ペトリネットによるワークフローのリファクタリングとシステム開発における活用の可能性

Petri Net Based Refactoring of Workflows and Its Applications in System Development

豊嶋伊知郎 Ichiro TOYOSHIMA

山口真悟 Shingo YAMAGUCHI

アブストラクト システム開発において対象のワークフローの正しさを保つことは、古くからの普遍的課題である。ペトリネットのサブクラスであるワークフローネットは、ワークフローのモデル化に適しており、その中でも構造がシンプルなものに対しては、各種の正しさを検証する解析技法が開発されている。本稿では、ワークフローネットの振舞いを保ったままシンプルな構造へ変換し、解析技法の適用を促すリファクタリング技術について解説する。また応用例やシステム開発との連携についても示す。

キーワード ワークフロー、ワークフローネット、リファクタリング、システム開発、ペトリネット

Abstract It is a universal problem to keep workflows correct in system development. Workflow nets, a subclass of Petri nets, are suited to creating a model of workflows. They have many verification algorithms for a simpler structure class. In this paper, a given workflow net to apply the verification algorithms. Furthermore, we show applications of the techniques in system development.

Key words workflow, workflow net, refactoring, system development, Petri net

1. はじめに

ワークフロー管理システムは幅広い事業分野に浸透しており⁽¹⁾、そのワークフローを適切に設計することの必要性は、古くからの普遍的課題である。またその妥当性を確認することが初期設計の一度のみで終わることは少ない。大半のシステムでは現実業務の変更に対応してワークフローにも手が加えられるため、確認作業も変更の都度実施する必要がある。1990年代以降の、企業合併や法制度変更による業務見直し、Web技術への対応を背景として、ワークフローの見直しと確認にかかるコストは増大している。

離散事象システムの研究分野では、1960年代からペトリネットと呼ばれる並行システムを取り扱う理論が発展しており、システム/ソフトウェアの設計と検証技術において多くの成果を上げている。ワークフローも複数のタスクが並行するシステムであり、ペトリネットで表現することが可能である。ワークフローネットはワークフローの表現に適したペトリネットのサブクラスであり、1990年代後半から van der Aalst などが中心となってペトリネット研究の一分野を形成している⁽²⁾。ワークフロー

ネットはスループットなどの性能評価にも活用できるが、様々な解析アルゴリズムを適用することで、ワークフローの論理的な正しさの検証を行うことができる。

これらの解析アルゴリズムは無論万能ではなく、複雑な構造のワークフローネットのクラスには効率良く適用することができない。もちろん設計の初期段階でワークフローネットを単純な構造に制限しておくことで、対処することが可能ではある。しかし前述のようにワークフローは変化し続けるため、対応するワークフローネットを設計者の意図どおりにシンプルな構造に保つことは容易ではない。

この問題に対して、近年筆者らは複雑なワークフローネットをその振舞いを保ったままシンプルなネットへ変更し、変更後のネットへ解析アルゴリズムを適用して元のネットの性質を判定する技術を提案している⁽³⁾⁽⁴⁾。このアプローチは、振舞いを保ったまま内部構造を変更するというコンセプトの点で、ソフトウェア工学における「リファクタリング」の概念と共通であり、ワークフローネットのリファクタリングと呼称する⁽⁵⁾、^(注1)。本解説ではそれらの成果を適用事例を交えて概説する。

まず2.でワークフローネットの数学的定義と関連する諸性質について例を交えて説明する。次に3.ではワークフローネットのリファクタリングの概念を振舞いの等価性を中心に考察し、分岐双模倣の概念を導入する。そしてリファクタリングの形式的定義を与える。4.ではリファクタリングのアプローチと方法を紹介する。5.では、4.で紹介したリファクタリングの応用を説明する。6.ではシステム開発への応用を見据え、デザイン

豊嶋伊知郎 正員 (株)東芝社会インフラシステム社
E-mail ichiro.toyoshima@toshiba.co.jp
山口真悟 正員：シニア会員 山口大学大学院理工学研究科
E-mail shingo@yamaguchi-u.ac.jp

Ichiro TOYOSHIMA, Member (Social Infrastructure Systems Company, TOSHIBA Corporation, Fuchu-Shi, 183-8511 Japan) and Shingo YAMAGUCHI, Senior Member (Graduate School of Science and Engineering, Yamaguchi University, Ube-Shi, 755-8611 Japan).

電子情報通信学会 基礎・境界サイエティ
Fundamentals Review Vol.9 No.4 pp.340-349 2016年4月
©電子情報通信学会 2016

(注1)：以降の本文中における「リファクタリング」の語は、特に断りのない限り後者を指す。

ツールとの連携方法の一端を紹介する。最後に 7. で、理論面と応用面の課題を整理し今後の方向性を示す。

2. ワークフローネットとその性質

本章ではワークフローネットと関連する諸概念の数学的基盤について述べる。

2.1 ワークフローの各用語

まず文献(6)を参考にワークフロー関連で必要な用語を整理する。

- アクション

ワークフローの内部的な処理や他のワークフローとのインタラクションを表すものであり、プロセスの最小構成単位である。

- プロセス

ワークフローの挙動を表すものであり、実行すべき順序で並べられたアクションの列に相当する。プロセスは選択的動作や並行動作を記述することもできる。

- インスタンス

プロセスに従って動作する実体を指す。例えば、発注処理では、ある物品の発注案件に相当する。現実のワークフローでは、1個のプロセスに従って動作する複数個のインスタンスが同時に存在することが一般的である。

2.2 ペトリネットとワークフローネット

ワークフローネットを導入するには、最初にペトリネット⁽⁷⁾⁽⁸⁾を定義する必要がある。

ペトリネット N は 3 項組 $N = (P, T, A)$ で与えられる。このとき P と T は、それぞれの要素を「プレース」と「トランジション」と呼ぶ重複のない有限集合である。また、 $A \subseteq (P \times T) \cup (T \times P)$ はアークの有限集合である。 N は P と T の分割による二部グラフと見ることできる。 x をプレースまたはトランジションとすると、前集合 $\bullet x$ 及び後集合 $x \bullet$ が各々 $\bullet x = \{y | (y, x) \in A\}$ and $x \bullet = \{y | (x, y) \in A\}$ として定義される。

ペトリネットの状態は、各プレースに存在するトークンと呼ばれる状態指定子の分布で表現される。このトークン分布をマーキングと呼び、 P 上の多重集合として表現する。すなわち N のマーキング M は $M = [p^{M(p)} | p \in P, M(p) \geq 0]$ として表す。例えば $[(p_3)^1, (p_7)^5, (p_9)^1]$ は p_3, p_7, p_9 に各々 1 個、5 個、1 個のトークンが存在し、それ以外のプレースにはトークンが存在しないというマーキングに対応する。ここで、トークンの個数を表すプレースの肩の数字を多重度と呼ぶ。マーキング M とトランジション t に対して $M \geq \bullet t$ ならば t は M で発火可能であるという。 t が発火すると M は $M - \bullet t \cup t \bullet$ へ遷移する。ここで、 $\geq$ は、各要素 p の多重度の比較において $\geq$ が成立している場合に成立する。ただし、 $\bullet t$ は各要素が多重度 1 の多重集合とみなす。-, $\cup$ は $\bullet t, t \bullet$ を多重集合とみなしたときの差集合と和集合を表す。マーキング M を M' へ遷移させる発火系列

$\sigma \in T^*$ が存在する場合、 M' は M から到達であるという。一般に $M[N, \sigma] M'$ と表記される。また M から到達なマーキング集合を $R(N, M)$ と表記する。

ワークフローネットはワークフローの制御の流れをモデル化するために、ペトリネットに幾つかの制限を加えて定義される。これらの制約はプロセスには始めと終わりがあることや、各アクションはシステム内で一意であることなどに相当する。

[定義 1] (ワークフローネット (Workflow net)⁽⁹⁾) 次の 3 条件を満たすとき、ラベル付きペトリネット $N = (P, T, A, \ell)$ はワークフローネットである。

- N はソースプレース p_I とシンクプレース p_O をそれぞれ一つずつ持つ。

- 全てのプレースとトランジションは p_I から p_O へのパス上にある。

- N のトランジションはラベル付き関数 $\ell: T \rightarrow A \cup \{\tau\}$ を持つ。ラベルには 2 種類あり A は外部アクションと呼ばれるシステム外部からその発火が観測可能なものの集合である。 τ は内部アクションと呼ばれるシステム外から発火が観測できないものを表す単一記号である。

□

前節で整理したワークフローの各概念とワークフローネットとの対比を試みる。「アクション」はトランジションとそのラベルに相当する。「プロセス」は、ネットの構造そのものと考えることができる。「インスタンス」は $[p_I]$ から到達可能なマーキングの集合である。

2.3 ワークフローネットのサブクラス

ワークフローネットには「Well-Structured (WS)」と「拡張自由選択」と呼ばれる二つの重要なサブクラスがある。ワークフローの選択的動作や並行動作は、ワークフローネットではパスの分岐として表現される。分岐開始点と合流点がプレース同士またはトランジション同士の場合は解析が容易であり、これが Well-Structured の語源となっている。これらの形式的定義のために、ネットの構造に関する幾つかの概念を導入する。

ワークフローネット N について p_O から p_I へ新しいトランジション t^* を付加して得られる有向路を持つネット $\bar{N} (= (P, T \cup \{t^*\}, A \cup \{(p_O, t^*), (t^*, p_I)\}, \ell \cup \{(t^*, \tau)\}))$ を N の short-circuited net と呼ぶ。

また構成要素の頂点に重複のない(閉)路 ρ を初等(閉)路と呼ぶ。 c を $\bar{N}$ 上の初等閉路、 ν を c の頂点 x から y への初等路とする。 ν が x, y 以外の頂点を c と共有していなければ、 ν は handle⁽⁷⁾と呼ばれる。次に c を $\bar{N}$ 上の初等閉路、 d を c の handle とし、 ξ を頂点 x から y への初等路とする。このとき ξ が x と y 以外の頂点を c, d と共有していなければ、 c と d 間の bridge⁽¹⁰⁾と呼ばれる。

handle と bridge は、始点と終点のプレース/トランジションの別により 4 通りに区別して扱われる(例: 始点がトランジションで終点がプレースの handle は TP-handle である)。WS

図 1 非巡回ワークフローネット間の包含関係

ワークフローネットの定義を以下に示す。

[定義 2] (Well-Structured (WS) ワークフローネット)

ワークフローネット N は、 $\bar{N}$ に PT-handle も TP-handle も存在しないとき、WS ワークフローネットである。

次に拡張自由選択ワークフローネット及び、そのサブクラスである自由選択ワークフローネットの定義を示す。

[定義 3] (自由選択 (Free Choice:FC) ワークフローネット)

ワークフローネット $N = (P, T, A, \ell)$ における任意の $p_i, p_j \in P$ について、 $p_i \bullet \cap p_j \bullet \neq \emptyset$ ならば $|p_i \bullet| = |p_j \bullet| = 1$ が成り立つならば N は自由選択である。

[定義 4] (拡張自由選択 (EFC) ワークフローネット)

ワークフローネット $N = (P, T, A, \ell)$ における任意の $p_i, p_j \in P$ について、 $p_i \bullet \cap p_j \bullet \neq \emptyset$ ならば $p_i \bullet = p_j \bullet$ が成り立つならば N は拡張自由選択である。

各サブクラスはラベルとは無関係に、ネットの構造にのみ注目して定義されていることに注意されたい。

これらのクラスについて閉路を持たない非巡回 (acyclic) なものは真の包含関係にある。非巡回 FC ワークフローネットは非巡回 EFC ワークフローネットの真部分集合である。非巡回 WS ワークフローネットは非巡回 FC ワークフローネットの真部分集合である。これらの関係を図 1 に示す。

現実世界のワークフローをモデル化するには、EFC ワークフローネットの複雑さが必要であると考えられており⁽⁹⁾、解析対象となるワークフローネットが EFC であることは所与の条件となる。EFC ワークフローネットの振舞いを保持したまま WS ワークフローネットに変換することで解析技法の適用が容易になる。

2.4 健全性と分岐双模倣性

本節ではワークフローネットの振舞いを議論する上で重要な性質である健全性と分岐双模倣性について述べる。

健全性はワークフローネットの論理的な正しさに相当する概念であり、ワークフローが不手際なく仕事を完了できることを表している⁽¹¹⁾。

[定義 5] (健全性 (soundness)⁽⁹⁾) ワークフローネット N は、以下の 3 条件を満たす場合、健全である。

- 任意の $M \in R(N, [p_I])$ から、 $[p_O]$ へ到達可能である。
- p_O にトークンがある場合には、他のプレースにはトークンは存在しない。
- $(N, [p_I])$ に不活性なトランジションは存在しない。

□

EFC ワークフローネットの健全性判定問題は、線形代数のアプローチによって多項式時間で判定できる⁽¹²⁾ またモデル検査による判定方法も研究が進んでいる⁽¹³⁾。

次にワークフローネットの振舞い同士を等しいとみなすための基準を導入する。ペトリネットは、トークンの配置が状態、またトランジションの発火が遷移に相当する状態遷移システムである。すなわちペトリネットのサブクラスであるワークフローネットにおいても、状態遷移システムの等価性基準である分岐双模倣性⁽¹¹⁾を考えることができる。

最初に準備としてワークフローネット $N = (P, T, A, \ell)$ のラベル付き可達グラフ $G = (V, E)$ を導入する：

- $V = R(N, [p_I])$ とする。 N の初期マーキングから到達可能な各マーキングが G のノードに対応する。
- $E = \{(M, \ell(t), M') \mid M, M' \in V, t \in T, M[N, t]M'\}$ とする。発火によるマーキングの遷移が枝に対応する。

ノード v' がノード v から α とラベル付された枝を通して到達できるならば $v[\alpha]v'$ と表記する。任意の数の τ (内部アクション) を通って到達できるならば $v[\tau^*]v'$ とする。また、 $\alpha = \tau \wedge v = v'$ または $v[\alpha]v'$ のいずれかの場合 $v[(\alpha)]v'$ と表記する。

[定義 6] (分岐双模倣 (branching bisimilarity)⁽¹¹⁾) G_X

をワークフローネット N_X の可達グラフ、 G_Y をワークフローネット N_Y の可達グラフとすると、 $\mathcal{R} \subseteq V_X \times V_Y$ をグラフ間の 2 項関係とする。 $\mathcal{R}$ は以下を満たすかつそのときに限り、 G_X と G_Y に関する分岐双模倣関係と呼ばれる。

- $uRv \wedge u[\alpha]u'$ であるとき、 $v[\tau^*]v'' \wedge v''[(\alpha)]v' \wedge uRv'' \wedge u'Rv'$ となるような $v', v'' \in V_Y$ が存在する。
- $uRv \wedge v[(\alpha)]v'$ であるとき、 $u[\tau^*]u'' \wedge u''[(\alpha)]u' \wedge u'Rv \wedge u'Rv'$ となるような $u', u'' \in V_X$ が存在する。
- p_O^X, p_O^Y はそれぞれ N_X, N_Y のシンクプレースを指すものとする。 uRv であるとき、 $(u = [p_O^X] \Rightarrow v[\tau^*][p_O^Y]) \wedge (v = [p_O^Y] \Rightarrow u[\tau^*][p_O^X])$ である。

$[p_I^X], [p_I^Y]$ をそれぞれワークフローネット N_X と N_Y のソースプレースとする。 $(N_X, [p_I^X])$ と $(N_Y, [p_I^Y])$ は、 $[p_I^X]$ と $[p_I^Y]$

図2 $(N_X, [p_I^X])$ と $(N_Y, [p_I^Y])$ の分岐双模倣関係の概念⁽¹⁴⁾. α は外部アクション, τ は内部アクションを表す.

の間に $[p_I^X] \mathcal{R} [p_I^Y]$ となるような分岐双模倣関係 $\mathcal{R}$ が存在するときに限り分岐双模倣であるという. □

ワークフローネットの分岐双模倣性の概念を図2に示す.

3. リファクタリングと リファクタリング可能性

従来ソフトウェア工学の分野では, 外部から見た振舞いを保存したまま内部の設計を見直すことをリファクタリング⁽¹⁵⁾と呼ぶ. ワークフローネットのリファクタリングも同様に, 分岐双模倣性の振舞いを保存したまま, ネットの構造をより簡単なものに変更する. ワークフローネットのリファクタリング問題は以下のように定義される.

[定義7](リファクタリング問題⁽⁵⁾)

入力: 非巡回 EFC ワークフローネット $N_X = (P_X, T_X, A_X, \ell_X)$
出力: 非巡回 WS ワークフローネット $N_Y = (P_Y, T_Y, A_Y, \ell_Y)$,
このとき $(N_X, [p_I^X])$ と $(N_Y, [p_I^Y])$ は分岐双模倣である. また τ 以外のラベルの一意性は保持されている. □

全ての非巡回 EFC ワークフローネットがリファクタリングできるわけではないため, ワークフローネットごとにその可能性を論ずる必要がある. リファクタリング可能性問題を, 決定問題として定義する.

[定義8](リファクタリング可能性問題⁽⁵⁾)

入力: 非巡回 EFC ワークフローネット $N_X = (P_X, T_X, A_X, \ell_X)$
出力 (yes/no): $(N_X, [p_I])$ と $(N_Y, [p_I])$ とが分岐双模倣である非巡回 WS ワークフローネット $N_Y = (P_Y, T_Y, A_Y, \ell_Y)$ が存在するか否か? このとき N_Y の τ 以外のラベルの一意性は保持されているものとする. □

どのような EFC ワークフローネットがリファクタリング可能であるのか, 別の言い方をすればリファクタリング可能性の判定条件が関心事である. 筆者らは対象のワークフローネットが健全であることが, その必要条件であることを明らかにしている⁽⁵⁾.

[性質1](リファクタリング可能性の必要条件⁽⁵⁾) N_X を非巡回 EFC かつ非 WS であるワークフローネットとする. N_X が非健全ならば, $(N_X, [p_I])$ と $(N_Y, [p_I])$ が分岐双模倣となる WS ワークフローネット N_Y は存在しない. □

与えられた EFC ワークフローネットが非健全である場合にはリファクタリングを試みる必要がなくなるという点で, 実用面からも重要な結果である.

4. リファクタリングの方法

筆者らはワークフローネットのリファクタリングに対して二つのアプローチを試みている. 一つは EFC ワークフローネットに特有の構造を置き換えるもの, もう一つはワークフローネットの冗長性を除去するものである.

4.1 構造を変換するアプローチ

WS ワークフローネットの定義は, その short-circuited net に PT-handle も TP-handle も存在しないことである. 対象となる EFC ワークフローネットに PT-handle が存在しないときには以下の定理を明らかにしている.

[定理1](リファクタリングの十分条件1⁽³⁾) N を健全かつ任意の内部アクション τ 以外のラベルに重複のない非巡回 EFC ワークフローネットとする. $\bar{N}$ が PT-handle を持たないとき, N は非巡回 WS ワークフローネットへリファクタリング可能である. □

定理1は, EFC 構造と呼ばれる PT クロス構造を FC 構造へ変換する操作を用いて, 構造的に証明することができる. この操作を ϕ_{Best} ⁽¹⁶⁾と呼ぶ(図3). ϕ_{Best} は TP-handle を除去する効果を持ち, 適用前後のネットが分岐双模倣性を保持することや, 適用により新たに PT-handle を生成しないため, ϕ_{Best} を有限回適用したものは元の EFC ワークフローネットと分岐双模倣である WS ワークフローネットとなる. 図4は ϕ_{Best} によるリファクタリングの前後を示した例である. 両者が分岐双

図3 ϕ_{Best} の適用例⁽⁴⁾

模倣関係にあることを図 5 に示す．

一方，定理 1 とは逆に TP-handle が存在しない場合は次の定理を明らかにしている．

[定理 2] (リファクタリングの十分条件²⁽³⁾) N を健全でかつ任意の内部アクション τ 以外のラベルに重複のない非巡回 EFC ワークフローネットとする． $\bar{N}$ が TP-handle を持たず，PT-handle は長さ 1 の TP-bridge を持つとき， N は非巡回 WS ワークフローネットへリファクタリング可能である． $\square$

こちらは，EFC 構造の双対である TP-cross 構造を変換する操作によって構造的に証明することができる．この操作を $\phi_{Desel}^{(12)}$ と呼ぶ (図 6)． ϕ_{Desel} は ϕ_{Best} とは逆に PT-handle を除去す

(a) 非巡回 EFC ワークフローネット N_1

(b) N_1 をリファクタリングした非巡回 WS ワークフローネット N_1'

図 4 ϕ_{Best} によるリファクタリング例⁽³⁾

図 5 $(N_1, [p_I])$ と $(N_1', [p_I])$ の分岐双模倣関係⁽³⁾

(a) TP-cross 構造

(b) ϕ_{Desel} による変換処理後

図 6 ϕ_{Desel} の適用例⁽⁴⁾

るが，同様に適用前後のネットが分岐双模倣性を保持し，適用により新たに TP-handle を生成しない．最終的に ϕ_{Desel} を有限回適用したものは適用前の EFC ワークフローネットと分岐双模倣である WS ワークフローネットとなる．

ワークフローネット中の EFC または TP-cross 構造を検出し， ϕ_{Best} , ϕ_{Desel} を適用することによってリファクタリングを行うアルゴリズムの詳細は文献 (3) を参照されたい．

4.2 冗長性を除去するアプローチ

ワークフローはビジネス上の変更に伴って形を変えていくが，その過程で設計者の意図しない冗長性が盛り込まれてしまうことがある．図 7 は当初，限定会員向けの発送サービスのワークフローネットが非会員向けに拡張されたケースである．単純な例ではあるが，このような変更は業務拡大に伴って日常的に発生すると考えられる．図 7 において p_3 は振舞いの点でタスクの完了に無関係なプレースであり冗長である．このようなプレースは Implicit Place と呼ばれる．

[定義 9] (Implicit Place⁽¹⁷⁾) ペトリネット (N, M_0) において，プレース p は以下を満たすとき，かつそのときに限り Implicit Place である： $\forall t \in p^\bullet, \forall M \in R(N, M_0) : M \geq \bullet t \setminus \{p\} \Rightarrow M \geq \bullet t$ ． $\square$

ワークフローネットから Implicit Place を除去しても，その前後のネットは分岐双模倣である⁽⁴⁾．Implicit Place を除去した結果が WS ワークフローネットとなっている場合は，適用後のネットはリファクタリング結果である．Implicit Place 除去の前後のネットが分岐双模倣関係にあることの例を図 8 に示す．図 8(b) のネットは WS ワークフローネットであり，本例の結果はリファクタリングとなっている．

具体的な Implicit Place を検出する多項式時間アルゴリズム

(a) 会員向け発送システムを表すワークフローネット N_1 .

(b) N_1 を非会員向けにも拡張したワークフローネット N_2 (p_3 が Implicit Place).

図7 ワークフローネットの拡張による Implicit Place の発生⁽⁴⁾

(a) Implicit place p_3 を持つワークフローネット N_2 .

(b) N_2 から p_3 を除去したワークフローネット N_3 .

(c) ($N_2, [p_I]$) と ($N_3, [p_I]$) の分岐双模倣関係.

図8 Implicit place 除去と分岐双模倣関係⁽⁴⁾

や、除去によるリファクタリングアルゴリズムは文献(4)を参照されたい。

4.3 統合アルゴリズム

前節までに二つのアプローチを紹介したが、これらを一連の処理として統合し、より一般的なワークフローネットを扱うことのできるアルゴリズムが望まれる。文献(18)では、前処理として Implicit Place を除去した後に $\phi_{Best}, \phi_{Desel}$ を適用するアルゴリズムを提案している。

5. リファクタリングの応用例

本章ではリファクタリングによって解析技法の適用が可能になる例を紹介する。

5.1 可達性問題

可達性問題とは、ワークフローネットのある特定の状態が与えられたときに、それが初期状態から到達可能か否かを決定する問題である。定義5より、健全なワークフローネットにおいては初期状態 $[p_I]$ から到達可能な全ての状態は、終了状態 $[p_O]$ へ到達可能でもある。

実務との対応から考えると、本問題の重要性が明確になる。あるワークフローにおいて、(複数個の場合もある)ワークフローインスタンスを保持したまま定義が変更された場合を想定する。多くのステークホルダーが関わるシステムでは、進行中の案件を中止させることは困難であり、システム変更の際にも切れ目なく処理を継続させなければならないことが、上記の事態を起こす背景である。一般にこのような変更は不具合を発生させるリスクが大きく、処理不能となるインスタンスが発生することが危惧される。このとき、移行後のインスタンスが、初期状態から移行後の状態へ遷移することができることが分かれば、その状態は終了状態へ遷移可能であることから構造変更は問題なく行われたと判断することができる。

非巡回 WS ワークフローネットの可達性問題は、次の性質2により多項式時間で解くことができることを明らかにしている。
[性質2] (可達性判定⁽¹⁹⁾) N を非巡回 WS ワークフローネットとする。以下の二つの条件が成り立つことが、マーキング M が初期マーキング $[(p_I)^k] (k \in \mathbb{N})$ から到達可能であるための必要十分条件である (i) $A_{\bar{N}}$ を $\bar{N}$ の隣接行列とする。 $\vec{M}, [(p_I)^k]$ はそれぞれのマーキングのトークン分布の列ベクトル表現である。このとき方程式 $A_{\bar{N}} \cdot X = \vec{M} - [(p_I)^k]$ が有理数解を持つ。
(ii) M は $\bar{N}$ の全ての空でないトラップ (trap)⁽⁷⁾ をマークしている。 □

性質2の結果は、対象のネットが WS ワークフローネットであることを前提としているため、残念ながら EFC ワークフローネットには適用できない。しかしリファクタリング技術により、EFC ワークフローネットを WS ワークフローネットへ変換すれば、性質2の結果を適用することが可能になる。

図9はあるWebサイトにおける会員登録処理を表すワークフローで、健全な非巡回 EFC ワークフローネット N_{reg} に “Data

図9 会員登録処理ワークフロー N_{reg} への処理追加⁽³⁾.

図10 リファクタリング結果 N_{reg}' とマーキング $M_{reg}^{(3)}$

Clear” という新たな処理が追加される状況を表している．このとき処理追加後の N_{reg} のマーキング $M = [p_1^2, p_2^2, p_8^2, p_9^2]$ が初期状態 $[(p_1^2)^2]$ から到達可能であることを検証したい^(注2)． N_{reg} は EFC ワークフローネットのため，性質 2 の解析技法を活用するにはリファクタリングして健全な非巡回 WS ワークフローネットに変換する必要がある．リファクタリングは文献 (18) の “3 gate Refactoring” アルゴリズムを用いて行うことができる．同アルゴリズムは Implicit Place や TP-cross 構造に対しても検出と除去を行うが本例ではどちらも構造も存在しないため，EFC 構造のみの除去となる．“3 gate Refactoring” は t_4^2, t_5^2, t_7^2 及び p_3^2, p_5^2, p_6^2 から成る EFC 構造を検出して ϕ_{Best} を適用し，分岐双模倣である構造と置き換える．変換後の WS ワークフローネット N_{reg}' を図 10 に示す．

(注2)：健全性の定義から， M が初期マーキングから到達できれば， M は必ず終了状態に到達する発火系列を持つ．実用上必要となるのは終了状態への到達性検証である．

ϕ_{Best} 操作前後の ワークフローネットは分岐双模倣関係であることにより，元の ワークフローネット N_{reg} と N_{reg}' も分岐双模倣つまり同じ振舞いを持つ．

性質 2 の解析技法を適用する．最初に次の方程式を解く．

$$\mathbf{A}_{N_{reg}'} \cdot \mathbf{X} = \vec{M}' - [(p_1^3)^2]$$

このとき方程式は整数解 $\mathbf{X} = (2, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0)$ を持つ．次にマーキング M' が N_{reg}' の空でないトラップをマークしていることを確認する． $\overline{P_{reg}'} \setminus \{p_1^3, p_2^3, p_8^3, p_9^3\} = \{p_1^3, p_3^3, p_4^3, p_5^3, p_6^3, p_7^3, p_{10}^3, p_O^3\}$ における極大トラップ $\overline{Q_{reg}'}$ は空集合である．ゆえに M' は全ての空でないトラップをマークしている．性質 2 の 2 条件を満たしているため， M' は $[(p_1^3)^2]$ から到達である． N_{reg}' と N_{reg} は分岐双模倣関係にあるため，この結果は M が $[(p_1^2)^2]$ から到達であることと等価である．実際に N_{reg} において，発火系列 $t_1^2 t_2^2 t_3^2 t_4^2 t_5^2 t_7^2 t_1^2$ が存在することが確かめられる．

リファクタリング可能な非巡回 EFC ワークフローネットであれば，本例のように解析技法の適用が可能になる．

5.2 状態数計算

ペトリネットは状態遷移システム的一种であり，その諸性質の解析には SPIN⁽²⁰⁾ (21) に代表されるモデル検査技術の活用が進んでいる．一方でモデル検査ツールで扱える状態数には計算機リソース上の上限があり，一定数以上の状態数を持つワークフローネットは検証することは難しい．ツール適用前に状態数 $|R(N, [p_I])|$ を知ることができれば，構造を分割する等の対策を講じることができる．この問題に対して bridge のない WS ワークフローネットの状態数を計算する多項式時間アルゴリズムを提案している⁽²²⁾ (23)．

一方 Implicit Place を用いたリファクタリングは，適用前後の状態数を変えないという結果を文献 (24) で得ている．

[性質 3] (状態数不変⁽²⁴⁾) N を健全な非巡回 EFC ワークフローネットとする， N' が N から Implicit Place を取り除くことで得られた WS ワークフローネットならば $|R(N, [p_I])| = |R(N', [p_I])|$ である． □

この性質に基づけば，リファクタリング後の WS ワークフローネットが所定の条件 (bridge レス) を満たせばその状態数を計算することが可能になる⁽²⁵⁾．つまり従来計算することが不可能であった，元の EFC ワークフローネットの状態数を得ることができる．

5.3 Response Property のチェック

Response Property⁽²⁶⁾ は事象の生起順序に関わる仕様の一種であり，ある事象 A が成立すれば必ず B が起こることを検証するものである．一例を挙げれば，「100 万円以上の物品購入が起票されれば，必ずマネージャ承認を得る」などである．これはワークフローネットでは，あるトランジション A が発火した後に必ずトランジション B が発火することに相当する．

文献(27)では WS ワークフローネットの Response Property 検証アルゴリズムを提案している。リファクタリング技術を活用することにより、WS ワークフローネットを超えたより広いワークフローネットへ適用することが可能となる。

6. システム開発への応用

本章ではリファクタリング技術の適用場面と利用可能なツールについて紹介する。

6.1 適用場面

ワークフローネットはワークフローの論理的振舞いを抽象化したモデルであり、ワークフロー管理システムの実装コードやデータベースなどの詳細設計に関する情報は持たない。リファクタリング技術と関連する解析アルゴリズムの適用は必然的に、要求定義と機能設計という上流部分が中心となる。図 11 にそのイメージを示す。具体的には、開発初期段階でワークフローネットのモデルを構築し、リファクタリングと解析アルゴリズムでその妥当性を確認する。これによりワークフローの論理的正しさに関する品質が担保されるため、後続の内部設計以降の各フェーズで発生したであろう後戻りを抑制することができる。

蛇足になるが、要求定義や機能設計の行為そのものはアジャイル等の開発モデルにおいても普遍的に存在するため、リファクタリング技術は古典的なウォーターフォールモデルでの活用に止まらず、上流設計全般においても有効に機能すると考える。

6.2 実行環境

リファクタリングを適用するには、計算機上にワークフローネットのモデルを実現する必要がある。

ワークフローネットのモデリングツールとしては、ドイツのカルスルーエ大学で開発されている WoPeD⁽²⁹⁾や、オランダのアイントホーフェン工科大学を中心に開発されている

図 12 YAWL のモデル作成画面 (図 4(a) の例に対応)

YAWL⁽³⁰⁾ 等がある。特に YAWL の実装系はワークフロー管理システムとしても動作するため、今後の実証研究のプラットフォームとして期待できる (図 12)。これらはいずれも GUI エディタを用いて XML ベースのワークフローモデル (前者は PNML⁽³¹⁾, 後者は YAWL) を生成できる。それらのワークフローモデルを使用し、リファクタリングと解析を行うことが可能になる。

WoPeD や YAWL は LGPL の下で開発されており、商用利用には一定の制約がある。一方で市場には企業によって開発されたワークフロー管理システムが流通している。ワークフローモデルの定義言語は複数の団体が標準を策定しているが、有力なものに、OASIS により策定された BPEL⁽³²⁾, OMG 下で管理されている BPMN⁽³³⁾, また WfMC による XPD⁽³⁴⁾ などがある。これらの表現形式に対応した商用ワークフローは、ワークフローモデルを抽出できるため、WoPeD/YAWL と同様にリファクタリングを適用することができる。

以上のようにワークフロー管理システムのインターフェースは研究者/開発者に対して開かれており、リファクタリング技術が活躍できる土壌は整っている。今後はパイロットプロジェクトによる効果測定や、レガシーシステムのモデルの解析を通じて普及を進めたい。

7. おわりに

本解説ではペトリネットを用いたワークフローのリファクタリングについて、その理論と応用事例を紹介した。冒頭ではワークフローのモデルとなるペトリネットのサブクラス、ワークフローネットと関連する重要な諸性質をまとめて更にワークフローネットのリファクタリングを定義し、適用条件とその手続きを示した。リファクタリングの応用事例としては、可達性解析・状態数計算・Response Property 検証の 3 例を紹介した。これらはいずれも EFC ワークフローネットを、リファクタリングアルゴリズムによって WS ワークフローネットに変換することで初めて解析可能になるケースである。

更に後段では、ワークフロー管理ツール YAWL を中心に、システム開発との橋渡しに関する情報を提示した。

最後にワークフローネットのリファクタリングの理論面と応用面の将来課題を述べることでまとめとしたい。

理論面で最も重要な課題は、リファクタリング可能性の必要十分条件の証明である。どのような EFC ワークフローネット

図 11 ワークフローリファクタリングの適用フェーズ (文献(28)を参考に作成)

図 13 リファクタリング可能性の必要条件と十分条件の関係。

がリファクタリング可能であるかを知ることは実用面からも大きな関心事である。

本解説執筆時点で得られている必要条件（健全性）と十分条件（ハンドル・Implicit Place）の間には図 13 のように隔たりがあり，これらの間隙に存在する必要十分条件を見出していく．また将来的にはリファクタリング結果の評価も必要である^(注3)．評価基準としてはブレース/トランジションの数や辺の複雑さが考えられるが，応用面に配慮すると，可読性に関する基準を設けることも必要であろう。

実用面の課題は，システム開発における適用場面の確立である．ワークフローのリファクタリングが最も有効に機能するフェーズをユースケースとともに提示することができれば，実務者による試行が容易になり，最終目的である開発コストの抑制の可能性が高まる．今後ワークフロー管理ツールとの連携と並行して進めていきたい。

ペトリネットによるワークフローのリファクタリングは，普及の前提条件が整いつつある．今後は実務者とシステム数理研究者の多彩な参入を期待したい。

文 献

- (1) ワークフローの実践，電気学会ワークフロー調査専門委員会（編），日科技連，1999．
- (2) W.M.P. van derAalst and K.M. vanHee, Workflow Management: Models, Methods, and Systems, MIT Press, 2002.
- (3) I. Toyoshima, S. Yamaguchi, and Y. Murakami, “Two sufficient conditions on refactorizability of acyclic extended free choice workflow nets to acyclic well-structured workflow nets and their application,” IE-ICE Trans. Fundamentals, vol.E98-A, no.2, pp.635–644, Feb. 2015.
- (4) I. Toyoshima, S. Yamaguchi, and J. Zhang, “Implicit places and refactoring in sound acyclic extended free choice workflow nets,” IEICE Trans. Fundamentals, vol.E99-A, no.2, pp.502–508, Feb. 2015.
- (5) S. Yamaguchi, “Refactoring problem of acyclic extended free-choice workflow nets to acyclic well-

structured workflow nets,” IEICE Trans. Inf. & Syst., vol.E95-D, no.5, pp.1375–1379, May 2012.

- (6) 戸田保一，飯島淳一，速水治夫，堀内正博，ワークフロー ビジネスプロセスの変革に向けて，日科技連，1998．
- (7) T. Murata, “Petri nets: properties, analysis and applications,” Proc. IEEE, vol.77, no.4, pp.541–580, 1989.
- (8) C. Cassandras and S. Laforture, Intruduction to Discrete Event System, Springer Verlag, 2007.
- (9) W.M.P. van der Aalst, “The application of petri nets to workflow management,” J. Circuits Syst. Comput., vol.8, no.1, pp.21–66, 1998.
- (10) J. Esparza and M. Silva, “Circuits, handles, bridges and nets,” Lect. Notes Comput. Sci., vol.483, pp.210–242, Springer-Verlag, Berlin, 1991.
- (11) K.M. vanHee, N. Sidorova, and M. Voorhoeve, “Soundness and separability of workflow nets in the stepwise refinement approach,” ICATPN, W.M.P. van der Aalst and E. Best, eds. Lect. Notes Comput. Sci., vol.2679, pp.337–356, Springer, 2003.
- (12) J.E. J. Desel, Free Choice Petri Nets, Cambridge University Press, 2005.
- (13) S. Yamaguchi, M. Yamaguchi, and M. Tanaka, “A model checkin method of soundness for acyclic workflow nets using the spin model checker,” Int. J. Information, vol.1, no.12, pp.163–172, 2009.
- (14) S. Yamaguchi, “On projection inheritance between acyclic extended free-choice workflow net and its subnet,” Proc. IEEE TENCON 2012, no.1, pp.1–6, 2012.
- (15) M. Fowler, Refactoring : Improving the Design of Existing Code, Addison Wesley, 1999.
- (16) E. Best and M.W. Shields, “Some equivalence results for free choice nets and simple nets and on the periodicity of live free choice nets,” Lect. Notes Comput. Sci., vol.159, pp.141–154, 1983.
- (17) W.M.P. van der Aalst and T. Basten, “Inheritance of workflows: an approach to tackling problems related to change,” Theor. Comput. Sci., vol.270, no.1-2, pp.125–203, 2002.
- (18) I. Toyoshima, S. Yamaguchi, and J. Zhang, “A refactoring algorithm of workflows based on petri nets,” Proc. The 4th International Congress on Advanced Applied Informatics (AAI), no.1, pp.79–84, 2015.
- (19) S. Yamaguchi and T. Hirakawa, “Polynomial time verification of protocol inheritance between acyclic extended free-choice workflow nets and their subnets,” IE-ICE Trans. Fundamentals, vol.E96-A, no.2, pp.505–513, Feb. 2013.
- (20) G.J. Holtzman, The Model Checker SPIN, Addison Wesley, 2003.
- (21) <http://spinroot.com/>
- (22) T. Susaki and S. Yamaguchi, “On process tree based calculation of the number of states in petri nets,” Proc. ITC-CSCC 2013, pp.84–87, 2013.
- (23) S. Yamaguchi, “The state number calculation problem for free-choice workflow nets is intractable,” Proc. ITC-CSCC 2014, pp.838–841, 2014.
- (24) J. Zhang, S. Yamaguchi, and I. Toyoshima, “On implicit place and its application to refactoring in acyclic extended free choice workflow nets,” Proc. ITC-CSCC 2015, pp.264–267, 2015.
- (25) S. Yamaguchi and M.A.B. Ahmadon, “Properties and decision procedure for bridge-less workflow nets,” IE-ICE Trans. Fundamentals, vol.E99-A, no.2, pp.509–512, Feb. 2016.
- (26) M.B. Dwyer, G.S. Avrunin, and J.C. Corbett, “Patterns in property specifications for finite-state verification,” Proc. the 21st International Conference on Software Engineering, ICSE ’99, pp.411–420, ACM, New York, USA, 1999.
- (27) M.A.B. Ahmadon and S. Yamaguchi, “State number calculation problem of workflow nets,” IEICE Trans. Inf. & Syst., vol.E98-D, no.6, pp.1128–1136,

(注3)：ワークフローネットの変換技術は近年研究が進んでおり，代表的なものに Polyvyanyy ら⁽³⁵⁾のものがある．相互の数学的比較は今後の課題である．

June 2015.

- (28) 浅井, 実践的ソフトウェア工学-実践現場から学ぶソフトウェア開発の勘所, 石田 (監修), 近代科学社, 2009.
- (29) <http://woped.dhbw-karlsruhe.de/woped/>
- (30) <http://www.yawlfoundation.org/>
- (31) <http://www.pnml.org/>
- (32) <https://www.oasis-open.org/committees/>
- (33) <http://www.omg.org/spec/BPMN/2.0/>
- (34) <http://www.wfmc.org/standards/xpdl>
- (35) A. Polyvyanyy, L. García-Bañuelos, D. Fahland, and M. Weske, "Maximal structuring of acyclic process models," *Comput. J.*, vol.57, no.1, pp.12–35, 2014.

(MSS 研究会提案, 平成 28 年 2 月 2 日受付)

豊嶋伊知郎 (正員)

平 16 早大・教育・数学卒・平 18 同大学院理工学研究科修士課程了。同年 (株) 東芝入社, 同社研究開発センターシステム技術ラボラトリーにて形式手法, ネット理論の研究に従事。平 27 山口大大学院理工学研究科博士後期課程了, 工博。現在同社 社会インフラシステム社にて電力需給管理システムの研究開発に従事。平 27 から本会システム数理と応用

研究専門委員会幹事。

山口真悟 (正員: シニア会員)

平 4 山口大・工・電子卒。平 6 同大学院工学研究科博士前期課程了。平 9 同大学院理工学研究科博士後期課程退学, 同大学・工・助手。現在同大学院理工学研究科准教授。工博。ネット理論, コンカレント工学, ソフトウェア工学などの研究に従事。平 19 米国イリノイ大シカゴ校訪問研究員。平 19-22 年度コンカレント工学研究専門委員会幹事。平 26 年度から本会基礎・境界サイエティ和・英論文誌編集委員会分野編集幹事。