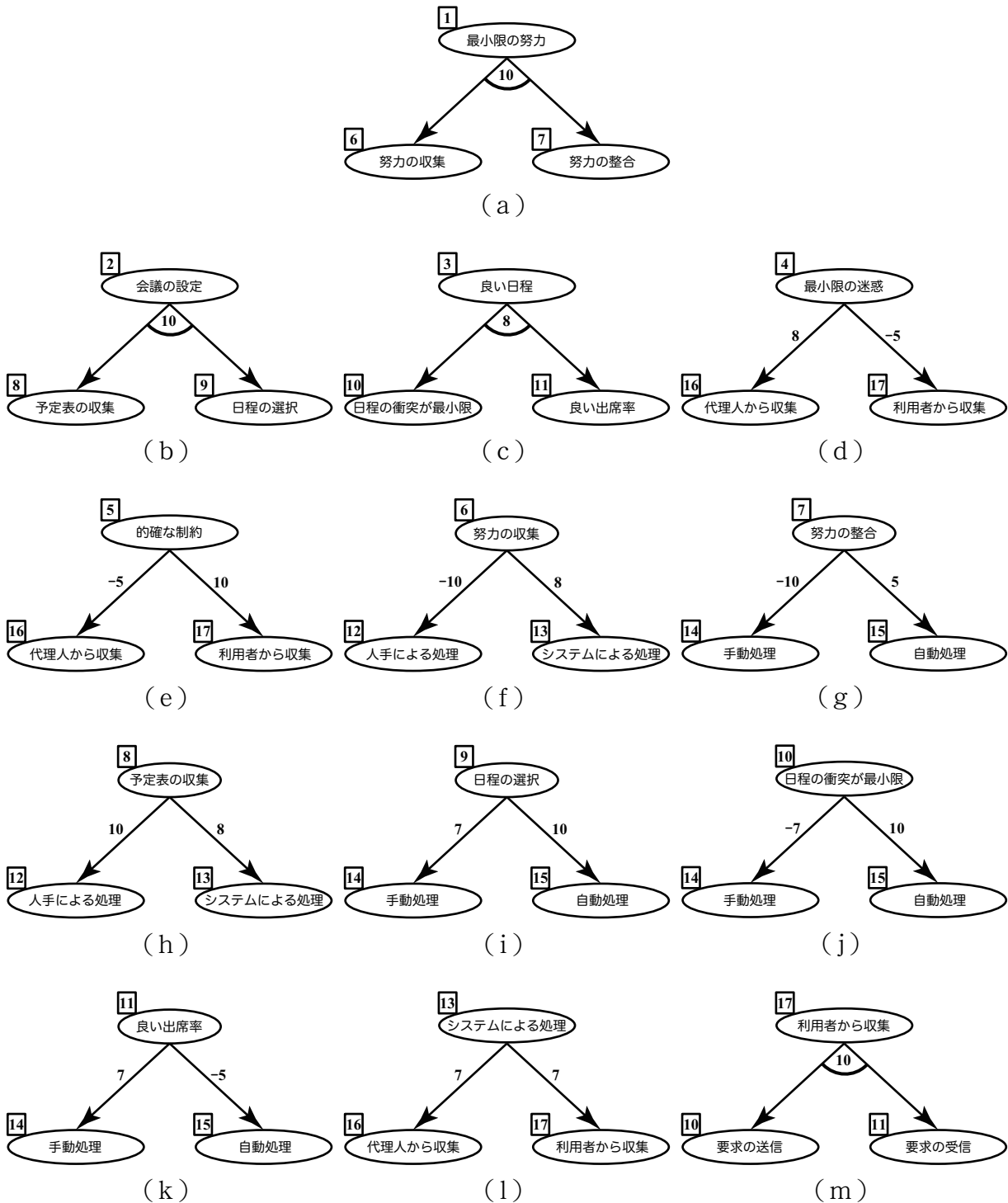


図 4.10: 図 4.8 の初期ゴールから要求までの経路を構成する 2 階層-部分ゴールグラフ

るものであり、最小限の子ゴールのみを選択しなければならないわけではない。むしろ、開発される情報システムのロバスト性の観点からは、開発コストや納期など種々の「プロジェクト制約」のもとで、なるべく多くの子ゴールを選択するべきである。なぜならば、ある要求を達成する情報システムの機能がただ1つだけであるとする、その機能に障害が発生した場合、その機能に対応する要求が達成されなくなり、その要求が達成されることによって達成されていたゴールが、達成されなくなってしまうからである。したがって、狭義のゴール選択は、厳密には、すべての子ゴールの組合せに対してそのゴール選択基準値を定め、その上で、最も大きなゴール選択基準値を持つ子ゴールの集合から順に、個々の2階層部分ゴールグラフ(1.7節(9)参照)の子ゴールごとに設定される「プロジェクト制約値」の上限の範囲内で、なるべく多くの子ゴールの集合を選択する問題である。ただし、プロジェクト制約値とは、プロジェクト制約の具体的な値を指すものとする。

狭義のゴール選択の対象となる2階層部分ゴールグラフの子ゴールの数が n 個だとすると、ゴール選択基準値を定める必要がある子ゴールの集合の数は $2^n - 1$ だけある。 n 個すべての子ゴールを要素に持つ集合を C とすると、1つ以上の要素を持つ C の部分集合の数 $(|\mathcal{P}(C) \setminus \{\emptyset\}|)$ だけ、ゴール選択基準値を定める必要があるからである(空集合 \emptyset をダミーゴール(1.7節(8)参照)に対応させることもできる。しかし、ダミーゴールは、定義上、狭義のゴール選択の候補にならないため、ゴール選択基準値を設定する意義はない)。 $2^n - 1$ だけある子ゴールの集合に対して、そのゴール選択基準値を計算する演算が必要になる(つまり、ゴールグラフは、広義のゴール選択(1.7節(10)参照)は、「要素還元的(reductive)」なシステムであるのに対して、狭義のゴール選択(1.7節(10)参照)は、「全体的(holistic)」なシステムである。両者はシステムとして階層をなしている。広義のゴール選択は、狭義のゴール選択の「スーパーシステム[浅居01, 出口09]」であり、狭義のゴール選択は、広義のゴール選択の「サブシステム[浅居01, 出口09]」である)ため、狭義のゴール選択には、指数時間を必要とする。特に、「プロジェクト制約値」を「重量」、「ゴール選択基準値」を「価値」に置換して考えれば、狭義のゴール選択は、まさしく「ナップサック問題」^{注2)}そのものである。ナップサック問題は、NP困難であることが知られているため、狭義のゴール選択もまたNP困難な問題である。

ナップサック問題では、「価値」はただ1つだけであるのに対し、狭義のゴール選択では、複数のゴール選択基準を考慮に入れる必要がある。たとえば、4.1.1節では、*gsl-dfs*のゴール選択基準として「貢献度」(2.3.1節(属性-1)参照)のみを考慮したが、貢献度以外に「顧客のニーズに関する妥当性」(3.1.1節参照)もまた重要なゴール選択基準である。さらに、貢献度と顧客のニーズに関する妥当性以外にも、理論上は無限にゴール選択基準を設定可能であり、より厳密に狭義のゴール選択を行うためには、より多くのゴール選択基準を設定する必要がある。したがって、狭義のゴール選択では、複数のゴール選択基準が設定されることを想定する必要がある。また、以上の事実は、「重量」に対しても同様にあてはまる。なぜならば、プロジェクト制約としては、「コスト」や「納期」を初めとする、複数の制約が考えられるためである。したがって、狭義のゴール選択は、一般に「重量」および「価値」が単一ではなく複数になるため、ナップサック問題よりも複雑な多目的最適化問題(多目的ナップサック問題)である。

以上より、現実の問題をなるべく厳密に考慮に入れつつ、多項式時間で狭義のゴール選択結果を得るア

ルゴリズムが必要となる。このアルゴリズムを構築する上で、種々の多属性意思決定方法 [Hwang 81] に基づくアルゴリズムを構築することが考えられる。また、ソフトウェア工学においては、「探索に基づくソフトウェア工学 (SBSE: Search Based Software Engineering)」と呼ばれる研究分野が 2001 年に発足し、現在に至るまで、競合する目的のバランスをとった上で、多項式時間で解を得るためのメタヒューリスティクス [久保 09, Michalewicz 10] が研究されている [Harman 09]。SBSE の諸研究成果は、上述のアルゴリズムを構築する上で参考になるだろう。

4.2 階層分析法に基づく形式的ゴール選択方法

ゴール選択アルゴリズム *gsl-dfs* (図 4.1) を適用する際には、子ゴール w の親ゴール v に対するゴール選択基準値を定める関数を定義する必要がある (4.1 節参照)。4.1 節では、子ゴールのゴール選択基準値として、貢献度を求める関数 $Cov(v, w)$ (式 (2.1)) を設定した場合の結果を示した。しかし、貢献度は顧客のニーズに関する妥当性 (3.1.1 節参照) との間にコンフリクトを持つ (3.1 節参照) ため、貢献度と顧客のニーズに関する妥当性の間のコンフリクトを考慮に入れた上で子ゴール w の親ゴール v に対するゴール選択基準値を定める関数が定義されるべきである。

この関数を、貢献度と顧客のニーズに関する妥当性を各々属性とする多属性意思決定方法として実現する場合を考える。多属性意思決定方法には、種々の方法が存在する [Hwang 81]。特に、階層分析法 (AHP: Analytic Hierarchy Process) [Saaty 80] は、これまでに多くの実問題に適用され、その有用性が実証されている方法である。そこで本節では、 w の親ゴール v に対するゴール選択基準値を定める関数として、階層分析法に基づいて形式的に狭義のゴール選択 (1.7 節 (10) 参照) を行う方法を提案する。

4.2.1 節では、階層分析法の概要について述べる。4.2.2 節では、階層分析法に基づく形式的ゴール選択方法について述べる。

4.2.1 階層分析法

階層分析法 (AHP: Analytic Hierarchy Process) [Saaty 80] では、階層構造の最上層に達成すべき目標を 1 つ設定し、これを「総合目的」と呼ぶ。最下層には最上層の目標を達成するための選択肢を設定する。各々の選択肢は「代替案」と呼ばれる。最上層と最下層の間の中間層には、総合目的に対する代替案の優先順位を決定するための属性を設定する。各々の属性は、「評価基準」と呼ばれる。評価基準は多階層でも構わない。このようにして作成される階層構造は「階層図」と呼ばれる。階層図の概念図を 4.11 に示す。

階層分析法は、総合目的に対する各評価基準の「重み」と、各評価基準に対する各代替案の「評価値」を定め、それらから決定した総合目的に対する各代替案の評価値 (「総合評価値」) によって、総合目的に対して最適な代替案を選択する多属性意思決定方法である。代替案 A_i ($i = 1, 2, \dots, m$) の総合評価値 $f(A_i)$ は、多属性意思決定において、総合評価値を求める際に最もよく用いられる、線形加重和によって評価される。すなわち、総合目的 G に対する評価基準 C_j ($j = 1, 2, \dots, n$) の重

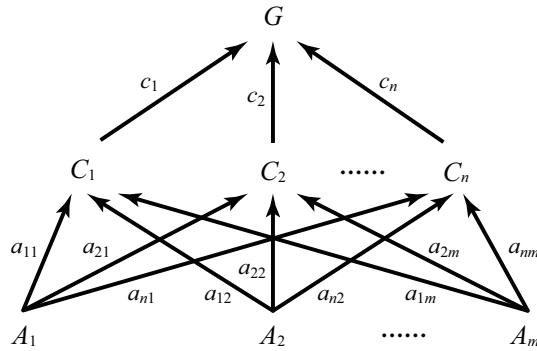


図 4.11: 階層図の概念図. G は総合目的, C_j は評価基準, A_i は代替案. c_j は, 総合目的に対する評価基準の重み, a_{ji} は, 評価基準に対する代替案の評価値

み c_j と, C_j に対する A_i の評価値 a_{ji} を用いて次式で求め, 総合評価値が最も高い代替案を最良の選択肢であると判断する.

$$f(A_i) = \sum_{j=1}^n c_j a_{ji} \tag{4.2}$$

c_j および a_{ji} の決定には, 一対比較行列を用いる. 一対比較行列は, 比較される評価基準または代替案の各々を各行および各列に配置した正方行列である. ここで, x_{ij} 成分は, 評価基準または代替案 X_i が同 X_j に対してどの程度重要であることを示す値である.

$$X = (x_{ij}) = \begin{matrix} & X_1 & X_2 & \dots & X_n \\ X_1 & \left(\begin{matrix} 1 & x_{12} & \dots & x_{1n} \\ x_{21} & 1 & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & 1 \end{matrix} \right) \end{matrix} \tag{4.3}$$

各 i, j に対して, x_{ij} と x_{ji} は互いに逆数の関係, すなわち, $x_{ij} = \alpha$ ならば $x_{ji} = \frac{1}{\alpha}$ (α は 0 でない定数) となるようにする. そのため, 主対角成分 x_{ii} の値は 1 となる. X_i と X_j の重みまたは評価値を, 各々 w_i, w_j とすると, $x_{ij} = \frac{w_i}{w_j}$ ならば $x_{ji} = \frac{w_j}{w_i}$ である. X に w_i のベクトル $w = (w_1, \dots, w_n)$ をかけると, $Xw = \lambda w$ が成立する. λ が X の固有値ならば, w は X の固有ベクトルである. 一方, X の階数は 1 であるから, それはただ 1 つの 0 でない固有値を持ち, 固有値の和が n に等しいことに注意すると, n が X のただ 1 つの 0 でない固有値であって, w は固有ベクトルであることがわかる. すなわち, w_i を求めることは, 固有ベクトルを求めることに帰着する.

ただし, あらかじめ w_i と w_j がわかっていない場合, この行列の各成分値を, 各評価基準または各代替案の比較 (一対比較) により決定する. 評価方法としては, 「基準尺度」[Saaty 80] (表 4.3) に基づく 5 段階評価が推奨されている. 本論文でもこの評価方法を採用する.

一対比較によって X を作成する場合, 必ずしも $x_{ij} = \frac{w_i}{w_j}$ になるとは限らない. そのため, 固有値も n になるとは限らない. そこで, 階層分析法では, $\sum_i w_i = 1$ となるように正規化された X の

表 4.3: 基準尺度 [Saaty 80]

重要度	定義
1	同程度重要
3	すこし重要
5	かなり重要
7	非常に重要
9	極めて重要

ただし, 2, 4, 6, 8 は中間のときに用い, 重要でないときは逆数を用いる.

主固有ベクトル (最大固有値に対する固有ベクトル) の成分として w_i を推定する. この計算法は「固有ベクトル法」と呼ばれる. しかし, 固有ベクトル法による計算は, 一般に複雑である. そのため, 簡易計算法として, 一対比較行列の各行の成分の幾何平均をとって $\sum_i w_i = 1$ となるように正規化する方法がある. この計算法は「幾何平均法」と呼ばれ, 固有ベクトル法の良い近似である. 特に, 幾何平均法は, $n \leq 3$ では, 固有ベクトル法と結果が一致する [森村 99]. 本論文では, $n \leq 3$ の場合のみを扱うため, 幾何平均法で十分である. したがって, 本論文では幾何平均法を採用する. 幾何平均法を用いた場合, w_i は次式で表される.

$$w_i = \frac{\sqrt[n]{\prod_{j=1}^n x_{ij}}}{\sum_{k=1}^n \sqrt[n]{\prod_{j=1}^n x_{kj}}} \quad (4.4)$$

X の最大固有値を λ_{\max} で表すことにすると, すべての i, j に対して $x_{ij} = \frac{w_i}{w_j}$ のとき, $\lambda_{\max} = n$ が成立する. このとき, 一対比較行列は完全に整合性を持つという. $n = 2$ の一対比較行列は常に完全な整合性を持っている. しかし, $n > 2$ の場合にはそうとは限らない [刀根 91]. 一対比較行列の整合性は, 次に示す「整合度指数」(C. I.: Consistency Index) [Saaty 80] を算出することにより調べられる.

$$\text{C. I.} = \frac{\lambda_{\max} - n}{n - 1} \quad \text{ただし, } \lambda_{\max} = \frac{\sum_{i=1}^n \lambda_i}{n}, \quad \lambda_i = \frac{\sum_{j=1}^n x_{ij} w_j}{w_i} \quad (4.5)$$

一対比較行列の整合性が良いほど C. I. の値は小さくなり, 完全に整合性を持つ場合には 0 となる. C. I. > 0.1 の場合には, 一対比較行列を再検討する.

4.2.2 階層分析法に基づく形式的ゴール選択方法

階層分析法に基づく形式的ゴール選択方法は, ゴール選択基準間のコンフリクト (2.4 節参照) がある 2 階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) に対して, 形式的に狭義のゴール選択 (1.7 節 (10) 参照) を行う方法である. そのため, 評価基準としては, ゴール選択基準を設定するの

が自然である。評価基準として「貢献度」(2.3.1節(属性-1)参照)と「顧客のニーズに関する妥当性」(3.1.1節参照)を設定した場合の階層図の例を図4.12に示す。階層分析法に基づく形式的ゴール

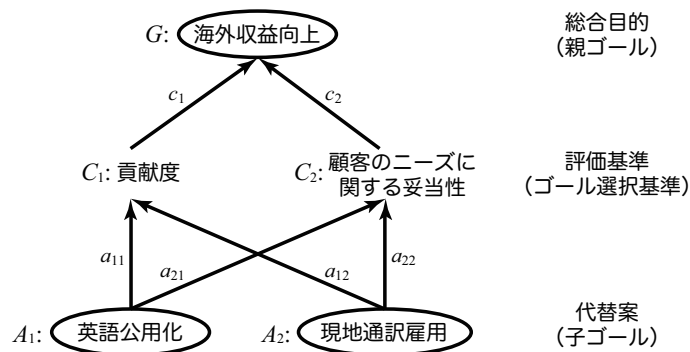


図 4.12: ゴール選択の階層図

ル選択方法の手順は次の通りである。(1) ゴールグラフ上の任意のゴールを1つ選定。(2) (1)で選定したゴールを親ゴールとする2階層-OR分解-部分ゴールグラフ(1.7節(9b)参照)を把握し、そのすべてのゴールに満足度行列を振る。(3) 階層図作成。(4) 総合目的(親ゴール)に対する各評価基準(ゴール選択基準)の重みを決定。(5) 各評価基準(ゴール選択基準)に対する各代替案(子ゴール)の評価値を決定。(6) 総合目的(親ゴール)に対する各代替案(子ゴール)の総合評価値を決定。

以下、各々の工程について、図3.1に対する適用結果と共に詳述する。

(1) ゴールグラフ上の任意のゴールを1つ選定

図3.1の「海外収益向上」を対象とする。階層分析法に基づく形式的ゴール選択方法は、「貢献度-妥当性-コンフリクト」(3.1.1節参照)など、「ゴール選択基準間のコンフリクト」(2.4節参照)がある場合にのみ必要となる。

(2) (1)で選定したゴールを親ゴールとする2階層-OR分解-部分ゴールグラフを把握し、そのすべてのゴールに満足度行列を振る

図3.1が対象となる。子ゴールの満足度行列は、図3.1の通りに振られたものとする^{注3)}。

(3) 階層図作成

階層図を作成する。図3.1に対して作成される階層図は、図4.12となる。総合目的には親ゴールを設定する。評価基準としては、貢献度(2.3.1節(属性-1)参照)と顧客のニーズに関する妥当性(3.1.1節参照)が設定されたものとする。代替案には子ゴールを設定する。

(4) 総合目的(親ゴール)に対する各評価基準(ゴール選択基準)の重みを決定

(4a) 総合目的に対する、評価基準の一对比較行列を作成

総合目的「海外収益向上」に対する評価基準の一对比較行列を作成する。この行列の各値は、ステークホルダの合意により決定されるものとする。ただし、一对比較される各々の評価基準の属性値を求める関数が定義可能な場合、評価基準 C_j に対する評価基

準 C_i の一対比較値は、次式で形式的に与えられるものとする。

$$Ctp(C_i, C_j) := \begin{cases} \text{rnd}_1\left(\frac{9}{10} \times Imp(f_{C_i}(C_i), f_{C_j}(C_j))\right) & \text{if } \text{rnd}_1\left(\frac{9}{10} \times Imp(f_{C_i}(C_i), f_{C_j}(C_j))\right) > 0, \\ 1 & \text{if } \text{rnd}_1\left(\frac{9}{10} \times Imp(f_{C_i}(C_i), f_{C_j}(C_j))\right) = 0, \\ 1/\text{rnd}_1\left(\frac{9}{10} \times |Imp(f_{C_i}(C_i), f_{C_j}(C_j))|\right) & \text{otherwise} \end{cases} \quad (4.6)$$

ただし、

$$Imp(f_{C_i}(C_i), f_{C_j}(C_j)) := \frac{\sum_{k=1}^m f_{C_i}(A_k) - \sum_{k=1}^m f_{C_j}(A_k)}{2m}, \quad i, j \in [1, n]$$

ただし、 f_{C_i} は、 C_i の属性値を求める関数。 f_{C_j} は、 C_j の属性値を求める関数。 $\text{rnd}_n(x)$ は、 $x \geq 0$ ならば、実数 x の小数点第 n 位を四捨五入した値。 $x < 0$ ならば、 $\text{rnd}_n(x) := -\text{rnd}_n(|x|)$ 。式 (4.6) から、 $Ctp(C_1, C_2) = 5$ であるため、総合目的に対する評価基準の一対比較行列は、図 4.13 となる。

G	C_1	C_2
C_1	1	5
C_2	1/5	1

図 4.13: 総合目的に対する評価基準の一対比較行列。 G は、総合目的「海外収益向上」、 C_1 、 C_2 は各々、評価基準「貢献度」、「顧客のニーズに関する妥当性」を表す

(4b) 総合目的に対する評価基準の一対比較行列の整合性判定

総合目的「海外収益向上」に対する評価基準「貢献度」と「顧客のニーズに関する妥当性」の重み (c_1, c_2) は、式 (4.4) から、 $(0.833, 0.167)$ ^{注4)}と求まる。また、 $n = 2$ であるため、 $C. I. = 0 (\leq 0.1)$ である (4.2.1 節参照)。そのため、総合目的に対する評価基準の一対比較行列の整合性に問題はない。よって、手順 (5) へ移る。

(5) 各評価基準 (ゴール選択基準) に対する各代替案 (子ゴール) の評価値を決定

(5a) 各評価基準に対する代替案の一対比較行列を作成

個々の評価基準ごとに代替案の一対比較行列を作成する。一対比較行列の各成分値は、ステークホルダ間の合意により決定されるものとする。ただし、評価基準を求める関数が定義可能な場合、評価基準 C のもとでの代替案 A_j に対する代替案 A_i の一対比較値

は、次式で形式的に与えられるものとする。

$$Atp(C, A_i, A_j) := \begin{cases} \text{rnd}_1\left(\frac{9}{10} \times Imp(f_C(A_i), f_C(A_j))\right) & \text{if } \text{rnd}_1\left(\frac{9}{10} \times Imp(f_C(A_i), f_C(A_j))\right) > 0, \\ 1 & \text{if } \text{rnd}_1\left(\frac{9}{10} \times Imp(f_C(A_i), f_C(A_j))\right) = 0, \\ 1/\text{rnd}_1\left(\frac{9}{10} \times |Imp(f_C(A_i), f_C(A_j))|\right) & \text{otherwise} \end{cases} \quad (4.7)$$

ただし、

$$Imp(f_C(A_i), f_C(A_j)) := \frac{f_C(A_i) - f_C(A_j)}{2}, \quad i, j \in [1, m]$$

ただし、 f_C は、 C の属性値を求める関数。 $\text{rnd}_n(x)$ は、 $x \geq 0$ ならば、実数 x の小数点第 n 位を四捨五入した値。 $x < 0$ ならば、 $\text{rnd}_n(x) := -\text{rnd}_n(|x|)$ 。 式 (4.7) から、 $Atp(C_1, A_1, A_2) = 3$ 、 $Atp(C_2, A_1, A_2) = \frac{1}{4}$ であるため、図 4.12 の各評価基準に対する代替案の一对比較行列は、図 4.14 となる。

(5b) 各評価基準に対する代替案の一对比較行列の整合性判定

各評価基準に対する代替案の一对比較行列の整合性を判定する。たとえば、評価基準の1つ「貢献度」に対する各代替案の評価値は、式 (4.4) を計算することによって、 $(a_{11}, a_{12}) = (0.75, 0.25)$ と求まる。また、 $n = 2$ であるため、 $C. I. = 0 (\leq 0.1)$ である (4.2.1 節参照)。もう一方の評価基準「顧客のニーズに関する妥当性」に対する各代替案の評価値も同様にして、 $(a_{21}, a_{22}) = (0.2, 0.8)$ と求まり、 $n = 2$ であるため、 $C. I. = 0 (\leq 0.1)$ である。各評価基準に対する代替案の一对比較行列の整合性がすべて 0.1 以下であるため、手順 (6) へ移る。

(6) 総合目的 (親ゴール) に対する各代替案 (子ゴール) の総合評価値を決定

総合目的に対する各代替案の総合評価値は、式 (4.2) から $(f(A_1), f(A_2)) = (0.65815, 0.34185)$ と求まる。 $f(A_1) > f(A_2)$ であるため、 A_1 がゴール選択されることになる。

C_1	A_1	A_2
A_1	1	3
A_2	1/3	1

(a)

C_2	A_1	A_2
A_1	1	1/4
A_2	4	1

(b)

図 4.14: 各評価基準に対する代替案の一对比較行列. (a), (b) は各々、評価基準「貢献度」、「顧客のニーズに関する妥当性」に対する代替案の一对比較行列である。 C_1 , C_2 は各々、評価基準「貢献度」、「顧客のニーズに関する妥当性」を表す。 A_1 , A_2 は各々、代替案「英語公用化」、「現地通訳雇用」を表す

4.2.2.1 考察

階層分析法に基づく形式的ゴール選択方法は、総合評価値を式(4.2)によって求める。ここで、式(4.2)は、 a_{ji} が定数ならば、任意の c_j に対して $f(A_i)$ は定数になるという性質を持つ。すなわち、次式が成立する。

$$(\forall k)(a_{ki} = a_{kj}) \Rightarrow f(A_i) = f(A_j) \quad (4.8)$$

ただし、 $i, j \in [1, m], k \in [1, n]$

a_{ji} が定数になり、式(4.8)が成立するのは、階層分析法に基づく形式的ゴール選択方法において、特別な場合である。この特別な場合に限り、総合評価値($f(A_i)$)は総合目的に対する評価基準の重み c_j によらず、常に一定となる。したがって、総合目的(親ゴール)に対して、代替案(子ゴール)はすべて同程度に重要であるため、子ゴールの任意の組合せを選択して構わない。よってこの場合、「すべての子ゴールを選択する」、「選択する子ゴールの組合せはランダムに決定する」など、任意のゴール選択方法が適用可能である。この場合、何らかのゴール選択方法をあらかじめ設定しておくことによって、4.2.2節(1)~(6)の手順を経るまでもなく、完全に形式的にゴール選択を行うことが可能である。ゴール g が選択し得る子ゴールの組合せの総数を $Tsg(g)$ とすると、 $Tsg(g) = \sum_{k=1}^n {}_n C_k = |\mathcal{P}(Sub(g)) \setminus \{\emptyset\}|$ が成立する。ただし、 ${}_n C_k$ は、 n 個の代替案から k 個選択する組合せの数。 $Sub(g)$ は、 g のすべての子ゴールを要素とする集合。

次に、貢献度と顧客のニーズに関する妥当性の他に、属性値を求める関数を定義することができない第三のゴール選択基準を考慮に入れる場合を考える。この例を図4.15に示す。図4.15では、図

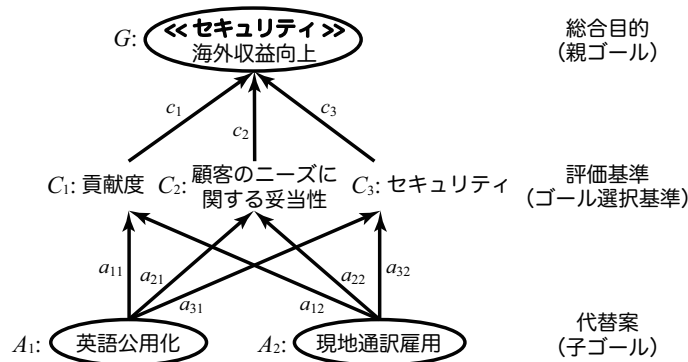


図 4.15: 意味タグをゴール選択基準に加えた場合のゴール選択の階層図

4.12に対して、新たに「セキュリティ」を評価基準として加えた階層図である。階層分析法を用いてゴール選択する既存の方法[Maiden 02, Ruhe 02, 井上 05, 山崎 06, アルダムラトフ 07]では、評価基準としてソフトウェアの品質特性、特に、非機能要求(1.7節(6)参照)を設定するものが多い。AGORA[Kaiya 02](2.3.1節参照)の持つ属性の1つである、意味タグ(2.3.1節(属性-3)参照)は、ソフトウェアの品質特性を、要求獲得の段階で、「ゴール特性」としてゴールに設定するものであり、特に非機能要求に基づいてゴール特性を定義することも可能である。そのため、意味タグはゴール選択基準として採用されることが望ましい。そこで、図4.15のように、総合目的となる親ゴールに意味タグが設定されており、かつ、階層分析法に基づく形式的ゴール選択方法を用い

てゴール選択を行う場合は、評価基準として親ゴールに設定されている意味タグをそのまま設定する（親ゴールに複数の意味タグが設定されている場合は、その各々を評価基準として設定する）。意味タグは、特定のゴールに設定されるため、特定のゴールに特化したゴール選択基準である。一方、貢献度と顧客のニーズに関する妥当性は、ゴール選択の必要がある任意のゴールに対して考慮されるべきゴール選択基準である。そのため、意味タグをゴール選択基準として考慮に入れた場合のゴール選択の階層図は、「貢献度」、「顧客のニーズに関する妥当性」、および「親ゴールに設定された意味タグ」が評価基準として設定される。図4.15では、親ゴールに意味タグ「セキュリティ」が設定されているため、貢献度、顧客のニーズに関する妥当性、セキュリティの3つが評価基準として設定されている。

特定の意味タグをゴール選択基準とする場合、その意味タグの属性値を定義する関数が定義可能であれば、式(4.6)によって、形式的に一对比較値が得られる。図4.15の評価基準「セキュリティ」は、開発対象の情報システムに応じて考慮すべき項目が変化するため、一般に適用可能な評価関数を定義できない。それ故、プロジェクトごとに関数を作成する必要がある。しかし、プロジェクトごとに関数を定義する場合、「セキュリティ」は抽象的なゴール選択基準であり、考慮すべき項目が多岐に渡ることから、評価関数を定義することは困難である場合が多いものと考えられる。そこで、ここでは、セキュリティの評価関数は作成されず、ステークホルダ間の合意に基づいて、一对比較値が決定されたものとする。仮想的な結果を、図4.16(a)に示す。この場合、総合目的「海外収益向上」に対する

G	C ₁	C ₂	C ₃
C ₁	1	5	1/9
C ₂	1/5	1	1/9
C ₃	9	9	1

(a)

G	C ₁	C ₂	C ₃
C ₁	1	3	1/9
C ₂	1/3	1	1/9
C ₃	9	9	1

(b)

図 4.16: 総合目的に対する評価基準の一对比較行列。(a)は、図 4.13 に基づいて最初に作成されたもの。(b)は、 $C. I. \leq 0.1$ となるように(a)を修正したもの。 G は、総合目的「海外収益向上」、 C_1, C_2, C_3 は各々、評価基準「貢献度」、「顧客のニーズに関する妥当性」、「セキュリティ」を表す

評価基準「貢献度」、「顧客のニーズに関する妥当性」、「セキュリティ」の重み(c_1, c_2, c_3)は、式(4.4)から、 $(0.151, 0.052, 0.797)^{\text{注4}}$ と求まる。しかし、式(4.5)から、 $C. I. = 0.147^{\text{注4}} (> 0.1)$ である。そのため、総合目的に対する評価基準の一对比較行列の整合性に問題があり、一对比較行列を再検討する必要がある。ここでは、再検討の結果、図4.16(b)が作成されたものとする。図4.16(b)は、式(4.4)から、 $(c_1, c_2, c_3) = (0.13, 0.062, 0.808)^{\text{注4}}$ であり、式(4.5)から、 $C. I. = 0.068^{\text{注4}} (\leq 0.1)$ である。そのため、図4.16(b)は、総合目的に対する評価基準の一对比較行列の整合性に問題がない。よって、手順(5)へ移る。

「セキュリティ」を評価基準とする場合、「現地通訳雇用」は、社内機密が海外の競合他社に漏洩する可能性が考えられることから、「英語公用化」に対して、優先順位は低くなる。ここでは、評価基準「セキュリティ」に対する代替案の一对比較行列が、ステークホルダ間の合意に基づいて、図4.17の通りに作成されたものとする。評価基準「セキュリティ」に対する各代替案の評価値は、式(4.4)を計算することによって、 $(a_{31}, a_{32}) = (0.9, 0.1)$ と求まる。また、 $n = 2$ であるため、 $C. I. = 0 (\leq 0.1)$

C_3	A_1	A_2
A_1	1	9
A_2	1/9	1

図 4.17: 評価基準「セキュリティ」に対する代替案の対比較行列. C_3 は, 評価基準「セキュリティ」を表す. A_1 , A_2 は各々, 代替案「英語公用化」, 「現地通訳雇用」を表す

である (4.2.1 節参照). 「セキュリティ」以外の評価基準「貢献度」および「顧客のニーズに関する妥当性」に対する各代替案の評価値は, $(a_{11}, a_{12}) = (0.75, 0.25)$, $(a_{21}, a_{22}) = (0.2, 0.8)$ であり, いずれも $C. I. = 0 (\leq 0.1)$ である (4.2.2 節 (5b) 参照). 各評価基準に対する代替案の対比較行列の整合性がすべて 0.1 以下であるため, 手順 (6) へ移る.

総合目的に対する各代替案の総合評価値は, 式 (4.2) から $(f(A_1), f(A_2)) = (0.8371, 0.1629)$ と求まる. よってこの場合, A_1 のみがゴール選択される.

4.3 ゴール選択基準の拡張

4.2.2 節で述べた「階層分析法に基づく形式的ゴール選択方法」は, 少なくともゴール選択基準間のコンフリクト (2.4 節参照) を起こすゴール選択基準の組の要素を各々属性とする. しかし, 各々の属性に妥当性 (1.7 節 (7) 参照) がない場合, この方法で選択されたゴールには妥当性がなく, 最終的に獲得される要求も妥当性を満たさないことになる. したがって, この方法を用いる際には, 各々のゴール選択基準 (属性) に対して, 妥当性があることを保証する必要がある. そのため, 各々のゴール選択基準は, なるべく拡張されることが望ましい.

コンフリクトを起こすゴール選択基準の組の例として, 「貢献度」 (2.3.1 節 (属性-1) 参照) と「顧客のニーズに関する妥当性」 (3.1.1 節参照) の対が挙げられる (3.1 節参照). したがって, この対の要素である, 貢献度と顧客のニーズに関する妥当性の各々の妥当性を保証する必要がある. この内, 顧客のニーズに関する妥当性に関しては, これを拡張する方法が提案されている [佐藤 12b]. 4.3.1 節では, この方法について述べる.

4.3.1 顧客のニーズに関する妥当性の拡張

顧客のニーズに関する妥当性を見積もる指標 $Cup(g)$ (式 (3.1)) は, ゴール g に対して振られた満足度行列のみから計算される. 満足度行列は特定のゴールと 1 対 1 対応があるため, $Cup(g)$ は特定のゴール g のみに着目して求まる値である. これに対して, 貢献度を見積もる指標 $Cov(g)$ (式 (2.2)) は, 特定のゴール g と g の親ゴールとの関係の中で定義される値である. 狭義のゴール選択 (1.7 節 (10) 参照) は, ある親ゴールに対して, 最適な子ゴールを選択する問題である. そのため, ゴール選択基準は, 貢献度のように, 親ゴールと子ゴールとの関係の中で見積もられるべきである. しかし, 任意の親ゴールに対して一意な値をとる $Cup(g)$ は, この観点からは, ゴール選択基準として不十分である. そこで, 本節では, ゴール g の顧客のニーズに関する妥当性を, g と g の親ゴール

との関係の中で見積もる指標 $Cup^E(g)$ の計算方法 [佐藤 12b] について述べる. そして, $Cup^E(g)$ は $Cup(g)$ の特別な場合であることを示し, $Cup^E(g)$ は $Cup(g)$ の拡張になっていることを明らかにする.

4.3.1.1 拡張された顧客のニーズに関する妥当性の計算手順

拡張された顧客のニーズに関する妥当性 $Cup^E(g)$ の計算手順は次の通りである [佐藤 13]. (1) ゴールグラフ上の任意のゴールを1つ選定. (2) (1) で選定したゴールを親ゴールとする2階層-OR分解-部分ゴールグラフ (1.7節 (9b) 参照) を把握し, そのすべてのゴールに満足度行列を振る. (3) 階層図作成. (4) 満足度行列に基づき, 総合目的 (親ゴール) に対する各評価基準 (顧客) の重みを決定. (5) 満足度行列に基づき, 各評価基準 (顧客) に対する各代替案 (子ゴール) の評価値を決定. (6) 総合目的 (親ゴール) に対する各代替案 (子ゴール) の総合評価値 (拡張された顧客のニーズに関する妥当性) を決定.

以下, 各々の工程について, 図 4.18 [新原 04, 佐藤 13] に対する適用結果と共に詳述する^{注5)}.

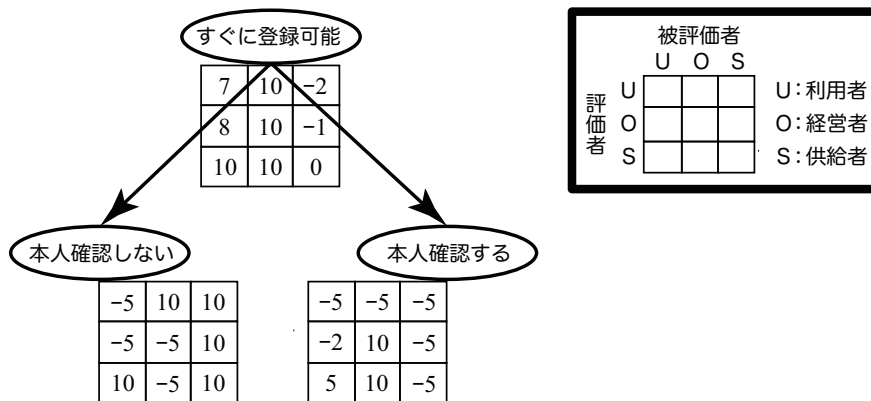


図 4.18: 適用対象の2階層-OR分解-部分ゴールグラフ [新原 04, 佐藤 13]

(1) ゴールグラフ上の任意のゴールを1つ選定

図 4.18 の「すぐに登録可能」を対象とする.

(2) (1) で選定したゴールを親ゴールとする2階層-OR分解-部分ゴールグラフを把握し, そのすべてのゴールに満足度行列を振る

図 4.18 が対象となる. 各ゴールの満足度行列は, 図 4.18 の通りに振られたものとする.

(3) 階層図作成

階層図を作成する. 図 4.18 に対して作成される階層図は, 図 4.19 となる [佐藤 13]. 総合目的には親ゴールを設定する. 図 4.18 の満足度行列から, ステークホルダは利用者, 経営者, 供給者の3者であることがわかり, このうち, 顧客に該当するステークホルダは, 利用者と経営者である (2.3.1節 (属性-2) 参照) ため, 評価基準として, これら2者を設定する. 代替案には子ゴールを設定する.

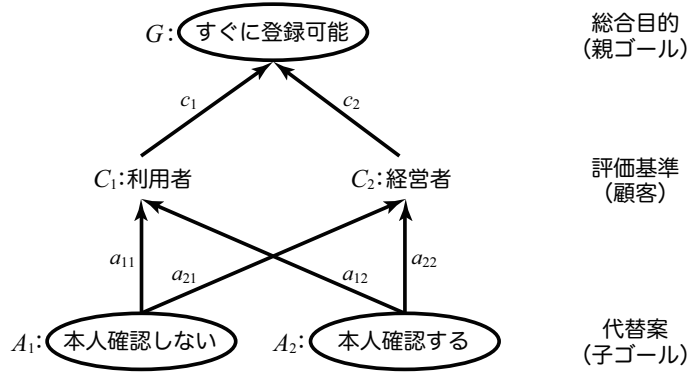


図 4.19: 提案方法の階層図 [佐藤 13]

(4) 満足度行列に基づき、総合目的（親ゴール）に対する各評価基準（顧客）の重みを決定

満足度行列から、総合目的に対する評価基準の一対比較行列を作成する。本論文では、総合目的または評価基準 x に関する、評価基準または代替案 y' に対する同 y の一対比較値は、次式で定義する関数 $Ptp(Stp(y, x), Stp(y', x))$ で与えられるものとする。

$$Ptp(Stp(y, x), Stp(y', x)) := \begin{cases} \text{rnd}_1\left(\frac{9}{10} \times Imp(Stp(y, x), Stp(y', x))\right) & \text{if } \text{rnd}_1\left(\frac{9}{10} \times Imp(Stp(y, x), Stp(y', x))\right) > 0, \\ 1 & \text{if } \text{rnd}_1\left(\frac{9}{10} \times Imp(Stp(y, x), Stp(y', x))\right) = 0, \\ 1 / \text{rnd}_1\left(\frac{9}{10} \times |Imp(Stp(y, x), Stp(y', x))|\right) & \text{otherwise} \end{cases} \quad (4.9)$$

ただし、

$$Imp(Stp(y, x), Stp(y', x)) := \frac{Stp(y, x) - Stp(y', x)}{2},$$

$$Stp(y, x) := \begin{cases} \frac{\sum_{s \in Stakeholder} m(x)_{s,y}}{|Stakeholder|} & \text{if } x = g, \\ \frac{\sum_{s \in Stakeholder} m(y)_{s,x}}{|Stakeholder|} & \text{otherwise} \end{cases}$$

ただし、 $m(g)_{s,s'}$ は、ステークホルダ $s \in Stakeholder$ から見た、ステークホルダ $s' \in Stakeholder$ にとっての、ゴール g の満足度。 $Imp(Stp(y, x), Stp(y', x))$ は、 $Stp(y', x)$ に対する $Stp(y, x)$ の重要度。 $\text{rnd}_n(r)$ は、 $r \geq 0$ ならば、実数 r の小数点第 n 位を四捨五入した値。 $r < 0$ ならば、 $\text{rnd}_n(r) := -\text{rnd}_n(|r|)$ 。 $Stp(y, x)$ は、ゴール g に振られた満足度行列において、すべてのステークホルダがステークホルダ s' に対して振った満足度の平均。ただし、 $x = g \Leftrightarrow y = s'$ 、 $x = s' \Leftrightarrow y = g$ 。

図 4.19 の総合目的に対する評価基準の一対比較行列 [佐藤 13] を図 4.20 に示す。作成過程は

次の通りである。

(4a) 総合目的に対する評価基準の対比較行列を作成

図4.19において、評価基準（顧客）は、「利用者」と「経営者」である。総合目的「すぐに登録可能」に対するこれら2つの評価基準の対比較行列は、式(4.9)から計算される。たとえば、1行2列成分は、 $Stp(C_1, G) = 8.333$ ^{注4)}、 $Stp(C_2, G) = 10$ より、 $Imp(Stp(C_1, G), Stp(C_2, G)) = -0.8335$ であるから、 $rnd_1(\frac{9}{10} \times Imp(Stp(C_1, G), Stp(C_2, G))) = -1 (< 0)$ となる。したがって、求める1行2列成分の値は、 $Ptp(Stp(C_1, G), Stp(C_2, G)) = 1$ と定まる。他の成分値も同様にして計算される。

G	C_1	C_2
C_1	1	1
C_2	1	1

図4.20: 満足度行列から作成される、総合目的に対する評価基準の対比較行列[佐藤 13]。
 G は、総合目的「すぐに登録可能」、 C_1 、 C_2 は各々、評価基準「利用者」、「経営者」を表す

(4b) 総合目的に対する評価基準の対比較行列の整合性判定

総合目的に対する評価基準の対比較行列の整合性を判定する。各評価基準の重みは、式(4.4)を計算することによって、 $(c_1, c_2) = (0.5, 0.5)$ と求まる。また、 $n = 2$ であるため、 $C.I. = 0 (\leq 0.1)$ である(4.2.1節参照)。総合目的に対する評価基準の対比較行列の整合性が0.1以下であるため、手順(5)へ移る。

(5) 満足度行列に基づき、各評価基準（顧客）に対する各代替案（子ゴール）の評価値を決定

満足度行列から、個々の評価基準ごとに代替案の対比較行列を作成する。この対比較行列の各成分値は、式(4.9)で与えられるものとする。図4.19の各評価基準に対する、代替案の対比較行列[佐藤 13]を図4.21に示す。このうち、評価基準の1つ「利用者」に対する代替案の対比較行列(図4.21(a))の作成過程は次の通りである。

(5a) 各評価基準に対する代替案の対比較行列を作成

図4.18において、代替案（子ゴール）は、「本人確認しない」と「本人確認する」である。評価基準の1つ「利用者」に対するこれら2つの代替案の対比較行列は、式(4.9)から計算される。たとえば1行2列成分は、 $Stp(A_1, C_1) = 0$ 、 $Stp(A_2, C_1) = -0.667$ ^{注4)}より、 $Imp(Stp(A_1, C_1), Stp(A_2, C_1)) = 0.3335$ であるから、 $rnd_1(\frac{9}{10} \times Imp(Stp(A_1, C_1), Stp(A_2, C_1))) = 0$ となる。したがって、求める1行2列成分の値は、 $Ptp(Stp(A_1, C_1), Stp(A_2, C_1)) = 1$ と定まる。他の成分値も同様にして計算される。

(5b) 各評価基準に対する代替案の対比較行列の整合性判定

各評価基準に対する代替案の対比較行列の整合性を判定する。たとえば、評価基準の1つ「利用者」に対する各代替案の評価値は、式(4.4)を計算することによって、

$(a_{11}, a_{12}) = (0.5, 0.5)$ と求まる。また、 $n = 2$ であるため、 $C. I. = 0 (\leq 0.1)$ である (4.2.1 節参照)。

「経営者」に対する代替案の一対比較行列も同様にして作成される。「経営者」に対する各代替案の評価値は、 $(a_{21}, a_{22}) = (0.333, 0.667)$ ^{注4)}であり、 $n = 2$ であるため、 $C. I. = 0 (\leq 0.1)$ である (4.2.1 節参照)。各評価基準に対する代替案の一対比較行列の整合性がすべて 0.1 以下であるため、手順(6)へ移る。

(6) 総合目的(親ゴール)に対する各代替案(子ゴール)の総合評価値(拡張された顧客のニーズに関する妥当性)を決定

総合目的に対する各代替案の総合評価値は、式(4.2)から、 $(f(A_1), f(A_2)) = (0.4165, 0.5835)$ と求まる。拡張されたゴール g の顧客のニーズに関する妥当性を $Cup^E(g)$ で表すことにすると、 A_1 と A_2 の拡張された顧客のニーズに関する妥当性は、 $Cup^E(A_1) = f(A_1) = 0.4165$ 、 $Cup^E(A_2) = f(A_2) = 0.5835$ である。 $Cup^E(g)$ をゴール選択基準とする場合、子ゴールのうち $Cup^E(g)$ が最大である A_2 、すなわち「本人確認する」が選択される。

C_1	A_1	A_2
A_1	1	1
A_2	1	1

(a)

C_2	A_1	A_2
A_1	1	1/2
A_2	2	1

(b)

図 4.21: 満足度行列から作成される、各評価基準に対する代替案の一対比較行列 [佐藤 13]。
(a), (b) は各々、評価基準「利用者」、「経営者」に対する代替案の一対比較行列である。 C_1, C_2 は各々、評価基準「利用者」、「経営者」を表す。 A_1, A_2 は各々、代替案「本人確認しない」、「本人確認する」を表す

4.3.1.2 評価

提案方法 (4.3.1.1 節参照) は、満足度行列 [Kaiya 02] (2.3.1 節 (属性-2) 参照) を一対比較行列に変換する関数 (式 (4.9)) を提供している。これにより、提案方法の階層図 (図 4.19) における各階層の要素間の一対比較行列は、すべて形式的に作成することができる。しかし、図 4.18 の親ゴールの満足度行列の各成分値は、筆者が恣意的に振ったものである (4.3.1.1 節参照) ため、この満足度行列を式 (4.9) によって変換した、総合目的に対する評価基準の一対比較行列 (図 4.20) には、筆者の恣意性が反映されている。この恣意性が、ゴール選択結果にどの程度影響するかを調べるために、 C_2 に対する C_1 の一対比較値 (図 4.20 の 1 行 2 列成分値) を、基準尺度 (表 4.3) が取り得るすべての値の範囲内で順に変化させた場合の、 $Cup^E(A_1)$ と $Cup^E(A_2)$ (4.3.1.1 節 (6) 参照) のグラフを作成した。結果 [佐藤 13] を図 4.22 に示す。各値の具体的な値 [佐藤 13] は表 4.4 の通りである。

図 4.22 と表 4.4 から、次のことがわかる。

- (1) すべての基準尺度で $Cup^E(A_1) < Cup^E(A_2)$ である。したがって、任意の総合目的に対する評価基準の一対比較行列に対して、 A_2 が選択される。これは、次に述べる性質による。式

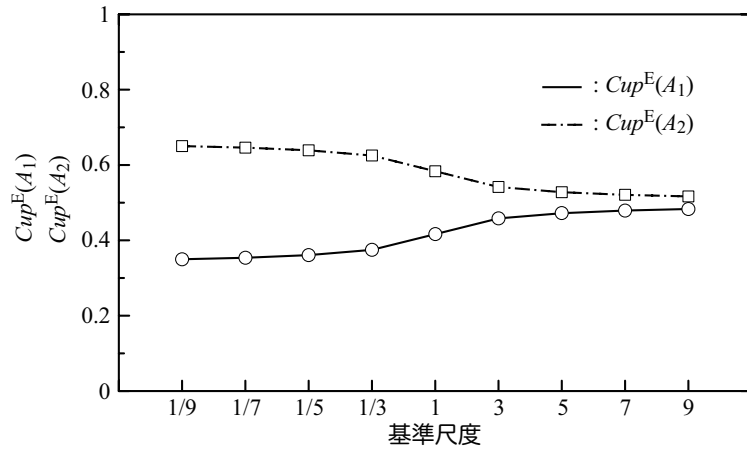


図 4.22: C_2 に対する C_1 の一対比較値に対する $Cup^E(A_1)$ と $Cup^E(A_2)$ のグラフ [佐藤 13].
実線は $Cup^E(A_1)$. 一点鎖線は $Cup^E(A_2)$

表 4.4: C_2 に対する C_1 の一対比較値に対する $Cup^E(A_1)$ と $Cup^E(A_2)$ [佐藤 13]

C_2 に対する C_1 の一対比較値	$Cup^E(A_1)$	$Cup^E(A_2)$
1/9	0.3497	0.6503
1/7	0.353875	0.646125
1/5	0.360889	0.639111
1/3	0.37475	0.62525
1	0.4165	0.5835
3	0.45825	0.54175
5	0.472111	0.527889
7	0.479125	0.520875
9	0.4833	0.5167

(4.2) は、各評価基準に対して、代替案の優先順位が一致していれば、 c_j はゴール選択に影響を与えず、 a_{ji} のみでゴール選択結果が決定されるという性質を持つ。すなわち、次式が成立する。

$$(\forall k)(a_{ki} \leq a_{kj}) \Rightarrow Cup^E(A_i) \leq Cup^E(A_j) \tag{4.10}$$

ただし、 $i, j \in [1, m], i \neq j, k \in [1, n]$

この例では、 $a_{11} = a_{12}$ かつ $a_{21} < a_{22}$ であるため、式 (4.10) より、 $Cup^E(A_1) < Cup^E(A_2)$ であることがわかる。よって、 A_2 が選択される。

(2) 基準尺度が 1 に近づくほど、隣り合う $Cup^E(g)$ の間の変化の割合の絶対値が大きく、逆に、1 から遠ざかるほど小さい。この結果は、総合目的に対する評価基準の一対比較値の偏りが小さい場合、ゴール選択結果は変更されやすく、逆に、大きい場合、ゴール選択結果は変更されにくいことを示している。

(1) より、総合目的に対する評価基準の一対比較行列がゴール選択結果に影響するのは、各評価基

準に対する代替案の優先順位が異なる場合である. そのため, 図4.21(a)の代わりに, 図4.21(b)とは評価基準に対する代替案の優先順位が異なる, 仮想的な一対比較行列[佐藤 13](図4.23(a)')を考える. この場合の C_2 に対する C_1 の一対比較値を, 基準尺度がとり得るすべての範囲内で順に変化させた場合の, $Cup^E(A_1)$ と $Cup^E(A_2)$ のグラフ[佐藤 13]とその具体的な値[佐藤 13]を各々図4.24と表4.5に示す. 図4.24と表4.5から, 基準尺度が1未満では A_2 が選択され, 1以上では A_1

表 4.5: C_2 に対する C_1 の一対比較値に対する $Cup^E(A_1)$ と $Cup^E(A_2)$ [佐藤 13]

C_2 に対する C_1 の一対比較値	$Cup^E(A_1)$	$Cup^E(A_2)$
1/9	0.3872	0.6128
1/7	0.40075	0.59925
1/5	0.423514	0.576486
1/3	0.4685	0.5315
1	0.604	0.396
3	0.7395	0.2605
5	0.784486	0.215514
7	0.80725	0.19275
9	0.8208	0.1792

C_1	A_1	A_2
A_1	1	7
A_2	1/7	1

(a)'

C_2	A_1	A_2
A_1	1	1/2
A_2	2	1

(b)

図 4.23: 各評価基準に対する代替案の一対比較行列[佐藤 13]. (b)は図4.21(b)と同じ行列. (a)'は, 評価基準に対する代替案の優先順位が(b)と異なる仮想的な行列

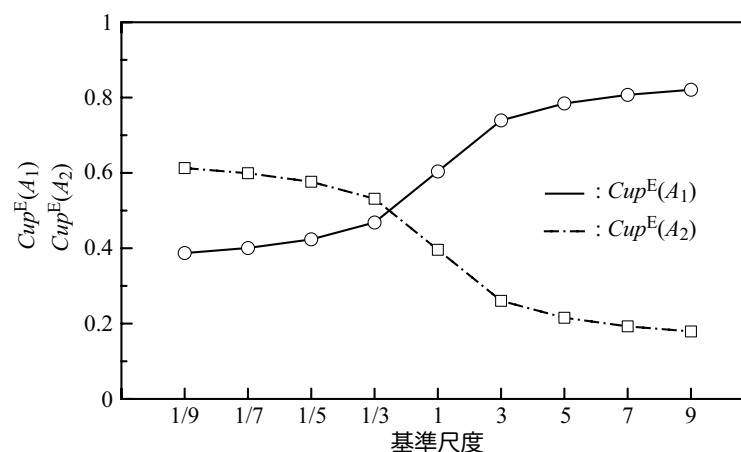


図 4.24: C_2 に対する C_1 の一対比較値に対する $Cup^E(A_1)$ と $Cup^E(A_2)$ のグラフ[佐藤 13]. 実線は $Cup^E(A_1)$. 一点鎖線は $Cup^E(A_2)$

が選択されることがわかる. よってこの場合, ゴール選択結果は総合目的に対する評価基準の一対比較行列に依存して変化する. これは, 次に述べる性質による. 式(4.2)は, 任意の j に対して c_j が一

定値ならば、ゴール選択に c_j は影響を与えないことを示している。すなわち、次式が成立する。

$$(\forall j)(c_j = \alpha) \Leftrightarrow \text{Cup}^E(A_i) = \alpha \times \sum_j a_{ji} \quad (4.11)$$

ただし、 α は定数、 $i \in [1, m]$, $j \in [1, n]$

基準尺度が1の場合、式(4.4)より、任意の j に対して、 $c_j = 0.5$ である。よってこの場合、式(4.11)が成立する。また、式(4.4)と式(4.9)から、 a_{ji} は $\text{Stp}(y, x)$ に対応する。そこで、任意の j に対して、 $c_j = 1$ とし、 a_{ji} の代わりに $\text{Stp}(y, x)$ を用いると、式(3.1)と式(4.2)から、次式が成立する。

$$\text{Cup}(g) = \frac{\sum_j \text{Stp}(y, x)}{|\text{Customer}|} \quad (4.12)$$

ただし、 $g = y = A_i$, $x = C_j$, $i \in [1, m]$, $j \in [1, n]$

提案方法において式(4.11)が成立するのは、基準尺度が1のときだけである。そのため、提案方法において、基準尺度が1の場合にのみ式(4.12)が成立する。したがって、 $\text{Cup}(g)$ は基準尺度が1の場合の $\text{Cup}^E(g)$ に対応する。

以上より、 $\text{Cup}^E(g)$ は $\text{Cup}(g)$ の特別な場合に対応しており、 $\text{Cup}^E(g)$ は $\text{Cup}(g)$ の拡張になっていることがわかる。この結果は、提案方法は、 $\text{Cup}(g)$ を拡張する計算方法として機能することを示すものである。

4.3.1.3 考察

$\text{Cup}(g)$ (式(3.1)) は、ゴール g に対して一意に定まる値である。一方、 $\text{Cup}^E(g)$ (4.3.1.1節(6)参照) は、ゴール g の親ゴールおよび、親ゴールの分解先となっているすべての子ゴールに依存して値が変化する。一般に、子ゴールは複数の親ゴールに隣接している場合があるため、 $\text{Cup}^E(g)$ は、複数の値を持ち得る。しかし、このことによる問題はない。顧客のニーズに関する妥当性は、ゴール選択基準として定義されている[佐藤 11] (3.1.1節参照) ため、 $\text{Cup}^E(g)$ は、ゴール選択基準として機能すればよいからである。 $\text{Cup}^E(g)$ がゴール選択基準として機能するためには、それが、ゴールグラフ上の任意の2階層-OR分解-部分ゴールグラフ(1.7節(9b)参照)の親ゴールに対して、その任意の子ゴールに一意な値を与えるような基準であればよい。提案方法では、式(4.2)、式(4.4)、式(4.9)の定義より、このことは保証されている。したがって、 $\text{Cup}^E(g)$ は、ゴールグラフ上の任意の2階層-OR分解-部分ゴールグラフに対して適用可能である。

$\text{Cup}^E(g)$ は、 $\text{Cup}(g)$ と同様に満足度行列のみから機械的に計算可能であるため、 $\text{Cup}(g)$ と比べて、計算上の付加的な人手コストは発生しない。ただし、 $\text{Cup}(g)$ は、ゴール g に対して振られた満足度行列のみから計算される(3.1.1節参照)のに対して、 $\text{Cup}^E(g)$ は、 g だけではなく、 g を子ゴールとする、2階層-OR分解-部分ゴールグラフを構成するすべてのゴールに対して振られた満足度行列から計算される(4.3.1.1節(6)参照)。したがって、 $\text{Cup}^E(g)$ は、 $\text{Cup}(g)$ と比べて、ゴール g 以外のゴールの満足度行列を作成する必要がある分だけ、適用上の人手コストがかかる。

$\text{Cup}(g)$ はゴール選択基準として定義されている[佐藤 11] (3.1.1節参照) ため、拡張された $\text{Cup}(g)$

である $Cup^E(g)$ もまた、ゴール選択基準として採用可能である。ゴール指向要求分析を用いて要求獲得を行う場合、ゴールグラフ上の2階層-OR分解-部分ゴールグラフに対して、 $Cup(g)$ ではなく、 $Cup^E(g)$ を用いた方が、より綿密に顧客のニーズに関する妥当性を検討した要求が獲得されることになる。これは、ゴール指向要求分析において、顧客のニーズを満足する要求の獲得に寄与する。したがって、獲得された要求に基づいて作成される要求仕様書の妥当性 [IEEE Standards Board 98] (Correctness) もまた、向上するものと考えられる。

4.4 本章のまとめ

4.1節では、ゴール選択アルゴリズム $gsl-dfs$ について述べた。 $gsl-dfs$ は、深さ優先探索 [Aho 83] に基づいて構築された。既存のゴールグラフに対して、ゴール選択基準として貢献度 (2.3.1節 (属性-1) 参照) を採用した場合について $gsl-dfs$ の評価を行い、 $gsl-dfs$ が正しく機能することを確認した。

4.2節では、「階層分析法に基づく形式的ゴール選択方法」を提案した。この方法は、狭義のゴール選択 (1.7節 (10) 参照) において、ゴール選択基準間のコンフリクト (2.4節参照) が存在する場合に必要となる。4.1.1節の評価結果から、 $gsl-dfs$ は、単一のゴール選択基準で狭義のゴール選択を行う場合は、既存のゴールグラフに対して機能することが確かめられた。しかし、複数のゴール選択基準で狭義のゴール選択を行う場合は、ゴール選択基準間のコンフリクトを考慮に入れる必要がある。提案方法を既存の2階層-OR分解-部分ゴールグラフに対して適用した結果、提案方法は形式的に狭義のゴール選択を行う方法として正しく機能することが確かめられた。

4.3節では、ゴール選択基準の拡張について述べた。4.2節で提案した形式的に狭義のゴール選択を行う方法は、ゴール選択基準の妥当性に依存してゴール選択結果が変化するため、ゴール選択基準そのものの品質が保証されていることが重要なためである。4.3.1節では、ゴール選択基準の1つである、ゴール g の顧客のニーズに関する妥当性 (3.1.1節参照) を見積もる指標 $Cup(g)$ (式 (3.1)) を階層分析法に基づいて拡張する方法について述べた。

後注

注1) ゴールグラフ上の個々の2階層-部分ゴールグラフ (1.7節 (9) 参照) に対する狭義のゴール選択 (1.7節 (10) 参照) に要する時間計算量が $O(x)$ であるとする、ただ1つの子ゴールを選択する場合、1本の枝に対して1つの2階層-部分ゴールグラフが対応するため、 $O(x)$ は枝1本あたりの時間計算量でもある。一方、ゴールグラフ上の枝の数が e であるとする、広義のゴール選択の時間計算量は、線形時間 $O(e)$ である (4.1.2節参照)。ここで、狭義のゴール選択結果が多項式時間で得られる場合、すなわち、 $O(x) = O(n^k)$ (ただし、ゴールグラフ上の個々の2階層-部分ゴールグラフの子ゴールの数を k 、子ゴール1つあたりの時間計算量を n とする) である場合、広義のゴール選択結果は、 $O(e) \times O(n^k) = O(en^k) = O(n^k)$ となり、やはり多項式時間で得られることがわかる。一方、狭義のゴール選択結果に指数時間を要する場合、すなわち $O(x) = O(2^n)$

である場合、広義のゴール選択結果は、 $\mathcal{O}(e) \times (2^n) = \mathcal{O}(e2^n) = \mathcal{O}(2^n)$ となり、指数時間を要することになってしまう。しかし、実用上のゴールグラフにおいて、2桁以上の子ゴールと隣接する親ゴールは、まず存在しない。そこで、子ゴールの数が1桁 ($k < 10$) であると仮定すると、個々の2階層部分ゴールグラフに対して、たかだか 2^{10} 回程度の計算をすればよいことになる。また、実用上のゴールグラフにおけるゴール数は 10^3 以下 [van Lamsweerde 04] であるため、ゴールグラフ中の2階層部分ゴールグラフの数は、だいたい $\frac{10^3}{y}$ (y は、個々の2階層部分ゴールグラフにおけるゴール数) であるから、 10^3 未満である。よって、狭義のゴール選択結果を得るのに指数時間を要するとしても、広義のゴール選択結果は、たかだか $10^3 \times 2^{10} = 1,024,000$ 回未満、つまり、およそ百万回未満の計算回数で済む。

注2) 各要素 i の重量 a_i と価値 c_i 、およびナップサックの重量の上限 b が与えられたとき、与えられた n 個の要素集合からいくつかを選び、選ばれた要素の重量の合計が b を超えないという条件のもとで、価値の合計を最大化する問題 [茨木 11].

注3) 貢献度 (2.3.1 節 (属性-1) 参照) と顧客のニーズに関する妥当性 (3.1.1 節参照) は、狭義のゴール選択 (1.7 節 (10) 参照) の必要がある任意のゴールに対して考慮されるべきゴール選択基準である (4.2.2.1 節参照)。この内、顧客のニーズに関する妥当性を見積もる指標として、 $Cup(g)$ (式 (3.1)) を用いる場合は、親ゴールに対して満足度行列を振る必要はない。一方、 $Cup^E(g)$ (4.3.1.1 節 (6) 参照) を用いる場合は、親ゴールに対して満足度行列を振る必要がある。

注4) 小数点第4位で四捨五入した値を示している。

注5) [新原 04] に掲載されているゴールグラフでは、子ゴール「本人確認しない」および「本人確認する」に対して満足度行列が振られているが、親ゴール「すぐに登録可能」に対して満足度行列は振られていない。本論文では、提案方法に必要なため、「すぐに登録可能」に対しても満足度行列を振る。「すぐに登録可能」に振られた満足度行列の各成分値は、筆者が恣意的に振った値である点に注意されたい。「本人確認しない」および「本人確認する」の満足度行列の各成分値は、[新原 04] のゴールグラフのものを踏襲している。

ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法

5.0 本章の概要

ゴール指向要求分析に基づく要求獲得を形式的に行う上で「ゴール選択基準間のコンフリクト」(2.4節参照)が問題となる。3章では、このコンフリクトを検出する方法と、その解決方法について述べた。一方、4章では、ゴール指向要求分析における形式的ゴール選択方法を提案した。本章では、3章および4章の内容を組み合わせることで、ゴール選択基準間のコンフリクトを考慮し、なるべく解決した上で、形式的に「広義のゴール選択」(1.7節(10)参照)を行う方法を提案する。5.1節では、この方法の詳細について述べる。5.2節では、既存の例題に対して提案方法を適用し、適切に機能することを確認する。5.3節では、本論文の補遺について述べる。

5.1 ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法

本節では、前章までに述べてきた方法を組み合わせることで、ゴール選択基準間のコンフリクトを考慮に入れた上で、広義のゴール選択(1.7節(10)参照)を形式的に行う方法を提案する。本論文では、この方法を「ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法」と呼ぶ。ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法のフローチャートを図5.1に示す。図5.1のうち、「形式的ゴール選択の準備」と「形式的ゴール選択」の詳細を、各々図5.2と図5.3に示す。図5.3のうち、「ゴール選択基準間のコンフリクト検出」の詳細を図5.4に示す。図5.2のうち、「属性の付与」は、ゴールグラフ中のゴールまたは枝に対して、ゴール選択上必要となる属性を付与する措置である。ゴール選択基準となる属性のうち、「貢献度」(2.3.1節(属性-1)参照)と「顧客のニーズに関する妥当性」(3.1.1節参照)は、常に考慮されるべきである(4.2.2.1節参照)。したがって、これら2つの属性は必ず付与する。ただし、顧客のニーズに関する妥当性は、「満足度行列」(図2.6)から算出される(3.1.1節および4.3.1節参照)属性であることから、ゴールグラフに対して直接付与することができない属性である。ゴールグラフに対して付与されるのは満足度行列であ

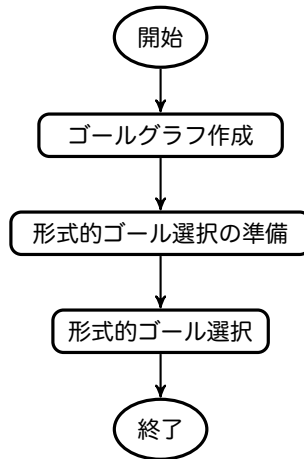


図 5.1: ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法

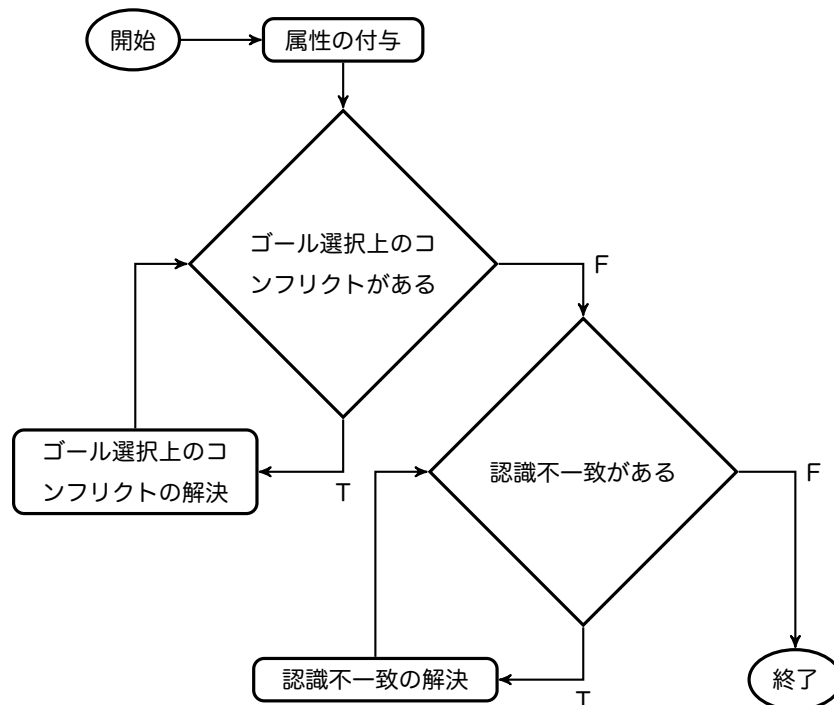


図 5.2: 形式的ゴール選択の準備

る点に注意されたい。「ゴール選択上のコンフリクト」およびその予防方法に関しては、2.3.1.2節で述べた通りである。ゴール選択上のコンフリクトとは、少なくとも1つの親ゴールに対して負の貢献度で接続しているゴールが達成されることである。 IG をすべての初期ゴール(2.2節参照)を要素とする集合、 SFG を広義のゴール選択(1.7節(10)参照)がなされた(要求として採用された)すべての最終ゴール(2.2節参照)を要素とする集合とすると、ゴール選択上のコンフリクトを持つゴールが、 $(\forall g \in IG, \forall g' \in SFG)Path_{(g, g')}$ (2.3.1.1節参照)の要素になる場合、広義のゴール選択がなされた最終ゴール(要求)を達成すると、初期ゴールの達成に必要となるゴールを阻害するゴールを達成することになってしまう。ゴール選択上のコンフリクトを解決することは、この事態を予防するための措置である。「ステークホルダ間の認識不一致」に関しては、2.3.1.3節で述べた通りで

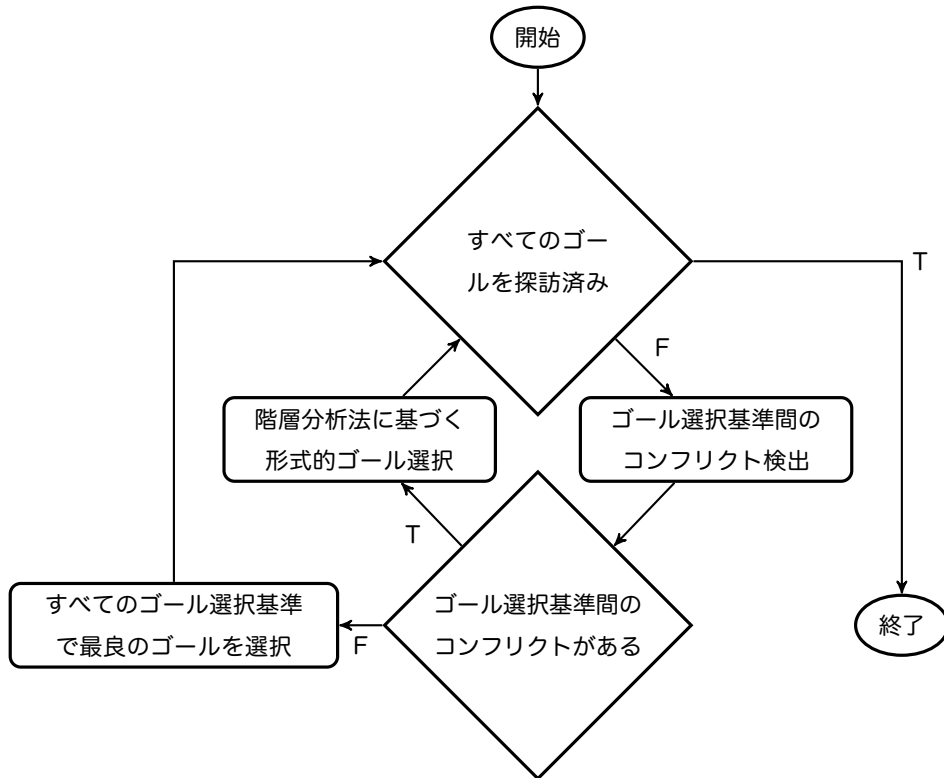


図 5.3: 形式的ゴール選択

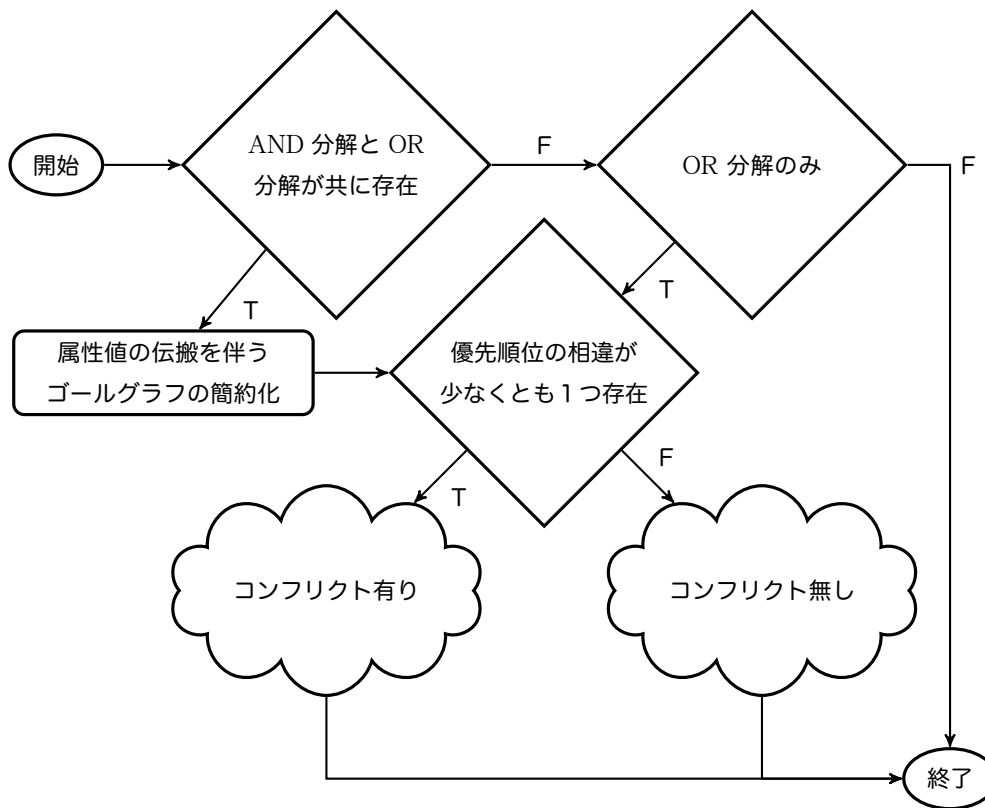


図 5.4: ゴール選択基準間のコンフリクト検出

ある。縦数値対立を持つ満足度行列に対しては、ステークホルダ間で話し合いを行い、被評価者の意図を理解した上で満足度行列を振り直す。これは、満足度行列の品質を保証するための措置である。満足度行列の品質を保証することは、満足度行列から定義されるゴール選択基準である $Cup(g)$ (式(3.1)) の品質を保証することになる。上述の通り、 $Cup(g)$ は、「貢献度」と共に常に採用されるべきゴール選択基準である、「顧客のニーズに関する妥当性」を見積もる指標である (3.1.1 節参照) ため、この操作は狭義のゴール選択 (1.7 節 (10) 参照) 上必須である。

図 5.3 のうち、「ゴール選択基準間のコンフリクト検出」を行う方法に関しては、3 章で述べた通りである。「階層分析法に基づく形式的ゴール選択」および「すべてのゴール選択基準で最良のゴールを選択」を行う方法に関しては、4 章で述べた通りである。上述の通り、ゴール選択基準のうち、貢献度と顧客のニーズに関する妥当性は、常に採用されるべきゴール選択基準である。そのため、ゴール選択基準間のコンフリクトのうち、「貢献度-妥当性-コンフリクト」(3.1 節参照) は、ゴール選択上常に考慮する必要があるコンフリクトである。

図 5.4 のうち、「属性値の伝搬を伴うゴールグラフの簡約化」に関しては、3.3 節で述べた通りである。上述の通り、貢献度-妥当性-コンフリクトは、ゴール選択上常に考慮されるべきであるため、AND 分解と OR 分解が共に存在する 2 階層-部分ゴールグラフ (1.7 節 (9) 参照) に対しては、貢献度-妥当性-コンフリクトの検出が行えるように、属性値の伝搬を伴うゴールグラフの簡約化を行う必要がある。

5.2 適用

「ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法」(5.1 節参照) を、図 2.10 に対して適用する。ゴール選択基準としては、「貢献度」(2.3.1 節 (属性-1) 参照) と「顧客のニーズに関する妥当性」(4.3 節参照) の 2 つを採用する。ただし、顧客のニーズに関する妥当性を見積もる指標としては、 $Cup^E(g)$ (4.3.1.1 節 (6) 参照) ではなく、 $Cup(g)$ (式(3.1)) を用いることにする。そのため、「ゴール選択基準間のコンフリクト」(2.4 節参照) は、「貢献度-妥当性-コンフリクト」(3.1 節参照) のみを考慮すればよい。顧客のニーズに関する妥当性は、満足度行列から計算される (3.1.1 節参照) ため、図 2.10 のゴールグラフに対しては、貢献度と満足度の 2 つを付与する必要がある。図 2.10 に対して貢献度と満足度の 2 つの属性を付与したゴールグラフは図 3.7 である。

まずは、図 3.7 に対して、「ゴール選択上のコンフリクト」(2.3.1.2 節参照) を検出し、検出された場合には、なるべく解決する。図 3.7 において、ゴール選択上のコンフリクトの検出対象となるゴールは、11 のみである。そこで、11 が負の貢献度で接続している少なくとも 1 つの、かつ、なるべく多くの親ゴールに対して正の貢献度を持ち、なおかつ、分解元の 11 に対しても正の貢献度を持つような 11 の子ゴールが存在するかどうかを調べる。11 が負の貢献度で接続している親ゴールは 4 のみであるから、4 と 11 に対して正の貢献度を持つような 11 の子ゴールが存在するかどうかを調べればよい。ここでは、そのような子ゴールは発見されなかったものとする。したがって、ゴール選択上のコンフリクト解決後のゴールグラフは、図 3.7 のままとする。

次に、図 3.7 に対して、「ステークホルダ間の認識不一致」(2.3.1.3 節参照) を検出し、必要ならばこれを解決する。図 3.7 中にステークホルダ間の認識不一致を持つゴールは多数存在するため、図

3.7 中のステークホルダ間の認識不一致を持つすべてのゴールに対して、これを解決する。ステークホルダ間の認識不一致が解決されたゴールグラフを図 5.5 に示す。ただし、満足度の修正は筆者が恣意的に行った。

次に、図 5.5 に対して、「形式的ゴール選択」(図 5.3) を行う。提案方法は、2 階層-OR 分解-部分ゴールグラフ (1.7 節 (9b) 参照) だけではなく、2 階層-部分ゴールグラフ (1.7 節 (9) 参照) が検出対象となる。すなわち、「AND 分解と OR 分解が共に存在」するゴールまで含めて貢献度-妥当性-コンフリクト検出を行う。この場合、図 5.5 中で「AND 分解と OR 分解が共に存在」するゴールは 4 (「製造工程の可視化」) のみであるため、このゴールに対して「属性値の伝搬を伴うゴールグラフの簡約化」(3.3 節参照) を行う。属性値の伝搬を伴うゴールグラフの簡約化後の図 5.5 を図 5.6 に示す。ただし、ゴールグラフの簡約化後に新たに現れたゴール「段取改善*マシンの汎用化」および「段取改善*製造付帯作業の削減」に対しては、「ゴールグラフの簡約化における添字の処理」(3.3.1 節参照) にしたがって、添字 8 と 9 を付番している。

ゴールグラフの簡約化後のゴールグラフ (図 5.6) 中で、ゴール選択基準間のコンフリクト (この場合、貢献度-妥当性-コンフリクト) を持つゴールは、3, 4, 7 の 3 つである (図 5.7)。これら 3 つのゴールに対しては、「階層分析法に基づく形式的ゴール選択」を行う。図 5.6 中のすべての 2 階層-部分ゴールグラフにおいて選択される子ゴールの一覧を、表 5.1 に示す。表 5.1 より、初期ゴール (1 (「生産コストの削減」)) から最終ゴールまでのパスは、「1->2->4->8」、「1->3->4->8」、「1->3->7->11」、「1->3->7->13」の 4 通りあることがわかる。よって、1 に対して獲得される要求は「8, 11, 13」となる (4.1 節参照)。

5.3 補遺

5.3.1 不確実性推論の形式的ゴール選択への応用

ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法 (5.1 節参照) は、ゴール選択基準間のコンフリクト (2.4 節参照) に関して、形式的に広義のゴール選択 (1.7 節 (10) 参照) を行うものである。しかし、広義のゴール選択の形式的定義 (式 (4.1)) によれば、広義のゴール選択結果は、狭義のゴール選択結果の総和であり、狭義のゴール選択は全体的 (holistic) な性質を持つシステムであるため、狭義のゴール選択結果を得るには指数時間を要するのであった (4.1.2 節参照)。狭義のゴール選択は、複数の評価基準 (ゴール選択基準) を考慮に入れる必要があるため、評価基準を属性とする多属性意思決定方法を用いることによって、多項式時間で狭義のゴール選択結果を得ることができる。本論文では、多属性意思決定方法の 1 つである「階層分析法」に基づいて、形式的に狭義のゴール選択を行う方法である、「階層分析法に基づく形式的ゴール選択方法」を提案した (4.2 節参照)。しかしながら、この方法は、作成にかかる人手コストが大きい属性に基づいて計算されるゴール選択基準である場合には、実用上の問題がある。たとえば、顧客のニーズに関する妥当性 (3.1.1 節参照) は、満足度行列という、作成にかかる人手コストが大きい属性に基づいて計算されるゴール選択基準である。この問題に対して、人工知能の分野で盛んに研究が行われてきたエキスパートシステムにおける推論エンジンに採用されている、種々の不確実性推論に用いられる確実性測

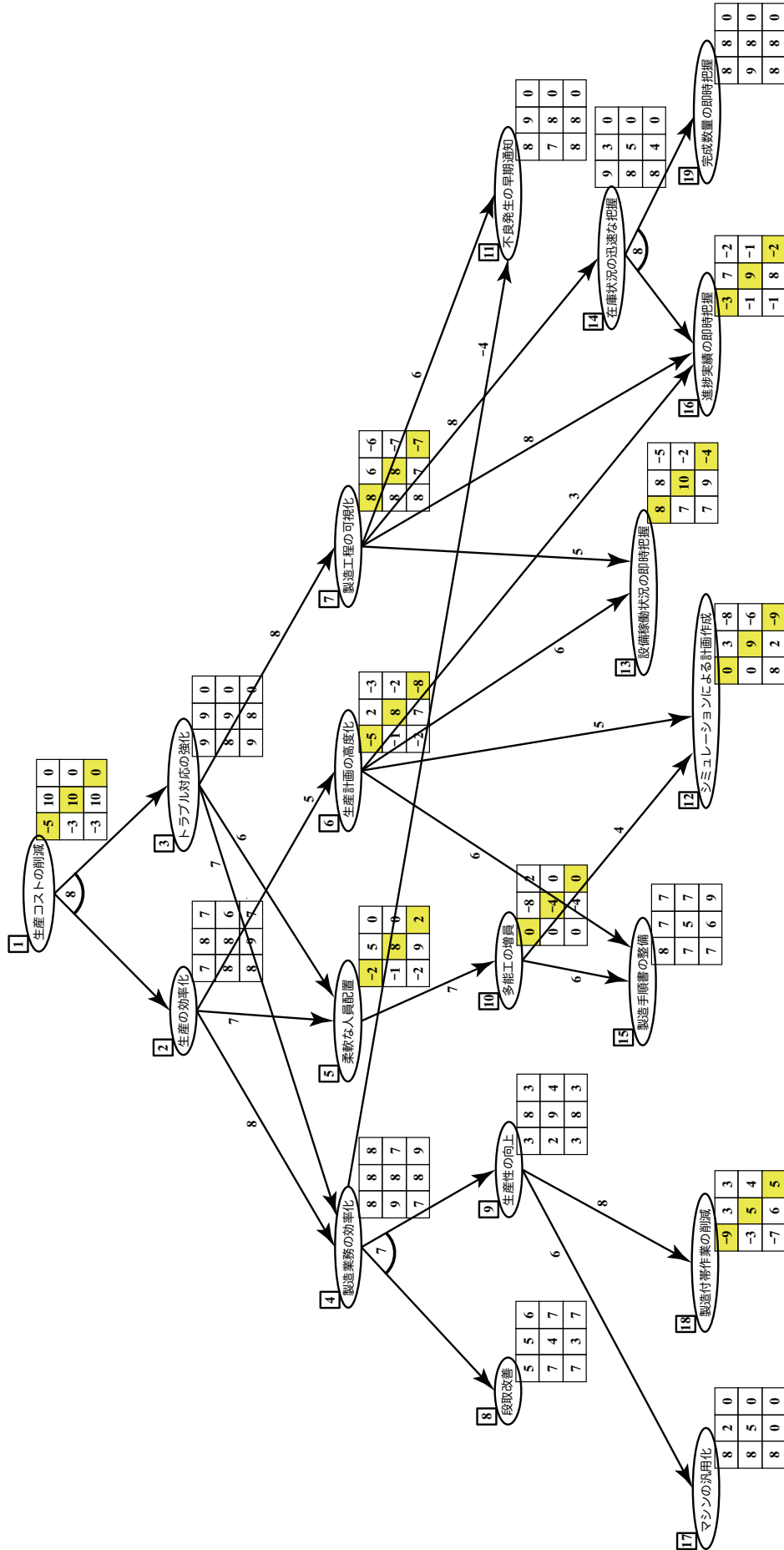


図 5.5: 図 3.7 中のステークホルダ間の認識不一致を持つゴールに対して、これを解決したゴールグラフ

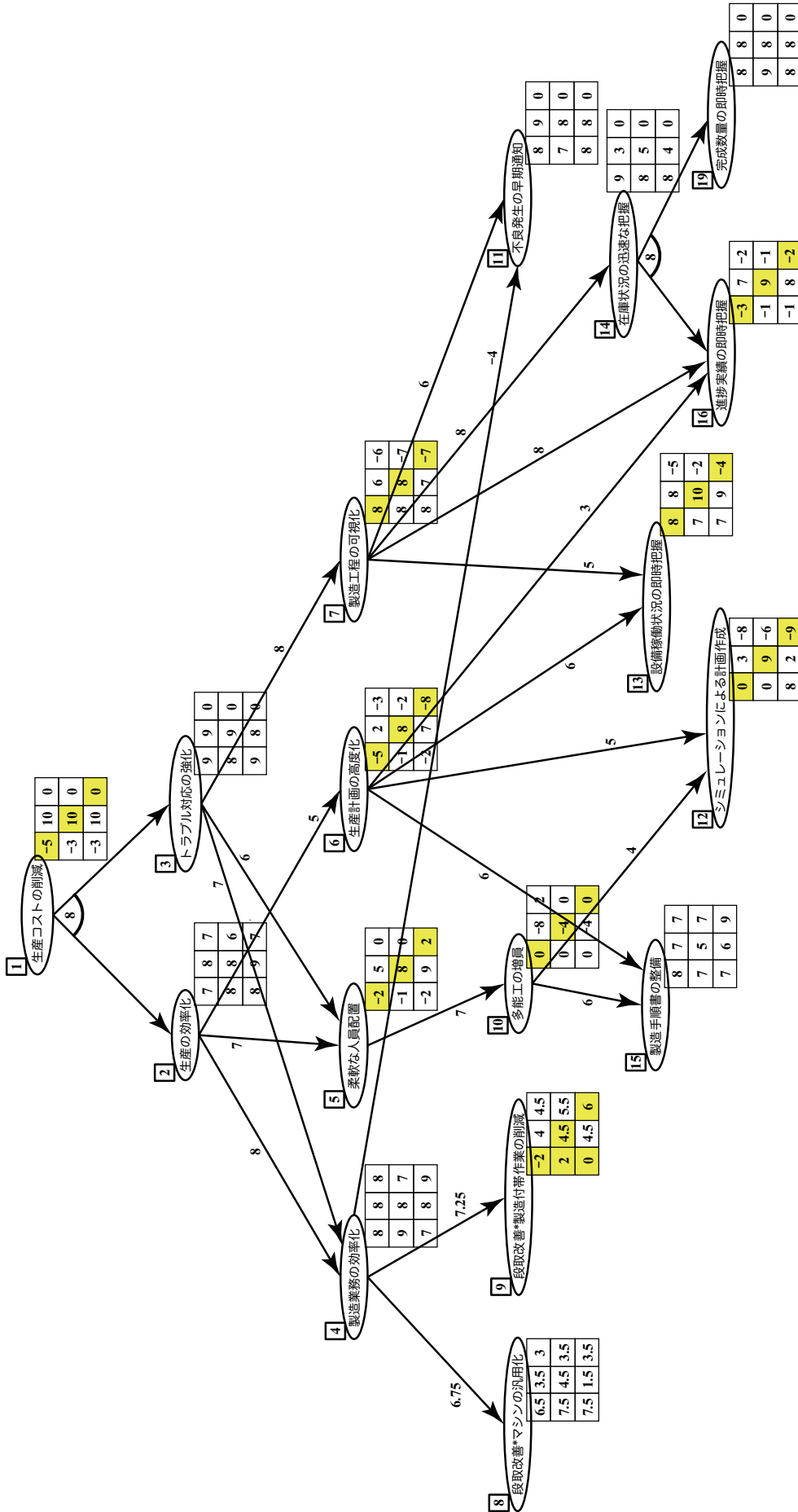


図 5.6: 図 5.5 に対して属性値の伝搬を伴うゴールグラフの簡約化を行った後のゴールグラフ

```
1: >_ccc-dfs.exe
2: 2は検出対象
3:          4と 5は優先順位の相違無し
4:          4と 6は優先順位の相違無し
5:          5と 6は優先順位の相違無し
6:          2はコンフリクト無し
7: 4は検出対象
8:          8と 9は優先順位の相違有り
9:          8と 11は優先順位の相違有り
10:         9と 11は優先順位の相違有り
11:         4はコンフリクト有り
12: 3は検出対象
13:         4と 5は優先順位の相違無し
14:         4と 7は優先順位の相違有り
15:         5と 7は優先順位の相違無し
16:         3はコンフリクト有り
17: 10は検出対象
18:        12と 15は優先順位の相違無し
19:        10はコンフリクト無し
20: 6は検出対象
21:        12と 13は優先順位の相違無し
22:        12と 15は優先順位の相違無し
23:        12と 16は優先順位の相違無し
24:        13と 15は優先順位の相違無し
25:        13と 16は優先順位の相違無し
26:        15と 16は優先順位の相違無し
27:        6はコンフリクト無し
28: 7は検出対象
29:        11と 13は優先順位の相違有り
30:        11と 14は優先順位の相違有り
31:        11と 16は優先順位の相違有り
32:        13と 14は優先順位の相違有り
33:        13と 16は優先順位の相違有り
34:        14と 16は優先順位の相違無し
35:        7はコンフリクト有り
```

図 5.7: 図 5.6 に対する貢献度-妥当性-コンフリクト検出結果

表 5.1: 図 5.6 中のすべての 2 階層-部分ゴールグラフにおいて選択される子ゴールの一覧

T	U	V	W	X	Y ^{注1)}	Z
1	2 3	AND				2, 3
2	4 5 6	OR	無	8 7 5	8 2.8 1.5	4
3	4 5 7	OR	有			4, 7
4	8 9 11	OR	有			8
5	10	OR ⁽¹⁾				10
6	12 13 15 16	OR	無	5 6 6 3	3.7 8.2 6.7 3.2	13, 15
7	11 13 14 16	OR	有			11, 13
10	12 15	OR	無	4 6	3.7 6.7	15
14	16 19	AND				16, 19

T: 親ゴール

U: 子ゴール

V: 2 階層-部分ゴールグラフの種類^{注2)}

W: 貢献度-妥当性-コンフリクトの有無

X: $Cov(g)$ Y: $Cup(g)$

Z: 選択される子ゴール

度を利用することで、貢献度（2.3.1節（属性-1）参照）のように「枝」（2.2節参照）に対して設定されるゴール選択基準と、顧客のニーズに関する妥当性のように、「ゴール」（2.2節参照）に対して設定されるゴール選択基準を各々1つ考慮に入れた上で、形式的に狭義のゴール選択を行うことができる。本論文では、この方法を「確実性測度に基づくゴール選択方法」と呼ぶ。以下、エキスパートシステムにおけるゴール指向とゴール指向要求分析の差異について述べた後、確実性測度の1つである「確信度」に基づくゴール選択方法を提案する。また、具体例に対して適用した結果を示し、提案方法が適切に機能することを確認する。

5.3.1.1 エキスパートシステムにおけるゴール指向とゴール指向要求分析の差異

エキスパートシステム(ES)は、主に専門家(エキスパート)から獲得した知識の集合である「知識ベース(Knowledge Base)」と、推論(reasoning, inference)^{注3)}を行う「推論エンジン(Inference Engine)」から成る[飯島 93](図5.8)。このうち、ゴール指向は、推論エンジンにおける推論に利用されている。これに対して、本論文で研究の対象としているゴール指向要求分析は、ESにおける「知識獲得」部分(図5.8)を対象とする分析方法である。本論文で構築した方法とESにおけるゴール指向の差異を述べる上で、まず「適用対象が異なる」ことは指摘しておく必要がある。

ESの知識獲得は、本論文で扱った要求獲得に敷衍して捉えることができる。つまり、両者を対比させて考えると、ESにおける知識獲得は、「専門家」から知識ベースに格納される知識（「事実」や「ルール」）を抽出することであるのに対し、要求工学における要求獲得は、「顧客」から要求仕様書として書かれるべき要求を抽出する作業である。知識獲得と要求獲得を対比させて考える場合、専門家とは、「開発者」ではなく「顧客」である。一般に、開発者は、情報システム開発の専門家ではあるけれども、顧客が従事している何らかの業務の専門家ではない。したがって、ESという情報システムを開発する場合、開発者が顧客から知識獲得を行うのである。開発者が知識を獲得する側であり、顧客が知識を獲得される側である。要求工学では、顧客が有している業務上の知識を「ドメイン知識」と呼び、マニュアル化されておらず、徒弟奉公的に育まれる暗黙知によって運営されている業務も現実にはしばしば見受けられることから、民族誌学(エスノグラフィ)などの社会科学における方法が要求獲得方法として必要とされているのであった(図2.3)。ESにおいては、知識獲得そのものが目的であるのに対し、要求獲得は、知識獲得を行った上でさらに、得られた知識を要求へと昇華させる作業が必要となる(2.1節参照)。

このように、要求獲得では、専門家(顧客)が持つ知識を獲得することは必須である。それ故、要求獲得は、ESにおける知識獲得を包含する作業であり、ESにおける知識獲得は、要求獲得に敷衍して捉えることができる。したがって、本論文の研究対象であるゴール指向要求分析は、要求獲得方法の1つであるため、ESに対して適用する場合、その適用対象は、知識獲得の部分になる。よって、ゴール指向要求分析をESの設計・構築に取り入れるとすれば、ゴール指向が利用されている推論エンジンの部分ではなく、知識獲得において利用されることになる。

一方、ESの推論エンジンにおいて用いられている推論方法は、ゴール指向要求分析におけるゴール選択に利用できる。ゴール指向要求分析の成果物であるゴールグラフは、AND/OR グラフであり、その個々のノードは「ゴール」と呼ばれ、達成すべき「目標」を表す(2.2節参照)。これに対して、

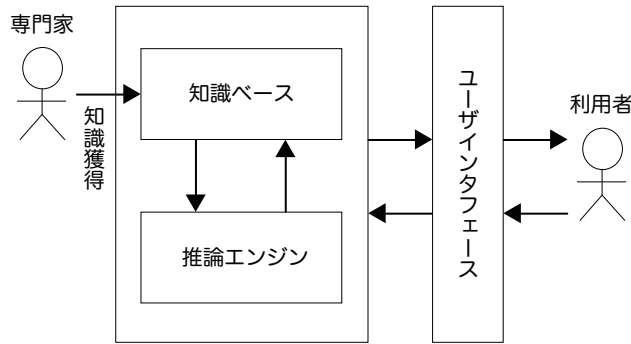


図 5.8: エキスパートシステムの基本構造 [飯島 93]

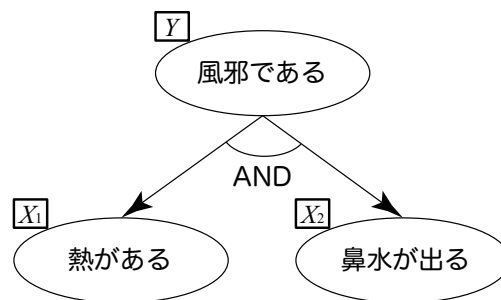


図 5.9: エキスパートシステムの推論に用いられる AND/OR グラフの例 [戸内 97]

ESにおけるAND/ORグラフは、個々のノードが現実のある「事実」を表し、すべての子ノードから結論である親ノードの確からしさ (fuzziness) がどの程度かを推論するために用いられる (ESにおけるAND/ORグラフの例を図5.9に示す。ゴールグラフは、最下位のゴールが最終ゴールであるのに対し、ESにおけるAND/ORグラフは、最上位のゴールが「最終ゴール」(図5.10)である。知識獲得に関心があるか、推論に関心があるかという、AND/ORグラフの用途の差異が顕著に表れている部分である)。この確からしさの推論方法は、「不確実性推論」と呼ばれ、これまでに種々の「確実性測度」に基づく方法が提案されている (たとえば、「確信度」[Shortliffe 75]に基づく方法、「主観確率」に基づく方法[Duda 76]、「Dempster-Shafer (デンプスター・シェーファー) の確率理論」[Dempster 67, Dempster 68, Shafer 76]に基づく方法、「ファジィ理論」[Zadeh 65]に基づく方法がある[飯島 93])。次の5.3.1.2節では、代表的なESであるMYCIN^{マイシン}において採用されている確実性測度である「確信度 (CF: Certainty Factor)」に基づいて、図5.9の親ノード (結論) である「風邪である」という事実の確からしさを推論する手順を示す。

5.3.1.2 確信度に基づくゴール選択方法

本節では、「確信度に基づくゴール選択方法」を提案する。まず最初に、確信度に基づくゴール選択方法の前提となる「確信度の定義」および「確信度の伝搬の定義」について述べる。確信度の定義は次の通りである。

表 5.2: 確信度の値の解釈 [薦田 97]

確信度の値	解釈
-1.0	絶対に……でない (definitely not)
-0.8	ほぼ確実に……でない (almost certainly not)
-0.6	たぶん……でない (probably not)
-0.4	……でないかもしれない (maybe not)
-0.2~0.2	わからない (unknown)
0.4	……であるかもしれない (maybe)
0.6	たぶん……である (probably)
0.8	ほぼ確実に……である (almost certainly)
1.0	絶対に……である (definitely)

確信度の定義 [薦田 97]

確信度は、「事実」や「ルール」の確からしさを定義するための尺度である。ルール $R := X \Rightarrow Y$ において、前提 X が真ならば結論 Y を信じる度合いを「 R の確信度」と呼び、 $CF(R)$ で表す。確信度は -1 以上 1 以下の実数値であり、負の値は否定的な信念を、正の値は肯定的な信念を意味し、その絶対値が大きいほど、より強く信じていることを表す。確信度の値の直感的な解釈は、表 5.2 に示す通りである。たとえば、図 5.9 において、“「熱がある」ならばたぶん「風邪である」” というルールは、次で表される。

$$\begin{aligned}
 R_1: X_1 &\Rightarrow Y \\
 CF(R_1) &= 0.6
 \end{aligned}
 \tag{5.1}$$

また、確信度は、ルールに対してだけでなく、ルールを構成する個々の事実に対しても用いられる。事実 X を信じる度合いを、「 X の確信度」と呼び、 $CF(X)$ で表す。たとえば、“「熱がある」かもしれない” という事実は、次で表される。

$$\begin{aligned}
 X_1: \text{熱がある} \\
 CF(X_1) &= 0.4
 \end{aligned}
 \tag{5.2}$$

なお、何らかの推論プロセスを経て得られた事実 X については、 X の確信度は、 X を導くに至った状況に依存するため、これを $CF(X, S)$ と表記する^{注4)}。ただし、 S は、事実 X を導くに至った根拠の集合である。

確信度を求める関数 CF の定義域は、コネクタ (5.3.1.3 節参照) の集合 C に対応する。

確信度の伝搬の定義は次の通りである。

確信度の伝搬の定義 [薦田 97]

確信度を用いた推論においては、前提となる事実やルールに付けられた確信度の値をもとに、導いた結論に対する確信度の値を求め、これを一連の推論プロセスに沿って順次伝搬することで、最

終的な結論の持つ確信度を決定する。確信度の伝搬の際には、次の3通りの場合の処理が必要になる。(1)ルールが単一の前提を持つ場合、(2)ルールが複数の前提を持つ場合、(3)同一の結論が独立の複数のルールに支持される場合。以下、それぞれについて処理手順を述べる。

(1) ルールが単一の前提を持つ場合

事実 X の確信度を $CF(X)$ 、ルール R の確信度を $CF(R)$ とすると、両者から導かれた結論である事実 Y の確信度 $CF(Y, \{X\})$ は、次式で定義される。

$$CF(Y, \{X\}) := CF(X) \times CF(R) \quad (5.3)$$

たとえば、式 (5.1) および式 (5.2) から、図 5.9 における結論 (親ゴール) 「風邪である」の確信度は、次の通り計算される。

$$\begin{aligned} CF(Y, \{X_1\}) &= CF(X_1) \times CF(R_1) \\ &= 0.4 \times 0.6 \\ &= 0.24 \end{aligned}$$

この結果は、表 5.2 より、「風邪であるかもしれない」という結論が導かれたことを示している。

(2) ルールが前提に複数の事実を持つ場合

分解関係 (AND 分解, OR 分解) の組合せによって、次の3通りの伝搬方法が定義される。

(2a) 前提が論理積 (連言, conjunction) のみから構成されている場合

$$\begin{aligned} R: \bigwedge_{i \in I} X_i \Rightarrow Y \quad (I \text{ は添字集合}) \\ CF(Y, \{X_i\}) := \min_{i \in I} (CF(X_i)) \times CF(R) \end{aligned} \quad (5.4)$$

(2b) 前提が論理和 (宣言, disjunction) のみから構成されている場合

$$\begin{aligned} R: \bigvee_{i \in I} X_i \Rightarrow Y \quad (I \text{ は添字集合}) \\ CF(Y, \{X_i\}) := \max_{i \in I} (CF(X_i)) \times CF(R) \end{aligned} \quad (5.5)$$

(2c) 前提が論理積と論理和の両方から構成されている場合

(2a), (2b) の組合せによって計算する。たとえば、

$$R: (X_1 \vee X_2) \wedge (X_3 \vee X_4) \wedge X_5 \Rightarrow Y$$

というルールに対しては、 Y の確信度は、次の通りに計算される。

$$\begin{aligned} CF(Y, \{X_1, X_2, X_3, X_4, X_5\}) \\ = \min \left(\max (CF(X_1), CF(X_2)), \max (CF(X_3), CF(X_4)), CF(X_5) \right) \times CF(R) \end{aligned}$$

(3) 同一の結論が独立した複数のルールに支持される場合

CF_1 と CF_2 が、異なる 2 つの独立したルールに支持されて導かれたある結論の確信度を表すとする、両者の「合成確信度」 CF_{COMB} は、次で定義される。

$$CF_{\text{COMB}} := \begin{cases} CF_1 + CF_2 \times (1 - CF_1) & \text{if } CF_1 > 0 \wedge CF_2 > 0, \\ \frac{CF_1 + CF_2}{1 - \min(|CF_1|, |CF_2|)} & \text{if } CF_1 \times CF_2 < 0, \\ CF_1 + CF_2 \times (1 + CF_1) & \text{if } CF_1 < 0 \wedge CF_2 < 0 \end{cases} \quad (5.6)$$

式 (5.6) は、「相乗効果」(式 (6.2b)), 「相殺効果」(式 (6.2c)) を考慮に入れた定義となっている。すなわち、同一の結論が独立した複数のルールに支持される場合には、ルールごとに (2a), (2b) の方法で計算した確信度よりも高い値が合成確信度として得られる。逆に、複数のルールが相反する結論を導いた場合には、ルールごとに (2a), (2b) の方法で計算した確信度よりも低い値が合成確信度として得られる。確信度に関する重要な点としては、複数の確信度を合成した場合の「合成確信度」が定義されているということである。そして、合成確信度は個々の確信度が正であるならば「単調性 (monotonicity)」および「優加法性 (superadditivity)」を満たす。つまり、個々の確信度が正であるならば、合成確信度は、合成される前の個々の確信度よりも高いだけでなく、個々の確信度の総和よりも高くなる。3 つ以上のルールから同じ結論が得られた場合には、その中から 2 つを選び、その合成確信度を求め、それに対してさらに次の確信度を合成するという操作を逐次的に繰り返すことで、全体の合成確信度を得られる。正負の確信度が混在する場合には、同符号の確信度の間で合成確信度を求め、最後にその 2 つを合成すればよい。

ところで、図 5.9 における結論 (親ゴール) 「風邪である」という事実に対して、確信度に基づく不確実性推論を行う。図 5.9 には、“「熱がある」 \Rightarrow 「風邪である」”，と“「鼻水が出る」 \Rightarrow 「風邪である」”という 2 つのルールが存在すると捉えるのは間違いである。なぜならば、AND/OR グラフは、全体である 1 つのルールに対応するからである。つまり、図 5.9 は、全体で“「熱がある」 \wedge 「鼻水が出る」 \Rightarrow 「風邪である」”という 1 つのルールに対応しており、これ以外のルールを表現するものではない。同一の結論に対して、複数のルールが存在する場合には、「COMB」(Combination) という分解関係で表現される。AND/OR 分解に COMB 分解が加わった場合の「AND/OR/COMB グラフ」の例を図 5.10 に示す。

AND/OR 関係に対する推論は、式 (5.4), 式 (5.5) より、ある確実性測度上で \min , \max 演算を採用でき、他に適切な方法はない [寺野 87]。したがって、COMB 関係に対する「合理的推論法」を、確実性測度の定義と共に定めることが必要となる。合理的推論法とは、事実が正しいならば、常に正しい結論を導く方法を意味する。式 (5.6) は、確実性測度「確信度」における合理的推論法である。しかし、確信度の欠点としては、「合理性の裏付けがない」ことが挙げられる [石塚 83a, 寺野 87]。上述の Dempster-Shafer の確率理論は、この欠点を克服し、主観にかかわる不確実性を理論的に扱うことを可能にするために提案されたものである [石塚 83a, 寺野 87]。

以上を踏まえると、図 5.9 は、確信度の伝搬の定義の (2a) に該当することがわかる。よって、式 (5.4) より、風邪であることの確信度は、添字集合を $I = \{1, 2\}$ とし、 $CF(R) = 0.8$, $CF(X_1) = 0.4$,

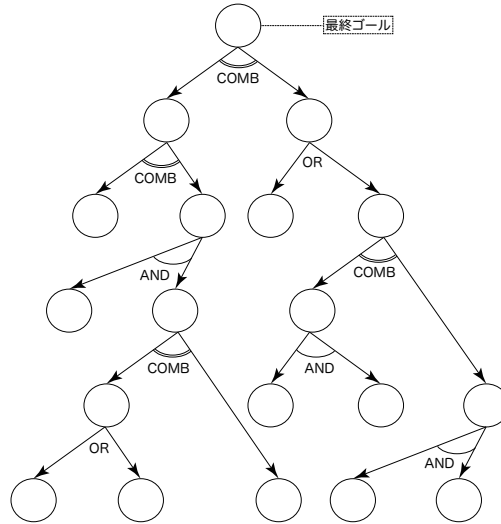


図 5.10: AND/OR/COMB グラフの例 [Ishizuka 82a, Ishizuka 82b, Ishizuka 83, 石塚 83b, 石塚 83c, 石塚 85, 寺野 87]

$CF(X_2) = 0.3$ とすると、次の通りに求まる。

$$\begin{aligned} CF(Y, \{X_1, X_2\}) &= \min_{i \in I} (CF(X_i) \times CF(R)) \\ &= 0.3 \times 0.8 \\ &= 0.24 \end{aligned}$$

この結果は、表 5.2 から、「風邪であるかもしれない」という結論が導かれたことを示している。

5.3.1.3 AND/OR グラフの形式的定義

AND/OR グラフは、ハイパーグラフ (V, C) として形式的に定義できる [Martelli 78, Nilsson 80, 大須賀 91]。ただし、 V はノードの集合、 C はコネクタの集合。コネクタとは、1 個の親ノードと子ノードの集合を結ぶハイパーアークの別名である。コネクタ C は、次の通り定義される。

$$C ::= \bigcup_{k=0}^m V^{k+1} \quad (5.7)$$

ただし、 $k = 1$ のコネクタを「1 コネクタ」、 $k = 2$ のコネクタを「2 コネクタ」のように呼ぶ。各 k コネクタは第 1 成分が親ノードで、それ以外が子ノードを表す、順序付き $(k + 1)$ -組、すなわち、 $(k + 1)$ -列である。定義上、素列 $s \in V^1$ は存在するが、空列 $s \in V^0$ は存在しない。1 コネクタは 1 つの OR アークに対応する。 k コネクタ ($k > 2$) は、1 つのノードからの k 個の AND アークに対応する。図 5.11 では、1 コネクタは、 (n_0, n_2) , (n_0, n_1) , (n_0, n_3) , (n_0, n_4) , (n_1, n_2) , (n_1, n_3) , (n_4, n_3) の 7 つあり、2 コネクタは、 (n_0, n_3, n_4) , (n_1, n_2, n_3) の 2 つある。 $k > 3$ のコネクタは存在しない。

AND/OR グラフの部分グラフ（「部分 AND/OR グラフ」）は、 $V' \subset V$, $C' \subset C$ に対して、 (V', C') で定義される。

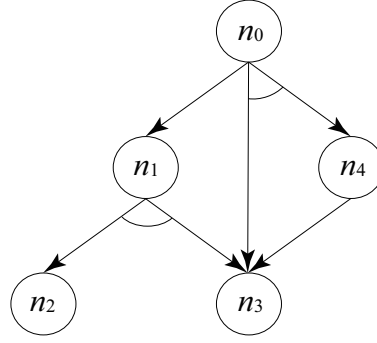


図 5.11: AND/OR グラフの例 [大須賀 91]

ループを除くコネクタ $C^\#$ と、サイクルを除くコネクタ C^\natural は、

$$C^\# := \bigcup_{k=0}^m V^{k+1} \setminus \Delta \quad \text{ただし, } \Delta := \{(v, v) \mid v \in V\} \quad (5.8)$$

$$C^\natural := \bigcup_{k=0}^m V^{k+1} \setminus \left\{ s \in \bigcup_{k=0}^m V^{k+1} \mid H(s) = L(s) \right\} \quad (5.9)$$

で定義される。ただし、 $H(s)$ と $L(s)$ は、列の先頭と末尾を取り出す関数であり、次の通り定義される [田中 08].

$$H : \begin{array}{ccc} \bigcup_{k=0}^m V^{k+1} & \longrightarrow & V \\ \Downarrow & & \Downarrow \end{array} \quad (5.10)$$

$$s = (s_1, s_2, \dots, s_{k+1}) \longmapsto s_1$$

$$L : \begin{array}{ccc} \bigcup_{k=0}^m V^{k+1} & \longrightarrow & V \\ \Downarrow & & \Downarrow \end{array} \quad (5.11)$$

$$s = (s_1, s_2, \dots, s_{k+1}) \longmapsto s_{k+1}$$

ループとサイクルを両方とも除くコネクタの集合 C^\flat は、

$$C^\flat := C^\# \cap C^\natural \quad (5.12)$$

で定義することができる。よって、ループのない AND/OR グラフ、サイクルのない AND/OR グラフ、ループおよびサイクルのない AND/OR グラフは、各々、 $(V, C^\#)$ 、 (V, C^\natural) 、 (V, C^\flat) で表される。

枝の集合 E の定義は、次の通りである。

$$E := \{c \in C \mid |T(c)| = 1\} \quad (5.13)$$

ただし, $T(c)$ は列の尾部を取り出す関数であり, 次の通り定義される [田中 08].

$$\begin{array}{ccc} T: & V^{k+1} & \longrightarrow & V^k \\ & \Downarrow & & \Downarrow \\ & s = (s_1, s_2, \dots, s_{k+1}) & \longmapsto & (s_2, \dots, s_{k+1}) \end{array} \quad (5.14)$$

5.3.1.4 ゴールグラフの形式的定義

AND/OR グラフ (V, C) とラベリング関数 L の対 (V, C, L) として, ゴールグラフを形式的に定義することができる. ただし, V はノードの集合, C はコネクタの集合, L はラベリング関数. ラベリング関数 L は, 次の通り定義される [田中 08].

$$\begin{array}{ccc} L: & V & \longrightarrow & \mathcal{P}(PV) \\ & \Downarrow & & \Downarrow \\ & v & \longmapsto & pv \end{array} \quad (5.15)$$

たとえば, 図 5.9 は, $PV = \{ \text{風邪である}, \text{熱がある}, \text{鼻水が出る} \}$ であり, $L(Y) = \{ \text{風邪である} \}$, $L(X_1) = \{ \text{熱がある} \}$, $L(X_2) = \{ \text{鼻水が出る} \}$ である. PV の命題としては, もちろん, 論理関係で結ばれた, 互いに素な複数の内容を記述することができる. たとえば, $pv = \{ \text{風邪である} \wedge \text{熱がある} \}$ は, 命題変数である. 多くのゴール指向要求分析では, ゴール分解を行う際には, 単一の内容を持つゴールのみを作成する. そのため pv は一点集合である. 複数の内容を持つゴール (pv が複数の原子命題から構成されているゴール) が現れるのは, 「ゴールグラフの簡約化」(2.3.2.3 節参照) を行った後に現れる特別なゴールだけである. また, pv が空集合の場合 ($pv = \emptyset$) は, ダミーゴールに対応する.

ゴールは, 達成されたかどうかを明確に判断できる「ハードゴール」と, 達成されたかどうかを明確に判断できない「ソフトゴール」に分割される (2.2 節参照). ハードゴールであれソフトゴールであれ, 達成されたか否かが判断可能な内容を持つため, 「ゴールの内容」は達成されたか否かが判断可能な命題である. そのため, 命題変数の集合を PV とすると, ゴールの内容を命題変数 $pv \in PV$ に対応させることができるのである. ここで, ハードゴールは, 達成されたかどうかを明確に判断できるため, 対応する命題変数は $pv \in \{0, 1\}$ である. しかし, ソフトゴールは, 達成されたかどうかを明確に判断できないため, 対応する命題変数は, $pv \in [0, 1]$ である必要がある. つまり, 命題変数 pv は, 古典論理ではなく, ファジィ論理である点に注意する必要がある.

ゴールの集合を G とすると, $G := (V, L)$ と定義すれば, 個々のゴール $g \in G$ は, $g = (v, L|_{\{v\}})$ で表される. よって, ゴールグラフ (V, C, L) は, (G, C) で表すこともできる. 前者の記法では, 部分ゴールグラフは, $V' \subset V$, $C' \subset C$ に対して, $(V', C', L|_{V'})$ と定義される. 後者の記法では, 部分ゴールグラフは, $G' \subset G$, $C' \subset C$ に対して, (G', C') と定義される. 2 階層部分ゴールグラフ (1.7 節 (9) 参照) は, 第一成分が一点集合の部分ゴールグラフに対応する.

ループのないゴールグラフ, サイクルのないゴールグラフ, ループおよびサイクルのないゴールグラフは, 式 (5.8), 式 (5.9), 式 (5.12) より, 各々, $(V, C^\#, L)$, (V, C^\natural, L) , (V, C^\flat, L) で表さ

れる。

5.3.1.5 貢献度の形式的定義

貢献度 (2.3.1 節 (属性-1) 参照) は, -10 から 10 までの整数値であり, ゴールグラフ上の「枝」 (2.2 節参照) に対して設定されるゴール選択基準の 1 つである。しかし, 貢献度は, 厳密には, 枝に対して設定されるのではなく, 「コネクタ」に対して設定される。よって, 貢献度の形式的定義は, 厳密には, 次の通りとなる。

$$\begin{array}{ccc} \text{Cov} : C & \longrightarrow & [-10, 10] \\ \cup & & \cup \\ c & \longmapsto & \text{Cov}(c) \end{array} \quad (5.16)$$

ここで重要なことは, 「ゴールの内容」, すなわち, 2階層部分ゴールグラフ (1.7 節 (9) 参照) の子ゴールの命題変数 (5.3.1.4 節参照) の間には「相乗効果」 (式 (6.2b)) や「相殺効果」 (式 (6.2c)) があるため, 枝の集合であるコネクタごとに貢献度を定める必要があるという点である。したがって, コネクタに対して設定された貢献度を, 相乗効果や相殺効果を考慮に入れた上で, 個々の枝に対してどのように配分するかが問題になる。この配分問題には, コネクタに対して貢献度を定める集合関数が必要となる。この集合関数の具体例としては, 「確信度」 (5.3.1.2 節参照) が挙げられる。今後, 確信度以外の種々の確実性測度や, 協力ゲーム理論, ファジィ理論などに基づいて, コネクタに設定される貢献度を, 個々の枝に対して適切に配分する方法を構築する必要がある。

5.3.1.6 確信度に基づく形式的ゴール選択方法

エキスパートシステムの推論エンジンにおいて採用されている不確実性推論をゴール選択に応用可能であることを示すために, 5.3.1.2 節で述べた不確実性推論方法の 1 つである「確信度」に基づいてゴール選択を行う具体例を示す。

確信度に基づくゴール選択の前提条件

- (前提-1) 確信度を与える関数 CF の定義域は, コネクタの集合 C に対応する。
- (前提-2) 複数の子ゴールが OR 分解で接続している場合はすべて COMB 分解 (5.3.1.2 節参照) であるものとする。したがって, 確信度の伝搬 (5.3.1.2 節参照) の計算に関して, (2) と (3) は一対の処理であり, いずれか一方のみが行われることはない。(3) \rightarrow (2) の順で計算を行うものとする。

図 2.4 のすべての最終ゴール (2.2 節参照) に対して満足度行列を設定したゴールグラフ (図 5.12^{注5)}) に対して確信度に基づく形式的ゴール選択を行う。ゴール「よい Web アカウント登録システム」を G_1 , 「アカウント登録が簡単」を G_2 , 「他人が勝手に登録しない」を G_3 , 「誰でもすぐに登録可能」を G_4 , 「パスワード認証」を G_5 , 「SSH による認証」を G_6 と表すことにする。

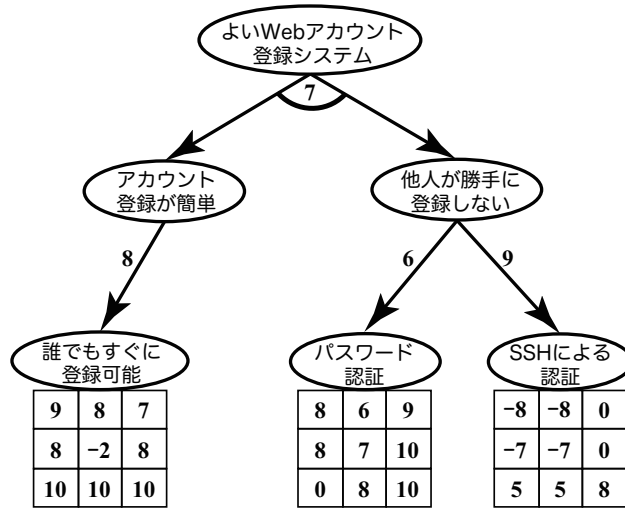


図 5.12: ゴールグラフ [Hayashi 12]

(1) $(\{G_2\}, \{G_4\})$ の 2 階層部分ゴールグラフ

$$CF(G_4) := \frac{Cup(G_4)}{10} \simeq 0.72 \qquad CF(R_2) := \frac{Cov(R_2)}{10} \simeq 0.8$$

確信度の伝搬の定義の (1) に該当するので,

$$CF(G_2, \{G_4\}) = CF(G_4) \times CF(R_2) \simeq 0.58$$

(2) $(\{G_3\}, \{G_5, G_6\})$ の 2 階層部分ゴールグラフ

$$CF(R_4) := \frac{Cov(R_3)}{10} = 0.6 \qquad CF(R_5) := \frac{Cov(R_4)}{10} = 0.9$$

確信度の伝搬の定義の (2c) に該当し, かつ, $CF(R_4) > 0 \wedge CF(R_5) > 0$ なので,

$$R_{(3,4)} := CF_{\text{COMB}}(R_3, R_4) = CF(R_3) + CF(R_4) \times (1 - CF(R_3)) = 0.96$$

確信度の伝搬の定義の (2b) に該当するので,

$$\begin{aligned} CF(G_3, \{G_5, G_6\}) &= \max(CF(G_5), CF(G_6)) \times CF(R_{(3,4)}) \\ &= \max\left(\frac{Cup(G_5)}{10}, \frac{Cup(G_6)}{10}\right) \times CF(R_{(3,4)}) \\ &\simeq 0.75 \end{aligned}$$

(3) $(\{G_1\}, \{G_2, G_3\})$ の 2 階層部分ゴールグラフ

確信度の伝搬の定義の (2a) に該当するので,

$$CF(G_1, \{G_2, G_3\}) = \min(CF(G_2), CF(G_3)) \times CF(R_1)$$

$$= (0.58, 0.75) \times \frac{\text{Cov}(R_1)}{10}$$

$$\simeq 0.41$$

以上、(1)~(3)より、確信度をゴール選択基準値に設定して、ゴール選択アルゴリズム *gsl-dfs* (図4.1) を適用すると、「 $G_1 \rightarrow G_2 \rightarrow G_4$ 」, 「 $G_1 \rightarrow G_3 \rightarrow G_5$ 」という結果が得られる。

確信度をゴール選択に用いる利点は、次の通りである。

確信度をゴール選択に用いる利点

- (1) 子ゴールの集合に対して、ゴールの内容に関する相乗効果および相殺効果を考慮に入れた上でゴール選択基準値を与える集合関数の具体例の1つである。
- (2) ゴール選択アルゴリズム *gsl-dfs* の「ゴール選択基準値を定める関数」(図4.2, 行9)としてCFを設定可能である。そのため、確信度は、*gsl-dfs* に自然に適合する。
- (3) 最終ゴールのみに満足度行列を振ればよい。満足度行列の作成には、多様なステークホルダに対してある程度のまとまった時間が必要となる。そのため、満足度行列は、作成上の人手コストが高い。たとえ実用的なゴール数 ($10^2 \sim 10^3$ [van Lamsweerde 04]) であっても、ゴールグラフ上のすべてのゴールに対して満足度行列を作成することは、プロジェクト全体の中で要求獲得にかかるコストの費用対効果に見合わなかったり、ステークホルダのスケジュール調整上困難な場合も考えられる。確信度に基づくゴール選択は、この問題を解決する実用上重要な方法である。
- (4) コネクタに対して設定されるゴール選択基準 H とゴールに対して設定されるゴール選択基準 G の対 (H, G) に対して、狭義のゴール選択 (1.7節(10)参照) が可能である。したがって、この対 (H, G) が、ゴール選択基準間のコンフリクト (2.4節参照) となる場合に、ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法として利用可能である^{注6)}。ただし、ゴール同士、枝同士の対に対しては、確信度に基づくゴール選択は、形式的ゴール選択方法としては利用できないという制約がある点に留意する必要がある。

5.3.2 閉路があるゴールグラフを仮定した場合の扱い

ゴールグラフは、定義上、無閉路有向グラフ (DAG: Directed Acyclic Graph) であるため、ゴールグラフ中に閉路 (ループ, サイクル) は存在し得ない (2.2節参照)。しかし、仮に閉路が存在する場合の扱いを考えた場合には、少なくとも「モデル検査 (Model Checking)」[Clarke, Jr. 99, Baier 08] を行うことによって、ゴールグラフ上の閉路の有無を判定することや、初期ゴールからある閉路に至るまでの経路を知ることができる。モデル検査は、「形式手法 (Formal Methods)」[Jackson 06, 高原 07, 中島 12, Gnesi 13] と呼ばれる、形式言語で作成された仕様がある性質を満たすかどうかを検証する方法の1つであり、昨今、ソフトウェア工学分野において精力的に研究が行われている [熊澤 11, 土屋 12]。形式手法は、情報システム開発の上流工程において誤りのな

い仕様を作成するための方法である。具体的には、要求獲得後の要求仕様記述で作成される要求仕様書や、要求仕様書に基づいて作成される詳細設計の仕様書に対して用いられる方法であり、要求獲得における応用は未だ盛んに行われていない。今後、閉路を伴うゴールグラフを考える必要がある問題が明らかになった場合は、閉路を伴うゴールグラフに対するモデル検査は、要求獲得に形式手法を応用する上で重要な研究対象になることは間違いない。以下、閉路があるゴールグラフに対するモデル検査の概要について述べる。

5.3.2.1 閉路があるゴールグラフに対するモデル検査

ゴールグラフは、 $M := (V, C, L)$ の3つ組で定義される (5.3.1.4 節参照)。しかし、個々の経路を検証する上で、分解関係は考慮する必要がないため、遷移関係はコネクタの集合 C ではなく、枝 E で十分である。すなわち、分解関係を考慮しないゴールグラフ (V, E, L) の Kripke 構造を考えればよい。

Kripke 構造では、 E は状態間の「遷移関係」を表し、任意の状態 $v \in V$ に対して、 vEv' となるような $e' \in E$ が存在するとき、 v から v' への遷移が存在すると考える。すなわち、ある時刻の状態が v のとき、次の時刻では、状態が v' となり得る。一般に二項関係 R において、常に $\forall s \exists s' sRs'$ が成立するとき、 R を「全域的 (total)」であるという [田中 08]。以降では、分解関係を考慮しないゴールグラフ $M = (V, E, L)$ における二項関係 E は全域的なものに限ることにする。

閉路があるゴールグラフの例を図 5.13 に示す。 $V = \{n_0, n_1, n_2, n_3, n_4\}$ であり、命題変数の集合は、 $PV = \{a, b, c, d, e\}$ であるとする。また、ラベリング関数により、 $L(n_0) = a, L(n_1) = b, L(n_2) = c, L(n_3) = d, L(n_4) = e$ として、各々のノードに命題 (ゴールの内容) が割り当てられるものとする。ここで、Kripke 構造の状態 v_0 から始まる無限長の有向歩道 (2.3.1.1 節参照) を

$$\sigma := (v_0, e_1, v_1, e_2, \dots,) \quad \text{ただし, } (\forall i)(e_i \in E), i \in [1, \infty) \quad (5.17)$$

とする。ただし、「 \dots 」は、無限個の1つ1つ並べることのできる要素の略記である。ここで、 $\sigma(j)$ ($j = i - 1$) は、 σ における j 番目の状態を表すとする。一般に、Kripke 構造 (分解関係を考慮しないゴールグラフ) M において、状態 i が時相論理式 p を満たすことは、

$$M, i \models p \quad (5.18)$$

で表される。モデル検査とは、この関係 (伴意関係) が成立するかどうかを判定することである。たとえば、現状態からのすべての経路 (有向歩道) において、常に p が成り立つことは、計算木論理 (CTL: Computational Tree Logic) を用いて、 AGp で表される。したがって、

$$M, i \models AGp \quad (5.19)$$

という伴意関係が、図 5.13 に対して成立するかどうかを判定 (モデル検査) する。たとえば、 $p = d$ かつ、現状態が $i = n_0$ (初期ゴール) であるとする。式 (5.19) は偽になる。反例は、次の3つの経路である。「 $n_0 \rightarrow n_4 \rightarrow n_4$ (ループ)」、 $\underline{n_0 \rightarrow n_3 \rightarrow n_0}$ (サイクル)」、 $\underline{n_0, n_1, n_2, n_2}$ (ループ)」。モ

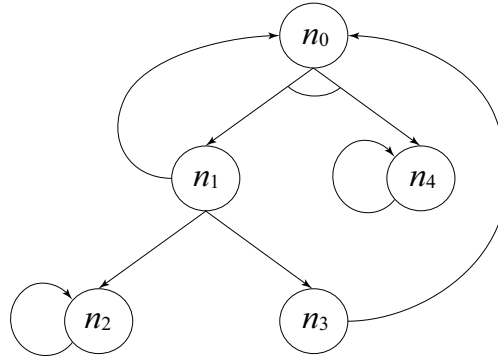


図 5.13: 閉路があるゴールグラフの例

デル検査器を用いることによって、これらの反例を自動的に得ることができる。

逆に、初期ゴール i から最終ゴールのうちの 1 つへ至る経路が存在するかどうかは、最終ゴールの内容が p であるとするとき、

$$M, i \models AGp \tag{5.20}$$

の真偽を判定すればよい。この伴意関係が偽ならば、反例として、初期ゴールから最終ゴールへ至るすべての経路を得ることができる。

5.3.2.2 閉路があるゴールグラフに対する *ccc-dfs* および *gsl-dfs* の構築

ゴールグラフは AND/OR グラフであり、AND/OR グラフは、元々は閉路を含むグラフとして提案された [Nilsson 71, Nilsson 80]。閉路がある AND/OR グラフに対する探索アルゴリズムは、これまでに種々提案されており [Chang 71, Nilsson 71, Pearl 84]、閉路がある AND/OR グラフに対する既存の探索アルゴリズムに基づいて、閉路があるゴールグラフに対する *ccc-dfs* (図 3.3) および *gsl-dfs* (図 4.1) を構築することができるものと考えられる。

5.3.3 数値計算やシミュレーションによる ゴール選択の傾向や一般的性質の導出可能性

5.3.3.1 ゴールグラフの深さに関連する指標の形式的定義

コンフリクト解決のためのグラフモデル (GMCR: Graph Model for Conflict Resolution) [Kilgour 87, Fang 89, Fang 93] において定義されている種々の指標に基づいて、ゴールグラフの階層の形式的定義を与える。

$g < g'$ の定義

$$g < g' :\Leftrightarrow g \neq g' \wedge g \in \bigcup_{i \in I} \left(\bigcup_{(g, g')\text{-道} \in \text{Path}_{(i, g')}} \text{Path}_{(i, g')} \right) \tag{5.21}$$

ただし、 G はゴールグラフ上のすべてのゴールの集合であり、 g および g' は、 G の要素である。 I は、初期ゴール (2.2 節参照) の集合であり、 G の部分集合である。 (g, g') -道は、ゴール g からゴール g' までの有向道 (2.3.1.1 節参照)。 $Path_{(g, g')}$ は、 g から g' へ到達可能なすべての (g, g') -道を要素とする集合 (2.3.1.1 節参照)。

$LL(g)$ の定義

$LL(g)$ を「ゴール g の深さ」と呼び、次の通り定義する。

$$LL(g) := |\{g' \mid g' < g\} \setminus I| + 1 \quad (5.22)$$

明らかに、 $g < g' \Rightarrow LL(g) < LL(g')$ が成り立つ。

$LL(G)$ の定義

$LL(G)$ を「ゴールグラフ G の深さ」と呼び、次の通り定義する。

$$LL(G) := \max \{ LL(g) \mid g \in G \} \quad (5.23)$$

St_k の定義

St_k ($k \in \{k \mid k \in \mathbb{Z}_{\geq 0}, 0 \leq k \leq LL(G)\}$, $\mathbb{Z}_{\geq 0}$ は 0 以上の自然数全体の集合) を「深さ k の階層」と呼び、次の通り定義する。

$$St_k := \{g \in G \mid LL(g) = k\} \quad (5.24)$$

明らかに、ゴールの集合 G は、 St_k を用いて、 $LL(G) + 1$ つの部分ゴールグラフ (1.7 節 (9) 参照) に分割できる。

5.3.3.2 ゴールグラフの深さと増加率

ゴールグラフ (G, C) (5.3.1.4 節参照) の数 N は、「ゴール数 ($|G|$)」と「 G の深さ ($LL(G)$)」によって定まる。たとえば、ゴール数が 3 の場合に考えられるすべてのゴールグラフは、図 5.14 に示す 12 通り存在する。ゴール数が 3 のゴールグラフに対して、ゴール指向要求分析におけるなんらかの一般的性質を検証するためには、これら 12 通りのゴールグラフの各々に対して、その性質が成立するか否かを調べることが必要になる。このように、ゴール数が少ない場合は、すべてのゴールグラフを容易に列挙することができるため、一般的性質を現実的な時間で検証することが可能である。しかし、ゴール数の増加に伴い、列挙するゴールグラフ数が膨大となるため、人手による検証は困難になる。実用規模のゴールグラフのゴール数 ($10^2 \sim 10^3$ [van Lamsweerde 04]) であっても、人手による検証は困難であるから、計算機シミュレーションは必須である。ただし、実用規模のゴールグラフのゴール数の範囲内であっても、 N が状態爆発してしまう場合には、計算機シミュレーションによって、ゴールグラフの一般的性質を検証することができない。そのため、 $(|G|, LL(G))$ に対する

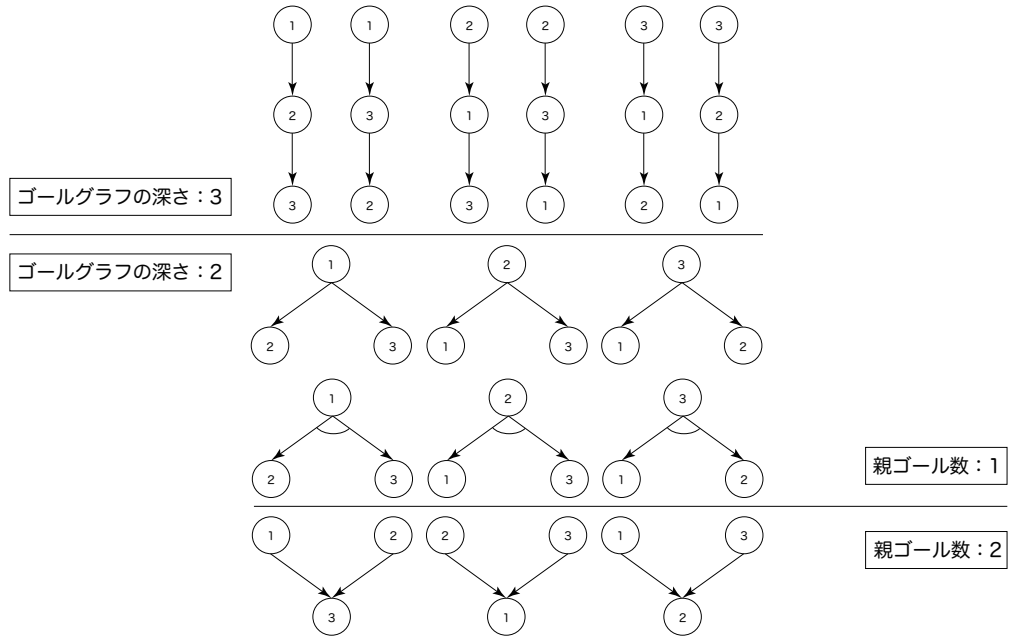


図 5.14: ゴール数が3のゴールグラフ

N の増加率についての知見を得る必要がある。そこでまずは、 $(|G|, LL(G))$ に対する N の増加率それ自体を計算機シミュレーションで明らかにすることが重要である。

ゴール指向要求分析における一般的性質の具体例として、本論文の研究対象である「ゴール選択」を取りあげる。ゴール選択に必要となるゴールグラフ数を $S (C N)$ とすると、 $(|G|, LL(G))$ に対する S の増加率は、 N の増加率よりも少ないものと考えられる。なぜならば、2階層-AND分解-部分ゴールグラフ (1.7節(9a)参照) および、ダミーゴール (1.7節(8)参照) を除いて子ゴールが1つだけの2階層-OR分解-部分ゴールグラフ (1.7節(9b)参照) に関しては、あらかじめ選択される子ゴールが決定されているため、検証する対象から外することができる。たとえば、図5.14の12通りのゴールグラフのうち、狭義のゴール選択について検討する必要があるのは、たかだか3通りの2階層-OR分解-部分ゴールグラフのみである。したがって、ゴール数が3の場合の広義のゴール選択 (1.7節(10)参照) に関して、ゴール選択の一般的性質を検証することは、狭義のゴール選択結果を得る演算を3回行うだけで可能である。しかし、ゴール数が4以上になると、2階層-AND/OR分解-部分ゴールグラフ (1.7節(9c)参照) と、ダミーゴールを除いて子ゴールが1つだけの2階層-OR分解-部分ゴールグラフが存在する。2階層-AND/OR分解-部分ゴールグラフは、「ゴールグラフの簡約化」(2.3.2.3節参照) を用いて、2階層-OR分解-部分ゴールグラフに変型可能であるため、やはりゴール選択の対象になる。したがって、ゴール数が4以上になると、2階層-AND分解-部分ゴールグラフを除くすべての2階層-部分ゴールグラフを考慮に入れる必要がある。以上のことから、一般に S の増加率が N の増加率に対して少ないことは明らかであるが、それほど差がない可能性もある。それほど差がない場合には、 S の増加率をなるべく低減するアルゴリズムが必要になる。

ゴール選択を行うゴールグラフを限定するために、「ゴール数が4未満ならば2階層-OR分解-部分ゴールグラフのみ」、「ゴール選択の対象となるゴールグラフは、 $LL(G) \geq 2$ のみ」、「 $|G| = LL(G)$ ならば、分解関係はOR分解のみ」といった性質もまた、ゴール選択を行うゴールグラフを限定する

上で重要になる。また、検証するゴールグラフを限定すること以外にも増加率を減らすための重要な性質がある。たとえば、「深さ k までのすべての深さのゴールグラフ数 = 深さ $k-1$ までのすべての深さのゴールグラフ数 + 深さ k のゴールグラフ数」という性質は、重複する計算をなるべく省略する「動的計画法 (dynamic programming)」[Aho 83]の観点から、特に重要になる。以上の知見や方法を用いて、 N の増加率に対する S の増加率をなるべく低減することが重要である。

5.4 本章のまとめ

5.1 節では、「ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法」を提案した。これは、2章~4章の内容を組み合わせることで構築された。具体的な手順は、フローチャートで示した。

5.2 節では、既存の例題に対してゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法を適用した結果について述べた。適用の結果、適切に機能することが確かめられた。

5.3 節では、本論文の補遺について述べた。5.3.1 節では、不確実性推論において用いられる確実性測度に基づいてゴール選択することの有用性について述べた。また、確実性測度の1つである確信度に基づくゴール選択方法を構築し、具体例に対して適用することで、有用性を確認した。5.3.2 節では、閉路があるゴールグラフを仮定した場合にモデル検査を行うことの展望について述べた。5.3.3 節では、数値計算やシミュレーションによる、ゴール選択の傾向や一般的性質の導出の可能性について述べ、ゴールグラフ数の増加率に関する知見をシミュレーションによって得ることが重要であることを論じた。

後注

注1) 小数点第3位を四捨五入した値を示している。

注2) 「AND」は、2階層-AND分解-部分ゴールグラフ (1.7節 (9a) 参照)。「OR」は、2階層-OR分解-部分ゴールグラフ (1.7節 (9b) 参照)を表す。「OR⁽¹⁾」は、子ゴールが1つのみの2階層-OR分解-部分ゴールグラフを表す。

注3) (知識ベースに格納されている)「事実」や「ルール」を用いて結論を導くプロセス [薦田 97]。

注4) X が状況依存の場合においても、 X を導くに至った状況が自明である場合には、便宜的に S を省略して $CF(X)$ と書くこともある [薦田 97]。

注5) [Hayashi 12] では、満足度行列は振られていない。満足度行列の個々の満足度は、著者が恣意的に振ったものである。

注6) ゴール選択基準は、設定の対象が2通りある。すなわち、(1) ゴール、(2) 枝またはコネクタのいずれかである。

6.0 本章の概要

本章では、本論文を総括する。6.1節では、本論文で得られた結果の要約を述べる。6.2節では、今後の課題について述べる。

6.1 結果の要約

本論文は、ソフトウェア工学の一分野である要求工学において盛んに研究が行われている「ゴール指向要求分析」に基づき、形式的にゴール選択（1.7節（10）参照）するための方法を構築した結果をまとめたものである。

2章では、3, 4, 5章で述べる方法の前提となる既存研究について述べた。2.1節では、情報システム開発における要求獲得の重要性について述べた。また、既存の要求獲得方法について概説し、ゴール指向要求分析はその中の1つに位置づけられることを述べた。他の要求獲得方法に対するゴール指向要求分析の特徴についても言及した。2.2節では、ゴール指向要求分析の概要について述べた。ゴール指向要求分析の黎明期から現在に至るまでに登場した様々なゴール指向要求分析法の特徴を、登場した年代当時のソフトウェア工学における問題意識や社会情勢と関連させて述べ、その発展経緯について概説した。2.3節では、既存のゴール指向要求分析法のうち、AGORA [Kaiya 02] と属性値に基づくゴール選択手法 [斎藤 06, 山本 07] について述べた。これらは共に、ゴールに属性を設定することによって、ゴール指向要求分析に基づく要求獲得を支援する。

3章では、ゴール選択基準間のコンフリクトを検出する方法について述べた。3.1節では、2つのゴール選択基準「貢献度」（2.3.1節（属性-1）参照） [Kaiya 02] と「顧客のニーズに関する妥当性」（3.1.1節参照） [佐藤 11] の間で「ゴール選択基準間のコンフリクト」（2.4節参照）が生じる場合があることを述べた。また、2階層-OR分解-部分ゴールグラフ（1.7節（9b）参照）においてゴール選択基準間のコンフリクトが存在する場合、形式的に「狭義のゴール選択」（1.7節（10）参照）ができない。そのため、このコンフリクトを持つゴールを検出することが重要であることを述べた。3.2節では、任意のゴールグラフに対して、ゴール選択基準間のコンフリクトの1つである「貢献度-妥当性-コンフリクト」（3.1節参照）を持つすべてのゴールを検出するアルゴリズム *ccc-dfs*（図3.3）につい

て述べた。ccc-dfs は、深さ優先探索 [Aho 83] に基づいて構築された。3.3 節では、ゴール g が貢献度-妥当性-コンフリクトを持つか否かを検出する指標 $Ccc(g)$ (式 (3.3)) が、その定義上、AND 分解と OR 分解を共に持つゴールに対してはこのコンフリクトを検出できないことから、これを検出可能にする方法を提案した。この方法は、「属性値の伝搬を伴うゴールグラフの簡約化」[佐藤 12c] と呼ばれる (3.3.1 節参照)。属性値の伝搬を伴うゴールグラフの簡約化は、「ゴールグラフの簡約化」(2.3.2.3 節参照) を行う際に、「貢献度の処理」(3.3.1 節 (1) 参照) と「満足度の処理」(3.3.1 節 (2) 参照) を追加したものである。

4 章では、ゴール指向要求分析における形式的ゴール選択方法について述べた。4.1 節では、ゴール選択アルゴリズム $gsl-dfs$ について述べた。 $gsl-dfs$ は、深さ優先探索に基づいて構築された。既存のゴールグラフに対して、ゴール選択基準として「貢献度」(2.3.1 節 (属性-1) 参照) を採用した場合について $gsl-dfs$ の評価を行い、 $gsl-dfs$ が正しく機能することを確認した。4.2 節では、階層分析法 [Saaty 80] に基づく形式的ゴール選択方法を提案した。この方法は、ゴール選択基準間のコンフリクト (2.4 節参照) が存在する場合に必要となる。4.1.1 節の評価結果から、 $gsl-dfs$ は、単一のゴール選択基準で狭義のゴール選択を行う場合は、任意のゴールグラフに対して機能することが確かめられた。しかし、複数のゴール選択基準で狭義のゴール選択を行う場合は、ゴール選択基準間のコンフリクトを考慮に入れる必要がある。提案方法を既存の 2 階層-OR 分解-部分ゴールグラフに対して適用した結果、提案方法は狭義のゴール選択を行う方法として正しく機能することが確かめられた。4.3 節では、ゴール選択基準の拡張について述べた。4.2 節で提案した、階層分析法に基づいて狭義のゴール選択を行う方法は、ゴール選択基準の妥当性 (1.7 節 (7) 参照) に依存して狭義のゴール選択結果が変化するため、ゴール選択基準そのものの品質が保証されていることが重要なためである。4.3.1 節では、ゴール選択基準の 1 つである、ゴール g の「顧客のニーズに関する妥当性」を見積もる指標 $Cup(g)$ (式 (3.1)) を拡張する方法について述べた。

5 章では、2, 3, 4 章で述べた内容を組み合わせることで、ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法を提案した。5.1 節では、「ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法」を提案した。これは、2 章~4 章の内容を組み合わせることで、ゴール指向要求分析において、ゴール選択基準間のコンフリクトを考慮に入れた「広義のゴール選択」(1.7 節 (10) 参照) を可能にすることを目的に構築された。ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法の具体的な手順は、フローチャートで示した。5.2 節では、既存の例題に対してこの方法を適用した結果について述べた。適用の結果、ゴール選択基準間のコンフリクトを考慮に入れた形式的ゴール選択方法が適切に機能することが確かめられた。5.3 節では、主に、不確実性推論の形式的ゴール選択への応用について述べた。不確実性推論に用いられる確実性測度のうち、「確信度」に基づく形式的ゴール選択方法を構築し、既存の例題に適用してその有用性を評価した。また、閉路があるゴールグラフを仮定した場合のモデル検査の有用性、および、数値計算やシミュレーションによるゴール選択の傾向や一般的性質の導出の可能性について述べた。

6.2 今後の課題

6.2.1 ゴール選択ゲームの構築

ゴール指向要求分析には、AND 分解と OR 分解という 2 つのゴール分解関係が存在する (2.2 節参照)。この内、AND 分解で接続された子ゴールの中には、無駄な子ゴールが存在する場合や、無駄な子ゴールは存在しないけれども、コスト制約上、すべての子ゴールが達成できない場合が考えられる。他方、OR 分解の場合も、ロバスト性向上などのために、複数の子ゴールを選択すべき場合が考えられる。こうした状況へ柔軟に対応するために、子ゴールを要素とする集合に対してゴール選択基準値を定めることができる枠組みが構築されるべきである。この枠組みとしては、次に述べる方法が考えられる。

協力ゲームの枠組みにおける特性関数形ゲームに基づき、「ゴール選択ゲーム」を、ゴール p とゴールの集合 $C \setminus \{p\}$ (ゴール p がゴールの集合 C に含まれることは、ゴールグラフ上にループが存在することを意味する。これは、無閉路であるというゴールグラフの定義 (2.2 節参照) に矛盾するため許されない) の対 (p, C) と特性関数 v の対 $((p, C), v)$ で定義する。ここで、 (p, C) は、 p を親ゴール、 C を p のすべての子ゴールの集合とする 2 階層部分ゴールグラフ (1.7 節 (9) 参照) を表す。 C の部分集合を S を「提携」と呼び、 C を特に「全員提携」と呼ぶ。 v は、 C の各部分集合 S のゴール選択基準値 (「提携値」) である。特性関数 v は、 C の部分集合全体 $\mathcal{P}(C)$ の上での実数値関数 $v: \mathcal{P}(C) \rightarrow \mathbb{R}$ であり、ゴール選択基準ごとに定義されるものとする。

ゴール選択ゲームの枠組みにおいて、 S の提携値を比較することは、ゴール選択基準値を比較することに相当する。そこで、今後の課題として、特性関数形ゲームにおいて提案されている種々の解概念に基づいて、ゴール選択ゲームの解概念を構築することが挙げられる。

一般に、現実の意思決定問題が表現された数学モデルにおいては、現実の状況に応じて変動する変数が存在する。そのため、その変数に具体的な値を代入し、変数の値の範囲内で変動させることによって変数と結果の間の関連性を調べる「感度分析」が行われることが多い。感度分析によって全体の知見を得るには、コンピュータシミュレーションによって網羅的な分析を行う必要がある。一方、変数に曖昧性や不確実性が内在することがあらかじめ明確に認識される場合には、ファジィ変数を含む形で定式化されることもある [西崎 02]。特に、協力ゲームの枠組みに変数の曖昧性や不確実性を考慮に入れようとする、提携値を正確に評価することが困難になることから、提携値をファジィ変数で表す「協力ファジィゲーム」(Cooperative Fuzzy Games) が提案されている [Aubin 81, Aubin 82] (協力ファジィゲームとの対比を明示したい場合、一般の協力ゲームは特に、「協力クリस्पゲーム」と呼ばれる)。協力ファジィゲームの解概念に関する研究は、国内外の研究者によって精力的に行われてきており、既に多数の報告がなされている [Butnariu 78, Butnariu 80, Sakawa 94, Tsurumi 01, Tijs 04, Butnariu 08]。

狭義のゴール選択 (1.7 節 (10) 参照) という意思決定問題において、常に考慮されるべきゴール選択基準 (4.2.2.1 節参照) である「貢献度」(2.3.1 節 (属性-1) 参照) および「顧客のニーズに関する妥当性」(3.1.1 節参照) は、ステークホルダの主観に基づく値であるため、上述のゴール選択ゲームにおいては、曖昧性や不確実性が内在する変数である。ゴールの対に「ゴール間のコンフリクト」(2.3.1 節 (属性-3) 参照) がある場合、すなわち、同時に達成不可能な内容を持つゴール同士は、提

携が不可能である。狭義のゴール選択においては、一般に、親ゴールを達成する上で、子ゴールの対の間に「相乗効果」や「相殺効果」が発生するため、子ゴールの集合のゴール選択基準値を定める際には、これらの効果を考慮に入れる必要がある。ゴール間のコンフリクトは、相殺効果の特別な場合として位置づけられる。相乗効果および相殺効果の定義は次の通りである。

ファジィ測度[多屋 07]

空でない有限集合 $X = \{1, 2, \dots, n\}$ 上のファジィ測度 μ は、次の2つの条件

$$\begin{cases} \mu(\emptyset) = 0 & (6.1a) \\ A \subset B \subset X \Rightarrow \mu(A) \leq \mu(B) & (6.1b) \end{cases}$$

を満たす X 上の非加法的集合関数である。式(6.1b)は、「単調性」と呼ばれる。

ファジィ測度 μ は加法性を満たす必要はないので、互いに素な X の部分集合 A と B に対して、 $\mu(A \cup B)$ と $\mu(A) + \mu(B)$ の大小関係は次の3通りに分割される。

$$A \cap B = \emptyset \Rightarrow \begin{cases} A \text{ と } B \text{ の間には相互作用がない。} & (6.2a) \\ \Leftrightarrow \mu(A \cup B) = \mu(A) + \mu(B) \\ A \text{ と } B \text{ の間には相乗効果がある。} & (6.2b) \\ \Leftrightarrow \mu(A \cup B) > \mu(A) + \mu(B) \\ A \text{ と } B \text{ の間には相殺効果がある。} & (6.2c) \\ \Leftrightarrow \mu(A \cup B) < \mu(A) + \mu(B) \end{cases}$$

式(6.2a)が成り立つとき、 μ は「加法的」(additive)であるという。式(6.2a)が成り立ち、かつ、式(6.2b)が成り立つとき ($\mu(A \cup B) \geq \mu(A) + \mu(B)$ が成り立つとき)、 μ は「優加法的」(superadditive)であるという。

人間による価値判断には、相乗効果および相殺効果があり、ファジィ測度を用いることによって、これらの相互作用を表現できる[出口 09]。ゴール選択ゲームにおいて、あるゴールの集合 S の提携値を求める特性関数 v は、 C の要素である各ゴール c のゴール選択基準値の総和をその値とする関数として定義できず、相乗効果や相殺効果を考慮に入れた値をとるファジィ測度として定義する必要がある。これはすなわち、ゴール選択ゲームにおいては、協力クリस्पゲームの枠組みにおいて、特性関数 v が優加法的であるという前提が成り立たないことを意味する。したがって、提携値の算出には、曖昧さや不確実性を含むゴール選択基準が用いられるため、ゴール選択ゲームの解概念は、特性関数 v がファジィ測度である協力ファジィゲームの解概念として構築される必要がある。この解概念の構築により、ゴール選択基準の曖昧性や不確実性まで考慮に入れたゴール選択を形式的に分析することが可能になる。

上述の通り、ファジィ測度は非加法的集合関数であり、非加法的集合関数を定めるには、空集合を除くすべての部分集合に対する値が必要となる。すなわち、非加法的集合関数は、システムを部分に還元せずにそのまま記述する。広義のゴール選択(1.7節(10)参照)は、狭義のゴール選択を要素とする要素還元可能なシステムとして定義された(式(4.1))。しかし、協力ファジィゲームとして

のゴール選択ゲームによって狭義のゴール選択を行う場合、狭義のゴール選択は、ファジィ測度としての特性関数 v によって算出された値をもとに行われるため、狭義のゴール選択の対象となるすべての子ゴールを要素とする集合の各々の部分集合に対する値が必要になる。すなわち、狭義のゴール選択は、子ゴールの部分集合のゴール選択基準値という要素に還元不可能である。よって協力ファジィゲームとしてのゴール選択ゲームに基づいて狭義のゴール選択を行う場合、ゴールグラフは、広義のゴール選択は「要素還元的」(reductive)であり、狭義のゴール選択は「全体的」(holistic)であるという、パラドキシカルな性質を持つシステムになる。

6.2.2 ゴール選択基準の精緻化

4.3節では、階層分析法に基づき、ゴールグラフ上のあるゴールに対して、「顧客のニーズに関する妥当性」(3.1.1節参照)を拡張する方法について述べた。この方法はゴール g の顧客のニーズに関する妥当性を見積もる指標 $Cup(g)$ (式(3.1))を拡張する方法であり、 $Cup(g)$ そのものに妥当性(1.7節(7)参照)があることを前提としている。顧客のニーズに関する妥当性は、狭義のゴール選択(1.7節(10)参照)を行う上で、常に考慮に入れるべきゴール選択基準の1つであり(4.2.2.1節参照)、その値の微小な変化によって、選択される子ゴールが変化する可能性がある。したがって、顧客のニーズに関する妥当性の妥当性を保証するためには、4.3節で述べた、 $Cup(g)$ を拡張するという方向性とは別に、 $Cup(g)$ そのものを精緻化することもまた必要である。

$Cup(g)$ を精緻化する方法としては、統計学の知見から $Cup(g)$ の再定義を行うことが挙げられる。満足度行列(図2.6)の個々の満足度は、定義上、個々のステークホルダが、各々の立場から振ったものである(2.3.1節(属性-2)参照)。ここで重要なのは、満足度が「記述統計学」における「代表値」であるという事実である。何故ならば、ここでいうステークホルダとは、特定の立場を持ったステークホルダの総体であって、個々のステークホルダを指すものではないからである。実際には、情報システムの開発プロジェクトに参画する特定の立場のステークホルダは、その全体のごく一部である。一般に、同一の立場のステークホルダの間にも属人性に依存する多様な価値観が存在し得るため、ごく一部のステークホルダが振った満足度がステークホルダの総体としての満足度を反映しているとは限らない。この問題への対処法として、「満足度行列の三次元拡張」が挙げられる。以下、三次元拡張された満足度行列を「三次元満足度行列」と呼ぶことにする。三次元満足度行列の例を(図6.1)に示す。また、三次元満足度行列と区別する目的で、従来の二次元の満足度行列を、以下では「二次元満足度行列」と呼ぶことにする。

三次元満足度行列では、二次元満足度行列にはない「奥行き」が存在するため、同一の立場の複数のステークホルダが満足度を振ることができる。それ故、三次元満足度行列においては、同一の立場の個々のステークホルダの満足度を表現することが可能である。以上の議論から、二次元満足度行列の各成分値は、三次元満足度行列における各成分の奥行き代表値として定義できる。満足度行列の三次元拡張は、統計学の見地からは、特定の立場のステークホルダに対して、「全数調査(悉皆調査)」のもとで個々の満足度を定めることを可能にするものである。しかし、現実には、要求獲得が困難になる大規模な情報システム開発になるほど、特定の立場のステークホルダの数も増大するため、満足度行列1つあたりの満足度の数が膨大となり、データの管理が困難になる恐れがあることから、適

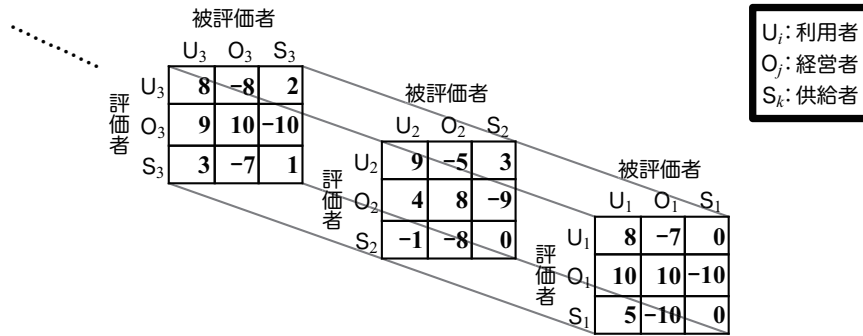


図 6.1: 三次元満足度行列の例

度な数の満足度で抑えられることが望ましい。また、実プロジェクトで作成されるゴールグラフ上のゴールの数は $10^2 \sim 10^3$ 程度である [van Lamsweerde 04] ため、個々のゴールに満足度を振る作業には、一般に膨大な時間を必要とする。したがって、通常業務の合間に、特定の立場のステークホルダ全員に満足度を振ってもらうことは、日程調整上極めて困難であり、現実的ではない。そのため、プロジェクトに参画する特定の立場のステークホルダの一部の人々（「標本」）から、なるべく特定の立場のステークホルダの総体（「母集団」）の満足度を正確に「推定」する「統計的推測」が必要となる。

推定の方法としては、「最尤法」がよく知られている。しかし、標本数が少数の「小標本」下では、満足度（「確率変数」）が母集団分布として一般に仮定される正規分布にしたがわない場合、最尤法を利用することは難しい。そこで、コンピュータシミュレーションに基づくノンパラメトリック最尤法である「ブートストラップ法」に基づいて推定量を算出し、その推定値として $Cup(g)$ を定義する方法を確立することが考えられる。ブートストラップ法を用いる利点として、標本数がごく少数の場合であっても、実用的な精度で推定を行うことが可能である点が挙げられる。したがって、特定の立場のステークホルダのうち、ごく一部のステークホルダしかプロジェクトに参画しない場合も十分考えられるため、ブートストラップ法を用いることは好ましい。ただし、推定量として、「最尤推定量」ではなく、「標本平均」、「標本モード」、「標本メディアン」、「標本トリム平均」などの推定量を採用した方が妥当性が増す場合も考えられる。一般的なプロジェクトに対して、どの推定量が最も妥当性があるのかを実験によって明らかにすることは、今後の重要な課題の1つとして挙げられる。

以上で述べた統計学に基づく $Cup(g)$ の精緻化方法の枠組みは、既に振られた満足度から、なるべく妥当性のある $Cup(g)$ を定義しようとするものである。したがって、振られた満足度それ自体が、個々のステークホルダの真の満足度を表していることが前提となる。しかし、満足度が真の満足度であるかどうかに関する保証は、工学的に厳密に評価可能な範疇を超えていると思われる。これについては、社会科学における諸方法に基づいて行われる必要があるだろう。

最後に、本節では $Cup(g)$ の精緻化に限定して述べたが、その他のゴール選択基準についても統計学に基づく精緻化が行われるべきであることを付記しておく。たとえば、「貢献度」（2.3.1 節（属性-1）参照）は、すべてのステークホルダに対して独立した立場を持つ分析者が独断で値を振る場合もあれば、分析者も含めたステークホルダの合意のもとで振られる場合もある。さらに、分析者が複数いる可能性もある。これらの理由から、分析者が複数である場合は、貢献度もまた満足度と同様に「二次元貢献度行列」、引いては「三次元貢献度行列」を作成し、上述の統計学における推定のもとで

その値が定められるべきである。ただし、貢献度は、その定義上、親ゴールと子ゴールの内容のみに依存して決定されるため、本来は、振る者に依存しない一意な値を持つ。この一意な値を定める方法としては、自然言語処理技術を導入することが考えられる。

6.2.3 ネットワーク分析の諸概念に基づくゴールの品質特性評価

ネットワーク（社会、通信、交通、電力など）分析において、「中心性」は主要な概念であり、様々な分野において用いられてきている [Blau 63, Coleman 66, Sade 72, Mariolis 75, Laumann 76, Pitts 78, Burt 82, Cook 86, Granovetter 95]. そして、過去 60 年間にわたり、種々の「中心性尺度」[Bonacich 72, Freeman 78, Borgatti 06] が提案されてきている [藤本 12]. 要求工学分野においても、ネットワーク分析に立脚した研究が行われている [Bryl 09, Lim 12] もの、ゴールグラフへの応用は十分に行われておらず、特に中心性に関する研究は著者の知る限りでは行われていない。ゴールの中心性は、ゴールの品質特性 [Kaiya 02, 鶴飼 10] を評価する指標になると考えられる。そのため、ゴールの中心性に関する基礎研究を行い、その知見を整備すると共に、ゴールの中心性とゴールの品質特性との関連を明らかにし、中心性尺度に基づきゴールの品質特性を評価する指標を確立することが今後の課題として挙げられる。また、中心性の双対概念として提案された「周辺性」[藤本 12] や、「コンフリクト解決のためのグラフモデル」[Kilgour 87, Fang 89, Fang 93] における「安定性」の概念もまた、ゴールの品質特性と関連する指標であると考えられる。中心性と並行して、これらの概念とゴールの品質特性との関連を明らかにする研究を行うことで、ゴールの品質特性を評価するための指標を確立することが望まれる。

また、中心性の高いゴールは、 IG をすべての初期ゴール (2.2 節参照) を要素とする集合、 SFG を広義のゴール選択 (1.7 節 (10) 参照) がなされた (要求として採用された) すべての最終ゴール (2.2 節参照) を要素とする集合とすると、ゴール選択上のコンフリクトを持つゴールが、 $(\forall g \in IG, \forall g' \in SFG) Path_{(g, g')}$ (2.3.1.1 節参照) の要素である可能性が高い。したがって、中心性の高いゴールは、要求変更に強い要求を獲得するという要求獲得を困難にする問題の 1 つ (2.1 節 (3) 参照) に対応するために、なるべく優先的に狭義のゴール選択 (1.7 節 (10) 参照) がなされるべきである。そのため、中心性 (周辺性) は、貢献度や顧客のニーズに関する妥当性と共に、狭義のゴール選択上常に考慮されるべきゴール選択基準である。今後の課題として、「階層分析法に基づく形式的ゴール選択方法」(4.2 節参照) の評価基準 (属性) として中心性や周辺性を設定するための評価関数の構築が挙げられる。

6.2.4 形式概念分析に基づくゴール選択基準の優先順位付け

ゴール選択基準は、ゴールグラフ中の特定のゴールに対して複数設定することが可能である (3 章および 4 章参照)。ゴール選択基準間のコンフリクト (2.4 節参照) を厳密に考慮する場合、すべてのゴール選択基準の対に対してゴール選択基準間のコンフリクトの有無を判定しなければならない。この対の数は、ゴール選択基準をノードとする完全グラフの辺の数に等しいため、ゴール選択基準の数

を n とすると, $\frac{n(n-1)}{2}$ 通りの対に対して, ゴール選択基準間のコンフリクトがあるかどうかを判定する必要がある. すなわち, 対の数は, n^2 に比例して急激に増加する.

あるゴール選択基準の対に対して, その親ゴールがゴール選択基準間のコンフリクトを持つか否かの判定は, *ccc-dfs* (図 3.3) を実装したプログラムを利用することで, 形式的に得られる. しかし, ゴール選択基準間のコンフリクトを持つ対の数が増えると, 判定結果を手で確認することが困難になるため, 追跡可能性 [IEEE Standards Board 98] (4.1 節参照) が低下する. また, 比較する対の全体数に比例して, ゴール選択基準間のコンフリクトを持つ対の数も一般に増加すると考えられる. ゴール選択基準間のコンフリクトを持つ対の数の増加は, 狭義のゴール選択 (1.7 節 (10) 参照) に関する妥当性 (1.7 節 (7) 参照) の基準が複雑化することを意味するため, 限定合理的な意思決定しか行うことができない人間にとって, 妥当性のあるゴール選択を行うことが困難になる.

ゴール選択基準間のコンフリクトを持つ対の数が多い場合, 階層分析法に基づく形式的ゴール選択方法 (4.2 節参照) を用いて狭義のゴール選択を形式的に行う場合にも問題がある. 階層分析法の特徴として一対比較 (4.2.1 節参照) を用いることが挙げられる. 一対比較は人間が主観的に行うため, 評価基準 (代替案) として設定されるものの数が多いと, やはり判断の誤りが起こりやすい. さらに, それ以前の問題として, 評価基準の一対比較は, 判断結果が不安定な場合が多い [木下 00a]. そのため, 評価基準 (代替案) の数は, 多くとも 1 桁に抑える必要があるとされている [木下 00b]. 属性数の増加は, 階層分析法以外の多属性意思決定方法についても共通する問題である. たとえば, 階層分析法と同様に多属性意思決定方法の 1 つである TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) [Hwang 81] では, 属性ごとに重み因子を主観的に決定する必要がある. TOPSIS を狭義のゴール選択に利用する場合は, 属性としてゴール選択基準を設定することになるため, ゴール選択基準の数が多いと, やはり判断の誤りが混入する可能性を排除できない.

以上で述べた背景から, 主観的に狭義のゴール選択を行うにせよ, 何らかの多属性意思決定方法を用いて形式的に狭義のゴール選択を行うにせよ, いずれにしても, 考慮に入れるゴール選択基準の数が 2 桁以上になる場合, 各々のゴール選択基準の間に優先順位を付け, ゴール選択基準の数を 1 桁に厳選する方法を確立する必要がある. このゴール選択基準に優先順位を付ける方法として, 「形式概念分析 (Formal Concept Analysis)」 [Ganter 99, 鈴木 07] を利用することができる.

形式概念分析の概要 [室伏 09]

形式概念分析とは, 束論に基づいて, 対象間, 属性間, 対象-属性間に階層順序を付けるデータ分析方法である. 形式概念分析では, 3つ組 (O, A, R) を「(形式) 文脈」と呼ぶ. ただし, O は対象の集合, A は属性の集合, R は, O と A の間の 2 項関係 ($R \subset O \times A$) である. $(o, a) \in R$ を oRa と書き, 「対象 o は属性 a を持つ」という関係を表す. 文脈 (O, A, R) は, $|O|$ 行 $|A|$ 列の「文脈表」で, oRa のとき o 行 a 列の成分に \times 印を記すことにより表現する. $X \subset O$ と $Y \subset A$ に対して, X', Y' を次で定義する.

$$\begin{cases} X' := \{a \in A \mid (\forall x \in X)(xRa)\} \\ Y' := \{o \in O \mid (\forall y \in Y)(oRy)\} \end{cases} \quad (6.3)$$

$X' = Y$ かつ $Y' = X$ であるとき、またそのときに限り、 (X, Y) を「(形式) 概念」、 X を「外延」、 Y を「内包」と呼ぶ。この定義から、 X と Y の間には、ガロア結合が成立する。文脈 $\mathbf{C} = (O, A, R)$ の概念全体の集合を $\mathcal{B}(\mathbf{C})$ で表す。

(X_1, Y_1) と、 (X_2, Y_2) を概念とすると、概念間の順序関係 \leq を次で定義する。

$$(X_1, Y_1) \leq (X_2, Y_2) :\Leftrightarrow X_1 \subset X_2 \Leftrightarrow Y_1 \supset Y_2 \quad (6.4)$$

(X_1, Y_1) を (X_2, Y_2) の「下位概念」、 (X_2, Y_2) を (X_1, Y_1) の「上位概念」と呼ぶ。順序関係 \leq は半順序であり、 \leq に関して $\mathcal{B}(\mathbf{C})$ は完備束になる。この束 $\mathcal{B} := (\mathcal{B}(\mathbf{C}), \leq)$ を「概念束」と呼ぶ。概念束は、概念の階層構造を表す。

任意の文脈 (文脈表) とその概念束は 1 対 1 対応する。また、概念束は Hasse 図として視覚的に表示することができる。概念束を Hasse 図として表示することで、概念間 (対象間, 属性間, 対象-属性間) の階層構造を視覚的に把握できる。文脈 (文脈表) から概念束を計算するアルゴリズムや、計算結果を Hasse 図として表示するツールが、数多く提案されている。

形式概念分析に基づいてゴール選択基準の優先順位付けを行う方法としては、たとえば次のような方法が考えられる。

まず、文脈を「多値文脈」[鈴木 07] に拡張する。多値文脈は、4 つ組 (O, A, V, R) で定義される。ただし、 O は対象の集合。 A は属性の集合。 V は属性値。 R は O, A, V の 3 項関係 ($R \subset O \times A \times V$) である。 $(o, a, v) \in R$ は、「対象 o が属性 a を持ち、かつ、属性 a の値が v である」という関係である。

次に、 O をステークホルダ (1.7 節 (4) 参照)、 A をゴール選択基準、 $V = [0, 10]$ であるような多値文脈を考える。そしてこの多値文脈を概念束の Hasse 図として表示する。概念束の Hasse 図は、概念が順序付けられているため、各々の概念に添字を設定することができる。属性の数が最も小さく ($\min |A|$)、ステークホルダの数が最も大きく ($\max |O|$)、属性値が最も大きい ($\max V$) 概念をこの順番で絞り込み、絞られた概念から順に優先順位を付ける。この操作を再帰的に繰り返すことで、概念束の Hasse 図上のすべての概念に優先順位を付けることができる。優先順位付けられた概念のうち、属性が一点集合の概念の優先順位が、ゴール選択基準の優先順位である。したがって、属性が一点集合の概念を優先順位順に並べた際の上位 10 概念を、ゴール選択基準として採用すればよい。この方法は、なるべく多くのステークホルダが、ゴールを達成する上でなるべく重要だと考えているゴール選択基準から順に優先順位を付けることに相当する。この方法の概念図を図 6.2 に示す。

今後の課題として、全探索アルゴリズム (例: 深さ優先探索 [Aho 83] (3.2 節参照) に基づいて、任意の概念束の Hasse 図に対して、ゴール選択基準の優先順位付けを行うことが可能なアルゴリズムを構築し、ゴール選択基準を厳選することを目的として、このアルゴリズムを実装したコンピュータプログラムを開発することが挙げられる。

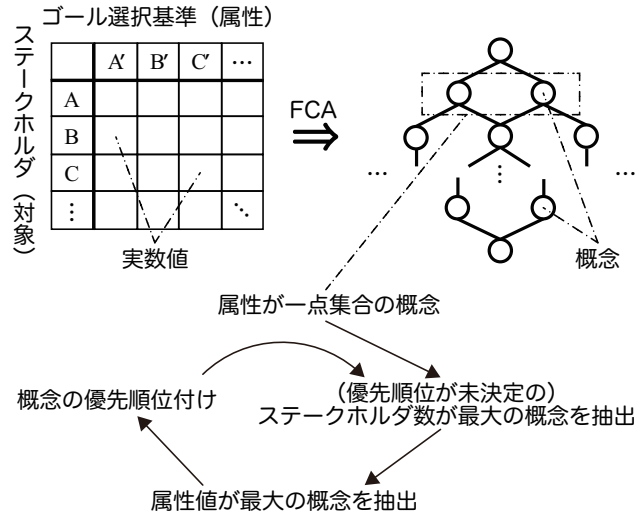


図 6.2: ゴール選択基準の優先順位付け方法の概念図

6.3 本章のまとめ

6.1 節では、本論文で得られた結果の要約を述べた。6.2 節では、今後の課題について述べた。6.2.1 節では、協力ゲームの枠組みにおける特性関数形ゲームに基づいて「ゴール選択ゲーム」を構築することの展望について述べた。6.2.2 節では、統計学の知見に基づいて、ゴール選択基準値を見積もる指標を精緻化することの展望について述べた。6.2.3 節では、ネットワーク分析の中心性などの諸概念に基づいてゴールの品質特性の評価方法を構築することの展望について述べた。6.2.4 節では、形式概念分析に基づいてゴール選択基準の優先順位付けを形式的に行う方法の展望について述べた。

謝辞

本研究は，東京工業大学大学院社会理工学研究科価値システム専攻猪原研究室において，猪原健弘教授のご指導の下に行われたものです．本研究の遂行にあたり，終始手篤いご指導，ご援助を賜りました，本研究科教授，猪原健弘先生に衷心より厚く御礼申し上げます．

猪原研究室の先輩として，喜多村正仁博士（現，日本学術振興会特別研究員（PD））には，研究室配属以来，一方ならぬご支援を頂きました．喜多村正仁博士に対して，ここに満腔の謝意を表します．

最後になりましたが，本論文をご査読頂きました，東京工業大学大学院社会理工学研究科価値システム専攻教授，木嶋恭一先生および往住彰文先生，ならびに，同准教授，金子宏直先生および中丸麻由子先生の各位におかれましては，本論文の瑕疵を微に入り細を穿ちご指摘頂き，また本論文を彫琢する上での貴重なコメントを頂戴しました．ここに深甚な謝意を表します．

本論文に幾許かの「価値」が宿っているとすれば，それは，ひとえに価値システム専攻という^{アカデミア}梁山泊において，以上各位のご薫陶に浴することができたからに他なりません．報恩謝徳の一念です．ここに学恩を銘記します．



参考文献

- [Aho 83] Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: *Data Structures and Algorithms*, Addison Wesley (1983), [邦訳]A. V. エイホ, J. E. ホップクロフト, J. D. ウルマン (著), 大野 義夫 (訳), データ構造とアルゴリズム, 培風館 (1987)
- [Anton 96] Anton, A. I.: Goal-Based Requirements Analysis, in *Proc. of the 2nd IEEE International Conference on Requirements Engineering (ICRE'96)*, pp. 136–144 (1996)
- [Aubin 81] Aubin, J.-P.: Cooperative Fuzzy Games, *Mathematics of Operations Research*, Vol. 6, No. 1, pp. 1–13 (1981)
- [Aubin 82] Aubin, J.-P.: *Mathematical Methods of Game and Economic Theory*, North-Holland, Amsterdam, revised edition (1982)
- [Baier 08] Baier, C. and Katoen, J.-P.: *Principles of Model Checking*, MIT Press (2008)
- [Basili 92] Basili, V. R.: Software Modeling and Measurement: The Goal/Question/Metric Paradigm, Technical report, University of Maryland, CS-TR-2956, UMIACS-TR-92-96 (1992)
- [Beck 04] Beck, K. and Andres, C.: *Extreme Programming explained—Embrace Change*, Addison-Wesley Professional, 2nd edition (2004), [邦訳]ケント・ベック (著), シンシア・アンドレス (協力), 長瀬 嘉秀 (監訳), テクノロジックアート (訳), XP エクストリーム・プログラミング入門—変化を受け入れる, 第2版, ピアソン・エデュケーション (2005)
- [Blau 63] Blau, P. M.: *The Dynamics of Bureaucracy: A Study of Interpersonal Relations in Two Government Agencies*, The University of Chicago Press, Chicago (1963)

- [Bleistein 06] Bleistein, S. J., Cox, K., Verner, J. and Phalp, K. T.: B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process, *Information and Software Technology*, Vol. 48, No. 9, pp. 846–868 (2006)
- [Boehm 81] Boehm, B. W.: *Software Engineering Economics*, Prentice Hall (1981)
- [Boehm 85] Boehm, B. W.: A spiral model of software development and enhancement, in *Proc. of the International Workshop on the Software Process and Software Environments (ISPW'85)*, pp. 22–42 (1985)
- [Boehm 96] Boehm, B. W. and In, H.: Identifying Quality—Requirement Conflicts, *IEEE Software*, Vol. 13, No. 2, pp. 25–35 (1996)
- [Bonacich 72] Bonacich, P.: Factoring and weighing approaches to status scores and clique identification, *Journal of Mathematical Sociology*, Vol. 2, No. 1, pp. 113–120 (1972)
- [Borgatti 06] Borgatti, S. P. and Everett, M. G.: A Graph-theoretic perspective on centrality, *Social Networks*, Vol. 28, No. 4, pp. 466–484 (2006)
- [Brooks 86] Brooks, F. P.: No Silver Bullet—Essence and Accidents of Software Engineering, in *Proc. of the IFIP 10th World Computing Conference*, pp. 1069–1076 (1986)
- [Bryl 09] Bryl, V., Giorgini, P. and Mylopoulos, J.: Designing socio-technical systems: from stakeholder goals to social networks, *Requirements Engineering*, Vol. 14, No. 1, pp. 47–70 (2009)
- [Burt 82] Burt, R. S.: *Toward a Structural Theory of Action: Network Models of Social Structure, Perception and Action*, Academic Press, New York (1982)
- [Butnariu 78] Butnariu, D.: Fuzzy games: A description of the concept, *Fuzzy Sets and Systems*, Vol. 1, No. 3, pp. 181–192 (1978)
- [Butnariu 80] Butnariu, D.: Stability and Shapley value for an n-persons fuzzy game, *Fuzzy Sets and Systems*, Vol. 4, No. 1, pp. 63–72 (1980)
- [Butnariu 08] Butnariu, D. and Kroupa, T.: Shapley mappings and the cumulative value for n-person games with fuzzy coalitions, *European Journal of Operational Research*, Vol. 186, No. 1, pp. 288–299 (2008)
- [Chalmers 99] Chalmers, A. F.: *What is This Thing Called Science?*, Open University Press, 3rd edition (1999), [邦訳] A. F. チャルマーズ(著), 高田 紀代志, 佐野 正博(訳), 科学論の展開—科学と呼ばれているのは何なのか?, 恒星社厚生閣, 新版(1985)

- [Chang 71] Chang, C. L. and Slagle, J. R.: An Admissible and Optimal Algorithm for Searching AND/OR Graphs, *Artificial Intelligence*, Vol. 2, No. 4, pp. 117–128 (1971)
- [Chastek 91] Chastek, G. J., Graham, M. H. and Zelesnik, G.: Requirements Engineering and Analysis Workshop Proceedings, Technical report, CMU/SEI-91-TR-30 (1991)
- [Checkland 81] Checkland, P.: *Systems Thinking, Systems Practice*, John Wiley & Sons (1981), [邦訳] 高原 康彦, 中野文平 (監訳), 新しいシステムアプローチ—システム思考とシステム実践, オーム社 (1985)
- [Checkland 90] Checkland, P. and Scholes, J.: *Soft Systems Methodology in Action*, John Wiley & Sons (1990), [邦訳] ピーター・チェックランド, ジム・スクールズ (著), 妹尾 堅一郎 (監訳), ソフト・システムズ方法論, 有斐閣 (1994)
- [Checkland 97] Checkland, P. and Holwell, S.: *Information, Systems and Information Systems—making sense of the field*, John Wiley & Sons (1997)
- [Clarke, Jr. 99] Clarke, Jr., E. M., Grumberg, O. and Peled, D. A.: *Model Checking*, MIT Press (1999)
- [Coleman 66] Coleman, J. S., Katz, E. and Menzel, H.: *Medical Innovation: A Diffusion Study*, Bobbs-Merrill, New York (1966)
- [Cook 86] Cook, K. S., Emerson, R. M., Gillmore, M. R. and Yamagishi, T.: The Distribution of Power in Exchange Networks: Theory and Experimental Results, *American Journal of Sociology (AJS)*, Vol. 89, No. 2, pp. 275–305 (1986)
- [Demarco 79] Demarco, T. and Plauger, P.: *Structured Analysis and System Specification*, Prentice Hall (1979), [邦訳] 高梨 智弘, 黒田 純一郎 (監訳), 構造化分析とシステム仕様—目指すシステムを明確にするモデル化技法, 日経 BP 出版センター (1994)
- [Dempster 67] Dempster, A. P.: Upper and Lower Probabilities Induced by a Multivalued Mapping, *The Annals of Mathematical Statistics*, Vol. 38, No. 2, pp. 325–339 (1967)
- [Dempster 68] Dempster, A. P.: A Generalization of Bayesian Inference, *Royal Statistical Society, Series B*, pp. 205–247 (1968)
- [Department of Defense 85] Department of Defense, : Military Standard—Defense System Software Development (1985)

- [Duda 76] Duda, R. O., Hart, P. E. and Nilsson, N. J.: Subjective Bayesian methods for rule-based inference systems, in *Proc. of the National Computer Conference and Exposition (AFIPS'76)*, pp. 1075–1082 (1976)
- [Fang 89] Fang, L., Hipel, K. W. and Kilgour, D. M.: Conflict Models in graph form: solution concepts and their interrelationships, *European Journal of Operational Research*, Vol. 41, pp. 86–100 (1989)
- [Fang 93] Fang, L., Hipel, K. W. and Kilgour, D. M.: *Interactive Decision Making: The Graph Model for Conflict Resolution*, Wiley-Interscience (1993)
- [Feather 87] Feather, M. S.: Language Support for the Specification and Development of Composite Systems, *ACM Transactions on Programming Language and Systems (TOPLAS)*, Vol. 9, No. 2, pp. 198–234 (1987)
- [Freeman 78] Freeman, L. C.: Centrality in Social Networks: Conceptual Clarification, *Social Networks*, Vol. 1, No. 3, pp. 215–239 (1978)
- [Ganter 99] Ganter, B. and Wille, R.: *Formal Concept Analysis: Mathematical Foundations*, Springer (1999)
- [Gilb 77] Gilb, T.: *Software Metrics*, Winthrop Publishers (1977)
- [Gilb 88] Gilb, T.: *Principles of Software Engineering Management*, Addison-Wesley Professional (1988)
- [Glinz 07] Glinz, M.: Requirements Engineering II: Conferences and Journals (2007), https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/requirements_engineering_II_hs07/RE_Journals_Conferences.pdf
- [Gnesi 13] Gnesi, S. and Margaria, T.: *Formal Methods for Industrial Critical Systems*, John Wiley & Sons (2013)
- [Goldratt 12] Goldratt, E. M. and Cox, J.: *The Goal: A Process of Ongoing Improvement*, North River Press, 3rd edition (2012), [邦訳] エリヤフ・ゴールドラット (著), 三本木 亮 (訳), 稲垣 公夫 (解説), ザ・ゴール—企業の究極の目的とは何か, ダイヤモンド社 (2001)
- [Granovetter 95] Granovetter, M.: *Getting a Job: A Study of Contacts and Careers*, The University of Chicago Press, Chicago, 2nd edition (1995)
- [Harman 09] Harman, M., Mansouri, S. A. and Zhang, Y.: Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical report, Technical Report of King's College London, TR-09-03 (2009)

- [Hartshorne 32] Hartshorne, C. and Weiss, P. (eds.): *Collected Papers of Charles Sanders Peirce, Volumes I and II: Principles of Philosophy and Elements of Logic*, Belknap Press of Harvard University Press (1932)
- [Hayashi 12] Hayashi, S., Tanabe, D., Kaiya, H. and Saeki, M.: Impact Analysis on an Attributed Goal Graph, *IEICE Trans. Inf. & Syst.*, Vol. E95-D, No. 4, pp. 1012–1020 (2012)
- [Hwang 81] Hwang, C.-L. and Yoon, K.: *Multiple Attribute Decision Making: Methods and Applications*, Springer (1981)
- [IEEE Standards Board 98] IEEE Standards Board, : IEEE Recommended Practice for Software Requirements Specifications, IEEE Std. 830-1998 (1998)
- [Inoue 10a] Inoue, K., Doncescu, A. and Nabeshima, H.: Discovering Rules by Meta-Level Abduction, in *Luc De Raedt (eds.), Inductive Logic Programming: Revised Papers from the 19th International Conference (ILP'09)*, Lecture Notes in Artificial Intelligence, Vol. 5989, pp. 49–64, Springer (2010)
- [Inoue 10b] Inoue, K., Doncescu, A. and Nabeshima, H.: Hypothesizing about Networks by Meta-Level Abduction, in *Proc. of the 20th International Conference on Inductive Logic Programming (ILP'10)* (2010)
- [Inoue 11] Inoue, K., Doncescu, A. and Nabeshima, H.: Hypothesizing about Casual Networks with Positive and Negative Effects by Meta-Level Abduction, in *Paolo Frasconi and Francesca A. Lisi (eds.), Inductive Logic Programming: Revised Papers from the 20th International Conference (ILP'10)*, Lecture Notes in Artificial Intelligence, Vol. 6489, pp. 114–129, Springer (2011)
- [Ishizuka 82a] Ishizuka, M.: An Extension of Dempster & Shafer's Theory to Fuzzy Set for Constructing Expert Systems, *生産研究*, Vol. 34, No. 7, pp. 312–315 (1982)
- [Ishizuka 82b] Ishizuka, M. and Fu, K. S.: Inference Procedures under Uncertainty for the Problem-Reduction Method, *Information Sciences*, Vol. 28, pp. 179–206 (1982)
- [Ishizuka 83] Ishizuka, M.: Inference Methods Based on Extended Dempster & Shafer's Theory for Problems with Uncertainty/Fuzziness, *New Generation Computing*, Vol. 1, pp. 159–168 (1983)
- [ISO 91] ISO/IEC 9126: Information Technology—Software product evaluation—Quality characteristics and guidelines for their use (1991)
- [ISO 08] ISO 9001:2008 Quality management systems—Requirements (2008)

- [Jackson 00] Jackson, M.: *Problem Frames: Analysing & Structuring Software Development Problems*, Addison-Wesley Professional (2000), [邦訳]マイケル・ジャクソン(著), 榊原 彰, 牧野 祐子(訳), プロブレムフレーム—ソフトウェア開発問題の分析と構造化, ソフトウェア開発の課題3, 翔泳社(2006)
- [Jackson 06] Jackson, D.: *Software Abstractions: Logic, Language, and Analysis*, MIT Press (2006), [邦訳]Daniel Jackson(著), 中島 震(監訳), 今井 健男, 酒井 政裕, 遠藤 侑介, 片岡 欣夫(共訳), 抽象によるソフトウェア設計—Alloyではじめる形式手法, オーム社(2011)
- [Jacobson 99] Jacobson, I., Booch, G. and Rumbaugh, J.: *The Unified Software Development Process*, Addison-Wesley Professional (1999), [邦訳]イヴァー・ヤコブソン, グラディ・ブーチ, ジェームズ・ランボー(著), 藤井 拓(監修), 日本ラショナルソフトウェア株式会社(訳), UMLによる統一ソフトウェア開発プロセス—オブジェクト指向開発方法論, 翔泳社(2000)
- [Jirotko 92] Jirotko, M., Goguen, J. A., Monk, A. F. and Gaines, B. R. (eds.): *Requirements Engineering—Social and Technical Issues*, Academic Press (1992)
- [Kaiya 02] Kaiya, H., Horai, H. and Saeki, M.: AGORA: Attributed Goal-Oriented Requirements Analysis Method, in *Proc. of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering (RE'02)*, pp. 13–22 (2002)
- [Kaiya 05] 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)
- [Kardasis 98] Kardasis, P. and Loucopoulos, P.: Aligning Legacy Information Systems to Business Processes, in *Proc. of the 10th International Conference on Advanced Information Systems Engineering (CAiSE'98)*, pp. 8–12 (1998)
- [Katzenstein 00] Katzenstein, G. and Lerch, F. J.: Beneath the Surface of Organizational Processes: A Social Representation Framework for Business Process Redesign, *ACM Transactions on Information Systems (TOIS)*, Vol. 18, No. 4, pp. 383–422 (2000)
- [Kavakli 02] Kavakli, E.: Goal Oriented Requirements Engineering: A Unifying Framework, *Requirements Engineering Journal*, Vol. 6, No. 4, pp. 237–251 (2002)
- [Kelly 98] Kelly, T. P.: *Arguing Safety—A Systematic Approach to Managing Safety Cases*, PhD thesis, Department of Computer Science, University of York (1998)

- [Kilgour 87] Kilgour, D. M., Hipel, K. W. and Fang, L.: The graph model for conflicts, *Automatica*, Vol. 23, No. 1, pp. 41–55 (1987)
- [Klir 91] Klir, G. J.: *Facets of Systems Science*, IFSR International Series on Systems Science and Engineering, vol.7, Plenum Publishing Corporation (1991)
- [Kokune 07] Kokune, A., Mizuno, M., Kadoya, K. and Yamamoto, S.: FBCM: Strategy modeling method for the validation of software requirements, *Journal of Systems and Software*, Vol. 80, No. 3, pp. 314–327 (2007)
- [Kruchten 03] Kruchten, P.: *The Rational Unified Process—An Introduction*, Addison-Wesley Professional, 3rd edition (2003), [邦訳] フィリップ・クルーシュテン(著), 藤井 拓(訳), ラショナル統一プロセス入門, 第3版, ピアソン・エデュケーション (2004)
- [Lapouchnian 05] Lapouchnian, A.: Goal-Oriented Requirements Engineering: An Overview of the Current Research (2005), <http://www.cs.utoronto.ca/~alexei/pub/Lapouchnian-Depth.pdf>
- [Larman 03] Larman, C.: *Agile & Iterative Development—A Manager’s Guide*, Addison-Wesley Professional (2003), [邦訳] クレーグ・ラーマン(著), ウルシステムズ 小高 慎治郎, 松田 直樹(監訳), 越智 典子(訳), 初めてのアジャイル開発—スクラム, XP, UP, Evo で学ぶ反復型開発の進め方, 日経BP社 (2004)
- [Laumann 76] Laumann, E. O. and Pappi, F. U.: *Networks of Collective Action: Perspective on Community Influence Systems*, Academic Press, New York (1976)
- [Lévi-Strauss 85] Lévi-Strauss, C.: *La Pensée sauvage*, Adler’s Foreign Books, revised edition (1985), [邦訳] クロード・レヴィ・ストロース(著), 大橋 保夫(翻訳), 野生の思考, みすず書房 (1976)
- [Lim 12] Lim, S. L. and Finkelstein, A.: StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation, *IEEE Transactions on Software Engineering (TSE)*, Vol. 38, No. 3, pp. 707–735 (2012)
- [Loucopoulos 95] Loucopoulos, P. and Karakostas, V.: *System Requirements Engineering*, McGraw-Hill (1995), [邦訳] Pericles Loucopoulos, Vassilios Karakostas(著), 富野 寿(監訳), 要求定義工学入門, 構造計画研究所 (1997)
- [Luhmann 73] Luhmann, N.: *Zweckbegriff und Systemrationalität: Über die Funktion von Zwecken in sozialen Systemen (suhrkamp taschenbuch wjssenschaft)*, Suhrkamp Verlag (1973), [邦訳] ニクラス・ルーマン(著), 馬場 靖雄, 上村 隆広(訳), 目的概念とシステム合理性—社会システムにおける目的の機能について, 勁草書房 (1990)

- [Maiden 02] Maiden, N. A. M., Pavan, P., Gizikis, A., Clause, O., Kim, H. and Zhu, X.: Making Decisions with Requirements: Integrating i* Goal Modelling and the AHP, in *Proc. of the 8th International Workshop on Requirements Engineering: Foundation for Software Quality* (REFSQ'02), pp. 24–35 (2002)
- [Mariolis 75] Mariolis, P.: Interlocking Directorates and Control of Corporations: The theory of Bank Control, *Social Science Quarterly*, Vol. 56, pp. 425–439 (1975)
- [Martelli 78] Martelli, A. and Montanari, U.: Optimizing Decision Trees Through Heuristically Guided Search, *Communications of the ACM* (CACM), Vol. 21, No. 12, pp. 1025–1039 (1978)
- [Michalewicz 10] Michalewicz, Z. and Fogel, D. B.: *How to Solve It: Modern Heuristics*, Springer, 2nd edition (2010)
- [Mylopoulos 92] Mylopoulos, J., Chung, L. and Nixon, B.: Representing and Using Non-Functional Requirements: A Process-Oriented Approach, *IEEE Transactions on Software Engineering* (TSE), Vol. 18, No. 6, pp. 483–497 (1992)
- [Nakatani 10] Nakatani, T., Tsumaki, T. and Tamai, T.: Requirements engineering education for senior engineers: Course design and its evaluation, in *Proc. of the 5th International Workshop on Requirements Engineering Education and Training* (REET'10), pp. 26–35 (2010)
- [Nilsson 71] Nilsson, N. J.: *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill (1971), [邦訳] 合田 周平, 増田 一比古 (訳), 人工知能—問題解決のシステム論, コロナ社 (1973)
- [Nilsson 80] Nilsson, N. J.: *Principles of Artificial Intelligence*, Morgan Kaufmann Publishers (1980), [邦訳] 白井 良明, 辻井 潤一, 佐藤 泰介 (訳), 人工知能の原理, コンピュータ・サイエンス研究書シリーズ 26, 日本コンピュータ協会 (1983)
- [Nuseibeh 00] Nuseibeh, B. and Easterbrook, S.: Requirements Engineering: A Roadmap, in *Proc. of the 22nd IEEE International Conference on Software Engineering—Future of Software Engineering* (ICSE'00), pp. 35–46 (2000)
- [Parmenter 07] Parmenter, D.: *Key Performance Indicators: Developing, Implementing, and Using Winning KPIs*, John Wiley & Sons (2007)
- [Pearl 84] Pearl, J.: *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley (1984)
- [Petrie 92] Petrie, C. J. (ed.): *Enterprise Integration Modeling—Proceedings of the First International Conference*, The MIT Press (1992)

- [Pitts 78] Pitts, F. R.: The Medieval River Trade Network of Russia Revisited, *Social Networks*, Vol. 1, No. 3, pp. 285–292 (1978)
- [Robinson 03] Robinson, W. N., Pawlowski, S. D. and Volkov, V.: Requirements Interaction Management, *ACM Computing Surveys* (CSUR), Vol. 35, No. 2, pp. 132–190 (2003)
- [Royce 70] Royce, W. W.: Managing the Development of Large Software Systems, in *Proc. of the IEEE WESCON*, pp. 1–9 (1970)
- [Ruhe 02] Ruhe, G., Eberlein, A. and Pfahl, D.: Quantitative WinWin—A New Method for Decision Support in Requirements Negotiation, in *Proc. of the 14th International Conference on Software Engineering and Knowledge Engineering* (SEKE'02), pp. 159–166 (2002)
- [Rumbaugh 90] Rumbaugh, J. R., Blaha, M. R., Lorensen, W., Eddy, F. and Premerlani, W.: *Object-Oriented Modeling and Design*, Prentice Hall (1990), [邦訳] J. ランボー, M. プラハ, W. プレメラニ, F. エディ, W. ローレンセン (著), 羽生田 栄一 (監訳), オブジェクト指向方法論 OMT—モデル化と設計, トッパン (1992)
- [Rumbaugh 98] Rumbaugh, J., Jacobson, I. and Booch, G.: *The Unified Modeling Language Reference Manual*, Addison-Wesley Professional (1998), [邦訳] ジェームズ・ランボー, イヴァー・ヤコブソン, グラディ・ブーチ (著), 石塚 圭樹 (監訳), 日本ラショナルソフトウェア株式会社 (訳), UML リファレンスマニュアル, ピアソン・エデュケーション (2002)
- [Russell 10] Russell, S. and Norvig, P.: *Artificial Intelligence: A Modern Approach*, Pearson, 3rd edition (2010), [邦訳] Stuart Russell, Peter Norvig (著), 古川 康一 (監訳), エージェントアプローチ人工知能, 共立出版 (2008)
- [Saaty 80] Saaty, T. L.: *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill (1980)
- [Sade 72] Sade, D. S.: Sociometrics of *Macaca mulatta*: I. Linkages and Cliques in Grooming Matrices, *Folia Primatologica*, Vol. 18, pp. 196–223 (1972)
- [Sakawa 94] Sakawa, M. and Nishizaki, I.: A lexicographical solution concept in an n-person cooperative fuzzy game, *Fuzzy Sets and Systems*, Vol. 61, No. 3, pp. 265–275 (1994)
- [Sato 09] Sato, T., Ishihata, M. and Inoue, K.: Constraint-Based Probabilistic Modeling for Statistical Abduction, in *Proc. of the 19th International Conference on Inductive Logic Programming* (ILP'09) (2009)

- [Schwaber 01] Schwaber, K. and Beedle, M.: *Agile Software Development with Scrum*, Prentice Hall (2001), [邦訳]長瀬 嘉秀, 今野 睦(監訳), スクラム・エバンジェリスト・グループ(訳), 株式会社テクノロジックアート(編集), アジャイルソフトウェア開発スクラム, ピアソン・エデュケーション(2003)
- [Shafer 76] Shafer, G.: *A Mathematical Theory of Evidence*, Princeton Univ. Press (1976)
- [Shortliffe 75] Shortliffe, E. H. and Buchanan, B. G.: A Model of Inexact Reasoning in Medicine, *Mathematical Biosciences*, Vol. 23, pp. 351–379 (1975)
- [Simon 57] Simon, H. A.: *Models of Man: Social and Rational—Mathematical Essays on Rational Human Behavior in a Social Setting*, Wiley (1957), [邦訳]ハーバート・A・サイモン(著), 宮沢 光一(監訳), 人間行動のモデル, 同文館出版(1970)
- [Sommerville 97] Sommerville, I. and Sawyer, P.: *Requirements Engineering—A Good Practice Guide*, John Wiley & Sons (1997), [邦訳]Ian Sommerville, Pete Sawyer(著), 富野 寿(監訳), 要求定義工学プラクティスガイド, 構造計画研究所(2000)
- [Tanabe 08] Tanabe, D., Uno, K., Akamine, K., Yoshikawa, T., Kaiya, H. and Saeki, M.: Supporting Requirements Change Management in Goal Oriented Analysis, in *Proc. of the 16th IEEE International Requirements Engineering Conference (RE'08)*, pp. 3–12 (2008)
- [The Standish Group 95] The Standish Group, : The Standish Group Report CHAOS (1995)
- [The Standish Group 09] The Standish Group, : CHAOS Summary 2009—The 10 Laws of CHAOS (2009)
- [Theyer 94] Theyer, R. H. and Dorfman, M.: *System and Software Requirements Engineering*, IEEE Computer Society (1994)
- [Tijs 04] Tijs, S., Brânzei, R., Ishihara, S. and Muto, S.: On cores and stable sets for fuzzy games, *Fuzzy Sets and Systems*, Vol. 146, No. 2, pp. 285–296 (2004)
- [Tsumaki 06] Tsumaki, T. and Tamai, T.: Framework for Matching Requirements Elicitation Techniques to Project Characteristics, *Software Process: Improvement and Practice*, Vol. 11, No. 5, pp. 505–519 (2006)
- [Tsurumi 01] Tsurumi, M., Tanino, T. and Inuiguchi, M.: A Shapley function on a class of cooperative fuzzy games, *European Journal of Operational Research*, Vol. 129, No. 3, pp. 596–618 (2001)

- [van Lamsweerde 91] van Lamsweerde, A., Dardenne, A., Delcourt, B. and Dubis, F.: The KAOS Project: Knowledge Acquisition in Automated Specification of Software, in *Proc. of the AAAI Spring Symposium Series, Design of Composite Systems*, pp. 59–62 (1991)
- [van Lamsweerde 01] van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour, in *Proc. of the 5th IEEE International Symposium on Requirements Engineering (RE'01)*, pp. 249–262 (2001)
- [van Lamsweerde 04] van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice, in *Proc. of the 12th IEEE International Requirements Engineering Conference (RE'04)*, pp. 3–6 (2004)
- [van Lamsweerde 09] van Lamsweerde, A.: *Requirements Engineering: From System Goals to UML Models to Software Specifications*, John Wiley & Sons (2009)
- [von Bertalanffy 69] von Bertalanffy, L.: *General System Theory—Foundations, Development, Applications*, George Braziller (1969), [邦訳]L. フォン・ベルタランフィ (著), 長野 敬, 太田 邦昌 (訳), 一般システム理論—その基礎・発展・応用, みすず書房 (1973)
- [Wilson 90] Wilson, B.: *Systems: Concepts, Methodologies, and Applications*, John Wiley & Sons, 2nd edition (1990), [邦訳]Brian Wilson (著), 根来 龍之 (監訳), システム仕様の分析学—ソフトシステム方法論, 共立出版 (1996)
- [Yamamoto 06] Yamamoto, S., Kaiya, H., Cox, K. and Bleinstein, S.: Goal Oriented Requirements Engineering: Trends and Issues, *IEICE Trans. Inf. & Syst.*, Vol. E89-D, No. 11, pp. 2701–2711 (2006)
- [Yamamoto 08] Yamamoto, K. and Saeki, M.: Attributed Goal-Oriented Analysis Method for Selecting Alternatives of Software Requirements, *IEICE Trans. Inf. & Syst.*, Vol. E91-D, No. 4, pp. 921–932 (2008)
- [Yu 97] Yu, E. S. K.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, in *Proc. of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, pp. 226–235 (1997)
- [Yu 98] Yu, E. S. K. and Mylopoulos, J.: Why Goal-Oriented Requirements Engineering (1998), <ftp://ftp.db.toronto.edu/pub/eric/REFSQ98.html>
- [Yu 11] Yu, E. S. K., Giorgini, P., Maiden, N. and Mylopoulos, J.: *Social Modeling for Requirements Engineering*, The MIT Press (2011)
- [Zadeh 65] Zadeh, L. A.: Fuzzy Sets, *Information and Control*, Vol. 8, pp. 338–353 (1965)

- [Zave 97] Zave, P.: Classification of Research Efforts in Requirements Engineering, *ACM Computing Surveys (CSUR)*, Vol. 29, No. 4, pp. 315–321 (1997)
- [REBOK 11] REBOK 企画 WG (編): 要求工学知識体系, 近代科学社 (2011)
- [浅居 01] 浅居 喜代治 (編著): 基礎 システム工学, オーム社 (2001)
- [アルダムラトフ 07] アルダムラトフ ジャマルト, 糸賀 裕弥, 大西 淳: ゴール指向要求分析法における合意形成手法, 電子情報通信学会技術研究報告 (信学技法), SS2007-10, Vol. 107, No. 99, pp. 7–12 (2007)
- [飯島 93] 飯島 淳一: 意思決定支援システムとエキスパートシステム, 日科技連出版社 (1993)
- [石塚 83a] 石塚 満: Dempster & Shafer の確率理論, 電子通信学会誌, Vol. 66, No. 9, pp. 900–903 (1983)
- [石塚 83b] 石塚 満: 建築物被害査定のエキスパート・システム, 情報処理学会論文誌, Vol. 24, No. 3, pp. 357–363 (1983)
- [石塚 83c] 石塚 満: 不確かな知識の取り扱い, 計測と制御, Vol. 22, No. 9, pp. 774–779 (1983)
- [石塚 85] 石塚 満: 曖昧な知識の表現と利用, 情報処理, Vol. 26, No. 12, pp. 1481–1486 (1985)
- [井上 92] 井上 克巳: アブダクションの原理, 人工知能学会誌, Vol. 7, No. 1, pp. 48–59 (1992)
- [井上 05] 井上 陽晴, 大西 淳: AHP を用いたゴール指向要求分析支援, 電子情報通信学会技術研究報告 (信学技法), SS2004-60, Vol. 104, No. 722, pp. 25–30 (2005)
- [井上 10] 井上 克巳: アブダクションとインダクション, 人工知能学会誌, Vol. 25, No. 3, pp. 389–399 (2010)
- [茨木 11] 茨木 俊秀, 片山 徹, 藤重 悟 (監修), 太田 快人, 酒井 英昭, 高橋 豊, 田中 利幸, 永持 仁, 福島 雅夫 (編集): 数理工学辞典, 朝倉書店 (2011)
- [鵜飼 10] 鵜飼 孝典, 林 晋平, 佐伯 元司: 属性つきゴールグラフにおけるゴールの品質特性, 第 17 回ソフトウェア工学の基礎ワークショップ (FOSE2010), pp. 5–14 (2010)
- [大須賀 91] 大須賀 節雄 (監訳): 人工知能大辞典, 丸善株式会社 (1991)
- [大西 02] 大西 淳, 郷 健太郎: 要求工学, ソフトウェアテクノロジーシリーズ 9, 共立出版 (2002)
- [大西 09] 大西 淳 (監修), 妻木 俊彦, 白銀 純子 (著): 要求工学概論—要求工学の基本概念から応用まで, トップエスイー基礎講座 2, 近代科学社 (2009)

- [川喜田 67] 川喜田 二郎：発想法—創造性開発のために，中公新書（1967）
- [川喜田 70] 川喜田 二郎：続・発想法—KJ法の展開と応用，中公新書（1970）
- [河本 95] 河本 英夫：オートポイエーシス—第三世代システム，青土社（1995）
- [木下 00a] 木下 栄蔵（編著）：AHPの理論と実際，日科技連出版社（2000）
- [木下 00b] 木下 栄蔵：入門AHP—決断と合意形成のテクニック，日科技連出版社（2000）
- [久保 09] 久保 幹雄，J. P. ペドロソ：メタヒューリスティクスの数理，共立出版（2009）
- [熊澤 11] 熊澤 努：高信頼性ソフトウェアシステムの実現のためのモデル修正技術に関する研究，PhD thesis，東京大学（2011）
- [薦田 97] 薦田 憲久，大川 剛直，安信 千津子：エキスパートシステムの設計と開発，昭晃堂（1997）
- [斎藤 06] 斎藤 忍，山本 修一郎：属性値に基づくゴール選択手法の提案と考察，経営情報学会誌，Vol. 15, No. 3, pp. 37–50（2006）
- [佐伯 01] 佐伯 元司：要求工学サーベイ（2001），<http://www.se.cs.titech.ac.jp/lecture/se/req-doc.pdf>
- [佐藤 10] 佐藤 泰介：統計的アブダクション，人工知能学会誌，Vol. 25, No. 3, pp. 400–407（2010）
- [佐藤 11] 佐藤 慎一，石川 冬樹，猪原 健弘：貢献度と顧客のニーズに関する妥当性の間のコンフリクト検出指標，ソフトウェアエンジニアリングシンポジウム2011 (SES2011) (2011)，ショートペーパー
- [佐藤 12a] 佐藤 慎一，猪原 健弘：貢献度と顧客のニーズに関する妥当性の間のコンフリクト検出アルゴリズム，コンピュータソフトウェア，岩波書店，Vol. 29, No. 4, pp. 251–257（2012），レター論文
- [佐藤 12b] 佐藤 慎一，猪原 健弘：階層分析法に基づいて顧客の影響度を考慮に入れた顧客のニーズに関する妥当性の計算方法，ソフトウェアエンジニアリングシンポジウム2012 (SES2012) (2012)，フルペーパー
- [佐藤 12c] 佐藤 慎一，猪原 健弘：貢献度と顧客のニーズに関する妥当性の間のコンフリクト検出における応用上の問題とその解決手法，コンピュータソフトウェア，岩波書店，Vol. 29, No. 3, pp. 77–90（2012）
- [佐藤 13] 佐藤 慎一，猪原 健弘：階層分析法に基づいて拡張された顧客のニーズに関する妥当性の計算方法，情報処理学会論文誌，Vol. 54, No. 2, pp. 765–773（2013）

- [新原 04] 新原 敦介, 河野 仁一, 海谷 治彦, 佐伯 元司: ゴール指向要求分析を用いたステークホルダの対立の検出, 情報処理学会研究報告, 2004-SE-144(14), p. 99-106 (2004)
- [鈴木 07] 鈴木 治, 室伏 俊明: 形式概念分析—入門・支援ソフト・応用, 知能と情報(日本知能情報ファジィ学会誌), Vol. 19, No. 2, pp. 103-142 (2007)
- [高原 07] 高原 康彦, 斎藤 敏雄, 旭 貴朗, 柴 直樹: 形式手法モデル理論アプローチ—情報システム開発の基礎, 日科技連出版社 (2007)
- [田中 08] 田中 譲(監修), 磯部 祥尚, 糸野 文洋, 櫻庭 健年, 田口 研治, 田原 康之: ソフトウェア科学基礎, トップエスイー基礎講座 1, 近代科学社 (2008)
- [田原 06] 田原 康之, 長野 伸一, 吉岡 信和, 本位田 真一: ゴール指向要求工学の体系化のための共通用語, 情報処理学会論文誌, Vol. 47, No. 2, pp. 596-608 (2006)
- [玉井 04] 玉井 哲雄: ソフトウェア工学の基礎, 岩波書店 (2004)
- [多屋 07] 多屋 優人, 室伏 俊明: 評価値に基づく相互作用を考慮したブートストラップ型主観的評価モデル, 知能と情報(知能情報ファジィ学会誌), Vol. 19, No. 1, pp. 47-56 (2007)
- [土屋 12] 土屋 達弘: 充足可能性判定を利用したモデル検査, コンピュータソフトウェア, Vol. 29, No. 1, pp. 19-29 (2012)
- [妻木 11] 妻木 俊彦: アダプティブ要求分析, 電子情報通信学会技術研究報告(信学技法), KBSE2011-27, Vol. 111, No. 211, pp. 19-24 (2011)
- [妻木 12] 妻木 俊彦: 要求工学: 現実と仮想をつなぐために, コンピュータソフトウェア, Vol. 29, No. 2, pp. 43-64 (2012)
- [寺野 87] 寺野 寿郎, 浅居 喜代治, 菅野 道夫(共編): ファジィシステム入門, オーム社 (1987)
- [出口 09] 出口 弘, 木嶋 恭一(編): エージェントベースの社会システム科学宣言—地球社会のリベラルアーツをめざして, 勁草書房 (2009)
- [戸内 97] 戸内 順一: 図解エキスパートシステム入門[新版], 日本理工出版会 (1997)
- [刀根 91] 刀根 薫: [増補] オペレーションズ・リサーチ読本, 日本評論社 (1991)
- [中島 12] 中島 震: 形式手法入門—ロジックによるソフトウェア設計, オーム社 (2012)
- [南波 09] 南波 幸雄: 企業情報システムアーキテクチャ, 翔泳社 (2009)
- [西崎 02] 西崎 一郎, 坂和 正敏: ファジィゲーム, 経営の科学オペレーションズ・リサーチ, Vol. 47, No. 5, pp. 302-308 (2002)

- [半田 12] 半田 久志：探索に基づいたソフトウェア工学：SBSE—ソフトウェア工学におけるメタヒューリスティクスの援用， 知能と情報（日本知能情報ファジィ学会誌）， Vol. 24， No. 6， pp. 224-229（2012）
- [廣松 98] 廣松 渉， 子安 宣邦， 三島 憲一， 宮本 久雄， 佐々木 力， 野家 啓一， 末木 文美士（編）：岩波哲学・思想事典， 岩波書店（1998）
- [藤本 12] 藤本 勝成：周辺性に基づくネットワーク形成とその安定性， 知能と情報（日本知能情報ファジィ学会誌）， Vol. 24， No. 4， pp. 901-908（2012）
- [古川 09] 古川 康一， 井上 克巳， 小林 郁夫， 諏訪 正樹：発想推論に基づく着眼点の発見， 人工知能学会第 23 回全国大会（JSAI'09）（2009）
- [室伏 09] 室伏 俊明：形式概念分析における並置を用いた分類表示—分岐分析への適用を例として， 第 25 回ファジィシステムシンポジウム， pp. 1-6（2009）
- [森村 99] 森村 英典， 刀根 薫， 伊理 正夫（監訳）：経営科学 OR 用語大辞典， 朝倉書店（1999）
- [山崎 06] 山崎 孝夫， 佐伯 元司：AHP による要求代替案選択手法， 電子情報通信学会技術研究報告（信学技法）， SS2005-83， Vol. 105， No. 597， pp. 1-6（2006）
- [山本 07] 山本 修一郎：～ゴール指向による!!～—システム要求管理技法， ソフト・リサーチ・センター（2007）
- [要求工学ワーキンググループ 07] 要求工学ワーキンググループ：要求定義で困ってませんか？—要求仕様書の品質に関する研究成果報告（2007）， <http://www.selab.is.ritsumei.ac.jp/~ohnishi/RE/rewg-tr1v2.pdf>