

# 資源のライフサイクルを意識した組織間ビジネスプロセスモデル のための オブジェクト指向ペトリネット開発支援 Object-Oriented Petri Net Development Support for Inter-Organization Business Process Model Conscious of the Life Cycle of Resources

藤巻 亮太<sup>†</sup>, 飯島 正<sup>‡</sup>

Ryota FUJIMAKI<sup>†</sup>, and Tadashi IIJIMA<sup>‡</sup>

<sup>†</sup>慶應義塾大学大学院 理工学研究科

<sup>‡</sup>慶應義塾大学 理工学部

<sup>†</sup>Graduate School of Science and Technology, Keio Univ.

<sup>‡</sup>Faculty of Science and Technology, Keio Univ.

## 要旨

業務プロセス管理において、時間制約の概念は重要な要素の一つである。著者らは、オブジェクト指向ペトリネットというモデルにより業務プロセスを記述し、そのモデルに時間表現を導入し例外処理の機能を追加することに試みている。ここでは、オブジェクト指向ペトリネットを導入したビジネスプロセスモデルとタイムアウト処理機能の開発の試みについて報告する。

## 1. はじめに

企業において業務改善を行うために、情報システムを用いた業務プロセス管理 (BPM; Business Process Management) は重要である。業務プロセスの自動化においては、タイムアウト処理いわゆる時間制約のルールを考慮する機会が多い。そこで本研究では、業務のモデリングの際に定義した時間制約のルールを対象を絞り、著者らが提案する業務プロセス表記法に対して時間表現と例外処理を導入することを示す。

近年、企業活動の範囲が広がり、複数の組織が連携して業務プロセスを実行することが多く、業務プロセスは巨大で複雑なものになる。この状況を踏まえ、これまで著者らは、組織間の協調と相互作用を扱うこと、ならびにアクター (実行者/サブシステム) を資源として扱うことを目的として、「オブジェクト指向ペトリネット」というモデルを業務プロセスの表記法として導入してきた。このモデルを用いることで、プロセスを構成要素の役割・機能ごとに分割してモジュール性を持たせ、それらの各要素同士の相互作用の記述のしやすさを示してきた [1][2]。

本研究では、著者らが提案する業務プロセスの表記法であるオブジェクト指向ペトリネットに対して、時間制約の表現を導入し、例外処理の機能を追加することを目標としている。

続く第2節では、対象とする時間制約のルールについて、第3節では、オブジェクト指向ペトリネットについて記述する。さらに第4節では、例外処理の表記方法について述べ、第5節で具体的な事例を示し、第6節で今後の課題について述べる。

## 2. 時間制約に関するルール

業務プロセスを効率よく管理するために、近年ではプロセスからルールを独立させて管理する業務ルール管理 (BRM; Business Rule Management) が重視されている。全てのルールは「定義付けのルール (構造ルール)」または「行動ルール (運用ルール)」のどちらかに分類される [5]。

定義付けのルールには、たとえば特定区間の運賃を定義するものがある。著者らはこれまで定義付けのルールに対し、if-then ルール形式の DSL [4] により表現し、さらにそれに加え、決定表を導入することで、業務ルールというドメインの知識の共有を実現するとともに、業務プロセス管理の効率化を実現してきた [3]。

一方、行動ルールとは、行為や行動に関する制約を表現することが多く、必要に応じて、判断や行動に関する評価をも含む。当事者の行為や行動を規制する制約条件の中で、時間制約のルールは重要であ

る。業務プロセスの自動化には、タイムアウト処理はほぼ不可欠といえるからである。そこで本研究では、時間制約に関する業務ルールを対象を絞って、オブジェクト指向ペトリネットによる業務プロセス表現に時間表現を導入し、例外処理を加えることを試みている。

### 3. オブジェクト指向ペトリネット

オブジェクト指向ペトリネットを業務プロセスを表現するモデルとして用いる。オブジェクト指向ペトリネットとは、オブジェクト指向の概念に基づいてモジュール性を取り入れたペトリネットの拡張モデルであり、著者らは、nets-within-nets 意味論に基づく参照ネット (Reference Net)[6] の考え方を採用している。ペトリネットで業務プロセスを記述する先行事例に YAWL[7] があるが、本研究は、この nets-within-nets 意味論に基づくモジュール性の導入に特徴がある。このモデルの例を示す (図 1)。

nets-within-nets 意味論 [6] とその拡張に基づくオブジェクト指向ペトリネットは次の 2 種類のトークンを持つ。

- 単純トークン (Black Token)
 

通常の P/T ネット (Place/Transition ネット) におけるトークンのことである。プレースに存在する単純トークンは、トークン数で示す。
- 参照トークン (Reference Token)
 

別のサブシステムを表現するサブネットへの参照 (reference) を持つトークンのことである。

図 1 の例では、クラス A のインスタンスへの参照を持っているトークンが参照トークンであり、クラス A のインスタンスを表現するペトリネットのプレース上に存在するトークンが単純トークンである。

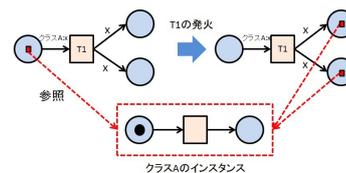


図 1: 参照トークンの例

この例で、参照トークンを持つ側のネットをシステムネットと呼び、参照トークンが参照している先のサブネットをオブジェクトネットと呼ぶ。このオブジェクトネットは、一つのオブジェクト (インスタンス) として識別される。

参照トークンも単純トークンと同様、ネットワーク中を遷移する。参照トークンがトランジションの発火によって複数個にコピーされる場合は、オブジェクトネットの実体ではなく、それへの参照がコピーされる点に注意して欲しい。

また、オブジェクトネットおよびシステムネット同士で、トランジションを同期発火 (interaction) の関係を持たせることも可能となっている。この同期発火が可能となるのは、全ての同期トランジションが発火可能な状態であることが条件である。nets-within-nets 意味論 [6] とその拡張に基づくオブジェクト指向ペトリネットにおいては、(a)interaction(相互作用):システムネットとオブジェクトネットが同期して遷移する場合、(b)transport(移送):システムネットの発火に際し同期して発火するオブジェクトネットがない場合、(c)autonomous(自律):システムネットとは独立に、オブジェクトネット内のトランジションが発火する場合の 3 通りの発火則が考えられる。

これらの特徴から、オブジェクト指向ペトリネットを業務プロセス表現を用いることで、以下が可能になる。

- 組織間にまたがる業務プロセス (inter-organizational business process) における組織間の協調表現
- アクター (実行者/サブシステム) を資源としてモデルに表現

### 4. 例外処理の表記法

ビジネスにおいて、予想外のことは日常茶飯事である。確固たるワークフローが確立されている場合であっても、様々な内部要因や外部要因によって例外の事態になることが多い。例えば、出張旅費を申請す

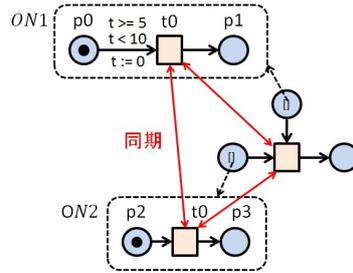


図 2: オブジェクト指向ペトリネットの例

るといふ単純なプロセスにおいても、領収書をなくしてしまうことや上司が休暇中であるといった状況が考えられる。ビジネスプロセスのモデリングにおいて、このような例外事態を無視することはできない。また、例外事態を表すだけでなく、例外が起きた場合の処理についても含めるべきだと考えた。本研究では、タイムアウトの処理だけでなく、様々な例外処理に対応できるような表記法を考案する。

#### 4.1. 基本的な例外処理の表記法

考案する表記法の中でも基礎的な表記法について説明する。例外処理を表記するためには、プロセスの中のどの部分において例外が起こりうるかを示す必要がある。また、一つのトランジションではなく複数の一連のトランジションにおいて例外処理を行えるようにするため、四角で囲まれたトランジションが例外処理の対象であるということ定義する。この四角は複数設けることができ、一つの四角の中に四角を設けられることもできる。

例外処理はこの四角の外に書くことで通常のプロセスと区別する。例外が発生したことを示すために各トランジションから例外のプレースを経てから例外処理のプロセスに移動する。

また、この四角の中に、例外が起きてしまったトークンの置き場を設ける。例外処理が行われて、不要になってしまったトークンがこの置き場に集まる。

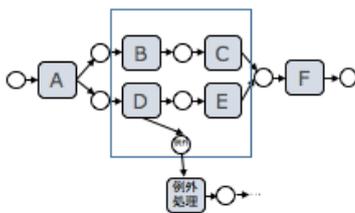


図 3: 基本的な表記法

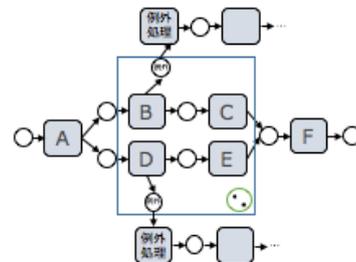


図 4: トークンの置き場

#### 4.2. ペトリネットへの時間表現の導入

あらゆるビジネスプロセスにおいて起こりうるタイムアウト処理に対応するために、時間表現をペトリネットに導入する。タイムアウトを判定するために、例外対象範囲を表す四角にも変数  $T$  を定義する。契約や状況によって一つ一つのトランジションに割ける時間と実際にかかる時間が違うため、トランジション一つずつに時間表現を設けるのではなく、例外対象の範囲に設ける。この  $T$  の数値と実際に過ぎている時間を比べ、 $T$  を超えてしまった場合に例外処理に移るとする。

#### 4.3. 状況別表記法

例外処理に当てはまるものはとても広範囲であり、一概に定義できないことが多い。そのため、あらゆる例外処理にも対応できるような表記法を考案する必要がある。そこで例外処理において多く発生すると考えられる状況について表記法を定義する。

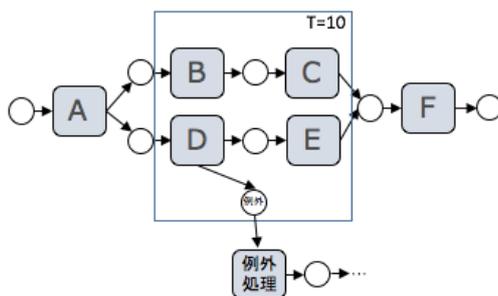


図 5: タイムアウト処理

例外処理において、処理を終えたあとにもとのプロセスや他のプロセスに戻るという状況が考えられる。また、処理の結果によっては、戻る場所が一つとは限らず複数の場合の可能性もあることも考えられる。よって、そのような場合は例外処理の時点で場合分けをし、戻るトランジションをそれぞれ指定する。

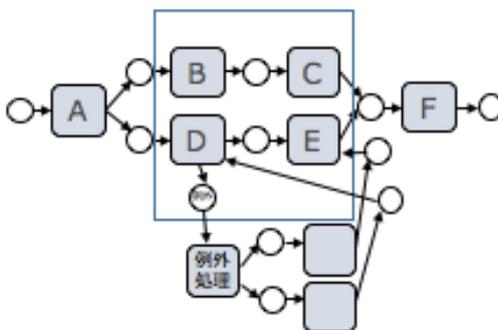


図 6: 戻る場所が複数の場合

一つの例外から戻る箇所が複数の場合もあるが、同じトランジションに複数の例外から戻るという状況も十分有り得る。このような場合を考慮し、四角の外から中に戻る前にプレースを経ることを定義する。これによって複数の例外からアークが一つのトランジションに集まるのではなく、一つの入口のようなプレースから一つのアークが届くだけで済むようになる。

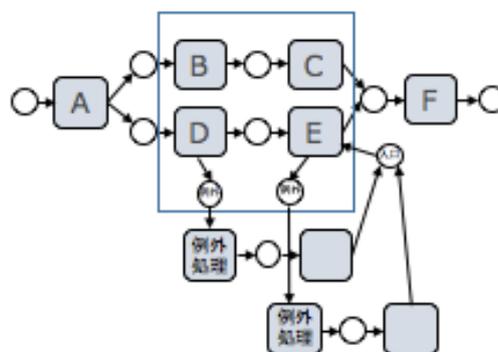


図 7: 複数の例外が一つに集まる場合

同様に、複数のトランジションから同じ種類の例外が起きることも予測できる。よって、このような場合は上記と同じように出口としての役割を果たす例外プレースを経て、四角の外の例外処理に移ることを定義する。

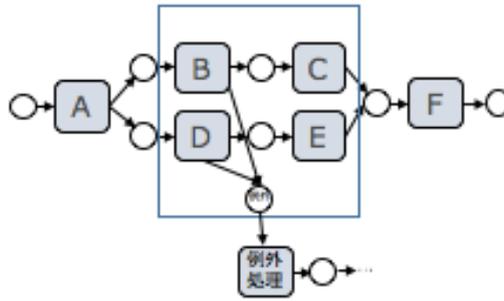


図 8: 複数のトランジションから同じ例外がある場合

## 5. 事例

本研究にて提案するビジネスプロセスモデルにおけるオブジェクト指向ペトリネットと例外処理の表記法を活用した具体的な例を示す。

### 5.1. オブジェクト指向ペトリネットの例

オブジェクト指向ペトリネットを導入した例として、企業における社員の出張旅費申請のプロセスを挙げる。企業における社員の出張旅費申請のプロセスにおいて、プロセスを構成する実体となるオブジェクトについて、社員、申請書、課長、財務部を定義した。それぞれをオブジェクトネットとして定義し、オブジェクト指向ペトリネットでもデル化すると、図 9 のように定義できる。定義した業務プロセスは次のような流れである。社員が出張旅費の申請書を作成し、申請書を課長に提出して、課長が受け付ける。課長は申請書の承認を行い、財務部へ申請書を提出して、財務部が受け付ける。財務部は申請書の内容を審査し、その結果を申請者である社員に通知し、社員が受信する。

オブジェクト指向ペトリネットを活用することによって、社員、申請書、課長、財務部などの資源やアクターを表現できるようになる。

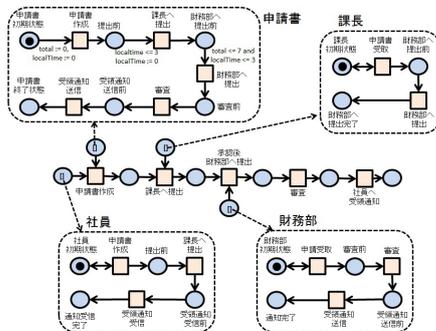


図 9: 出張旅費申請のプロセス

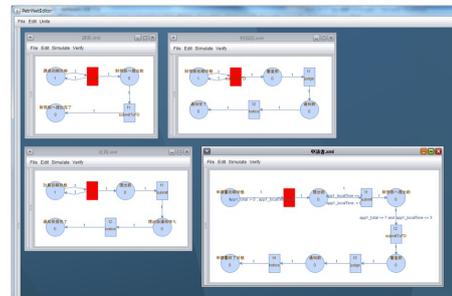


図 10: エディタ上でのオブジェクトネットの記述

### 5.2. 例外処理の例

例外処理の例としては、企業における製品の出荷のプロセスを挙げる。流れとしては、受注を受けたあと、在庫の引き当てを行い、出荷依頼と出荷、請求書発行を同時並行に行い、顧客に届ける。このビジネスプロセスをペトリネットでもデル化すると図 11 のようになる。ここで、考えられる例外として、タイムアウトと出荷した製品の中に欠品が出た場合を挙げる。タイムアウト処理に関しては、顧客の返答によって次のアクションが変わるため、場合分けをしてある。また、欠品の際の返品とタイムアウト処理における割引のあとは両方共請求書を発行しないとイケないため、戻る場所が同じであり、同じ入口から四角に入っている。このように例外が発生することだけでなく、発生後の処理についてもビジネスプロセスとしてモデル化が可能である。

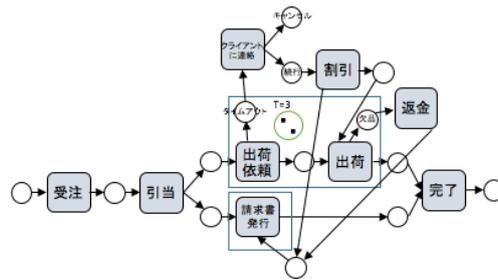


図 11: 例外処理の具体例

## 6. 今後の課題

今後としては、まずオブジェクト指向ペトリネットに例外処理の表記法を融合させる。それに伴い、通常のペトリネットでは発生しない、オブジェクト指向ペトリネット特有の例外に対応できる表記法も新たに定義する。表記法も複数の案を考案し、評価実験を行うことによって、ビジネス・プロセスにおいて最適な例外処理の表記法を定める。

## 参考文献

- [1] 飯島 正: “アスペクト指向ワークフロー変換 ～ オブジェクト指向ペトリネットによるワークフロー表現への適用 ～,” 技術報告 (知能ソフトウェア工学研究会), Vol.112, Vol.165, pp.1-6, 電子情報通信学会, 2012.
- [2] 飯島 正, 片山 輝彦, 金子 良太, 高橋 貴大: “オブジェクト指向論理ペトリネットを使った業務プロセス/業務ルール管理,” 第 8 回 全国大会・研究発表大会, 情報システム学会, 2012
- [3] 飯島 正, 秦 良平, 金子 良太: “オブジェクト指向ペトリネットとルールに基づく業務プロセスの理解支援,” 第 9 回 全国大会・研究発表大会, 情報システム学会, 2013
- [4] Debasish Ghosh, (監訳) 佐藤 竜一: “実践プログラミング DSL ～ ドメイン特化言語の設計と実装のノウハウ,” 翔泳社, 2012.
- [5] ロナルド・G・ロス: “アジャイル経営のためのビジネスルールマネジメント入門,” 日経 BP 社, 2013.
- [6] Rüdiger Valk: “Object Petri nets Using the nets-within-nets paradigm,” LNCS 3098, pp.819–848, Springer, 2004.
- [7] W.M.P. van der Aalst and A.H.M. ter Hofstede: “YAWL: Yet Another Workflow Language,” QUT Technical Report, FIT-TR-2002-06, Queensland University of Technology, Brisbane, 2002.
- [8] “SPIN”,  
<http://spinroot.com/spin/whatispin.html>