

ゴール指向要求分析に基づく
ビジネスプロセスの構築と検証に関する研究

堀田 大貴

電気通信大学大学院情報システム学研究科
博士（工学）の学位申請論文

2017年3月

ゴール指向要求分析に基づく
ビジネスプロセスの構築と検証に関する研究

博士論文審査委員会

主査	大須賀 昭彦	教授
委員	田中 健次	教授
委員	古賀 久志	准教授
委員	田原 康之	准教授
委員	石川 冬樹	客員准教授

著作権所有者

堀田 大貴

2017

Business Process Construction and verification Methods Based on Goal Oriented Requirements Analysis

Hiroki Horita

Abstract

Information systems are used in various companies and government agencies for supporting business. In this situation, it is needed to cooperate with the development of information systems and design of business processes. To design business process to achieve business goals of the organization is important, and it is needed to support the business process using information systems that requirements of stakeholders are reflected. Using goal models and business process models is valuable methods for development of information systems and design of business processes. However, environments are continuously changed by various reasons, therefore, the development of information systems and design of business processes are also continuously conducted for dealing with the environmental change. Moreover, it is difficult to define requirements of information systems and business processes in the complex and changing environments.

In order to cope with the above-mentioned problems, it is needed to support verification methods for detecting environmental changes and support design of business processes that can achieve business goals. Existing researches dealt with these problems, but these are still difficult challenges.

In this research, we dealt with these challenges and proposed following two methods: 1) Transformation method for KAOS goal model to business process model. 2) Verification support method for business process execution logs using decision tree. These methods can support design of business process and detect problems of executed business processes. We evaluated these methods conducting case studies using london ambulance service, telephone repair process and etc.

ゴール指向要求分析に基づくビジネスプロセスの構築と検証に関する研究

堀田 大貴

概要

情報システムは様々な企業や官公庁で利用されており、業務を支援している。このような状況では、実際の業務において真に有用な情報システムを開発するためには、情報システムの開発とビジネスプロセスの設計をそれぞれ独立して行うのではなく、組織の目標を達成するためのビジネスプロセスを設計し、それに合わせてビジネスプロセスの実行を効率的に支援するための情報システムを構築する必要がある。これらの設計・構築は要求を体系的・論理的に記述できるゴールモデルや、ビジネスプロセスの流れを記述できるビジネスプロセスモデルを用いることで、効果的に行うことができる。しかし、設計時において前提としていた組織を取り巻く環境は法律の改正や市場の変化等の理由によって変化するため、情報システムやビジネスプロセスは1度構築するだけでは十分ではなく、継続的に現環境において適切なものとなっているのかを検証し、不適切であれば改善する必要がある。また、このように複雑で変化する環境においては、情報システムやビジネスプロセスに求められる要件定義を行うことは難しい。

上記のような問題に対処するためには、環境変化が発生しているか確認するために、情報システムの実行ログが望ましい性質を満たしているか検証する技術や、組織の目標やビジネスプロセスに関するモデルを効率的に構築する技術が必要であり、研究が行われているが依然困難である。既存研究においては、実行ログの分析手法については、一般的に時相論理によって成り立つべき性質や成り立つべきでない性質を記述して検証を行うが、時相論理の記述は数理論理学の知識が不足している者やドメイン知識が不足している場合においては、正確に記述することが難しいという問題がある。また、モデルの構築については、組織を取り巻く様々な側面を記述した複数のモデルの整合性がとれた状態で構築する手法が不十分である。

本研究で提案するアプローチはこれらの課題の解決を目指し、以下の2つの内容に取り

組んだ. : (1) ゴール指向要求分析手法 KAOS によるゴールモデルからビジネスプロセスモデルを導出する手法, (2) 決定木を利用したビジネスプロセス実行ログの検証支援手法. これらを用いることで, 要求を的確にビジネスプロセスに反映すること, 実行されたビジネスプロセスの問題点を把握することができる. これらの提案手法はロンドンにおける救急車配備システムや電話の修理プロセス等を題材にケーススタディを行いそれぞれ2つの提案手法について評価し, 有効性を確認できた.

目次

第1章	序論	1
1.1	本研究の背景	1
1.2	本研究の目的と貢献	4
1.2.1	目的	4
1.2.2	貢献	5
1.3	本論文の構成	6
第2章	背景	9
2.1	ゴールモデル	9
2.1.1	KAOS	9
2.1.2	リファインメントパターン	11
2.2	ビジネスプロセスマネジメント	14
2.2.1	ビジネスプロセスモデル	16
2.2.2	プロセスマイニング	16
2.2.3	ビジネスプロセスの検証	17
2.2.4	データマイニングの利用	21
第3章	リファインメントパターンを利用したKAOSゴールモデルからBPMNモデルへの変換	23
3.1	はじめに	23
3.2	提案手法	25
3.2.1	提案手法適用対象	25
3.2.2	提案手法概要	25

3.2.3	変換アルゴリズム	27
3.2.4	各リファインメントパターンを利用した変換	27
3.3	ケーススタディ	29
3.3.1	LAS 概要	30
3.3.2	LAS におけるケーススタディ	30
3.3.3	bCMS 概要	32
3.3.4	bCMS に関するケーススタディ	33
3.3.5	卸—メーカー間における取引業務概要	34
3.3.6	卸—メーカー間における取引業務に関するケーススタディ	35
3.3.7	ケーススタディにおける考察	36
3.4	考察	37
3.4.1	生成できる BPMN モデルの特徴	37
3.4.2	ゴールとアクティビティの同等性	37
3.4.3	変換ルールの妥当性	38
3.4.4	OR 分解の選択	39
3.4.5	提案手法適用範囲	39
3.5	関連研究	40
3.5.1	ゴールモデルからビジネスプロセスモデルへの変換	41
3.5.2	ビジネスプロセスモデルからゴールモデルへの変換	41
3.5.3	ゴールモデルとビジネスプロセスモデルの関連付け	42
3.5.4	リファインメントパターンの活用	42
3.5.5	企業における取り組み	43
3.6	まとめと今後の課題	43
第 4 章	機械学習手法を利用したビジネスプロセス実行ログの検証支援手法	45
4.1	はじめに	45
4.2	提案手法	47
4.2.1	イベント実行順序関係に着目した特徴量抽出と学習	47
4.2.2	LTL 式の生成	51

4.3	評価	52
4.3.1	電話修理プロセス	53
4.3.2	構築された決定木と論理式	54
4.3.3	予測結果と検証結果の比較	54
4.4	考察	56
4.5	関連研究	57
4.6	まとめと今後の課題	58
第5章	関連研究	59
5.1	環境変化への対応方法に関する研究	59
5.1.1	ゴールモデルの修正方法に関する研究	59
5.1.2	ビジネスプロセスモデルの修正方法に関する研究	60
5.1.3	実行時における適応	60
5.2	超上流工程を対象とした手法	62
第6章	結論	63
6.1	本研究のまとめ	63
6.1.1	課題・目的	63
6.1.2	提案内容	64
6.1.3	評価内容	64
6.2	今後の課題	65
6.2.1	モデル修正手法	65
	謝辞	67
	参考文献	69
	付録	79
.1	付録: bCMS のゴールモデル	79
.2	付録: 卸メーカー間取引業務のゴールモデル	79

目次

2.1	KAOSモデルからBPMNモデルへの変換パターン	11
2.2	ビジネスプロセスライフサイクル [2]	15
2.3	BPMNモデル記述例	17
2.4	プロセスマイニング概要 [2]	18
3.1	提案手法概要	26
3.2	変換アルゴリズム	27
3.3	LASに関するゴールモデル	32
3.4	LASに関するゴールモデルを提案手法により変換したBPMNモデル	32
3.5	変換結果まとめ	35
3.6	卸メーカー取引業務プロセス	36
4.1	プロセスマイニングの概要 [3]	46
4.2	提案手法の概観	48
4.3	トレースのイベント実行順序関係に基づいて構築した決定木	51
4.4	決定木の例	52
4.5	各ステップにおける比較と他手法との比較	56
4.6	決定木による予測と他分類器による予測との比較	56
1	bCMSのゴールモデル	80
2	卸メーカー間取引業務のゴールモデル	81

表 目 次

2.1	シンプルなイベントログの例	21
2.2	制約の例	21
4.1	特徴量としてイベント順序関係を持ったトレースの例	49
4.2	人手で記述した論理式を用いた検証結果	55
4.3	決定木による予測結果	55
4.4	新たに生成した論理式を用いた検証結果	55

第1章 序論

本章では，本研究の背景を述べた後，本論文の目的と貢献を説明する．その後，本論文の構成について述べる．

1.1 本研究の背景

近年，情報システムは企業や官公庁等の様々な組織において利用されており，業務を支援する役割を担っている．情報技術を活用する場面は増え続けており，今後もこのような傾向が継続することが予想される．このような状況では，情報システムの開発とビジネスプロセスの設計はそれぞれ独立して構築するのではなく，組織の目標を達成するためのビジネスプロセスを設計し，それに合わせてビジネスプロセスの実行を効率的に支援するための情報システムを構築する必要がある．情報システムとビジネスプロセスを合わせて検討することは近年重要視されてきており，ビジネスアナリシスの知識体系である BABOK[28] や，ビジネスプロセスの可視化や改善を支援する BPM ツールの普及が進んでいる．

BABOK や BPM ツールの普及が進行している背景には，対処すべき複数の理由がある．その理由としてまず挙げられるのは組織を取り巻く急激な環境変化である．ここでの環境変化とは，法律改正，市場動向の変化，顧客行動の変化，戦略的転換など様々なものを指す．組織が活動を通してビジネスゴールを達成し続けるためには，環境変化が発生する度にビジネスプロセスや情報システムを修正し，コスト削減や効率化することが求められる．また，そのような複雑で変化する状況においては，情報システムやビジネスプロセスに求められる要件定義を行うことは難しい．組織が行うべき活動をビジネスプロセスとして定義し，それを支援する情報システムを構築するための要件定義が必要であるが，複雑な状況においては，要件の抜けや漏れが発生しやすく，問題が開発の後工程において発生した場合には前工程に戻り，修正する必要があるため，時間や資金的なコストが超過するのみ

ならず，構築したビジネスプロセスや情報システムがステークホルダにとって望ましくないものになる可能性がある．そのため，ビジネスプロセスや情報システムを効率的に構築するための手段が求められている．

環境変化に対応するための要件を把握するためには，人手で経営者，管理者や現場での業務に従事する者に対してインタビュー等を行い，現在の状況を把握する (as-is 分析) することから始めるのが一般的である．しかし，上記のように組織を取り巻く状況が複雑化することによって，正確かつ迅速に状況を把握することは困難である．また，インタビュー等を用いた分析を行うと，作業者の主観が入りやすい，例外的な業務処理は問題が発生しやすいにも関わらず，インタビューなどでは検出しにくい，などの危険もはらんでいる [68]．そのため，組織が要件を定義し，環境変化に対応していくためには，客観的な分析手法が必要となる．この役割を担う重要な要素がモデルとデータである．以下において本研究におけるモデルとデータについてそれぞれ説明する．

モデルとは，対象世界の概念構造を描くことである．情報システムによって経営活動の支援を行うためには，経営者や現場の担当者が認識しているそれぞれの対象世界の概念構造を情報システムにおいて再現し，操作可能にする必要がある [67]．情報システム開発において用いられる UML[49] やビジネスプロセスの標準である BPMN[51] をはじめとして，様々なモデルが利用されている．何らかの対象をモデル化することは，自然言語よりもより厳密に記述できるという利点や，グラフィカルに表現することでステークホルダとのコミュニケーションを促進できるなどの効果がある [61]．本研究ではモデルとしてゴールモデル [61] とビジネスプロセスモデル [64] を利用する．ゴールモデルとは要求をゴールとして系統的・論理的に記述できるモデルである．抽象的なゴールをより具体的なゴールへ，親ゴールの達成には子ゴールすべてが達成される必要がある AND 分解や，親ゴールの達成には子ゴールのいずれかが達成されることが条件となる OR 分解を通して分解される．ゴールモデルは，従来からよく使用されている情報システムがユーザと対話するシナリオを記述する UML ユースケース図や，状態遷移について記述する UML ステートマシン図等とは異なり，目的について記述できるという点が異なっており，近年注目されている．情報システムを開発するにあたっては，上流工程において，どのようなものが本当に求められているのかをはっきりと記述せずに，開発を進めてしまったがために，後の工程におい

て、要件の変更が発生したり、それに起因した開発の遅れによる予算の超過等大きな問題が起こっている。それを解決する手段として、ゴールモデルを利用することは有効な手段であると考えられている。NTT データではゴールモデルは Model-Oriented methodology for Your Awareness (MOYA)[72] として要求定義によるガイドラインとして実際に用いられており、いくつものプロジェクトにおいて有効性が確かめられている。

一方で、ビジネスプロセスモデルは組織におけるビジネスゴールを達成するために行われるべきイベントの流れや条件分岐について記述するものである。作業手順を記したモデルであるともいえる。ビジネスプロセスモデルを用いることは、業務の大まかな流れをビジネスアナリストが記述することから始まり、ビジネスプロセスを情報システムとして実行するエンジニアや、ビジネスプロセスの管理を行う担当者をつなぐ役割を担う。

モデルを用いることで効果的な分析を行うことができる一方で、モデルを用いた分析には限界がある。それは、モデルが現実を反映しており、正確に構築されていなければ、モデルに立脚した分析が不適切なものになってしまうということである。上記のように、組織を取り巻く環境は変化するため、その中で正確に状況を捉えたモデルを構築することは難しい。このような問題を解決するためには、モデル構築を支援する方法論が必要であるとともに、現実に行われたものを記録したデータを用いて現実をとらえることが有効である。

データとは、ここでは情報システムを実行した結果として記録されたものである。情報技術の発展により、大量の情報を記録するコストが下がり、データ分析を行うことで何らかの知見を得る試みが活発になってきている。また、データを利用することで環境変化に応じて情報システムやビジネスプロセスを修正するために必要な客観的な情報を得ることができる。本研究において用いるデータとは情報システムによって記録されるビジネスプロセスの実行ログである。このようなデータの分析はプロセスマイニング [1] と呼ばれ、ビジネスプロセスの実行ログを分析することでプロセスを発見、監視、改善することができる [2]。プロセスマイニング技術として、富士通は現状業務・システムのプロセスを可視化する技術 (BPM-E : Business Process Management by Evidence) を開発した。BPM-E を用いることで詳細な業務知識や膨大なヒアリングを最小限に抑え、システムに手を入れることなく、データベースの情報から IT 化された業務プロセスを可視化することが可能となった [68]。また、アイントホーフェン工科大学の Aalst らによって開発された LTL checker は

ビジネスプロセスにおいて観測された振る舞いが期待された振る舞いに適合しているか検証することができる [3]. LTL checker を用いることでモデル上ではなく、実際に行われたビジネスアクティビティを監視することで組織を取り巻く環境変化を検知することができる. 4-eyes principle (複数人による確認体制) を確認することは, LTL checker を用いることで検証できる一例である. これは2つのアクティビティが実行される時, 同一の人物が両方のアクティビティを実行してはならないという原則である. 例えば, 備品購入申請を行うというアクティビティとそれを承認するアクティビティは同一の人物によって行われてはならないことを指す.

このように, 要求分析・定義のためにモデルを用いることと, データを記録して分析することで, 実際に実行されたものを把握することは, 情報システムやビジネスプロセスについて正確な要件定義を行い, 環境変化への対応を促進する. しかし, モデルを用いた分析を行う上で, 正確なモデルが構築されていることは非常に重要であるが, 本研究で用いるゴールモデルやビジネスプロセスモデルはそれぞれ別々の役目を担っており, 整合性が保たれた両モデルを構築することは難しい. ビジネスプロセスは組織のゴールを達成するものであると仮定されるが [34], ゴールモデルによって記述された要求を満たすことができるビジネスプロセスモデルを構築する方法として依然確立された方法は存在しない. また, 記録したデータを分析する際に, LTL checker を用いることは有効な手段だが, 分析対象のドメイン (保険申請等) についてあまり詳しくない者や, 数理論理学の知識に乏しい者が LTL によって検証したい性質を正確に検証することは難しい. そのため, これらを支援する手段が必要である.

1.2 本研究の目的と貢献

本節では, 本研究の目的と貢献について述べる.

1.2.1 目的

本研究の大きな目的は以下の2点である.

1. 要求をビジネスプロセスへ反映する

2. 環境変化を検知する

目標1に関する内容は、本論文の3章に対応している。組織のビジネスプロセスにおいて、ビジネスプロセスが組織のゴールを達成することができるのか確認するためには、要求とビジネスプロセスの関係性を明確に記述する必要がある。これを支援するための手段が必要である。

目標2に関する内容は、本論文の4章に対応している。組織がビジネスゴールを達成し続けるためには、組織が環境に応じて戦略やビジネスプロセスを変化させる必要があり、そのためには、まず環境変化を検知する必要がある。これを支援するための手段が必要である。

1.2.2 貢献

1.2.1節において説明した2つの目的へ対応するための本研究における貢献は以下の2つである。

1. リファインメントパターンに基づきゴールモデルによって記述した要求をビジネスプロセスモデルへ反映する手法
2. ビジネスプロセス実行ログにおけるイベントの実行順序関係に基づいて決定木学習を行いユーザの意図を反映した論理式を生成する手法

目的1については、1.1節において説明したように、モデルを用いて組織のビジネスゴールやビジネスプロセスを記述することは、厳密性や理解しやすさなどの点において有効である。しかし、誰しもが記述したい内容をモデルを用いて正確に表現することは難しい。構築したモデルの記述に抜け・漏れや矛盾があるとモデルを用いた分析が不適切なものとなる。また、モデルはそれぞれ異なる役割を持っているため、複数のモデルを用いる場合はモデル間の整合性が取れていることが求められる。本研究では、組織のゴールを論理的・体系的に記述できるゴールモデルと、イベントの流れや条件分岐を記述できるビジネスプロセスモデルに着目し、ゴールモデルによって記述した組織の要求をビジネスプロセスモデルへ反映する手法を提案している。そのために、本研究ではゴール分解ガイドラインで

あるリファインメントパターンに着目している。リファインメントパターンを用いることで、ゴール分解の難しさを軽減するとともに、厳密にゴールを記述することができる。また、リファインメントパターンを用いて分解されたゴール間の関係性に基づいてビジネスプロセスモデルにおけるイベント間の関係性（イベントの実行順序や分岐）と関連付けるルールと変換アルゴリズムを提案している。そうすることで、ゴールモデルにおいて記述された情報を用いてビジネスプロセスモデルにおいて行われる処理の位置づけを明確化するとともに詳細なモデルを構築できる。複数の題材を用いてケーススタディを行い有効性を示した。

目的2については、1.1節において説明したように、ビジネスプロセスの実行に関するデータを分析することで、LTL checkerのような手法を用いて検証を行うことで、組織において実行された物事を把握することができる。しかし、LTL checkerを用いた検証を行うためには、LTL（線形時相論理）を用いてビジネスプロセスにおいて成り立つべき性質を記述しなければならないが、ドメイン知識や数理論理学の知識が乏しいユーザーにとっては意図を正確に論理式として記述することが難しいという問題がある。そのような場合は検証結果がユーザーの意図に基づいていない不適切なものとなり、間違った意思決定につながる可能性がある。そのため、ユーザーの意図を論理式に反映するための支援があることが望ましい。本研究では、ビジネスプロセス実行ログにおけるイベントの実行順序関係に基づき、決定木学習を行うことで、十分にユーザーの意図を反映していない論理式よりも、より正確な論理式を生成する手法を提案している。そうすることで、ビジネスプロセスにおいて実行されていることを正確に把握することができる。この提案手法について電話修理プロセスの実行ログを用いて、新たな論理式を生成し有効性を確認した。

1.3 本論文の構成

本論文の構成は以下のとおりである。まず、2章において3章、4章における内容の背景や関連する技術としてゴールモデルやビジネスプロセスマネジメントについて述べる。2.1節では、本研究で主に用いているゴール指向要求分析法であるKAOSや、3章における手法において核として用いるリファインメントパターンについて述べる。2.2節では、ビジネスプロセスマネジメントについて3章においてゴールモデルから導出されるビジネスプロ

セスモデルや、4章に関する内容であるプロセスマイニングについて述べる。

続いて、3章、4章において、提案手法について述べる。まず、3章では、ゴール指向要求分析法である KAOS に基づいて記述された要求をビジネスプロセスモデルへ変換する手法について述べる。本手法はゴールモデルの分解ガイドラインであるリファインメントパターンを用いており、各パターンごとにゴールモデルからビジネスプロセスモデルへ変換するためのルールと、それらを用いて変換するためのアルゴリズムを説明する。続く4章では、LTL checker を用いてビジネスプロセス実行ログを検証する際に、論理式がユーザーの意図を正確に反映していない場合に機械学習手法である決定木を用いることで、より正しい論理式を生成し、正確な検証結果を得るための手法について述べる。各ログにおいて記録されたイベントの実行順序関係に着目して決定木を用いて学習し、さらに構築された決定木の構造に基づいて新たな論理式を生成する方法について説明する。

5章では、情報システムやビジネスプロセスが様々な場面において環境変化に対応するために必要だが、本研究では扱えなかった部分に関する研究について述べる。最後に、6章でまとめと今後の課題、展望について述べる。

第2章 背景

本章では，本論文で提案する内容に関連した背景について説明する．まず，2.1節において，ゴールモデルについて説明する．続いて，2.2節で，ビジネスプロセスマネジメントについて説明する．

2.1 ゴールモデル

本節では，ゴールモデルおよび，ゴールモデルにおけるゴール分解のガイドラインであるリファインメントパターンについて説明する．

2.1.1 KAOS

ゴール指向要求工学は要求工学の研究において潜在するシステム要求の動機を理解や適切な問題に対処するために，適切なシステムができているか確かめる手段として大きな注目を浴びている [27]．

ゴール指向要求工学では，ステークホルダの要求をゴールとして捉え，より具体的なゴールへと分解していくことで要求を導出する．正しい環境過程において，正しいソフトウェア要求を得ることは正しいソフトウェアを開発するための重要な条件である [60]．ゴールモデルには KAOS や i star [65] など様々な種類がある．KAOS は代表的なゴール指向要求分析法の1つであり，ゴールモデル，オペレーションモデル，エージェントモデル，オブジェクトモデルの4モデルを活用し，要求を実現するための操作，操作を行うエージェント，入出力されるオブジェクトを記述し要求分析を行う．本研究においてはゴールモデルのみを取り扱うため，ゴールモデルに関してのみ以下に記述する．

ゴールモデルの構築方法には親ゴールに HOW と問うことによって，どのような手段（子ゴールの組み合わせ）によって親ゴールを達成するのか確認する方法と子ゴールに対して

WHY と問うことによって子ゴールを何故達成しなければならないのか確認し親ゴールを導出する方法がある [62]. 本研究ではリファインメントパターンを KAOS ゴールモデルのゴール分解に利用しており, これは HOW と問うことによって手段となる子ゴールを導出する方法である.

ゴール分解には2種類ある. AND 分解は親ゴールを達成するためにはその子ゴールがすべて達成される必要がある分解法である. AND 分解に求められる性質として, 完全性, 整合性, 最小性がある.

完全性は以下の式によって表される.

$$\{G_1, G_2, \dots, G_m, DOM\} \models G \quad (2.1)$$

全ての子ゴール (G_1, G_2, \dots, G_m) と, 全てのドメインプロパティやドメイン仮定 (DOM) が満たされると, 親ゴール (G) も満たされる. ここで, ドメインプロパティとは, 環境において常に成り立つことが分かっている性質である. ドメイン仮定とは, 環境において成り立つことが望ましい性質である.

整合性は以下の式によって表される.

$$\{G_1, G_2, \dots, G_m, DOM\} \not\models false \quad (2.2)$$

子ゴール, ドメインプロパティ, ドメイン仮定は互いに矛盾しないことを表す.

最小性は以下の式によって表される.

$$\{G_1, \dots, G_{j-1}, G_{j+1}, \dots, G_m, DOM\} \not\models G \quad (2.3)$$

子ゴールのうち一つでも欠落している場合, 親ゴールの満足が保証されないことを表す. リファインメントパターンを用いてゴール分解を行うことはゴールモデルがこれらの性質を持つことに繋がる.

一方, OR 分解はその子ゴールのいずれか1つが達成されれば親ゴールが達成されるような分解法である.

ゴールを達成するためにはエージェントの協調が必要である. KAOS におけるエージェントとはソフトウェアコンポーネントやデバイス, 人間などである. ゴールとエージェントを関連付けて記述することによってゴール達成責任を持つエージェントを明確化する.

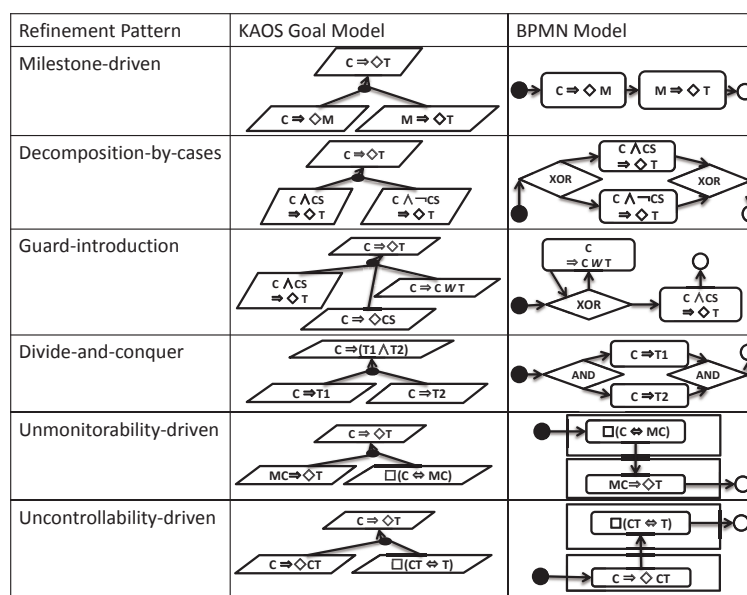


図 2.1: KAOS モデルから BPMN モデルへの変換パターン

2.1.2 リファインメントパターン

適切にゴール分解を行うことは難しい事が知られている。何らかの支援なくゴール分解を行うことは一般的に不十分になりがちであり、矛盾を含むことさえある [62]。ゴール分解を適切に行う手段としてフォーマルメソッドの利用が考えられる。しかし、フォーマルメソッドの利用は難しくコストが高い [18]。このような課題を解決するために KAOS では 6 種類のリファインメントパターンを用いることで、より適切なゴール分解を行うことができる。リファインメントパターンは [13],[36], [60] において時相論理を用いて形式的に記されている。各パターンは正しく、完全であることが証明されている [36]。パターンを再利用することはパターンの正しさ、完全さの再利用を伴う。そのためリファインメントパターン利用者は数学的な側面をあまり意識することなく適切なゴール分解を行うことができる [18]。以下において図 2.1 を用いつつ各リファインメントパターンを説明する。

Milestone-driven refinement pattern

Milestone-driven Refinement Pattern は、前提条件 C が成り立つ場合に、最終的にある条件 T を満足させる、すなわち時相論理では「 $(C \Rightarrow \Diamond T)$ 」と表されるゴールに対し、マイ

ルストーン条件 M_1, \dots, M_n を設定して、これらを順に満足させる、すなわち「 $C \Rightarrow \diamond M_1$ 」, 「 $M_1 \Rightarrow \diamond M_2$ 」, \dots , 「 $M_{n-1} \Rightarrow \diamond M_n$ 」, 「 $M_n \Rightarrow \diamond T$ 」に AND 分解するリファインメントパターンである。親ゴール達成（ターゲット条件への到達）のためのマイルストーン条件がある場合に用いる。図 2.1 は時相論理によって記述されたマイルストーン条件が 1 つの場合の Milestone-driven refinement pattern によるゴール分解である。

Decomposition-by-cases pattern

Decomposition-by-cases pattern はターゲット状態に到達する異なるケースがある場合に用いられるリファインメントパターンである。これらのケースは互いに素であり、状態空間全体を網羅している。Decomposition-by-cases pattern によって親ゴールに対し分解条件 CS が設定され、親ゴールを分解した子ゴールがそれぞれ「 $CS \Rightarrow \diamond T$ 」, 「 $\neg CS \Rightarrow \diamond T$ 」となる。分解条件 CS が成り立つ場合も、成り立たない場合もターゲット条件を達成できることがわかる。これらの子ゴールは互いに素であるためどちらかだけが達成される。

Guard-introduction pattern

Guard-introduction pattern は、ターゲット状態に到達するためにガード条件が必要な異なるケースに場合分けされる場合に用いられるリファインメントパターンである。Guard-introduction pattern によって、親ゴールはガード条件達成ゴール、ガード条件ゴール、ガード条件未達成ゴールに分解される。

図 2.1 に Guard-introduction pattern によるゴール分解を時相論理を用いて記述している。「 $C \wedge CS \Rightarrow \diamond T$ 」は current condition(C) とガード条件 (CS) が成り立つときそのうちターゲット条件 T が成り立つことを表している。「 $C \Rightarrow \diamond CS$ 」は current condition(C) が成り立つときそのうちガード条件 (CS) が成り立つことを表している。「 $C \Rightarrow C \wedge \neg T$ 」は T が成り立つまでは C が成り立ち続けることを表している。つまり、T が成り立っていなくても C が成り立っていればそのうち CS が成り立って結果的に T も成り立つことを意味する。

Divide-and-conquer pattern

Divide-and-conquer pattern は、結合しているゴールがより小さいゴールに AND 分解できる場合に用いられるリファインメントパターンである。Divide-and-conquer pattern によって結合した状態にある親ゴールを子ゴールに分解される。これらはどちらも達成されなければいけないゴールである。

図 2.1 に時相論理によって Divide-and-conquer pattern によるゴール分解を記述している。親ゴールである「 $C \Rightarrow (T_1 \wedge T_2)$ 」が達成されるためには 1 つ目のターゲット条件 (T_1) と 2 つ目のターゲット条件 (T_2) が両方成り立つ必要がある。このゴールがそれぞれ 1 つ目のターゲット条件 (T_1) と 2 つ目のターゲット条件 (T_2) が成り立つためのゴールである。「 $C \Rightarrow T_1$ 」と「 $C \Rightarrow T_2$ 」が両方達成されれば、親ゴールである「 $C \Rightarrow (T_1 \wedge T_2)$ 」も達成される。

Unmonitorability-driven refinement pattern

Unmonitorability-driven Refinement pattern は、ゴールの達成を担当しているエージェントでは、監視すべき状態を監視することができない場合に、そのような状況を解決するために用いられるリファインメントパターンである。親ゴールの達成を担当しているエージェントがゴールを達成するための条件を監視できない場合、ゴール達成の部分的な責任が監視できるエージェントに割り振られる。Unmonitorability-driven refinement pattern によって親ゴールが子ゴール、期待に分解される。期待は Software-to-be の環境におけるエージェントの責任のもとにあるゴールである。その場合環境エージェント（センサエージェント）が期待、ソフトウェアエージェントが子ゴールの達成責任を負う。このようにゴールの責任が複数のエージェントに分割される。モニタリングした情報を活用してソフトウェアが動作することになる。

図 2.1 に時相論理によって Unmonitorability-driven refinement pattern によるゴール分解を記述している。親ゴールである「 $C \Rightarrow \diamond T$ 」が達成されるためには子ゴールである、「 $\square(C \Leftrightarrow MC)$ 」と「 $MC \Rightarrow \diamond T$ 」が両方達成される必要がある。ここで MC とは監視すべき対象を監視可能である状態を表す。C の成否と MC の成否が一致すれば ($\square(C \Leftrightarrow MC)$)、C が成り立てば MC も成り立つので、そのうち T が達成され、親ゴールである「 $C \Rightarrow \diamond T$ 」が達成される。

Uncontrollability-driven refinement pattern

Uncontrollability-driven refinement pattern は親ゴールの達成を担当しているエージェントでは制御すべき状態を制御することができない場合において、そのような状況を解決することを狙いとしたパターンである。親ゴールを Uncontrollability-driven refinement pattern によって分解した場合、ソフトウェアエージェントにより達成される子ゴールと、アクチュエータエージェント等の環境エージェントによって達成される期待になる。アクチュエータエージェントが動作するためにはソフトウェアエージェントによる指示が必要である。

図 2.1 に時相論理によって Uncontrollability-driven refinement pattern によるゴール分解を記述している。親ゴールである「 $C \Rightarrow \diamond T$ 」が達成されるためには子ゴールである、「 $\square(CT \Leftrightarrow T)$ 」と「 $C \Rightarrow \diamond CT$ 」が両方達成される必要がある。ここで CT はコントロールすべき対象をコントロール可能である状態を表す。 C が達成されたとき、そのうち CT が達成され、 CT の成否と T の成否が一致すれば ($\square(CT \Leftrightarrow T)$)、親ゴールである「 $C \Rightarrow \diamond T$ 」が達成される。

2.2 ビジネスプロセスマネジメント

ビジネスプロセスとは、実行されるイベントの集まりであり、ビジネスゴールを実現するために組織・技術的環境を強調させるために必要となる [64]。また、ビジネスプロセスをビジネスプロセスモデル (2.2.1 にて説明) という形でドキュメント化することは非常に重要であり、ビジネスプロセスにおける知見を得るための効果的な手段である。更に近年では、米国におけるサーベンス・オクスリー法 (SOX 法) のような法律によってビジネスプロセスをモデル化することが義務付けられる可能性もあるといわれている。

上記のような法律改正や顧客行動の変化、市場の変化、内部統制など様々な要因で組織を取り巻く環境は変化する。そのような状況下において、ビジネスゴールを達成し続けるためには、環境変化が発生した場合にはその都度対応する必要がある。安定して対応するためには、そのための方法論があることが望ましい。それはビジネスプロセスマネジメントと呼ばれ、ビジネスプロセスに「分析」「(再)設計」「実装」「(再)設定」「実行」「適応」「診断」というマネジメントサイクルを回し、継続的に経営・業務改善プロセスを行う

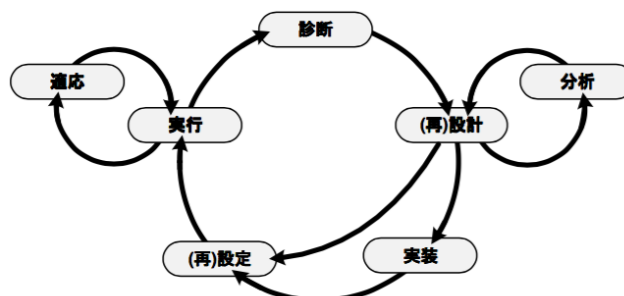


図 2.2: ビジネスプロセスライフサイクル [2]

ものである。

上記のようにビジネスプロセスを構築し、管理することを目的としてビジネスプロセスライフサイクルについて説明する。図 2.3 はビジネスプロセスライフサイクルを表しており、このようなプロセスによってビジネスプロセスは管理される。まず、分析工程では、組織を取り巻く環境などを分析し、どのようなビジネスゴールを達成するためにどのようなビジネスプロセスを構築すべきなのかを検証する。次に設計工程では分析工程において検討したビジネスプロセスを BPM N などのビジネスプロセスモデルへモデル化し形式的に記述する。実装工程では、設計工程において構築したビジネスプロセスモデルを中心に情報システムとして実装する。または、設計工程において、サービス指向アーキテクチャ (SOA) のような技術を用いて情報システムを小さいサービス単位のコンポーネントを組み合わせることで構築する。実行工程では、構築した情報システムを用いてビジネスプロセスの実行を支援するとともに実行に関するデータを記録する。適応工程では、ビジネスプロセス実行時に発生する環境変化に対応してビジネスゴールを達成するために、柔軟にビジネスプロセスを変更する。そして、診断工程においては実行時に記録された情報を分析し、ビジネスプロセスが実行された情報が適切なものだったのか確認する。この結果を踏まえて、再設計工程において再度ビジネスプロセスモデルを適切なものへ修正する。

2.2.1 ビジネスプロセスモデル

ビジネスプロセスのモデリング技法はこれまでに多く提案されている。ペトリネットやUML アクティビティ図, BPMN など様々なものが利用されている。本研究では BPMN を用いるため, 本節では BPMN について記述する。BPMN(Business Process Model and Notation) [51] は, ビジネスプロセスを表記する OMG 標準化仕様の表記法である。BPMN は主に処理や分岐を表すフローノードとそれらの順序関係であるシーケンスフローによって構成される。アクティビティによってビジネスプロセスにおいて行われる処理を表し, シーケンスフローによってアクティビティが行われる順序を定める。更に, アクティビティが記述されるスイムレーンを分けることで, 処理を実行する主体ごとに行われるアクティビティを分けて記述できる。処理の分岐はゲートウェイによって記述する。AND Gateway は並列して行う処理を表し, XOR Gateway はいずれか1つの処理を実行することを表す。BPMN を用いればビジネスにおいていつ (when), 誰が (who), 何を (what) 行うのか記述することができる。一方で, なぜ (why) 処理を行うのか記述することはできない。この問題はなぜ (why) という側面を記述できるゴールモデルを用いることで, 対処することができる。本手法により, リファインメントパターンによりゴールの関係を明確化することで, BPMN モデルにおけるアクティビティが実行されるべき根拠を明確にできる。

図 2.3 は 3 章におけるケーススタディに用いている BPMN モデルを簡素化したものである。通報のあった場所へ救急車が到達するまでの流れを表している。まず, 「通報を受ける」というアクティビティが実行され, その後, 「現場へ向かう救急車を選択する」というアクティビティが実行される。アクティビティの間に記述されているシーケンスフローによって, 順序関係を表す。その後, XOR Gateway によって, 行うべき処理が分岐し, アクティビティである「ステーションにいる救急車が現場へ向かう」, 「路上にいる救急車が現場へ向かう」のいずれかが実行される。

2.2.2 プロセスマイニング

近年では, 組織において情報システムが多く利用され, 多くの情報が記録される。記録された情報を分析することで様々な知見を得ることができる。このような分析はビジネス

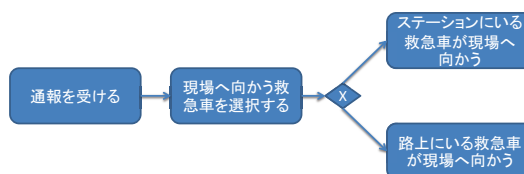


図 2.3: BPMN モデル記述例

インテリジェンス (BI) とも呼ばれ、多くのツールが利用されている。さらには、データ駆動型社会までもが求められている [73]。本研究において利用するデータはビジネスプロセスにおいて実行されたイベントに関するものであり、この分析はプロセスマイニングと呼ばれる。プロセスマイニング技術は情報システムにおいて記録されたイベントログと呼ばれるデータを分析し、プロセスを発見、監視、改善することに活用できる。図 2.4 はプロセスマイニングの概要である。左上に記述された世界は、組織の業務プロセスや業務に従事する人々、マシンなどの我々が存在する世界を表している。この世界は抽象化したものが左下のモデルである。モデル化することで着目する点を限定し、分析をしやすくできる。そして、ビジネスプロセスライフサイクルにおける実装工程のように、世界を抽象化したモデルについてソフトウェアシステムとしての仕様を定め、実装する。このソフトウェアシステムを業務に用いることで、実行されたイベントがデータとして記録される。これが右下に記述されたイベントログである。このイベントログを分析することで、ビジネスプロセスの as-is モデルを自動構築するプロセス発見や、イベントログとビジネスプロセスモデルの差異を分析する適合性検査、モデルにイベントログの内容を付加する強化などを行うことができる。

2.2.3 ビジネスプロセスの検証

ビジネスゴールを達成するためのビジネスプロセスの設計は難しい。妥当なビジネスプロセスが設計されているか確認するためには、ビジネスプロセスモデルを検証するという手段が有効である。ビジネスプロセスモデルの検証は 2.2 節において記述したビジネスプロセスライフサイクルにおける複数の工程において行われる。

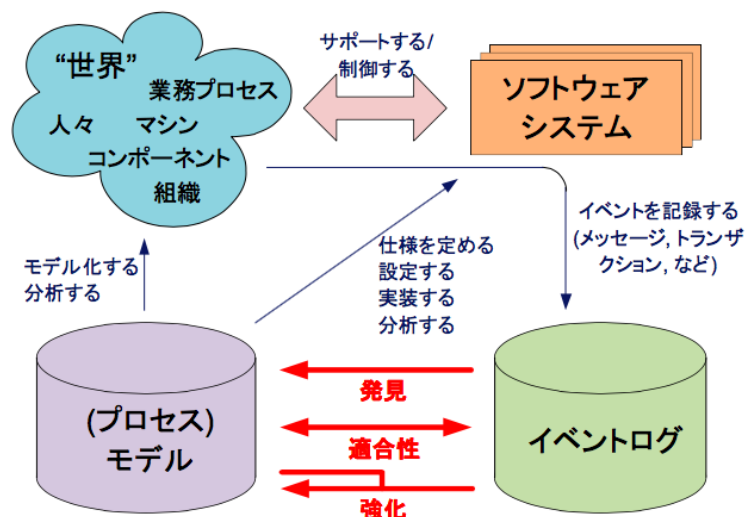


図 2.4: プロセスマイニング概要 [2]

1. 設計時における検証

設計時に行う検証は、設計したビジネスプロセスモデルが仕様を満たしているものなのか検証するために行う。モデル検査を行うことで、設計したモデルが望ましい性質を満たしたものであるのか自動的かつ網羅的に検証することができる。適切なモデル設計がなされなかった場合の問題点として、デッドロックとライブロックがある。デッドロックとは、システムの動作中の全実行単位に関し、各単位が他の単位のクリティカルセクション完了を待っている状態にあるために、どの単位もそれ以上実行不可能になることをいう [71]。ライブロックとは、デッドロックとは異なり、1つ以上の実行単位が、実行を継続しているが、その内容が他のクリティカルセクション完了を待つための処理であるため、実質的に動作が進行していない状況のことである [71]。

また、時間や資源に関する性質や、確率的な性質に関する検証手法も研究が行われている。綿引らは、ビジネスプロセスにとって重要な性質である時間及び資源に関する制約を付加した BPMN を、形式的な時間オートマトンモデルに変換し、モデル検査ツールを用いて性質を網羅的に検証する手法を提案した [74]。Mendt らは不確実な

情報を含んだビジネスプロセスモデルを検証するために、確率的モデル検査ツール PRISM を用いてビジネスプロセスモデルを検証する手法を提案した [43].

2. 実行時における検証

実行時に行う検証は、ビジネスプロセスにおける実行時における決定を支援する (オペレーショナルサポート) ために行われ、現在注目を浴びている [4]. ビジネスプロセスオペレーションにおいて、現在までに実行され、蓄積されたイベントログを分析し、あらかじめ定めたビジネスプロセスモデルやビジネスルールから逸脱した振る舞いが観測された場合にすぐさま検知し、対処法を推薦したり、将来起こりうることを予測したりすることができる。実行時における検証については、5.1.3 節においてより詳細に記述する。

3. 実行後 (診断時) における検証

診断時に行う検証は、ビジネスプロセス実行が何らかの性質やルールに一致しているか検証するものである [20]. 診断時における検証は大きく 2 つに分けられ、目的に応じて使い分ける。

(a) 適合性検査

Rozinat らはビジネスプロセスにおける振る舞いを記述したイベントログとコントロールフローを記述したビジネスプロセスモデルがどれくらい適合しているのかを表す適合性検査技術を開発した [56]. この技術によって、イベントログがモデルのどの部分との相違があるのか確認することができる。また、フィットネスを測定することで、与えられたモデルがイベントログに見られる挙動をどのくらい良く許容するのか判断する基準となる [2]. Rozinat らが適合性検査に用いるビジネスプロセスモデルとしてペトリネットを選択している一方で、Molka らはビジネスプロセスモデルの標準である BPMN モデルを用いるアプローチを提案している [44]. また、Leoni らはビジネスプロセスモデルとして Petri net with data (DPN-net) を利用することで、ビジネスプロセスのコントロールフローのみならず、リソースやデータを含めた複数の観点に着目した適合性検査手法を提案しており、検証できる対象を広げている [35].

(b) 特定の性質に関する検証

適合性検査では、ビジネスプロセスモデルとイベントログとの差分を主にコントロールフローに着目して検出することができるが、特定の性質に着目して検証を行うことは難しい。また、近年ではコントロールフロー以外も検証対象にした Leoni らの手法などが提案されているが、主に対象にしているのはコントロールフローであり、データやリソースに関する検証にはあまり適していない。この問題に対処するためには、データやリソースに関する性質を Event Calculus[45] や線形時相論理 (LTL)[53] 等によって形式的な言語として表現し、検証に用いるのが有効である。本研究では特定の性質について検証するための手段として LTL を利用しているための LTL を利用した検証について記述する。

LTL は時間的に変化する性質を記述するための特定の性質を記述することができる。LTL を用いて形式的な検証方法は検証対象が望ましい性質を満たしているかどうかを自動的に網羅的に検証することができる。LTL は古典的な論理子や時間的な操作子などのワークフロータスクにおけるシーケンスについて制約を記述する能力を提供する [24]。プロセスマイニング分野においては [3] は LTL checker と呼ばれる LTL を用いた検証方法である。それは PRoM というプロセスマイニングのためのオープンソースフレームワークにおいて利用できるプラグインである。

我々は LTL checker をビジネスプロセスの検証に用いるので、簡単な例を用いて説明する。表 2.1 は簡単なビジネスプロセスの実行ログを表している。それぞれのログは ID とトレースと呼ばれる実行されたイベントシーケンスを持っている。それぞれのイベントは1つのアルファベット (A, B, C, D, E) によって表現される。例えば、ID 1 はイベント A, B, C, D がそれぞれ順番に実行されたことを表す。表 hyourei2 はワークフローにおける制約を表している。それぞれの制約は自然言語と線形時相論理によって記述されている。LTL checker は表 2.1 における各トレースが表 2.2 における制約を満足するか検証することができる。ID1, 2 は両方共すべての制約を満たすが、ID 3 はイベント D または E がトレースにおいてそのうち実行されるべきだという性質を満たさない。少なくともこ

の性質を満たすためにはイベント D またはイベント E のいずれかが実行されている必要がある。

表 2.1: シンプルなイベントログの例

ID of Traces	Traces
1	ABCD
2	ABED
3	ABC

表 2.2: 制約の例

Constraint	Formal Constraint
あるトレースにおいて A が実行されたらそのうち B が実行される	$A \Rightarrow \diamond B$
あるトレースにおいて D または E が実行される	$\diamond(D \vee E)$

2.2.4 データマイニングの利用

プロセスマイニングは2本の柱によって成り立っている：(a) プロセスモデリング及び分析，(b) データマイニング。本節では本研究で利用しているデータマイニング技術である決定木学習について説明する。決定木学習は説明変数に基づくインスタンスの分類を目的とした教師あり学習である。教師あり学習とはラベル付けされたデータを学習することで、従属変数となる各インスタンスのラベルを予測するものである。決定木学習では、データを学習することで決定ルールを学習することができ、分類結果をラベルとして持つカテゴリカルな従属変数が木構造でグラフィカルに表される [1]。本論文では、決定木学習のインスタンスとなるのは、ビジネスプロセス実行ログのトレースであり、また従属変数は望ましい性質が満たされているかどうかを表す。トレースのような構造データを学習するために特徴量をどのように決めるのかは定かではない。我々は各トレースにおけるイベントの実行順序関係の特徴量として用いる。このようにすれば、各トレースにおけるイベントの順序関係を記述することができる。

第3章 リファインメントパターンを利用したKAOSゴールモデルからBPMNモデルへの変換

3.1 はじめに

近年、ビジネス環境は、コスト、内部統制、サービス改善等様々な理由で変化する。そのような状況下では、ステークホルダの要求を的確に捉え、変化に対応する必要がある。更に、ビジネス活動において、情報システムは幅広く用いられており、ビジネスプロセスにおいて必要とされるソフトウェアの開発が求められている。

変化への対応や、システム、ソフトウェアに対する要求を効率的に分析するためには、ゴール指向要求分析法や、ビジネスプロセスモデルを利用することができる。ゴール指向要求分析法は要求をゴールモデルとして表現し、システムやソフトウェアに対する個別の要求を表すゴールの分解を繰り返すことによって、要求とその達成手段を詳細化していく手法である。ゴール指向要求分析法の中でも、KAOS[61]はリファインメントパターンを用いてゴール分解を系統的・論理的に行えることが知られている。一方、ビジネスプロセスモデルはビジネスプロセスの流れや条件分岐を表すモデルであり、その中でもBPMNが標準として普及している。Varaらはゴールモデルとビジネスプロセスモデルを併用することによって、情報システムの要求を獲得する手法を提案している[38]。ゴールモデルによってなぜゴール（目的）を達成しなければならないのかを分析し、ビジネスプロセスモデルを用いて何をどのように行うか分析するという両モデリング手法の利点を活かすことによって多角的に要求を捉えることができる。また、Varaらはシステム開発においてゴールモデルやビジネスプロセスモデルを用いる際、どのような場合にそれらを用いるべきなのか論じている。Varaらは組織が明確な作業手順を定めていない場合はビジネスプロセスモデル

24第3章 リファインメントパターンを利用したKAOSゴールモデルからBPMNモデルへの変換

よりも先にゴールモデルを作るほうが良いと主張している [38]. その理由として, (1) ビジネスプロセスを一から作ることの困難性, (2) ゴールモデルを作成することによってシステム要求が組織の目標を満たしているか評価できる点, (3) ゴールモデルが作業手順の可変性分析に利用しやすい点を挙げている.

そこで, 本研究では, ステークホルダの要求を, 的確にビジネスプロセスモデルへ反映するために, ゴールモデルにより記述されたステークホルダの要求を, BPMN モデルへ変換する手法を提案する. ゴールモデルをビジネスプロセスモデルへ変換する研究は複数存在する. しかし, リファインメントパターンを利用したゴールモデルから BPMN モデルへの変換方法はいまだ論じられていない. リファインメントパターンに基づいて分解したゴールモデルには静的な要求だけでなく振る舞いに関する情報も含まれている. しかし, ここでは振る舞いに関しては暗黙的な情報しか記述されておらず, 振る舞いを明示的に示してはいない. そこで本研究では振る舞いに関する暗黙的な情報を BPMN の形式で明示的に示す工夫を行うことで, ゴールモデルから BPMN モデルを導出した. また, リファインメントパターンに着目することで, エージェントによって責任とそれを含むパターンから各エージェントの担当部分とエージェント間の情報交換の流れを, それぞれのスイムレーンとその間のフローとして明確化することができる. 更に, 近年, リファインメントパターンを利用したソフトウェア開発におけるモデル構築 [16][25], 形式仕様記述 [41], リスク評価方法 [13] が提案されており, ゴールモデルをリファインメントパターンに基づいて構築することがソフトウェア開発における様々な工程に活用できることが示されている. そのため, KAOS ゴールモデルにおけるゴール分解ガイドラインであるリファインメントパターン [61] に基づいて分解されたゴールを, BPMN モデルへ体系的に変換するための手法を提案する. 6種類のリファインメントパターンを用いて定義されるゴール分解の関係と BPMN モデルにおける要素との関係を表すルールを作成し, 手続き的な変換アルゴリズムを用いることで, KAOS ゴールモデルを BPMN モデルへ変換する. 本提案手法により, KAOS ゴールモデルによって系統的に記述された要求と, その実現手順の整合性の確保が可能となる. 複数のケーススタディを行うことで, KAOS モデルにおけるゴール間の関係を, 適切に BPMN モデルへ反映できることを示した.

本章では, 3.2 で提案手法について説明する. 3.3 で提案手法のケーススタディを行う.

3.4で考察を記述し，3.5で関連研究を示す．最後に3.6でまとめと今後の課題を示す．

3.2 提案手法

本節ではKAOSゴールモデルにより記述された要求をBPMNモデルへ体系的に変換する手法を説明する．それによってBPMNモデル構築の難しさを軽減し，要求(KAOS)とビジネスプロセス(BPMN)の整合性を確保する．3.2.1において，本提案手法の適用対象を示す．どのようなゴールモデルに対して本手法が適用可能であるのかを記す．3.2.2において，提案手法の概要を記す．3.2.3において，複数のリファインメントパターンが用いられたKAOSゴールモデルをどのようにBPMNモデルへ段階的に変換していくのか記す．3.2.4においてリファインメントパターンによって分解されたKAOSゴールモデルをいかにBPMNモデルの要素と対応付け，変換するのか記す．

3.2.1 提案手法適用対象

本研究ではOR分解による要求選択肢を選択済み，かつ，すべてのゴール分解がいずれかのリファインメントパターンを用いて行われているゴールモデルを対象としている．OR分解による要求選択肢を選択済みとは，同一のゴールを達成するための手段が複数あり，選択可能であるとき，いずれかの達成手段が選択されていることを表す．あるゴールを達成するために手動で行う達成手段とソフトウェアを用いて自動で行う達成手段がある場合，手動で行うか，自動で行うか決定すると，OR分解による要求選択肢を選択したことになる．

3.2.2 提案手法概要

図3.1に提案手法の概要を示す．図のゴールモデルは親ゴールAがMilestone-driven refinement patternによって子ゴールBとCに分解されている．変換ルールによって子ゴールB，CはそれぞれアクティビティB，Cに変換される．アクティビティ実行順序はMilestone-driven refinement patternによって示されるゴール達成順序と同じくB，Cである．更に子ゴールBはUncontrollability-driven refinement patternによって子ゴールDと期待Eに分解されている．ゴールを達成する部分的な順番はD，Eとなる．この関係が変換ルール

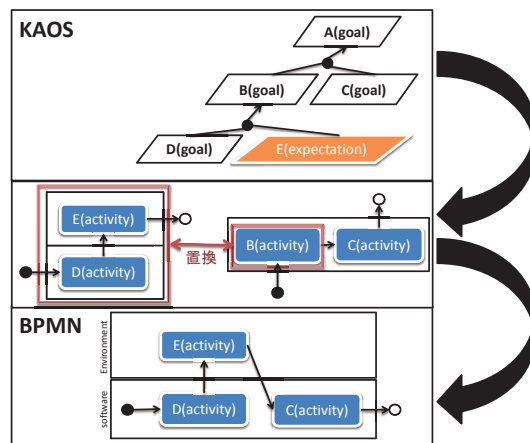


図 3.1: 提案手法概要

を用いて変換されアクティビティ実行順序は D, E だと示される。その後、ゴール B の達成はその子ゴール D, E を達成することに等しいため変換先の BPMN モデルにおいてアクティビティ B と D, E を置換する。その結果アクティビティ実行順序は D → E → C となる。下の BPMN モデルはその様子を表している。アクティビティ D, C と E を実行するエージェントは異なるため、エージェントごとにレーンを分けて記述する。本研究では KAOS ゴールモデルから BPMN モデルに変換するにあたってゴール分解をガイドするリファインメントパターンを利用している。上記のように、リファインメントパターンを用いて構築されたゴールモデルであれば、ゴールモデルの構造から振る舞いに関する情報が読み取れるため、ゴールモデルから特定の BPMN モデルを導出することができる。一方で、リファインメントパターンを用いずに構築したゴールモデルからはゴールモデルの構造を利用して特定の BPMN モデルを導出することができない。これについて図 3.1 の例を用いて説明する。まず、リファインメントパターンを用いてゴール分解を行っている場合は、ゴール A がゴール B, C に分解されたときにこれらを達成するために実行されるアクティビティの順番が B → C だと特定することができる。一方で、ゴール A がリファインメントパターンを用いずに分解された場合はアクティビティ B, C を実行する順番をゴールモデルの構造から特定することができず、この場合は B → C, C → B, B · C への AND 分岐, B · C への OR 分岐になる場合がありうる。

3.2.3 変換アルゴリズム

本節では複数のリファインメントパターンが利用されているゴールモデルを BPMN モデルへ段階的に変換するアルゴリズムを説明する。変換は図 3.2 のアルゴリズム T() にしたがって行う。まずトップゴールから開始して (図 3.2 の 1 行目), 各ゴールのリファインメントパターンに着目しルールに従って BPMN モデルへの変換を行う (図 3.2 の 4 行目)。変換されたアクティビティと対応するゴールが葉ゴールでないならば, 再度ルールに従って変換を行った後アクティビティを置換する (図 3.2 の 5, 6 行目)。ここで, 「置換」とは「親ゴールに関するシーケンスフローの接続関係の子ゴールへ継承すること」を指す。これを変換されていないゴールがなくなるまで繰り返すことで KAOS ゴールモデルを BPMN モデルへ変換する。段階的変換の例として図 3.1 を用いて説明する。図 3.1 のゴールモデルのトップゴールは A であり, これは Milestone-driven refinement pattern で B, C に分解されているため, 変換すると B, C の順でアクティビティが実行される。次にゴール B は葉ゴールではないため更に変換を行う。ゴール B は Uncontrollability refinement pattern によって D, E に分解されており, ルールによって BPMN モデルへ変換したものをアクティビティ B と置換する。その結果が図 3.1 の BPMN モデルである。アクティビティ B がアクティビティ D, E に置換されていることがわかる。

```
1 T(): T1(G0: トップゴール)
2 T1(G): if (G: 葉ゴール) return G
3   else {
4     b = Gの分解に変換ルールを適用した結果;
5     for each G': Gの子ゴール
6       b中のG'をT1(G')で置換
7   }
```

図 3.2: 変換アルゴリズム

3.2.4 各リファインメントパターンを利用した変換

本節ではリファインメントパターン 6 種類を用いたゴール分解と BPMN モデルとの対応付けを図 2.1 を用いつつ説明する。なお, この節による変換は図 3.2 の 4 行~6 行目に対応

28第3章 リファインメントパターンを利用した KAOS ゴールモデルから BPMN モデルへの変換
している。

Milestone-driven refinement pattern を用いた変換

図 2.1 のように KAOS ゴールモデルにおける子ゴールがそれぞれ BPMN モデルにおけるアクティビティへ変換される。アクティビティの実行順序は KAOS ゴールモデルにおいて定められたマイルストーン条件の達成順序と同一である。図 2.1 においてはマイルストーン条件が1つの場合であるため、マイルストーン条件達成のアクティビティとターゲット条件達成のアクティビティが順に実行される。

Decomposition-by-cases pattern を用いた変換

図 2.1 のように KAOS ゴールモデルにおける子ゴールがそれぞれ BPMN モデルにおけるアクティビティへ変換される。Decomposition-by-cases pattern はケース条件の成否によって場合分けするリファインメントパターンであるため、アクティビティは XOR gateway によって分けられ、どちらかのみが実行される。

Guard-introduction pattern を用いた変換

図 2.1 のように KAOS ゴールモデルにおけるガード条件達成時ゴール ($C \wedge CS \Rightarrow \diamond T$)、ガード条件未達成時ゴール ($C \Rightarrow CWT$) はそれぞれ BPMN モデルにおけるアクティビティへ変換される。これらのアクティビティの実行は XOR gateway を用いて表されるガード条件の成否によって分岐する。C 及び CS が成り立っていれば、ガード条件達成時ゴール ($C \wedge CS \Rightarrow \diamond T$) が変換されたアクティビティが実行され、C 及び CS が成り立っていない場合はガード条件未達成時ゴール ($C \Rightarrow CWT$) が変換されたアクティビティが実行される。こちらの場合は、CS が成り立つまで XOR gateway の前へ戻り、CS の達成を待つ。

Divide-and-conquer pattern を用いた変換

図 2.1 のように KAOS ゴールモデルにおける子ゴールがそれぞれ BPMN モデルにおけるアクティビティへ変換される。Divide-and-conquer pattern は結合している状態にあるゴー

ルをより詳細なゴールへ分解する場合に用いられるリファインメントパターンであり、分解された子ゴールは両方達成されなければならない。そのため、この関係を BPMN モデルにおいて AND gateway によって表す。すなわち、アクティビティは両方実行される。

Unmonitorability-driven refinement pattern を用いた変換

図 2.1 のように KAOS ゴールモデルにおける子ゴールがそれぞれ BPMN モデルにおけるアクティビティへ変換される。Unmonitorability-driven refinement pattern は監視すべき状態が監視できない状態を解決するために、ゴール達成の責任が分割されるリファインメントパターンであり、子ゴールを達成するエージェントはそれぞれ異なる。その関係を BPMN モデルにおいては環境エージェントとソフトウェアエージェントへレーン分割することによって表す。ソフトウェアエージェントは環境エージェントによって監視された情報を利用して動作するため、環境エージェントによるアクティビティ($\square(C \Leftrightarrow MC)$)が実行された後に、ソフトウェアエージェントによるアクティビティ($MC \Rightarrow \diamond T$)が実行される。

Uncontrollability-driven refinement pattern を用いた変換

図 2.1 のように KAOS ゴールモデルにおける子ゴールがそれぞれ BPMN モデルにおけるアクティビティへ変換される。Uncontrollability-driven refinement pattern は制御すべき状態が制御できない状態を解決するために、ゴール達成の責任が分割されるリファインメントパターンであり、子ゴールを達成するエージェントはそれぞれ異なる。その関係を BPMN モデルにおいては環境エージェントとソフトウェアエージェントへレーン分割することによって表す。環境エージェントはソフトウェアエージェントから出力される情報を基に動作するため、ソフトウェアエージェントによるアクティビティ($C \Rightarrow \diamond T$)が実行された後に、環境エージェントによるアクティビティ($\square(CT \Leftrightarrow T)$)が実行される。

3.3 ケーススタディ

提案手法の有効性を検証するためにケーススタディを行った。題材として Cailliau らの研究 [10] で用いられている London Ambulance Service (LAS) のゴールモデル (ゴール数 :

17 個) と Cailliau らの研究 [11], [12] で用いられている barbados Car crash Management System (bCMS) のゴールモデル (ゴール数: 60 個), 卸—メーカー間取引におけるビジネス・プロセス・モデル調査研究所レポート [75] より, 著者が作成したゴールモデル (ゴール数: 25 個) を用いる.

評価方法として, 変換結果である BPMN モデルがゴールモデルにおいて想定されているシナリオを満たせるのか確認すること, 及び, BPMN モデルに対してモデル検査を行うことによって, ゴールモデルにおいて想定される性質を満たせるのか形式的に検証することで評価を行う. 後者においてはモデル検査ツールである SPIN を用いて, 仕様記述言語である PROMELA によって記述された BPMN モデルが, LTL (Linear Temporal Logic) によって記述される性質を満たせるのか検証する. BPMN モデルを PROMELA 化するためには Janssen らの研究 [29] を少し修正して適用した. [29] ではアクティビティを表す変数の真偽を変更することによってアクティビティの実行を表す. 検証に用いる性質はトップゴールを直接的に表す性質, 及び重要な性質を用いる.

3.3.1 LAS 概要

LAS は目的地に救急車が到着することを目的とするシステムである. そのためにはまず, 事故の状況等を救急隊員に知らせる必要がある. 救急隊員は聞き取った情報を定められたフォームに落とし込む. その後, どの救急車が現場へ向かうのか決定される. 現場へ向かう救急車は路上にいる場合やステーションにいる場合が考えられる. ステーションにいる救急車が現場へ向かう場合は, その救急車に乗り込む救急隊員に対して指示が伝えられ, プリントアウトされる. その後, 救急車が現場へ到着することで目的が達成される.

3.3.2 LAS におけるケーススタディ

まず, LAS に関するゴールモデルの変換について記述する. このゴールモデルは [19] より構築されたものである. このゴールモデルは何らかの事故が起きた時, それを解決できるということに関するものであり, 時相論理を用いて記述されている.

$\text{IncHappened} \Rightarrow \diamond \text{IncResolved}$ というトップゴールは 7 回分解されている. 1, 3, 4, 5, 7 回目の分解は Milestone-driven refinement pattern に基づいて分解されており, 2, 6 回目

の分解は Decomposition-by-case pattern に基づいて分解されている。

このゴールモデルを提案手法を用いて BPMN モデルへ変換した。図 3.4 は変換結果である。アクティビティの数字は図 3.3 のゴールモデルにおける数字と対応している。生成された BPMN モデルと LAS に関するシナリオである「要請があったとき、救急車が路上にいる場合はそのまま現場に向かい、ステーションにいる場合は order に従って現場に向かう」との整合性を検証する。救急車が路上にいる場合は BPMN において「(1) 事故が起こり通報される, (5) 通報された内容をフォームに落とし込む, (6) フォームに落とし込まれたものを確認し必要な手段を選択する, (7) 路上にいる救急車が動員される, (4) 動員されることが決まった救急車が現場に向かう」の順でアクティビティを実行することで現場の状況が伝えられた路上を走っている救急車が現場に向かえることを示している。救急車がステーションにいる場合は BPMN において「(1) 事故が起こり通報される, (5) 通報された内容をフォームに落とし込む, (6) フォームに落とし込まれたものを確認し必要な手段を選択する, (8) ステーションにいる救急車が選ばれ, order が送られる, (9) order の内容が印刷される (10) order に従った救急車が動員される (4) 動員されることが決まった救急車が現場に向かう」の順でアクティビティを実行することで現場の状況を伝えられたステーションにいる救急車が order にしたがって現場に向かえることを示している。以上の結果から、提案手法はゴールモデルが表している事故が起きた時、救急車が現場へ向かい解決するまでの流れや条件分岐を BPMN モデルへ適切に反映していると考えられる。

次に、図 3.4 の BPMN モデルを PROMELA 化し、SPIN を用いてモデル検査を行った。検証には以下に示す 2 つの性質を用いた。

性質 1. 起きた事故はそのうち解決される。

この性質は以下の LTL 式で記述できる。

$\square((\text{事故が通報される}) \Rightarrow \diamond(\text{起きた事故が解決される}))$

性質 2. ステーションにいる救急車が order に従い現場へ向かう。

この性質は以下の LTL 式で記述できる。

$\square(\text{ステーションにいる救急車が選択される} \Rightarrow \diamond(\text{order を手にする} \wedge \diamond \text{救急車が現場へ向かう}))$

以上の性質を検証した結果、図 3.4 の BPMN モデルはゴールモデル中の性質を満たしたモ

32第3章 リファインメントパターンを利用した KAOS ゴールモデルから BPMN モデルへの変換

デルとなっていることを確認できた。

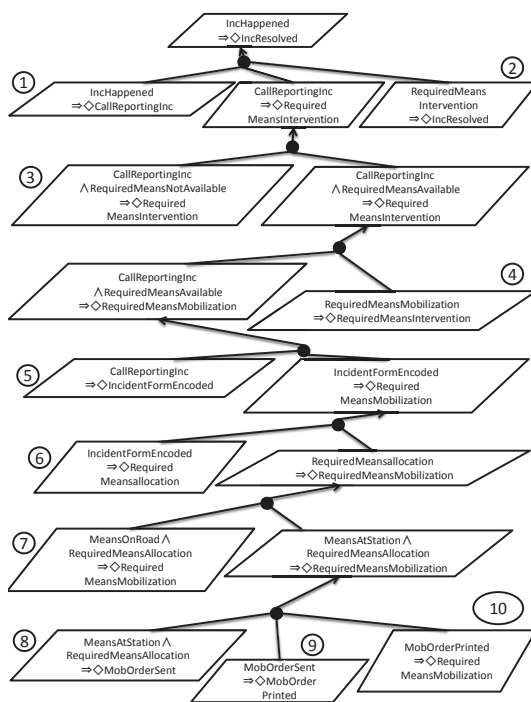


図 3.3: LAS に関するゴールモデル



図 3.4: LAS に関するゴールモデルを提案手法により変換した BPMN モデル

3.3.3 bCMS 概要

bCMS は fire station coordinator (FSC) と police station coordinator (PSC) の間で行われるコミュニケーションを支援することにより事故に対処するシステムである。FSC、PSC 内部のコミュニケーションはこのシステムの対象外である。coordinator のタスクに関係す

る事故の情報は常に最新の状態に保たれなければならない。FSC, PSC は協調していつ、どこに、どのように消防車を送るのか決定し、事故に対処する。

3.3.4 bCMS に関するケーススタディ

次に, barbados Car crash Management System (bCMS) に関するゴールモデルの変換について記述する。[12]にはbCMSに関するゴールモデル全体が記されている。本研究で対象としているのは behavior goal の変換であるため, [12] から該当する部分のみを抽出し整形した。[12]のゴールモデルはOR分解によって manual system, automated system (centralized system, distributed system) の3つに分かれている。Manual system は system-as-is に相当し, ソフトウェアによる支援は行わず, 人間が作業を行うシステムである。一方, automated system はシステムにおける一部をソフトウェアにより支援・自動化したものである。本提案手法はOR分解による選択肢が選択されたゴールモデルを対象としているため, automated system である distributed system をすべてのOR分解において選択した。

[12]に記載されているbCMSのゴールモデルは[15]を基に作られたものである。このゴールモデルはリファインメントパターンが明示的に記されておらず, そのまま各リファインメントパターンをすべてのゴール分解へ当てはめることはできない。そのため, 著者がゴールモデルをすべてのゴール分解がいずれかのリファインメントパターンを用いて分解されたものとなるように意味が変化しないよう留意して整形した。

[12]にはまとまった大きなゴールモデルは記載されていない。しかし, 断片的な複数のゴールモデルにおける同一の親ゴールと葉ゴールを繋ぎ合わせることによって大きなゴールモデルが得られる。繋ぎ合わせたゴールモデルを整形したものが図1である。図1のゴールモデルの親ゴールが Achieve [Crisis Resolved When Reported] であり, Achieve [Crisis Resolved When Reported] というゴールが各リファインメントパターンを用いて分解されている。付録図1のゴールモデルを提案手法によって変換した結果が図3.5である。

LASのケーススタディと同様に, まず, ゴールモデルにおけるシナリオがBPMNモデルにおいて満たせるのか確かめることによって有効性を検証する。シナリオは[15]に記載されているbCMSのメインシナリオを使用する。メインシナリオは[15]のとおり, 7つのパー

34第3章 リファインメントパターンを利用した KAOS ゴールモデルから BPMN モデルへの変換

トにより構成されている。メインシナリオは事故が発生し、PSC と FSC がコミュニケーションを確立し、coordinator を同定してから、PSC と FSC が事故が終息したことを同意するまでの流れを表している。メインシナリオ 1, 5 に関しては今回使用したゴールモデルに含めていない部分であるため、検証に使用するシナリオから除外する。図 3.5 におけるアクティビティの流れを確認すると、適切にゴールモデルにおけるシナリオを反映できていることが確認できる。Unmonitorability-driven refinement pattern, Uncontrollability-driven refinement pattern によって、ソフトウェアエージェントと環境エージェントにレーンをわけることによって、ソフトウェアと外部環境のインタラクションを表現している。

次に、図 3.5 の BPMN モデルを PROMELA 化し、SPIN を用いてモデル検査を行った。検証には以下に示す 2 つの性質を用いた。

性質 1. 事故の情報が消防と警察の間で共有されたら、事故が解決される。

この性質は以下の LTL 式で記述できる。

$\square ((\text{警察の情報を登録する}) \ \&\& \ (\text{消防の情報を登録する}) \ \&\& \ (\text{両者の情報が共有される})) \Rightarrow \diamond (\text{事故が解決される})$

性質 2. 消防と警察の間でルートの合意がなければ、現場へ向かわない。

この性質は以下の LTL 式で記述できる。

$\square (!(\text{消防がルートに合意する} \ \&\& \ \text{警察がルートに合意する}) \ W(\text{現場へ向かう}))$

以上の性質を LTL によって記述し、検証した結果、図 3.5 の BPMN モデルはゴールモデル中の性質を満たしたモデルとなっていることを確認できた。

3.3.5 卸一メーカー間における取引業務概要

卸一メーカー間における取引業務では卸・メーカー間の商品の受発注、物流、請求・支払い等が行われる。製造業や流通業における情報システムを用いた業務支援は未だ不十分であり、迅速な取引情報の交換や情報の共有を低コストで行える情報システムの開発が望まれている。

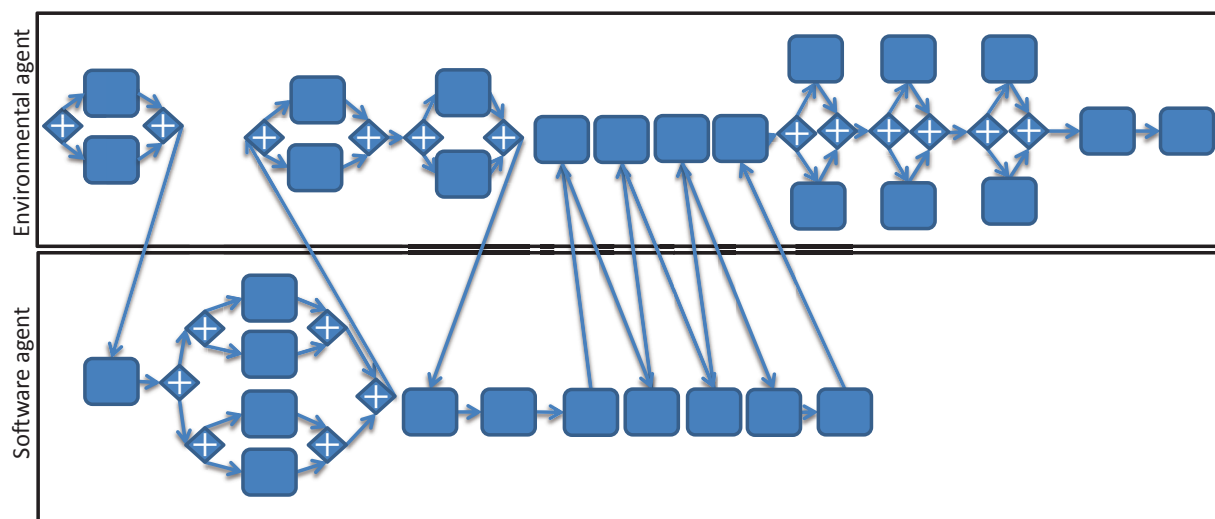


図 3.5: 変換結果まとめ

3.3.6 卸一メーカー間における取引業務に関するケーススタディ

次に、卸一メーカー間における取引業務に関するケーススタディを行った。既存のゴールモデルを用いず、実際に企業が直面している課題である [75] に記載されている情報から著者がリファインメントパターンに基づいてゴール分解を行うことで、KAOS ゴールモデルを構築した。付録図 2 ゴールモデルに対して提案手法を用いて変換した結果が図 3.6 である。LAS や bCMS のケーススタディと同様に、まず、ゴールモデルにおけるシナリオが BPMN モデルにおいて満たせるのか確かめることによって有効性を検証する。[75] において卸一メーカー間におけるビジネスプロセスに関して、全体的な流れや受発注や物流等に分割した流れが記載されている。これらと図 3.6 の BPMN モデルを照らし合わせると適切にシナリオを満たせることが確認できた。また、[75] に記載されているプロセスモデルと同様の流れであることが確認できた。次に、図 3.6 の BPMN モデルを PROMELA 化し、SPIN を用いてモデル検査を行った。検証には以下に示す 2 つの性質を用いた。

性質 1. 欠品連絡がなされたら、注文が修正されなければならない。

この性質は以下の LTL 式で記述できる。

$\square((\text{欠品連絡を行う}) \Rightarrow \diamond(\text{注文が修正される}))$

性質 2. 仕入計上処理と売上計上処理が両方実行されたら、照合チェックが行われなければならない。

この性質は以下の LTL 式で記述できる。

$\square((\text{仕入計上処理を行う}) \ \&\& \ (\text{売上計上処理を行う}) \Rightarrow \diamond(\text{照合チェックを行う}))$

以上の性質を LTL によって記述し、検証した結果、図 3.6 の BPMN モデルはゴールモデル中の性質を満たしたモデルとなっていることを確認できた。

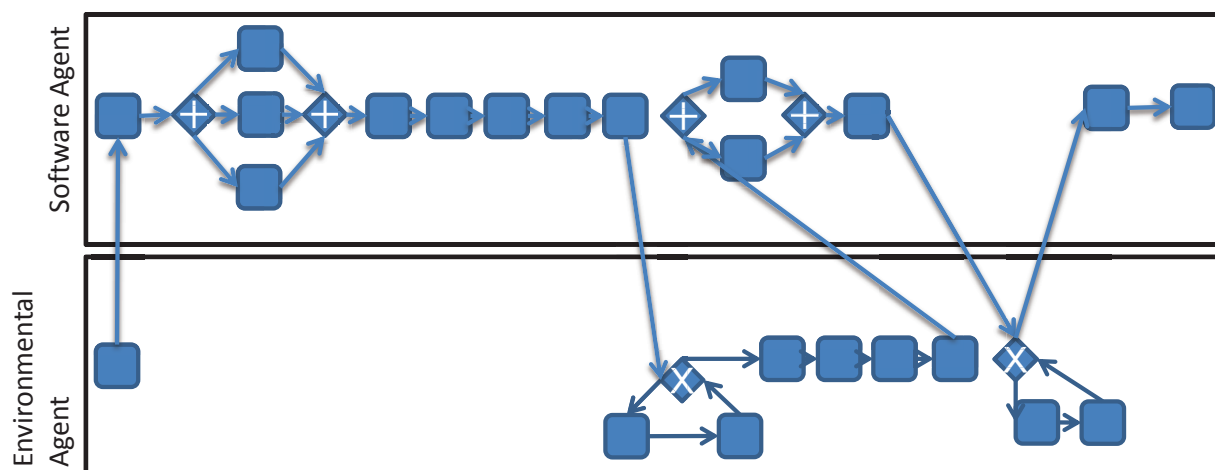


図 3.6: 卸メーカー取引業務プロセス

3.3.7 ケーススタディにおける考察

評価に用いた LAS, bCMS, 卸メーカー取引業務いずれの場合においても、リファインメントパターンを用いて記述された KAOS ゴールモデルにおけるゴール間の関係を、順序や条件分岐の関係を保ち、適切に BPMN モデルへ反映することができた。リファインメントパターンはゴール分解において頻出するゴール間の関係性をパターン化し、時相論理によって記述したものである。ゴールモデルをリファインメントパターンを用いて構築することは、ゴールモデルが 2.1.1 節で述べた AND 分解の性質を持つと共に他のモデルに変換する際にも有効に活用できると考えられる。更に LAS に関するゴールモデル、bCMS に関するゴールモデル、卸メーカー取引業務に関するゴールモデル等、大きさの異なる

ゴールモデルの内容を適切に BPMN モデルへ反映することができた。このことからモデルの規模に関わらず、本手法によってゴールの関係を BPMN モデルへ反映できるといえる。

3.4 考察

3.4.1 生成できる BPMN モデルの特徴

本手法ではリファインメントパターンによってゴールが分解されているゴールモデルを対象としている。そのため、生成した BPMN モデルにはゴールモデルにおけるゴール間の関係が反映されている。ゴールモデルはシステムの目的とその実現手段を記述するために用いるものであり、システムの流れを記述するためのものではない。ゴールモデルにおいては、特に葉ゴールにおいて上位の要求を表すゴールを実現するための手段が記載されており、それは本研究のように BPMN モデルにおいて記述すべきアクティビティと同等のものだと考えることができる。しかし、ゴールモデルは直接的にシステムの振る舞いを記述するために用いるモデルではないため、複雑な流れを記述することはできない場合が考えられる。本研究はゴールモデルによって記述した要求を漏れなく BPMN 初期モデルに反映することを目的としている。そのため、複雑な BPMN モデルを構築する際は本手法によって生成された BPMN モデルを更に精緻化したほうが合理的である。ゴールモデルはシステムの流れの記述には向かないからである。ゴールモデルは上記のように、システムの目的とその実現手段を表すモデルであり、構造的にシステムの流れを明示的に記述することはできない。一方、ビジネスプロセスモデルはシーケンスフローやゲートウェイを用いてシステムの流れを明示的に記述することが可能である。

3.4.2 ゴールとアクティビティの同等性

3.4.1 節において記述したように、ゴールモデルにおけるゴール、特に葉ゴールはアクティビティと同等のものだと考えることができる。これは、BPMN モデルにおけるアクティビティとは、ゴールモデルにおけるオペレーションと同等のものだという仮定を置いているからである。オペレーションとは目的を満たすためにシステム（人やソフトウェアを含む）が提供すべきサービスである。つまり、ゴールを達成するためにはオペレーショ

ンを実行する必要がある。十分に分解されたゴール（葉ゴール）であれば、ゴールとアクティビティは一对一で関連付けることができる。ゴールをアクティビティへ変換するとは、ゴールを達成するためのオペレーションはアクティビティと同一であるという仮定を行っているということである。

3.4.3 変換ルールの妥当性

本節では、ゴール達成・開始の前後関係、アクティビティの開始・終了の前後関係が、変換前後において同一のものであるのかについて記述する。本手法は変換ルールを利用することでKAOSゴールモデルをBPMNモデルへ変換を行う。KAOSゴールモデルにおいてリファインメントパターンを用いてゴール分解が行われることで記述されるゴールを達成・開始する順番は、変換後であるBPMNモデルにおけるアクティビティを終了・開始する順番と同様だと考えられる。Milestone-drivenパターンを例に説明する。Milestone-drivenパターンは親ゴール達成（ターゲット条件Tへの到達）のために必要な中間的な条件（マイルストーン条件M1,M2,M3…）がある場合に用いる。図1のように親ゴール($C \Rightarrow \diamond T$)が成り立つためには、子ゴール1 ($C \Rightarrow \diamond M$)、子ゴール2 ($M \Rightarrow \diamond T$)が順に達成される必要がある。上記の時相論理式で記述したゴールは、親ゴール($C \Rightarrow \diamond T$)（現在条件Cが成り立つとき、そのうちターゲット条件Tが成り立つ。）達成のために、子ゴール1 ($C \Rightarrow \diamond M$)（現在条件Cが成り立つとき、そのうちマイルストーン条件Mが成り立つ。）が実行された後に、子ゴール2 ($M \Rightarrow \diamond T$)（マイルストーン条件Mが成り立つときそのうちターゲット条件Tが成り立つ。）が達成される必要がある。時相論理式を見ればわかるように、子ゴール2はマイルストーン条件Mが成り立っている場合にそのうちターゲット条件Tが成り立つというものである。つまり、これはゴールの終了に関する前後関係が規定されるとともに、開始に関する前後関係も規定されることを意味する。アクティビティに関しても同様である。Milestone-drivenパターン以外の場合であっても時相論理式で記述されているように、終了に関する前後関係、開始に関する前後関係が共に規定される。しかし、規定された前後関係の通りに必ずしもアクティビティが実行されるとは限らない。現実には、あるゴールの達成を待たずに、アクティビティを開始することは起こりうるが、本稿においてはゴールモデルにおける特定の前後関係を仮定してアクティビティの順序へ反映している。例と

して Milestone-driven パターンを取り上げる。上記のように Milestone-driven パターンにおける変換では順序があると仮定しているが、場合によっては各アクティビティの実行が前後することはありうる。そのような場合においては、ある望ましい順序がゴールモデル上に記述されていると仮定して、その順序を BPMN モデルへ反映するとみなす。

3.4.4 OR 分解の選択

3.2.1 節において記述したように本研究では OR 分解による要求選択肢を選択済みのゴールモデルを対象としている。しかし、通常開発すべきシステムのステークホルダは複数存在し、各ステークホルダによってシステムに搭載すべき要求の優先順位が異なる場合が考えられる。そのような場合、ステークホルダ間の利害関係の対立によって、ゴール選択を適切に行うことは難しい。ゴールの選択方法に関しては本研究の範囲外であるが、ゴールを選択するための手法はいくつか提案されている。適切にゴール選択を行うための手法として Kaiya らの AGORA (Attributed Goal-oriented Requirements Analysis)[30] が挙げられる。AGORA はゴール指向要求分析法の一種であり、ゴールの属性値として貢献度と満足度を持つ。満足度を用いてステークホルダ間の要求の不一致を同定することでゴール選択を行うことができる [31]。また、斎藤らの手法 [66] ではゴールに寄与度と有効度を割り振ることでどのような理由でゴールが絞り込まれていったのか根拠を表すことができる。

3.4.5 提案手法適用範囲

本節では、本手法を適用すべき対象と適用できない場合について記述する。

本手法はゴールモデルを変換することによりビジネスプロセスモデルを構築している。よって、本手法の適用対象となりうるのはゴールを達成するために手順を踏む必要があるシステムである。ゴールを達成するために人やソフトウェアが行うべき動作をゴールモデルによる要求の洗練によって獲得し、それらをビジネスプロセスモデルへ変換することで、ゴール達成のための手順を明示的に示せるとともに、プロセスにおいて行うべきことをソフトウェアと人等の環境エージェントで分割することができる。本手法はそのような目的での使用に適していると考えられる。

次に本手法を適用できない場合について記述する。

リファインメントパターンのいずれにも当てはまらない場合

本手法はKAOSゴールモデルにおけるリファインメントパターンに着目しBPMNモデルへ変換を行う。そのため、リファインメントパターンが明示されているKAOSゴールモデルのみを変換対象としている。KAOSと、リファインメントパターンの考案者であるLamsweerdeは6種類のリファインメントパターンを定めている[61]。しかし、ゴール分解において考えられるパターンを6種類で網羅することはできず、どのリファインメントパターンにも当てはまらない分解が起こりうる。そのような場合、本手法ではBPMNモデルへ変換することができず、ゴール達成の順番はゴールの意味を考慮して決定する必要がある。

ハードゴール以外の要素の変換

本手法では変換対象をハードゴールに限定している。これはリファインメントパターンによるゴール分解はハードゴールのみを対象としているからである[61]。その他のKAOSゴールモデルにおける要素としては、ドメインプロパティや非機能要求などがある。共に振る舞いを表す要素ではなく、BPMNモデルにおいて対応付けられる要素がないため本手法においては変換しない。しかし、非機能要求はプロセスに影響を与える場合がある。非機能要求には時間的・空間的計算量や入出力サイズ等の性能に関する要求がある。例として書籍検索システムの非機能要求である「書籍検索の応答時間は1秒より短くなければならない」が挙げられる。このような非機能要求はBPMNモデルにおいてアクティビティとしては表せないが、アクティビティの制約として表すことができる。このように非機能要求をBPMNモデルへ反映することは、より詳細なシステムプロセスの構築に繋がると考えられ、今後の課題として挙げられる。

3.5 関連研究

関連研究としてはゴールモデルとビジネスプロセスモデルに関する研究4種類及び、企業における取り組みを取り上げる。

3.5.1 ゴールモデルからビジネスプロセスモデルへの変換

Sun らはゴールモデルをビジネスプロセスモデルに変換する手法を提案している [59]. Sun らの手法ではビジネスプロセスモデルにおけるアクタを単一のものとして扱っているが、本手法ではリファインメントパターンを用いてゴールの達成責任を分割することによって、レーンを分けることができる.

Nagel らは KAOS モデルをビジネスプロセスモデルへ変換する手法を提案している [47]. KAOS モデルを拡張し、オペレーションを実行する順番に関する制約を付加することで、KAOS モデルにおける要素の関連をビジネスプロセスモデルへ反映している. Nagel らの手法では KAOS モデルにおけるゴールを達成するエージェントが考慮されていないため、ビジネスプロセスモデルにおいて複数のレーンを使い分けることはできないが、本手法ではリファインメントパターンを用いてゴールの達成責任を分割することによって、レーンを分けることができる.

Koliadis らはゴールモデルの一種である i*フレームワークにおける変化を BPMN モデルへ反映する手法を提案している [33]. Koliadis らの手法ではゴールモデルの意味をくみ取って変換しなければならない. 一方で本手法ではリファインメントパターンを用いることでより効率的な変換が可能である.

3.5.2 ビジネスプロセスモデルからゴールモデルへの変換

Vara らはビジネスプロセスモデルをゴールモデルへ変換するガイドラインを提案している [38]. ビジネスプロセスモデルにおける要素パターンとゴールモデルとのマッピングを、22 個のガイドラインを用いることによって行う.

Boness らはビジネスプロセスモデルと類似している UML アクティビティ図を用いて、ゴールモデルの作成支援に関する研究を行っている [9]. これらの研究は本研究とは異なりゴールモデルからビジネスプロセスモデルへの体系的な変換手法を提案してはいない.

3.5.3 ゴールモデルとビジネスプロセスモデルの関連付け

Koliadis ら, Cornax ら及び Ruiz らはビジネスプロセスモデルとゴールモデルを関連付ける方法論を提案している [32], [17], [57]. これらの研究はビジネスプロセスモデルとゴールモデルを関連付けることを対象としており両者のギャップ削減を図っている.

Gröner らは記述論理を用いて自動でゴールモデルとビジネスプロセスモデル間の妥当性を検証する研究を行っている [23]. それによってユーザの要求を満たした実行可能プロセスを検知できる. これらの研究は本研究とは異なりゴールモデルからビジネスプロセスモデルへの体系的な変換手法を提案してはいない.

Nagel らは KAOS モデルから検証可能なビジネスプロセス品質制約を自動で生成するパターンベースの手法を提案している [46]. 生成された制約を用いてビジネスゴールとビジネスプロセスの整合性を検証することができる.

3.5.4 リファインメントパターンの活用

リファインメントパターンはゴール分解を行うために考案されたものであるが, それ以外にも様々な利用法が考案されている. これらを用いて, ゴールモデルを起点として情報システムを開発することは開発効率化の有効な手段になると考えられる.

Sombat らは KAOS モデルを UML クラス図へ変換する手法を提案した [16]. この研究ではリファインメントパターンに基づくゴールモデルを, OCL 制約が記述された UML クラス図へ変換している.

Honda らは KAOS モデルを用いて定義した要求をユースケースモデル, ロバストネスモデルへ変換する方法を提案した [25]. それにより要求モデル・設計モデル間のギャップの削減を支援する.

Cailliau らはリファインメントパターンに基づくリスク評価フレームワークを提案している [13]. この研究ではリファインメントパターンを用いて分解された確率的に達成されるゴールに対してパターンごとに確率を計算することで, 親ゴールの達成確率を計算することができる.

Matoussi らはリファインメントパターンが用いられた KAOS モデルを Event-B モデル

へ変換するアプローチを提案している [41]. それによってソフトウェア開発にフォーマルメソッドを活用する際, 困難な点である初期フォーマルモデル構築の困難さを軽減することができる.

3.5.5 企業における取り組み

企業におけるゴールモデル, ビジネスプロセスモデルに関する取り組みとしては, NTTデータが提唱した MOYA[72] があげられる. MOYA は NTT データにおけるシステム開発標準である TERASOLUNA の要求定義におけるガイドラインの 1 つである. MOYA の狙いはステークホルダの気付きを導き, 要求定義の品質を向上させることであり, 関係者, 課題, 目的, 手段, 業務をゴールモデルや UML を用いて分析し可視化することがより品質の高い要求定義に繋がると考えられている. MOYA は大きく 2 つのステップに分けられる. ステップ 1 ではゴールモデル等を用いてビジネス目的の確認を行う. ステップ 2 では UML を用いてビジネスのモデル化を行う. ステップ 1 で構築したゴールモデルの情報をステップ 2 へ反映させることで, ゴールを達成できるビジネスプロセス, ユースケース等のモデル構築を支援することができる. MOYA は実際のプロジェクトにおいて多く利用されており, ゴールモデルによって手段と目的を構造化・可視化することで, 全体最適となる解決策を選定することができた, 各要件間の関連性や達成しようとしている課題と目的との関係を明確にすることができたという評価が実際に MOYA を適用した担当者から得られている. 我々の提案手法によってリファインメントパターンを利用したゴールモデルをビジネスプロセスモデルの構築に用いることで, MOYA におけるビジネスプロセスモデル構築を体系的に行うことができると考えられる.

3.6 まとめと今後の課題

本研究では, KAOS ゴールモデルのリファインメントパターンに基づいて記述された要求を, システムの流れを表す BPMN の初期モデルへ体系的に変換する手法を提案した. またケーススタディを行い提案手法の有効性を示した. その結果 KAOS モデルで記述した要求を達成すべき順番が BPMN モデルにおいて適切に記述できることを示した. これ

44第3章 リファインメントパターンを利用した KAOS ゴールモデルから BPMN モデルへの変換

により、適切な要求定義のための支援ができると考える。今後の課題としては、本手法をツールとして実装することが挙げられる。それによって変換の際に人為的なミスを排除するとともに短い時間で BPMN モデルを生成することができる。すべてのゴール分解がリファインメントパターンを用いて行なわれ、図 3.2 の変換アルゴリズムを実装すれば自動で BPMN モデルへの変換を行うことが可能だと考える。評価に関しては分野や規模などが異なる様々なケーススタディに本手法を適用することで有効性を検証したい。また、モデル変換の過程を OMG が定めたモデル変換の標準 QVT[50] を用い、Nwokeji らの提案した KAOS メタモデル [48] を利用することで変換をより俯瞰的に示したい。また、形式手法 [23] を用いることで本手法の有効性を様々な角度から検証していきたい。適用範囲の拡大に関しても課題として挙げられる。非機能要求の利用は検討すべき課題である。モデル駆動開発における非機能要求の利用方法は [8] においてまとめられている。

第4章 機械学習手法を利用したビジネスプロセス実行ログの検証支援手法

4.1 はじめに

ITの成長と普及に伴い、Process-Aware Information Systems (PAISs), ERP (Enterprise Resource Planning) systems, CRM (Customer Relationship Management) 等によって効率的なビジネスオペレーションの支援が可能になっている。これらの情報システムは情報を日々記録することができる。これらのログを分析することは組織を取り巻く環境変化が激しい現代において、プロセス改善のための有効な手段となる。

プロセスマイニング [1] は情報システムの実行ログから知識を抽出するための有効な手段である。図 4.1 はプロセスマイニングの概要を表している。情報システムは組織の業務を支援し、イベントログを記録する。イベントログはビジネスプロセスについての実際の情報を持っているため、分析することで意思決定を支援できる。ビジネスプロセスはペトリネットや BPMN のようなモデルによって表現され、またビジネスルールのような制約を持つ。イベントログとモデルや制約を比較することは実行されたビジネスプロセスの問題を理解することを促進する。それはビジネスプロセスライフサイクルの診断フェーズに該当する。このフェーズでの分析結果が次の再設計フェーズにおいて活かされる。

本章ではプロセスマイニングにおける検証について取り扱う。ここでの検証とは各ビジネスプロセスのログ（トレース）において、ある性質が満たされるのかを確かめることである。ビジネスプロセスのログを対象としている研究は主に2つの分野にわけられる。1つ目は適合性検査である。これはイベントログとビジネスプロセスモデル間の整合性を検証するものである。2つ目は特定の性質を形式的な言語を用いて記述し検証するものである。我々はこの研究において後者を取り扱う。LTL checker [3] はその有力な手段であり、各トレースにおいて線形時相論理 (LTL) によって記述された性質が満たされるのか検証するこ

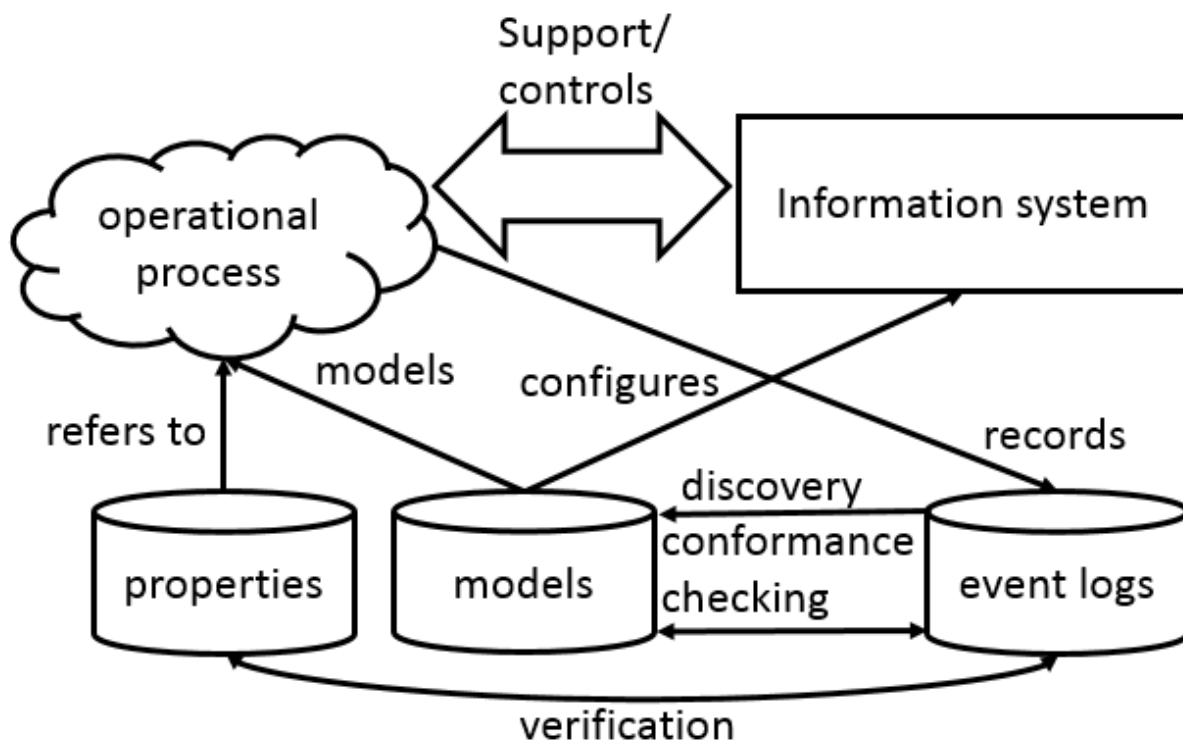


図 4.1: プロセスマイニングの概要 [3]

とができる。しかし、ユーザーの意図を十分に反映した論理式を記述することは難しい。ユーザーの意図を十分に反映しない論理式を用いて検証を行った場合は、検証結果は意図したものとはならず、不適切な検証結果に基づいた意思決定は、誤ったものとなる可能性があり、望ましくない。ユーザーの意図を正確に論理式へ反映するためには、分析対象のドメインについて理解することと、数理論理学の知識が必要だが、これらを十分に習得することは難しい。それゆえ、我々はドメイン知識や数理論理学の知識が十分ではないユーザーを支援し、検証する手法を提案する。我々の手法は各トレースのイベントの実行順序として表現される部分構造に着目し、決定木学習を行うことで、各トレースが特定の性質を満たしているのか予測することができる。また、決定木の構造を用いることで、より正確な論理式を生成することができる。これにより、より正確な検証を行うことができる。

我々の手法の有効性はいくつかのケーススタディーを行って評価した。我々は公的に利用可能な電話修理プロセスのログを用いた。その結果は高い正解率であり、適切な論理式を生成できることを示した。

この論文は以下のように構成される。4.2では提案手法を説明する。4.3では評価結果を説明する。4.4は考察を記述する。4.5では関連研究を説明する。4.6では本節をまとめる。

4.2 提案手法

この節では、提案手法を説明する。提案手法を用いることで、イベントログの検証を行う際に用いられた不十分な論理式を補い、より正確な論理式を生成することができる。既存手法は十分なドメイン知識や数理論理学の知識を持っていないユーザーが検証に使用する論理式を記述するのが困難であるという問題がある。我々の手法はこの問題に対してイベントログの各トレースにおけるイベントの実行順序関係に着目し、決定木を用いた予測を行うことで対処する。

図4.2は提案手法の概観である。このプロセスによって、トレースは決定木による予測と生成された論理式を用いた検証によって分類される。このプロセスではまず、LTLで記述された論理式を用いてイベントログを検証する。もし入力されたトレースが入力された論理式を満たすのなら、これらのトレースが真のトレースに分類され、満たさないのなら、偽のトレースに分類される。我々はLTL checker[3]をこの部分における検証に用いる。次に、各トレースのイベント実行順序として表される特徴量が抽出される。それはアルゴリズム1に沿って行われる。次に決定木に基づく予測が行われる。決定木は決定木学習フェーズにおいて訓練データから構築される。これは4.2.1において説明する。次に決定木の構造に基づいて論理式を構築する。決定木は分類ルールを表しているため、論理式化することができる。最後にLTLによる検証が生成された論理式を用いて行われる。その結果は最初に記述された論理式を検証に用いる場合よりも高い精度による検証を行うことができる。

4.2.1 イベント実行順序関係に着目した特徴量抽出と学習

この節は図4.2における”イベント実行順序関係に着目した特徴量抽出”，”決定木学習”，”決定木による予測”に対応する。決定木学習を用いた予測を行うことでユーザーの意図を反映していない論理式を検証のために用いた際に発生する誤分類を発見することができる。決定木学習は教師あり学習技術であるため、予測モデルを構築するためには訓練

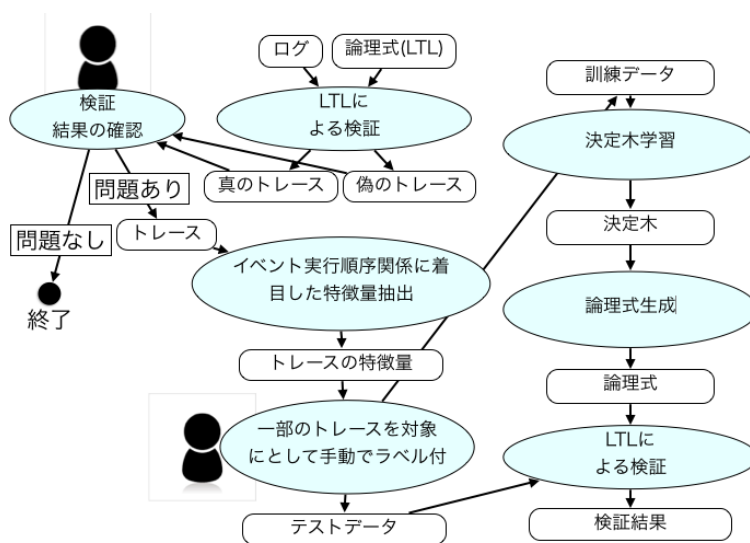


図 4.2: 提案手法の概観

データが必要となる。訓練データとしては検証したい性質が真となるトレースの一部と偽となるトレースの一部を使用し、人手で各トレースに対してラベルを付ける。決定木学習のためには Classification And Regression Tree (CART) [37] というアルゴリズムを用いる。

訓練データはイベントの実行順序関係として各トレースの部分構造から構築される。ビジネスプロセスのトレースは構造データであり、ベクトルデータではない。しかし、多くの機械学習技術はベクトルデータを学習に使用する。それゆえ、構造データとして表現されているイベントログをベクトルデータへ各トレースのイベント実行順序関係に着目して変換する。各トレースは複数のイベント実行順序関係へ変換され、各トレースにおけるイベント実行順序関係の集合が特徴集合となる。学習に使用するそれぞれのトレースは検証したい性質を真に満たしているかどうかを表したラベルを持つ。それゆえ、各トレースはイベント順序に関する特徴とラベルを持っている。各イベント名は1つのアルファベット (A, B, C 等) に置き換えられる。学習した結果構築された決定木は各トレースの真偽を表す条件分岐となる。

表 4.1 は特徴ベクトルの例であり、各トレースはクラス (真または偽) を表すラベルを持っている。各列はトレース ID, イベントシーケンス, イベント順序関係の特徴 (AB, AC 等), ラベル (true, または false) を持っている。ログの特徴集合は全種類のイベント実行順序関係であり、すべてのトレースにおけるイベント集合から2つ取る順列を計算する

表 4.1: 特徴量としてイベント順序関係を持ったトレースの例

trace ID	trace	AB	AC	AE	AF	...	label
1	ABCEF	1	1	1	1	...	true
2	ABCD	1	0	1	0	...	false
3	ABBA	2	0	0	0	...	false

ことで得られる。これは以下の式により記述できる。

A : 全トレースにおけるイベントの組

$$\text{特徴の種類} = P(A, 2)$$

$P(n, r)$ は $n(n-1)(n-2)\dots(n-r+1)$ のように計算できる。次に、各トレースから特徴量を抽出する方法を説明する。例えば、表 4.1 においてトレース 1 は "ABCEF" という順番でイベントが実行されている。それぞれのアルファベットはイベントログにおいて記録されたイベントを表している。表 4.1 におけるイベント実行順序関係の列 (AB, AC など) の集合が特徴集合となっている。各イベント実行順序関係の列において、その値が各トレースごとに計算される。トレース 1 はいくつかのイベント実行順序関係を持っており、(AB, AC, AE, AF, BC, BE 等。これらは Algorithm 1 において FeatureListOfEachLog1 として記述される) これらは algorithm 1 における "step 1: 各トレースからイベント順序関係を抽出する" によって出力される。

このアルゴリズムは各トレースである log_i を入力とする。各 log_i において、 $event_{ij}$ はイベント名とトレースにおけるイベントの場所を表す。例えば、 $event_{i0}$ は log_i の最初に実行されたイベントであることを表す。各 $event_{ij}$ はトレースにおいてそのイベントの後に実行される $event_{ij}$ とペアにされる。この処理は 8 行目に書かれている。 $event_{ij} + event_{ik}$ は $event_{ij}$ と $event_{ik}$ ペアとして統合することを表し、両イベントがこの順番で実行されたという意味である。もしそのペアが特徴集合の要素である特定のものに等しければ、該当する値がインクリメントされる。このようなプロセスを通して決定木学習のためのベクトル形式のデータが得られる。

学習された決定木はラベルがつけられていない各トレースの真偽を予測することができる。図 4.3 は決定木の例であり、この木は 2 回分岐している。根ノードでは分解条件は

Algorithm 1 各トレースからの特徴量の抽出

```

1: Input:
    $log_0, \dots, log_n$  : 配列として表現されるトレース群
    $log_i = \langle event_{i0}, \dots, event_{im} \rangle$ : トレースはイベントの集まりによって構成される
    $AllFeatures$  : すべてのイベント順序関係の集合
2: for  $i = 0$  to  $n$  do
3:   //step 1: 各トレースからイベント順序関係を抽出する
4:    $FeatureListOfLog_i = []$ 
5:    $Results = []$  : 各ログの特徴を表すベクトルデータ
6:   for  $j = 0$  to  $m - 1$  do
7:     for  $k = j + 1$  to  $m$  do
8:        $FeatureListOfLog_i.append(event_{ij} + event_{ik})$ 
9:     end for
10:  end for
11:  //step 2:各特徴の値を計算する
12:  for  $L = 0$  to last do
13:    for  $O = 1$  to last do
14:      if  $AllFeatures_L == FeatureListOfLog_O$  then
15:         $ValueOfFeatureListOfLog_O += 1$ 
16:      end if
17:    end for
18:  end for
19:   $results.extend(FeatureListOfEachlog_O)$ 
20: end for
21: Output:  $Results$ 

```

$EH \leq 0.5$ であり、これはもし EH が実行されたら、偽に分岐し、そうでないなら真に分岐するということを表す。同様のことがより下のノードにおいても実行される。左側は真の場合の分岐であり、右側は偽の場合の分岐である。1つのクラスは各パスによって学習された結果であり、クラスラベルが1の場合と0の場合の両方の数は50 (50, 48 + 2) である。

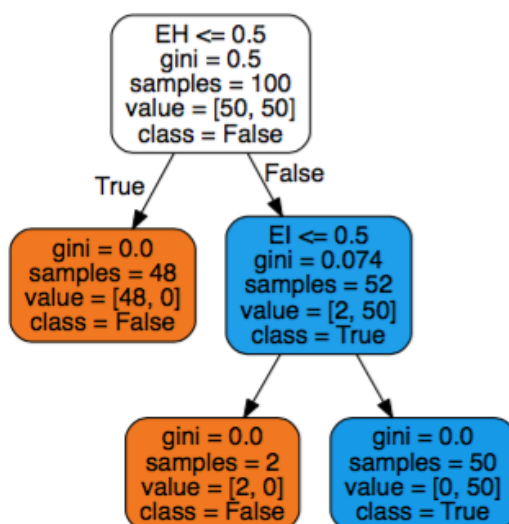


図 4.3: トレースのイベント実行順序関係に基づいて構築した決定木

4.2.2 LTL 式の生成

4.2.1 節において記述した手法によって構築された決定木は論理式の生成に利用される。これはよりユーザーの意図を正確に反映した論理式を生成するために行われ、これはより正確な検証に貢献する。

次に、我々は決定木を論理式へ変換する方法を説明する。我々はイベント実行順序関係によって順次分岐していく決定木を使用する。この決定木では、クラスラベルが1である（真である）葉ノードに注目し、すべての条件分岐の連言をとる。条件分岐は図 4.3 の $EH \leq 0.5$ のように表現され、これはイベント E が実行された後にイベント H が実行されることを表す。もしこれが真ならば、論理式は $\diamond(E \wedge \diamond H)$ となり、偽であれば、 $\neg \diamond(E \wedge \diamond H)$ となる。加えて、すべてのパスの選言は a pass (“class label = 1”) として表される。その結果、論理式はクラスラベルが1となる条件をすべての網羅できる。つまり、決定木において、ク

ラストラベルが1となる部分のみが論理式へ変換される。決定木から構築される論理式の一般形は以下のとおりである。

$$\forall \{ \wedge (\text{根ノードから"クラストラベル} = 1\text{"である葉ノードへのパス}) \}$$

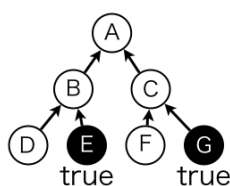


図 4.4: 決定木の例

次に生成された論理式の一般形を説明する。図 4.4 は決定木であり、各アルファベットはノードの名前を表す。黒いノードは真のラベルを表す。それゆえ、新しい論理式を作るためにパス $(A \rightarrow B \rightarrow E)$ 、パス $(A \rightarrow C \rightarrow G)$ が使用される。A,B と A,C はそれぞれ連言によって接続される。真になるためのパスは2つであるため、これらのパスは選言によって接続される。このように、次の論理式が図 4.4 の論理式から構築される。

$$(A \wedge \neg B) \vee (\neg A \wedge \neg C)$$

この論理式を検証に用いることで、初期論理式を用いるよりも、よりユーザーの真の意図を反映した結果が得られる。このフェイズは図 4.2 における下部の LTL による検証に対応する。

4.3 評価

この章では、提案手法の有効性をケーススタディを行うことと、他手法と比較することによって示す。データセットと対応する論理式は 4.3.1 において説明する。4.3.2 では決定木を構築し、新たな論理式を生成する。4.3.3 では人手で記述した論理式による検証、決定木による予測、生成した論理式による検証、他手法を用いた場合の正解率を比較する。

4.3.1 電話修理プロセス

我々は公的に利用可能であり、いくつかの研究において評価で使用された実績のある電話修理プロセスのログ¹を評価のために用いる。このログは1104件のトレースにおける12種類のイベントから11855個のイベントが記録されており、各トレースは電話修理のプロセスを表している（登録、故障原因分析、修理結果の確認、各案件のアーカイブなど）。

我々は電話修理プロセスに関する4種類の正しい論理式と間違っただ論理式のペアを評価に用いた。次の論理式はその一部である。

correct : $\diamond (\text{activity} == \text{Repair}(\text{Complex})\text{start}$
 $\wedge \diamond (\text{activity} == \text{Repair}(\text{Complex})\text{complete}$
 $\wedge \diamond (\text{activity} == \text{Inform User}$
 $\wedge \diamond (\text{activity} == \text{Archive Repair}))))$

incorrect : $\diamond (\text{activity} == \text{Repair}(\text{Complex})\text{start}$
 $\wedge \diamond (\text{activity} == \text{Repair}(\text{Complex})\text{complete})$
 $\wedge \diamond (\text{activity} == \text{Inform User})$

Correct はユーザーの意図を正確に反映した望ましい性質を表し、Incorrect はユーザの意図を正確に反映していない論理式であることを表す。そのような状況は不十分なドメイン知識や数理論理学の知識を持っている場合に起こりうる。両方の論理式は似ているが、これらはそれぞれ異なる意味を持つ。それゆえ、両方の検証結果は異なる結果を示す。正確な論理式はアクティビティ”Repair (Complex) start”(A)が実行されたらその内アクティビティ”Repair (Complex) complete”(B)が実行され、その内アクティビティ”Inform User”(C)が実行され、そのうちアクティビティ”Archive Repair”(D)が実行されるという論理式であり、そのためこれらの4つのイベントについて許容される順番は1つしか存在しない(ABCDがこの順番で実行される)。ここで、許容されるとはすなわち検証した結果、真に分類されることである。一方で不正確な論理式はイベントABCがそれぞれ実行されるというものである。この論理式はこれら3つのイベントについて実行されるべき順番を規定していないため、ABC,BAC,CBA等様々な順番が許容される。更にIncorrectの論理式ではイベントDの実行については記述されていない。つまり、Incorrectの論理式にはイ

¹<http://www.processmining.org/logs/start>

イベント実行順序の制約と条件の見落としという2つの問題がある。これらの論理式は LTL checker を用いて検証される。これは図 4.2 において左上部に記述された LTL による検証に該当する。

4.3.2 構築された決定木と論理式

決定木を構築するために、我々はログを訓練データとテストデータに分割する。訓練データは真のログと偽のログからランダムで選択され、望ましい性質が満たされているか満たされていないかを表すラベルを手動で付ける。これらの訓練データを用いて決定木が CART アルゴリズム [37] によって scikit-learn [52] を用いて構築される。図 4.3 は訓練データから構築された決定木である。この決定木は 4.2.1 節において説明されている。

次に、我々は 4.2.1 節において説明された決定木から再構築された論理式を示す。この決定木から生成された論理式は以下である。

```

(◇ ((activity == Repair(Complex)complete
  ∧ ◇ ( activity==InformUser)))
  ∧ ◇ ((activity == Repair(Complex)complete
  ∧ ◇ (activity==ArchiveRepair))

```

4.3.3 予測結果と検証結果の比較

この節では図 4.2 における我々の提案手法における各フェーズ間での比較と他手法との比較を行う。表 4.2 は人手で記述した論理式を用いた検証結果である。表 4.3 は決定木を用いた場合の予測結果であり、表 4.4 は生成した論理式を用いて検証した結果である。表 4.3 と表 4.4 の両方が表 4.2 よりも良い結果を示した。図 4.5 は人手で記述した論理式を用いた検証、決定木による予測、生成した論理式による検証、他手法の結果を比較したものである。他手法は我々が行った研究 [26] であり、決定木を予測のために使っているが、学習に用いる特徴集合が本章において提案する手法とは異なる。その特徴集合は適合性検査を行い、それぞれのイベント (invisible transition を含む) が各トレースにおいて実行されたか否かを表すものである。比較結果は正解率 $(TP + TN / FP + FN + TP + TN)$ を比較したものである。その結果は、我々の手法 (イベント実行順序関係に着目した決定木による

予測, 生成した論理式による検証) の正解率は人手で記述した論理式による検証や他手法を用いた場合よりも高いことがわかった. このように特徴量としてイベントの実行順序関係を用いた決定木による予測や決定木の構造に基づく論理式の生成は有効であることが確認できた.

表 4.2: 人手で記述した論理式を用いた検証結果

		actual	
		T	F
classified	T	226	431
	F	0	447

表 4.3: 決定木による予測結果

		actual	
		T	F
classified	T	226	17
	F	0	861

表 4.4: 新たに生成した論理式を用いた検証結果

		actual	
		T	F
classified	T	226	17
	F	0	861

次に, 我々は他のケーススタディの結果を説明する. 次の論理式は使用した正確な論理式である. これは先ほど記述した正確な論理式と似ているが, "Archive Repair" が記述されていないという違いがある. それゆえ, これはより単純な例題であるといえる. 一方で, 不正確な論理式は上に記述したものと同じである. その結果は決定木による予測と新たに生成した論理式を用いた両方において完全な分類であることを示した. それゆえ, シンプルな論理式を用いた場合は我々の手法はより正確に分類できることがわかった.

correct : $\diamond (\text{activity} == \text{Repair}(\text{Complex})\text{start})$
 $\wedge \diamond (\text{activity} == \text{Repair}(\text{Complex})\text{complete})$
 $\wedge \diamond (\text{activity} == \text{Inform User}))$

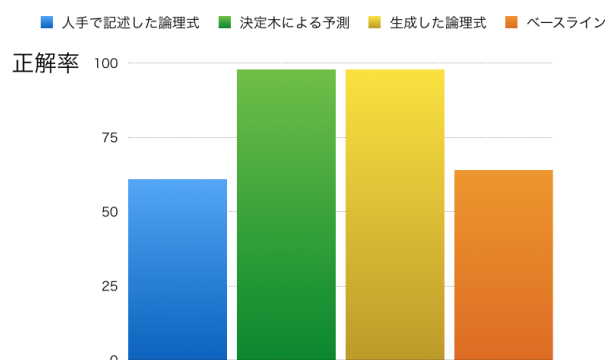


図 4.5: 各ステップにおける比較と他手法との比較

4.4 考察

この節では我々は提案手法の妥当性や限界についてディスカッションする。

我々は決定木を分類器として使用した。ベクトルデータを対象とする機械学習手法であれば、サポートベクターマシン (SVM) やナイーブベイズなどを用いた場合も分類することは可能である。特にカーネル法 [58] と SVM は構造データの学習のための重要な方法である。カーネル法は構造データの種類に応じたカーネルを用いることで明示的に特徴集合を構築することなく予測することができる。これらの正解率を比較した結果は、決定木はナイーブベイズを用いた場合よりも高く、SVM とは近いというものであり、表 4.6 はその比較結果を表している。しかしこれらの手法はより妥当な論理式を生成するという目的には不十分である。なぜなら、これらの2つの手法は決定木のような構造を持っていないためである。我々の手法では決定木の木構造を用いることで論理式を生成している。

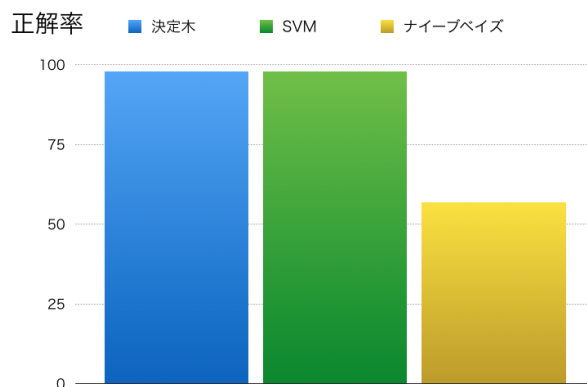


図 4.6: 決定木による予測と他分類器による予測との比較

我々の手法によって生成した論理式を用いた検証は良い結果を示した。しかし、それは理想的な論理式を使用した場合と比べると完璧ではない。その論理式は冗長であり、理想的なものとは必ずしも同一の意味を持つわけではない。より適切で短い論理式を得るための手法は将来の課題である。

我々の手法は適用できる範囲に制限があると思われる。我々はトレースのイベント実行順序関係に着目して特徴集合を構築した。それゆえ、我々はトレースにおけるイベント実行順序に関する情報のみを利用している。しかし、イベントログはタイムスタンプやリソースなどのデータを持っており、これらに関連した制約（例えば、電話の修了は1日以内に終了しなければならない等）については行えない。扱える情報の種類を拡大することは将来の課題である。

4.5 関連研究

ビジネスプロセスの検証は様々な側面を持つ。本節では、いくつかの関連研究を説明する。

ビジネスプロセスモデルとイベントログの差異を検証する適合性検査は様々な側面から研究されている。[56]はプロセスモデルとイベントログとのフィットネスを測定することができる。[35]はコントロールフロー、データ、リソースなどの複数の種類の情報を用いて適合性検査を行う。[44]では、多くの適合性検査手法はペトリネットベースのプロセスモデルを用いている一方で、BPMN モデルを用いた検証手法を提案している。

Medeirosらはsemantic LTL checkerを提案している[42]。それはLTL checkerを拡張し、意味的に等しい要素をオントロジーを用いて表現することで、より効率的に検証することができる。Maggiらは決定木を用いたゴール達成を予測する手法を提案した[40]。この手法はLTLを用いた検証を行うことで実行時におけるゴール達成確率を予測することができる。我々の手法は実行時を対象としたMaggiらの手法とは異なり、決定木と新たに生成した論理式を用いて後方的な検証を行っている。

4.6 まとめと今後の課題

この章では、我々はLTLを用いた検証についての新しいプロセスマイニングにおける検証支援に関する手法を提案した。それはユーザーの不十分なドメイン知識や数理論理学の知識を補うことができる。我々は提案手法を他の手法と比較し、ケーススタディを行うことで有効性を示した。我々の提案手法はトレースのイベント実行順序関係に着目して論理式で検証した場合の真偽を予測する事ができる。更に決定木から構築した論理式が検証の際に有効に使用できることを示した。

将来の課題は、より正確で簡潔な論理式を構築することと、タイムスタンプやリソース等扱える情報の範囲を拡大することである。

第5章 関連研究

本章ではビジネスプロセスや情報システムを環境変化へ対応させるために重要であるが、本研究では直接的には対象としていない研究について記述する。

5.1 環境変化への対応方法に関する研究

本研究は法律改正や市場動向の変化等の環境変化への対応を意図して、ビジネスプロセス実行ログの検証支援に関する研究を行っている。しかし、環境変化に対応するためには、検証等の分析を行うことで現状を把握するだけでは不十分であり、分析結果に応じて、ビジネスプロセスや戦略を修正する必要がある。本節ではその方法について記述する。

5.1.1 ゴールモデルの修正方法に関する研究

環境変化の発生や要求の見落としなどの理由によってゴールが達成できないことがわかった場合、ゴール達成条件を緩和したり、別のゴールに変更するなどの対策が必要である。Lamsweerdeらはゴールモデル中に障害が記述されている場合に、ゴールモデルを修正する手法を提案した [63]。それにより、ゴールモデルにおいて局所的に障害に対応して修正することができる。Cailliauらはゴール達成の障害やその対策が記述された KAOS ゴールモデルを修正する方法を提案した [14]。それにより、障害モデルを利用したりリスク分析を行った結果を踏まえて、より妥当なゴールモデルを構築することができる。Arlajehらは論理ベースの機械学習手法 [55] を用いてゴールモデルを自動で修正する手法を提案している [7]。それにより、ゴールモデル中に記述された障害に対して対策ができたゴールモデルを自動で構築することができる。

5.1.2 ビジネスプロセスモデルの修正方法に関する研究

イベントログの情報を利用して、ビジネスプロセスモデルを修正する手法が提案されている。Fahlandらはイベントログとビジネスプロセスモデルの差異を分析し、イベントログに合わせて、ビジネスプロセスモデルを修正する手法を提案した [21]。実際に観測された情報であるイベントログから自動でビジネスプロセスモデルを構築する手法はプロセス発見として、以前から研究されていたが [5][39]、これらの手法を用いた場合は既存のビジネスプロセスモデルとの差異が大きくなる可能性がある。一方で、Fahlandの手法 [21] のように、イベントログの情報を用いて既存のビジネスプロセスモデルを修正するアプローチであれば、実行された状況を反映し、かつ既存のビジネスプロセスモデルとの差分を小さくできる。また、FrancescomarinoらはFahlandらの手法を多目的最適化手法を用いて、ビジネスプロセスモデルの修正コストを少なく、かつ多くの振る舞いを許容できるように改善している [22]。また、PolyvyanyyらもFahlandらの手法を発展させており、ビジネスプロセスモデルをより短時間で修正する手法を提案している [54]。

5.1.3 実行時における適応

本研究では、ビジネスプロセスライフサイクルにおいて設計工程と診断工程を対象としており、環境変化に対して事後的に記録されたデータを検証し、対応するための支援を行うことを目的としている。一方で、即時に環境変化を検知し、実行時において適応する必要がある場面もありうる。本節では、そのような場合における方法論について、ソフトウェア工学における手法と、ビジネスプロセスマネジメントにおける手法にわけて記述する。

ソフトウェア工学における方法論

情報システムの実行時において環境変化に対応するための手法はソフトウェア工学においては自己適応システムとして盛んに研究が行われている。自己適応システムは実行時に起きる変化を検知し、要求を満たし続けるようソフトウェアの構造・振る舞いを自身で変更する [70]。その手段として、MAPEループの概念を用いた適応エンジンを用いることは有力な手段である。MAPEループとは、環境の観測 (Monitor)、環境の分析 (Analyze)、

適応の計画 (Plan), 適応の実行 (Execute) を行い, 外部環境を観測し, それに応じた対応を行うものであり, 様々な種類に適用するためのパターンが複数提案されている.

ビジネスプロセスマネジメントにおける方法論

プロセスマイニングに関する研究は従来オフラインにおいてイベントログから知識を抽出し, ビジネスプロセスの支援を行うことを目的としていた. しかし, 近年ではビジネスプロセスにおける実行時における決定を支援する (オペレーショナルサポート) 手法が注目を浴びている [4]. オペレーショナルサポートには下記の3種類がある.

1. 検知

検知とは, 実行されているイベントがモデルとは合致しない内容であることを検知することを指す. 例えば, イベントAが実行されてから1週間以内にイベントBが実行されなければならない状況において, 1週間以内に実行されなかった場合などは責任を持つ管理者へ通知される.

2. 予測

予測とは, 現在までに観測されている情報を分析し, 未来に生じるものを予測することを指す. 例えば, イベントABが実行された場合において, 最終的にプロセスが終了するまでの日数が10日であることなどを予測できる. 時間に関するだけでなく, コストやある結果が生じる確率, 利用可能な資源など様々なものが考えられる.

3. 推薦

推薦とは, 現在までに実行されている情報を分析し, 次に行うべき内容を推薦することを指す. これは予測と似ているが, 行うべき具体的な内容を提示するところが異なる. また, 推薦の内容は次に行うべきイベント提示に限らず, 資源配分なども対象となる.

これらの技術を用いることで, ビジネスプロセス実行中において, 環境変化が起きた場合でも適切に行動するための支援を行うことができる.

5.2 超上流工程を対象とした手法

本研究においては、主にオペレーショナルな側面についてモデル化や、データ分析を行っているが、環境変化に対応するためには、より高次元の戦略に関する分析も行う必要がある。経営、事業戦略と情報システムやビジネスプロセスとの整合性が取れていなければ、ビジネスゴールを達成することは難しい。そのための有効な手段が、ゴールモデルの一種である GQM+Strategies である。GQM+Strategies は目標 (G)、質問 (Q)、メトリクス (M) によって、何のデータを収集すべきものかを示すものであり、目標を決定して、その目標を系統だって測定へと具体化する [69]。このアプローチを用いることで、目標が結果として成功したのかを確認することができる。

第6章 結論

本章では，本研究における課題，提案手法についてまとめ，最後に今後の課題について記述する．

6.1 本研究のまとめ

本研究では，以下に示す2つの内容について取り組んだ．

1. ゴールモデルによって記述した要求をビジネスプロセスモデルへ反映する手法
2. 機械学習技術を用いてビジネスプロセス実行ログの検証支援を行う手法

本節では，これらを行う目的と課題，それらを解決するために提案した内容，評価方法についてまとめる．

6.1.1 課題・目的

法律改正や市場動向の変化等の環境変化が発生した場合において，組織において達成することが期待されるゴールが脅かされていないかを把握し，必要であれば，対処する必要がある．そのためには，ビジネスプロセスや情報システムについての要求を定義するために，ゴールモデルやビジネスプロセスモデルを用いることや，ビジネスプロセス実行ログを分析し，現実に何が実行されているのかを明確に把握することが必要である．具体的に取り組んだ課題は以下である．

1. ゴールモデルによって定義した要求を漏れなくビジネスプロセスモデルへ反映する方法

2. ユーザの意図をより正確に反映した論理式を生成し、より正確にビジネスプロセス実行ログを検証する手法

本研究では、これらに取り組むことを目的としている。

6.1.2 提案内容

上記の問題1に関しては、ゴール指向要求分析手法 KAOS により定義した要求をビジネスプロセスモデルとして標準的に利用されている BPMN モデルへ反映する手法を提案した。ゴールモデルにおけるゴール分解をパターン化した6種類のリファインメントパターンを用いた KAOS ゴールモデルから BPMN モデルへ変換するためのルールと、変換アルゴリズムを提案した。

上記の問題2に関しては、LTL によってビジネスプロセスにおいて成り立つべき性質を記述し、ビジネスプロセス実行ログの検証に用いる際に、ユーザーのドメイン知識や数理論理学の知識が不足していた場合に起こる可能性のある論理式の記述にミスがあり、ユーザーの意図を正確に反映できていない場合に対して、機械学習アルゴリズムの一種である決定木を用いて、各実行ログのイベントに関する部分構造に着目することで、新たな論理式を生成し、正確な検証を行う手法を提案した。

6.1.3 評価内容

問題1については、London Ambulance Service, Car Crash Management System, 卸メーカー間取引プロセスという3つの題材を用いて提案手法の適用実験を行い有効性を評価した。評価方法として、ゴールモデルにおいて想定されるシナリオが提案手法を用いて構築されたビジネスプロセスモデルにおいて再現されているか確認することと、モデル検査という形式的な検証手法を用いてゴールモデル中において成り立つ性質がビジネスプロセスモデルにおいても成り立っているのか確認することによって、ゴールモデルの内容が適切にビジネスプロセスモデルに反映されているのかを確認した。その結果、いずれのケーススタディにおいても適切にゴールモデルの内容が反映されていることが確認できた。

問題2について、電話修理プロセスを題材として用いて適用実験を行い提案手法の有効性を評価した。評価方法として、電話修理プロセスのログを検証する際に用いるユーザの意図を十分に反映していないLTLによって記述した論理式を用いて検証が行われた場合において、提案手法によって新たな論理式を生成することで、より正確に検証できるのかを確認した。その結果、提案手法によって論理式を新たに生成することで、より高い正解率を出すことができた。また、その論理式はイベント実行順序や条件を見落としていた論理式をうまく補完することができた。

6.2 今後の課題

6.2.1 モデル修正手法

提案手法ではビジネスプロセスへ正しく要求を反映することと、ビジネスプロセスにおいて要求が反映されているか確認することを目的として、ゴールモデルによって定義された要求をビジネスプロセスへ反映する手法と、ビジネスプロセス実行ログの検証支援を行う手法について取り組んだ。しかし、環境変化に対応して情報システムやビジネスプロセスを改善するためには、検証結果を踏まえて、適切に修正するための方法が必要である。Fahlandらはイベントログとビジネスプロセスモデルの差異を分析し、イベントログの内容を満たしているビジネスプロセスモデルへ修正する手法を提案している。しかし、修正すべきなのはビジネスプロセスモデルだけでなく、ゴールモデルなどの別のモデルもまた同様に修正すべきである。ゴールモデルの修正についてはalrajehらの研究[7]などにおいて取り扱われているが、実行ログの内容を用いてビジネスプロセスモデルとゴールモデルを修正した場合において両モデルの整合性が取れているかは定かではない。Grönerらはゴールモデルとビジネスプロセスモデルの整合性を検証する手法[23]を提案しているが、このような手法を用いても検証結果を踏まえて両モデルを修正する方法は定かではない。そのため、両モデルを連携して修正を行い、修正後においても整合性を保つ手法が必要である。手段として、alrajehらの論理ベース機械学習を用いた自動修正を行うフレームワーク[6]などを利用して、各モデルの修正をもう片方のモデルへ自動で反映し、整合性を保つ手法などが今後の課題となる。

謝辞

本論文を執筆するにあたり，ご多忙の中多大なるご指導をしていただいた電気通信大学大学院情報システム学研究科 大須賀昭彦教授，同 田原康之准教授に感謝いたします。同じく，同 中川博之元助教（現大阪大学准教授），同 清雄一助教，同 本田耕三氏，協力研究員である東芝ソリューション販売（株）平山秀昭氏と（株）東芝早瀬健夫氏，大須賀・田原研究室の皆様には，貴重なご意見とご指導を頂き感謝致します。また，審査を快く引き受けてくださいました大学院情報システム学研究科の田中健次教授，古賀久志准教授，石川冬樹客員准教授に感謝申し上げます。

参考文献

- [1] W. M. P. v. d. Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.
- [2] W. M. P. v. d. Aalst, A. Adriansyah, et al. プロセスマイニングマニフェスト (最終版). 2012.
- [3] W. M. P. v. d. Aalst, H. T. de. Beer, and Boudewijn F. van Dongen. Process mining and verification of properties: An approach based on temporal logic. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences CoopIS, DOA, and ODBASE 2005*, pages 130–147, 2005.
- [4] W. M. P. v. d. Aalst, M. Pesic, and M. Song. Beyond process mining: From the past to present and future. In *Proc. of 22nd International Conference on Advanced Information Systems Engineering, (CAiSE'10)*, pages 38–52, 2010.
- [5] W. M. P. v. d. Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1128–1142, 2004.
- [6] D. Alrajeh, J. Kramer, A. Russo, and S. Uchitel. Automated support for diagnosis and repair. *Commun. ACM*, 58(2):65–72, 2015.
- [7] D. Alrajeh, A. van Lamsweerde, J Kramer, A Russo, and S Uchitel. Risk-driven revision of requirements models. In *Proc. of the 38th International Conference on Software Engineering, (ICSE'16)*, pages 855–865, 2016.

- [8] D. Ameller, X. Franch, and J. Cabot. Dealing with non-functional requirements in model-driven development. In *Proc. of 18th IEEE International Requirements Engineering Conference (RE'10)*, pages 189–198. IEEE, 2010.
- [9] K. Boness and R. Harrison. Goal sketching with activity diagrams. In *Proc. of 3rd the Third International Conference on Software Engineering Advances(ICSEA '08)*, pages 277–283. IARIA, 2008.
- [10] A. Cailliau. Model transformations and refactoring for goal-oriented models. Master's thesis, UCLouvain, Louvain-la-Neuve, 2012.
- [11] A. Cailliau, C. Damas, B. Lambeau, and A. van Lamsweerde. Modeling car crash management with kaos. In *Comparing Requirements Modeling Approaches Workshop (CMA@RE'13)*, pages 19–24. IEEE, 2013.
- [12] A. Cailliau, C. Damas, B. Lambeau, and A. van Lamsweerde. Modeling car crash management with kaos, 2013. <http://kaos.info.ucl.ac.be/bcms.html>.
- [13] A. Cailliau and A. van Lamsweerde. A probabilistic framework for goal-oriented risk analysis. In *Proc. of 20th IEEE International Requirements Engineering Conference (RE'12)*, pages 201–210. IEEE, 2012.
- [14] A. Cailliau and A. van Lamsweerde. Integrating exception handling in goal models. In *Proc. of IEEE 22nd International Requirements Engineering Conference, (RE'14)*, pages 43–52, 2014.
- [15] A. Capozucca, B.H.C. Cheng, G. Georg, N. Guelfi, P. Istoan, and G. Mussbacher. Requirements definition document for a software product line of car crash management systems, 2012.
- [16] S. Chanvilai, K. Honda, H. Nakagawa, Y. Tahara, and A Ohsuga. Goal-oriented approach to creating class diagrams with ocl constraints. In *Proc. of the 27th ACM Symposium on Applied Computing(SAC'12)*, pages 1051–1056, 2012.

- [17] M. Cortes Cornax, A. Matei, E. Letier, S. Dupuy-Chessa, and D. Rieu. Intentional fragments: Bridging the gap between organizational and intentional levels in business processes. In *Proc. of the 10th Confederated International Conferences On the Move to Meaningful Internet Systems (OTM'12)*, pages 110–127. Springer, 2012.
- [18] R. Darimont and A. van Lamsweerde. Formal refinement patterns for goal-driven requirements elaboration. In *Proc. of the 4th ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE'96)*, pages 179–190. ACM, 1996.
- [19] T. C. Directorate. Report of the inquiry into the london ambulance service, 1993.
- [20] M. El Kharbili, A. K. A. de Medeiros, S. Stein, and W. M. P. v. d. Aalst. Business process compliance checking: Current state and future challenges. *MobIS*, 8:107–113, 2008.
- [21] D. Fahland and W. M. P. v. d. Aalst. Model repair - aligning process models to reality. *Inf. Syst.*, 47:220–243, 2015.
- [22] C. D. Francescomarino, R. Tiella, C. Ghidini, and P. Tonella. A multi-objective approach to business process repair. In *Proc. of 12th International Conference on Service-Oriented Computing, (ICSOC'14)*, pages 32–46, 2014.
- [23] G. Gröner, M. Asabi, B. Mohabbati, D. Gasevic, F. Silva Parreiras, and M Boskovic. Validation of user intentions in process models. In *Proc. of the 24th International Conference on Advanced Information Systems Engineering (CAiSE'12)*, pages 366–381. Springer, 2012.
- [24] Arthur H. M. ter. Hofstede, W. M. P. v. d. Aalst, M. Adams, and N. Russell, editors. *Modern Business Process Automation - YAWL and its Support Environment*. Springer, 2010.

- [25] K. Honda, H. Nakagawa, Y. Tahara, and A. Ohsuga. Goal-oriented robustness analysis. In *Proc. of the 10th Joint Conference on Knowledge - Based Software Engineering (JCKBSE'12)*, pages 171–180. IOS Press, 2012.
- [26] H. Horita, H. Hirayama, Y. Tahara, and A. Ohsuga. Goal achievement analysis based on LTL checking and decision tree for improvements of PAIS. In *Proc. of the 31st Annual ACM Symposium on Applied Computing, (SAC'16)*, pages 1214–1218, 2016.
- [27] J. Horkoff and E. Yu. Analyzing goal models: different approaches and how to choose among them. In *Proc. of the 26th ACM Symposium on Applied Computing (SAC'11)*, pages 675–682. ACM, 2011.
- [28] A IIBA. a guide to the business analysis body of knowledge (babok guide). *International Institute of Business Analysis (IIBA)*, 2015.
- [29] W. Janssen, R. Mateescu, S. Mauw, and J Springintveld. Verifying business processes using spin. In *Proc. of 4th International SPIN Workshop (SPIN'98)*, pages 21–36. Springer, 1998.
- [30] H. Kaiya, H. Horai, and M. Saeki. Agora: Attributed goal-oriented requirements analysis method. In *Proc. of 10th Anniversary IEEE Joint International Conference on Requirements Engineering (RE'02)*, pages 13–22. IEEE, 2002.
- [31] H. Kaiya, D. Shinbara, J. Kawano, and M. Saeki. Improving the detection of requirements discordances among stakeholders. *Requir. Eng.*, 10(4):289–303, 2005.
- [32] G. Koliadis and A. Ghose. Relating business process models to goal-oriented requirements models in kaos. In *Proc. of the 1st Pacific Rim Knowledge Acquisition Workshop (PKAW'06)*, pages 25–39. Springer, 2006.
- [33] G. Koliadis, A. Vranesevic, M. Bhuiyan, A. Krishna, and A. Ghose. Combining i^* and bpmn for business process model lifecycle management. In *Proc. of the Business Process Management Workshops*, pages 416–427. Springer, 2006.

- [34] P. Kueng and P. Kawalek. Goal-based business process models: creation and evaluation. *Business Process Management Journal*, 3(1):17–38, 1997.
- [35] Massimiliano de. Leoni and W. M. P. v. d. Aalst. Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming. In *Proc. of 11th International Conference on Business Process Management, (BPM'13)*, pages 113–129, 2013.
- [36] E Letier and A van Lamsweerde. Agent-based tactics for goal-oriented requirements elaboration. In *Proc. of the 24th International Conference on Software Engineering (ICSE'02)*, pages 83–93. ACM, 2002.
- [37] W. Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.
- [38] J. Luis de la Vara, J. Sánchez, and O. Pastor. On the use of goal models and business process models for elicitation of system requirements. In *Proc. of Enterprise, Business-Process and Information Systems Modeling - 14th International Conference, BPMDS 2013, 18th International Conference ENNSAD2013 (BPMDS'13/EMMSAD'13)*, pages 168–183. Springer, 2013.
- [39] F. M. Maggi, R. P. J. C. Bose, and W. M. P. v. d. Aalst. Efficient discovery of understandable declarative process models from event logs. In *Proc. of 24th International Conference on Advanced Information Systems Engineering, (CAiSE'12)*, pages 270–285, 2012.
- [40] F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini. Predictive monitoring of business processes. In *Proc. of 26th International Conference on Advanced Information Systems Engineering, (CAiSE'14)*, pages 457–472, 2014.
- [41] A. Matoussi, F. Gervais, and R. Laleau. A goal-based approach to guide the design of an abstract event-b specification. In *Proc. of the 16th IEEE International Conference*

- on Engineering of Complex Computer Systems (ICECCS'11)*, pages 139–148. IEEE, 2011.
- [42] Ana Karla Alves de Medeiros, W. M. P. v. d. Aalst, and C. Pedrinaci. Semantic process mining tools: Core building blocks. In *Proc. of 16th European Conference on Information Systems, (ECIS'08)*, pages 1953–1964, 2008.
- [43] T. Mendt, C. Sinz, and O. Tveretina. Probabilistic model checking of constraints in a supply chain business process. In *Proc. of 14th Business Information Systems - 14th International Conference, (BIS'11)*, pages 1–12, 2011.
- [44] T. Molka, D Redlich, M. Drobek, A. Caetano, X. Zeng, and W. Gilani. Conformance checking for bpmn-based process models. In *Proc. of Symposium on Applied Computing, (SAC'14)*, pages 1406–1413, 2014.
- [45] M. Montali, F. M. Maggi, F. Chesani, P. Mello, and W. M. P. v. d. Aalst. Monitoring business constraints with the event calculus. *ACM TIST*, 5(1):17, 2013.
- [46] B. Nagel, C. Gerth, G. Engels, and J. Post. Ensuring consistency among business goals and business process models. In *Proc. of the 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC'13)*, pages 17–26. IEEE, 2013.
- [47] B. Nagel, C. Gerth, J. Post, and G. Engels. Kaos4soa - extending kaos models with temporal and logical dependencies. In *Proc. of the CAiSE'13 Forum at the 25th International Conference on Advanced Information Systems Engineering*, pages 9–16. CEUR-WS.org, 2013.
- [48] J.C. Nwokeji, T. Clark, and B.S. Barn. Towards a comprehensive meta-model for kaos. In *Proc. of the 3rd International Workshop on Model-Driven Requirements Engineering (MoDRE'13), 2013*, pages 30–39. IEEE, 2013.
- [49] OMG. Unified modeling language. <http://www.uml.org/>.

- [50] OMG. Meta object facility (mof) 2.0 query/view/transformation. specification version1.0, April 2008.
- [51] OMG. Business Process Model and Notation (BPMN), Version 2.0, January 2011.
- [52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.
- [53] A. Pnueli. The temporal logic of programs. In *Proc. of 18th Annual Symposium on Foundations of Computer Science, Providence, 1977*, pages 46–57, 1977.
- [54] A. Polyvyanyy, W. M. P. v. d. Aalst, A.H.M. t. Hofstede, and M. Wynn. Impact-driven process model repair. 2016.
- [55] O. Ray. Nonmonotonic abductive inductive learning. *J. Applied Logic*, 7(3):329–340, 2009.
- [56] A. Rozinat and W. M. P. v. d. Aalst. Conformance checking of processes based on monitoring real behavior. *Inf. Syst.*, 33(1):64–95, 2008.
- [57] M. Ruiz, D. Costal, S. España, X. Franch, and O. Pastor. Gobis: An integrated framework to analyse the goal and business process perspectives in information systems. *Inf. Syst.*, 53:330–345, 2015.
- [58] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [59] Z. Sun, J. Wang, K. He, S. Xiang, and D. Yu. A model transformation method in service-oriented domain modeling. In *Proc. of the 21st Australian Software Engineering Conference (ASWEC'10)*, pages 107–116. IEEE Computer Society, 2010.
- [60] A. van Lamsweerde. Requirements engineering: From craft to discipline. In *Proc. of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'08)*, pages 238–249. ACM, 2008.

- [61] A. van Lamsweerde. *Requirements Engineering - From System Goals to UML Models to Software Specifications*. Wiley, 2009.
- [62] A. van Lamsweerde, R. Darimont, and P. Massonet. Goal-directed elaboration of requirements for a meeting scheduler: problems and lessons learnt. In *Proc. of the 2nd IEEE International Symposium on Requirements Engineering (RE'95)*, pages 194–203. IEEE, 1995.
- [63] A. van Lamsweerde and E. Letier. Handling obstacles in goal-oriented requirements engineering. *IEEE Trans. Software Eng.*, 26(10):978–1005, 2000.
- [64] M. Weske. *Business Process Management - Concepts, Languages, Architectures, 2nd Edition*. Springer, 2012.
- [65] E. S. K. Yu. Towards modeling and reasoning support for early-phase requirements engineering. In *Proc. of 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, pages 226–235, 1997.
- [66] 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*, 15(3):37–50, dec 2006.
- [67] 児玉公信. 情報システム設計における概念モデリング (<レクチャーシリーズ> 知能ソフトウェア工学 [第5回]). *人工知能学会誌*, 25(1):139–146, jan 2010.
- [68] 西村比朗志, 岩貞正樹, 金井剛. データ分析による現状業務の可視化–業務プロセス可視化技術: Bpm-e (特集 it コアサービス). *Fujitsu*, 59(1):26–32, 2008.
- [69] 著者: V. Basili, A. Trendowicz, M. Kowalczyk, J. Heidrich, C. Seaman, J. Munch, D. Rombach, 共著: 鷺崎 弘宜, 小堀 貴信, 新谷 勝利, 松岡 秀樹 監訳. *ゴール&ストラテジ入門 残念なシステムの無くし方*. オーム社, 2015.
- [70] 鄭顕志, 清水遼, 高橋竜一, 石川冬樹. 自己適応ソフトウェアのための自己適応性設計に関する研究動向. *コンピュータ ソフトウェア*, 31(1):1.49–1.59, 2014.

- [71] 田中 讓 (監修), 磯部 祥尚 (著), 桑野 文洋 (著), 櫻庭 健年 (著), 田口 研治 (著), 田原 康之 (著), 本位田 真一 (シリーズ監修). ソフトウェア科学基礎最先端のソフトウェア開発に求められる数理的基礎 (トップエスイー基礎講座). 近代科学社, 2008.
- [72] 萩原淳, 斎藤忍. 目的と手段の断絶を解消するビジネスモデリング方法論の実践—気づきから要求を導く—. 情報処理学会デジタルプラクティス, 4(2):169–177, apr 2013.
- [73] 飯島正, 田端啓一, 斎藤忍. プロセスマイニング・サーベイ (第 01 回: 概要と基本概念). 情報システム学会誌, 11(02):20–53, 2016.
- [74] 綿引健二, 石川冬樹, 平石邦彦. 時間, 資源の制約をもつビジネスプロセスの形式検証. 電子情報通信学会論文誌 *D*, 96(8):1878–1891, 2013.
- [75] 財団法人 流通システム開発センター. 酒類・加工食品と日用品・化粧品業界の卸—メーカー間取引におけるビジネス・プロセス・モデル調査研究報告書, 2005.

付録

.1 付録: bCMS のゴールモデル

図1はケーススタディで用いたbCMSのゴールモデルである。ゴールを分解する際に用いるリファインメントパターンがゴール間に記述されている。黒の菱形はゴール，白の菱形は期待，黒の六角形は環境エージェント，白の六角形はソフトウェアエージェントを表す。

.2 付録: 卸メーカー間取引業務のゴールモデル

図2はケーススタディで用いた卸メーカー間取引業務のゴールモデルである。

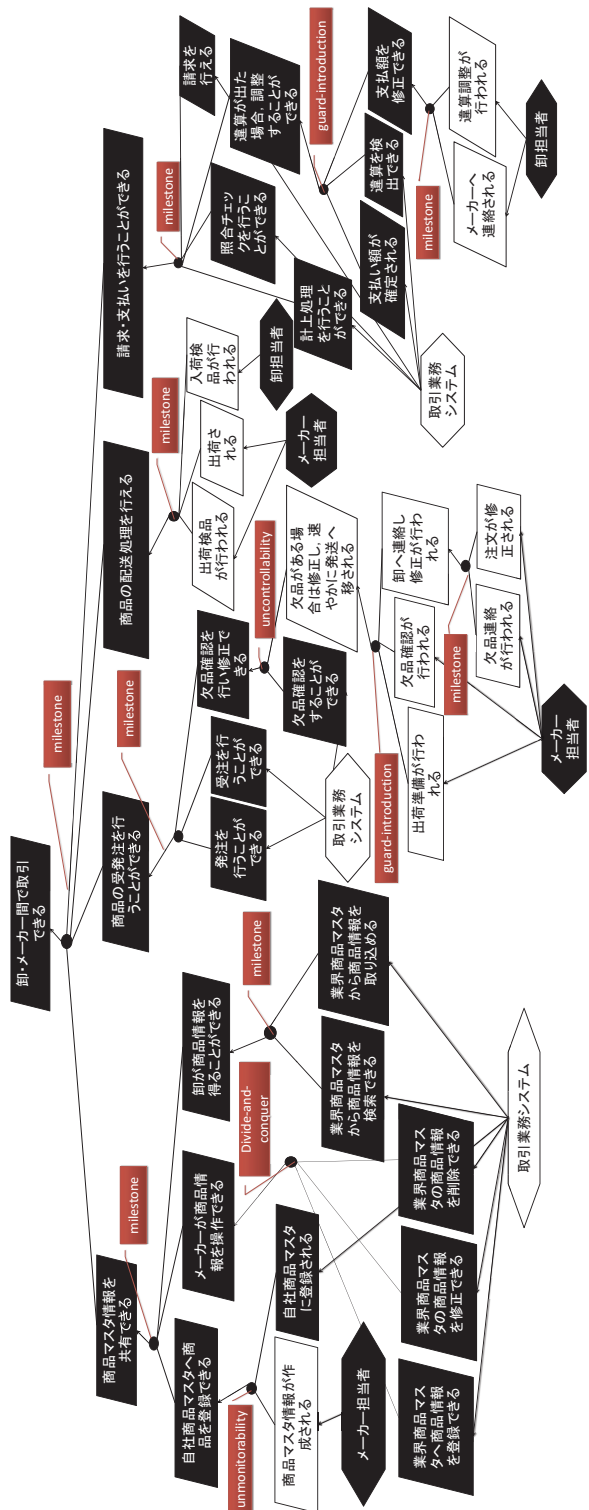


図 2: 卸・メーカー間取引業務のゴールモデル

研究業績

学術雑誌

1. 堀田大貴, 本田耕三, 平山秀昭, 清雄一, 中川博之, 田原康之, 大須賀昭彦: リファインメントパターンを利用した KAOS ゴールモデルから BPMN モデルへの変換, 日本ソフトウェア科学会コンピュータソフトウェア, VOL.32, NO.4, pp141-160 (2015.11)

国際会議

1. Hiroki Horita, Kozo Honda, Yuichi Sei, Hiroyuki Nakagawa, Yasuyuki Tahara, Akihiko Ohsuga: Transformation Approach from KAOS Goal Models to BPMN Models Using Refinement Patterns, 29th Annual ACM Symposium on Applied Computing, (SAC 2014), pp.1023-1024, (2014/3)
2. Hiroki Horita, Hideaki Hirayama, Yasuyuki Tahara and Akihiko Ohsuga: Towards Goal-Oriented Conformance Checking, International Conference on Software Engineering and Knowledge Engineering (SEKE 2015), pp.722-724, (2015.7)
3. Hiroki Horita, Hideaki Hirayama, Yasuyuki Tahara, Akihiko Ohsuga: Goal Achievement Analysis based on LTL Checking and Decision Tree for Improvements of PAIS, 31th Annual ACM Symposium on Applied Computing (SAC 2016), pp.1214-1216, (2016.4)
4. Hiroki Horita, Hideaki Hirayama, Takeo Hayase, Yasuyuki Tahara, Akihiko Ohsuga, Business Process Verification and Restructuring LTL Formula Based on Machine

Learning Approach in book "Studies in Computational Intelligence" (Selected papers from 15th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2016)), pp.89-102, (2016.6)

5. **Hiroki Horita**, Hideaki Hirayama, Takeo Hayase, Yasuyuki Tahara, Akihiko Ohsuga, Process Mining Approach Based on Partial Structures of Event Logs and Decision Tree Learning, 5th IIAI International Congress on Advanced Applied Informatics (IIAI AAI 2016), pp113-118, (2016.7)

査読付国内シンポジウム

1. **堀田大貴**, 本田耕三, 清雄一, 中川博之, 田原康之, 大須賀昭彦, "リファインメントパターンを利用した KAOS モデルから BPMN モデルへの変換", 第 20 回ソフトウェア工学の基礎ワークショップ (FOSE2013), pp.273-282, (2013.11)

関連論文の印刷公表の方法及び時期

学術雑誌

1. 全著者名：堀田大貴，本田耕三，平山秀昭，清雄一，中川博之，田原康之，大須賀昭彦
論文題目：リファインメントパターンを利用した KAOS ゴールモデルから BPMN モデルへの変換
印刷公表の方法及び時期：日本ソフトウェア科学会コンピュータソフトウェア，VOL.32，NO.4，2015 年 11 月
(第 3 章)

国際会議

2. 全著者名：Hiroki Horita，Kozo Honda，Yuichi Sei，Hiroyuki Nakagawa，Yasuyuki Tahara，Akihiko Ohsuga
論文題目：Transformation Approach from KAOS Goal Models to BPMN Models Using Refinement Patterns
印刷公表の方法及び時期：Proceedings of 29th Annual ACM Symposium on Applied Computing, (SAC 2014), pp.1023-1024, ACM, March 2014
(第 3 章)
3. 全著者名：Hiroki Horita，Hideaki Hirayama，Yasuyuki Tahara and Akihiko Ohsuga
論文題目：Towards Goal-Oriented Conformance Checking
印刷公表の方法及び時期：Proceedings of International Conference on Software Engineering and Knowledge Engineering (SEKE2015), pp.722-724, Knowledge Systems

Institute, July 2015

(第4章)

4. 全著者名 : **Hiroki Horita**, Hideaki Hirayama, Yasuyuki Tahara and Akihiko Ohsuga
論文題目 : Goal Achievement Analysis based on LTL Checking and Decision Tree for Improvements of PAIS
印刷公表の方法及び時期 : Proceedings of 31th Annual ACM Symposium on Applied Computing, (SAC 2016), pp.1214-1216, ACM, April 2016
(第4章)
5. 全著者名 : **Hiroki Horita**, Hideaki Hirayama, Takeo Hayase, Yasuyuki Tahara, Akihiko Ohsuga
論文題目 : Business Process Verification and Restructuring LTL Formula Based on Machine Learning Approach
印刷公表の方法及び時期 : in book "Studies in Computational Intelligence" (Selected papers from 15th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2016)) , pp.89-102, Springer, June 2016
(第4章)
6. 全著者名 : **Hiroki Horita**, Hideaki Hirayama, Takeo Hayase, Yasuyuki Tahara, Akihiko Ohsuga
論文題目 : Process Mining Approach Based on Partial Structures of Event Logs and Decision Tree Learning
印刷公表の方法及び時期 : Proceedings of 5th IIAI International Congress on Advanced Applied Informatics (IIAI AAI 2016), pp.113-118, IEEE, July 2016
(第4章)

本論文との関連の詳細

章	節	関連論文番号	関連する内容
3章		1, 2	リファインメントパターンを利用したKAOSゴールモデルからBPMNモデルへの変換
4章		3, 4, 5, 6	機械学習手法を利用したビジネスプロセス実行ログの検証支援

著者略歴

堀田 大貴（ほりた ひろき）

- 1989年11月23日 富山県富山市に生まれる
- 2008年3月 富山第一高等学校 卒業
- 2008年4月 青山学院大学 社会情報学部 社会情報学科 入学
- 2012年3月 青山学院大学 社会情報学部 社会情報学科 卒業
- 2012年4月 国立大学法人 電気通信大学 大学院 情報システム学研究科
社会知能情報学専攻 博士前期課程 入学
- 2014年3月 国立大学法人 電気通信大学 大学院 情報システム学研究科
社会知能情報学専攻 博士前期課程 修了
- 2014年4月 国立大学法人 電気通信大学 大学院 情報システム学研究科
社会知能情報学専攻 博士後期課程 入学
- 2017年3月 国立大学法人 電気通信大学 大学院 情報システム学研究科
社会知能情報学専攻 博士後期課程 修了予定