



技術解説

ペトリネットによる 離散事象システムのモデル化と制御

高井重昌*

Modeling and Control of Discrete Event Systems Using Petri Nets

Key Words : discrete event system, modeling, control, Petri net

1 はじめに

現代のシステムはコンピュータを中心に構成されているため、システム内の離散的な信号・情報に着目し、その動作を適切に解析・制御する必要がある。このような背景から、システムの振舞いの離散的な側面を取り扱う離散事象システムに関する研究がシステム制御、計算機科学などの分野で行われている。離散事象システムは、事象の生起により、状態が離散的に遷移するようなシステムの総称であり、離散事象システムとしてみなすことができるシステムの例として、生産システム、シーケンス制御システム、オペレーティングシステム、データベースシステム、通信システムなどがあげられる [1]。

離散事象システムの数学モデルとして、オートマトン、ペトリネット、及びそれらの拡張モデルなどが用いられているが、特にペトリネットは、システムの並行性、非同期性、分散性、非決定性といった離散事象システムの特性を図的に表現できる、という利点を有する。ペトリネットは 1962 年に、C. A. Petri の博士論文において、オートマトン間の通信機構のモデルとして提案され、当初は計算機科学の分野での理論的研究が中心であった。その後、離散事象システムのモデルとして注目されるようになり、システム制御の分野においても、理論・応用の両面

から活発に研究が進められた。現在では、ペトリネット理論はほぼ確立した段階となり、ペトリネットに基づいたシステム設計・解析のための様々なツールも開発されている。

本稿では、ペトリネットによる離散事象システムのモデル化、制御に関する基本事項について紹介することを目的とする。まず、システムを如何に離散事象システムとして捉えるかということ、簡単な生産ラインを例に説明する。そして、ペトリネットの定義、及びペトリネットモデルの例を述べる。最後に、ペトリネットを用いた離散事象システムの制御の一手法を紹介する。

2 離散事象システムとしての捉え方

システムを離散事象システムとして捉えるためには、離散状態、その遷移を引き起こす事象、および状態遷移規則を定める必要がある。ここでは、簡単な生産ラインを例にあげて、対象システムを如何に離散事象システムとして捉えるかを説明する。

例として、図1に示されるような生産ラインを考える。この生産ラインでは、部品はまずバッファ1へ到着する。そして、機械1で処理されてバッファ2へ運ばれる。さらに、機械2で処理された後に搬出されるとする。ただし、モデルの簡単化のため、機械1, 2はそれぞれ2個以上の部品を同時に処理することはないとし、バッファ1, 2の容量はともに無限とする。



図1：生産ライン



*Shigemasa TAKAI

1966年6月生

神戸大学大学院工学研究科システム工学
専攻修士課程 (1991年)現在、大阪大学 工学研究科 電気電子
情報工学専攻 教授 博士(工学) シス
テム理論

TEL : 06-6879-7693

FAX : 06-6879-7263

E-mail : takai@eei.eng.osaka-u.ac.jp

このシステムを以下のように離散事象システムとして捉える。まず、システムの状態として、機械の状態とバッファにある部品の個数に注目する。すると、システムの状態は機械1, 2の状態 x_{M1}, x_{M2} とバッファ1, 2にある部品の個数 x_{B1}, x_{B2} からなる4項組 $x = (x_{M1}, x_{M2}, x_{B1}, x_{B2})$ で表現できる。機械1, 2の状態として、

- I_1, I_2 : 機械1, 2が休止中
- W_1, W_2 : 機械1, 2が稼動中

を考えると、状態集合 X は

$$X = \{(x_{M1}, x_{M2}, x_{B1}, x_{B2}) \mid x_{M1} \in \{I_1, W_1\}, x_{M2} \in \{I_2, W_2\}, x_{B1}, x_{B2} \in \{0, 1, 2, \dots\}\}$$

となる。例えば、状態 $(W_1, I_2, 0, 2) \in X$ は、機械1が稼働中、機械2が休止中、バッファ1には部品がなく、バッファ2には2個の部品がある状況を表している。機械の状態 x_{Mi} ($i = 1, 2$) はシンボル値 I_i と W_i のいずれかをとり、バッファの状態 x_{Bi} は非負整数値をとる離散変数である。このように離散事象システムにおける離散状態は、一般にシンボル値や離散的な数値で表現できる。

システムの離散状態は、事象の生起に駆動されて遷移する。そこで、離散状態の遷移を引き起こす事象集合 Σ を $\Sigma = \{a, s_1, c_1, s_2, c_2\}$ とする。ここで、

- a はバッファ1への部品の到着
- s_1, s_2 はそれぞれ部品がバッファ1, 2から機械1, 2へ運ばれ処理開始
- c_1 は機械1が処理を終了し、部品をバッファ2へ搬送
- c_2 は機械2が処理を終了し、部品を搬出

をそれぞれ意味するとする。そして、これらの事象の生起による状態遷移は以下ようになる。

- 事象 a が生起すると

$$(x_{M1}, x_{M2}, x_{B1}, x_{B2}) \rightarrow (x_{M1}, x_{M2}, x_{B1}+1, x_{B2})$$
- $x_{M1} = I_1, x_{B1} > 0$ のとき s_1 が生起すると

$$(I_1, x_{M2}, x_{B1}, x_{B2}) \rightarrow (W_1, x_{M2}, x_{B1}-1, x_{B2})$$
- $x_{M1} = W_1$ のとき c_1 が生起すると

$$(W_1, x_{M2}, x_{B1}, x_{B2}) \rightarrow (I_1, x_{M2}, x_{B1}, x_{B2}+1)$$
- $x_{M2} = I_2, x_{B2} > 0$ のとき s_2 が生起すると

$$(x_{M1}, I_2, x_{B1}, x_{B2}) \rightarrow (x_{M1}, W_2, x_{B1}, x_{B2}-1)$$
- $x_{M2} = W_2$ のとき c_2 が生起すると

$$(x_{M1}, W_2, x_{B1}, x_{B2}) \rightarrow (x_{M1}, I_2, x_{B1}, x_{B2})$$

このように、離散状態、事象および状態遷移規則を定めることにより、生産ラインを離散事象システ

ムとして捉えることができる。

3 ペトリネット

本章では、離散事象システムを図的に表現できる数学モデルであるペトリネットを紹介し、その基礎的事項について述べる。

3.1 ペトリネットの定義

ペトリネットは4項組 $PN = (P, T, A, M_0)$ で定義される。ここで、

- P はプレースと呼ばれる節点の有限集合
- T はトランジションと呼ばれる節点の有限集合
- $A: (P \times T) \cup (T \times P) \rightarrow \{0, 1, 2, \dots\}$ はプレースとトランジションの間の接続を表す関数
- $M_0: P \rightarrow \{0, 1, 2, \dots\}$ は初期マーキング

である。システムの状態は各プレースに存在するトークン数で表現する。プレースにおけるトークン分布はマーキングと呼ばれる関数 $M: P \rightarrow \{0, 1, 2, \dots\}$ で表わされる。つまり、マーキング M のもとで、各プレース $p \in P$ には $M(p)$ 個のトークンが存在する。初期マーキング M_0 は、システムにおける初期のトークン分布であり、初期状態を表している。そして、状態遷移を引き起こす事象をトランジションで表現する。トランジションが発火（事象の生起に対応）することにより、マーキングが変化し、システムの状態遷移を表す。

図的には、

- プレース $p \in P$ は \circ で表現
- トランジション $t \in T$ は $|$ で表現
- プレース p とトランジション t の接続関係は $p(t)$ から $t(p)$ への $A(p, t)$ ($A(t, p)$) 本のアーク \rightarrow で表現
- トークンは \bullet で表現

される。ペトリネットの図的表現の例を以下に示す。

例1 ペトリネット $PN = (P, T, A, M_0)$ において、

$$P = \{p_1, p_2, p_3\}$$

$$T = \{t_1, t_2, t_3, t_4\}$$

$$A(p, t) = \begin{cases} 2 & \text{if } p = p_1, t = t_2 \\ 1 & \text{if } (p = p_1, t = t_1) \text{ or } (p = p_2, t = t_2) \\ & \text{or } (p = p_2, t = t_3) \text{ or } (p = p_3, t = t_4) \\ 0 & \text{otherwise} \end{cases}$$

$$A(t, p) = \begin{cases} 3 & \text{if } t = t_2, p = p_3 \\ 1 & \text{if } (t = t_1, p = p_1) \text{ or } (t = t_1, p = p_2) \\ & \text{or } (t = t_3, p = p_3) \\ 0 & \text{otherwise} \end{cases}$$

$M_0(p_1) = 2, M_0(p_2) = 1, M_0(p_3) = 0$
 ならば、初期マーキング M_0 のもとでの図的表現は
 図2となる。

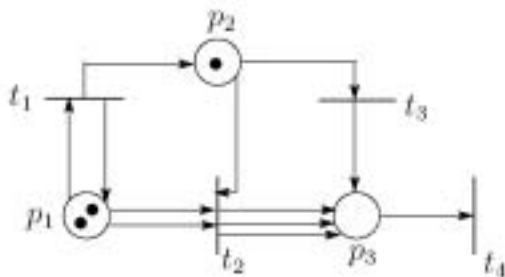


図2：ペトリネット

次に、システムの状態遷移を表現するための、トランジションの発火規則について説明する。マーキング M において、つぎの条件が満足されるとき、トランジション $t \in T$ は発火可能であるという。

$$\forall p \in P; M(p) \geq A(p, t)$$

つまり、各プレース $p \in P$ において、 t へのアークの数 $A(p, t)$ 以上のトークンが存在するならば、 t は発火可能である。例えば、図2においては、 t_1, t_2, t_3 が発火可能である。発火可能なトランジション t が発火するとマーキングが変化する。マーキングが M から M' に変化したとすると、

$$\forall p \in P; M'(p) = M(p) - A(p, t) + A(t, p)$$

が成り立つ。つまり、各プレース $p \in P$ において、 t へのアークの数 $A(p, t)$ だけトークンが取り去られ、 t からのアークの数 $A(t, p)$ だけトークンが加えられる。例えば、図2において t_1 が発火した場合を考える。 p_1 では、 $A(p_1, t_1) = 1$ 個のトークンが取り去られ、 $A(t_1, p_1) = 1$ 個のトークンが加えられる。 p_2 では、 $A(p_2, t_1) = 0$ のため、トークンは取り去られず、 $A(t_1, p_2) = 1$ 個のトークンが加えられる。また p_3 では、 $A(p_3, t_1) = A(t_1, p_3) = 0$ より、 t_1 が発火してもトークン数は変化しない。よって、 t_1 発火後のマーキングは図3のようになる。図3では、 t_1, t_2, t_3 が発火可能であるが、 t_2 が発火すると図4のようになる。

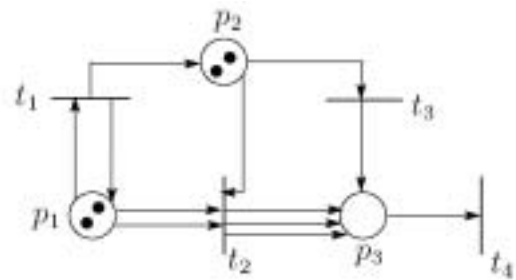


図3： t_1 が発火した後のマーキング

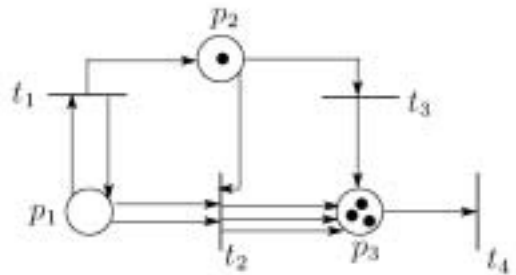


図4： t_2 が発火した後のマーキング

なお、ペトリネットでは、プレース内の各トークンは区別されず、その個数だけが意味をもつことに注意されたい。

注意1 各マーキングにおいて、発火可能なトランジションは一般に複数個存在しうる。本稿では、複数のトランジションの同時発火は考えていないが、発火規則を拡張することで、容易にトランジションの同時発火を扱うことができる。

3.2 生産ラインのモデリング例

2章で考えた生産ラインは図5のペトリネットでモデル化できる。ここで、各プレースとトランジションの意味は以下のとおりである。

- p_1 : バッファ1に存在する部品の数
- p_2 : 機械1が稼動中
- p_3 : 機械1が休止中
- p_4 : バッファ2に存在する部品の数
- p_5 : 機械2が稼動中
- p_6 : 機械2が休止中
- t_1 : 部品到着
- t_2 : 機械1の処理開始
- t_3 : 機械1の処理終了
- t_4 : 機械2の処理開始

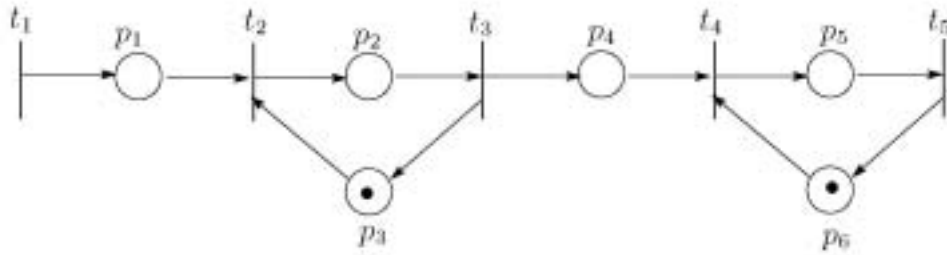


図5：生産ラインのペトリネットモデル

- t_5 ：機械2の処理終了と部品の搬出

バッファ1と2に存在する部品数は、それぞれプレース p_1 と p_4 のトークン数で表現できる。また、機械1の状態は、 p_3 にトークンが存在するときは休止中、 p_2 にトークンが存在するときは稼働中として表現できる。機械2の状態も同様に、 p_5 と p_6 に存在するトークンで表すことができる。図5のマーキングは、機械1と2が共に休止中であり、バッファ1と2が共に空である状況を表している。

注意2 2章では簡単のため、機械1,2はそれぞれ2個以上の部品を同時に処理することはないとしたが、図5のペトリネットにおいて、 p_3 のトークン数を n_1 、 p_6 のトークン数を n_2 とすると、機械1,2がそれぞれ、 n_1 個、 n_2 個まで部品を同時に処理できる場合を表現することができる。この場合、 p_2 (p_5)のトークン数は機械1(機械2)で処理されている部品の数を表し、 p_3 (p_6)のトークン数は、さらに機械1(機械2)で処理可能な部品数を表している。

また、2章の生産ラインの例では考えなかったが、二つの部品が機械によって組み立てられて、一つの部品になる処理は図6のように表現できる。さらに、部品が二つの機械のいずれかによって処理される場合は図7のように表現できる。上述の例にも示されたとおり、ペトリネットは生産システムと相性がよいモデルとして知られている。

最後に、離散事象システムのモデルとしてのペトリネットの利点をまとめておく。離散事象システムの振舞いは状態遷移図でも記述することができるが、状態遷移図ではシステムの状態が一つの節点で表現されるため、その物理的意味をとらえにくい。さらに状態遷移構造がシステムの物理的な構造を必ずし

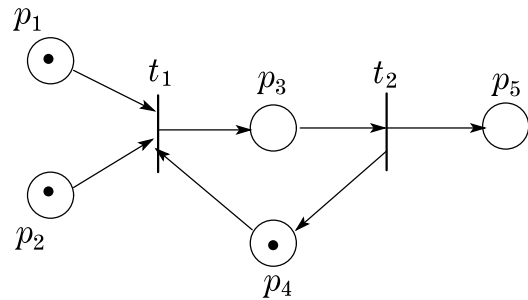


図6：二つの部品の組立て

も表すものではなく、状態遷移図を見ただけではシステムの構造やそこでの物・情報の流れを把握することは難しい。一方、ペトリネットでは、システムの状態はプレースにおけるトークン分布により表現するため、例えば上述の生産ラインの例において、プレース p_1 と p_4 のトークン数がバッファ1と2にある部品数を表す、といったように、状態の物理的意味をつかみやすい、という利点がある。また、ネット構造を用いることで、システムの構造を表現でき、さらにトークンの移動により、物・情報の流れを把握しやすく、シミュレーションなどにも適したモデルであるといえる。

4 ペトリネットの制御問題

離散事象システムにおいては、共有資源の排他制御やバッファのオーバーフローの回避といった制御問題が存在する。本章では、ペトリネットの枠組みにおいて、このような制御問題を解決する方法について述べる。プレースの個数を n とし、 $P = \{p_1, p_2, \dots, p_n\}$ としたとき、マーキングは n 次元の非負整数ベクトル

$$M = [M(p_1) \ M(p_2) \ \dots \ M(p_n)]^T$$

で表すこともできる。ここで、 T はベクトルの転置を表すとする。本章では以下、マーキング M はこのように n 次元列ベクトルとして表現されるとする。

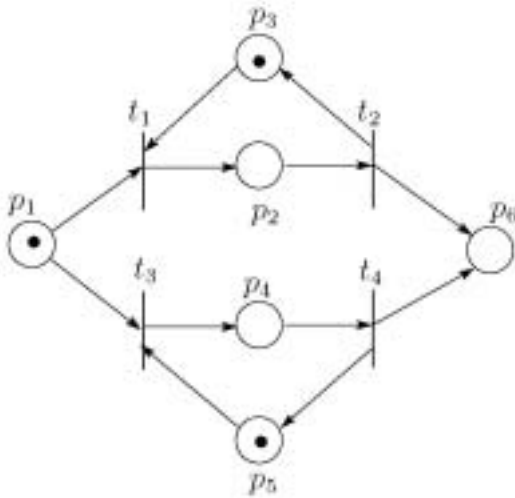


図7：二つの機械のいずれかによる処理

システムにおける制御仕様はトークンの重み付き和がある値を超えない、すなわち、

$$w^T M \leq b \quad (1)$$

の形式で与えられるとする。ここで、 w は重みを表す n 次元整数ベクトルであり、 b は正の整数である。例えば、上述の排他制御やバッファのオーバーフローの回避に関する制御仕様はこのような形式で与えられる。そして、(1) 式が満足されないようなマーキングに到達しないように、トランジションの発火を適切に制御することが制御問題である。なお、以下では、初期マーキング M_0 において (1) 式が満足される、つまり $w^T M_0 \leq b$ が成り立つとする。

ペトリネット PN におけるプレースとトランジションの接続関係を接続行列と呼ばれる $n \times m$ 行列 D_{PN} で表す。ここで、 n はプレースの個数、 m はトランジションの個数である。 D_{PN} の i, j 要素 $[D_{PN}]_{i,j}$ はつぎのように定義される。

$$[D_{PN}]_{i,j} = A(t_j, p_i) - A(p_i, t_j)$$

つまり、 $[D_{PN}]_{i,j}$ は、トランジション t_j の発火によるプレース p_i のトークン数の変化を表している。

(1) 式の制約が満足されるように、ペトリネット PN に制御プレースと呼ばれる新たなプレース p_c 、および p_c とトランジションの間のアークを付け加える。 p_c と各トランジションの間の接続を表す m 次元整数ベクトル d_c と制御プレースの初期トークン数 m_{c0} を次式で求める [2]。

$$d_c = -w^T D_{PN} \quad (2)$$

$$m_{c0} = b - w^T M_0 \quad (3)$$

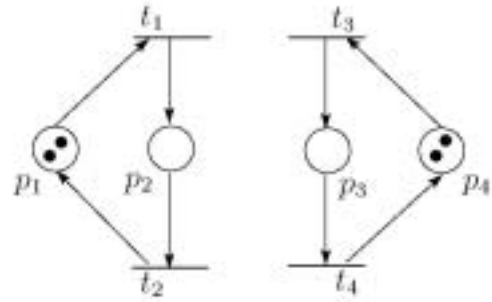


図8：書き込み/読出しプロセスのペトリネットモデル

ベクトル d_c の各要素 $[d_c]_j$ ($j = 1, 2, \dots, m$) は制御プレース p_c とトランジション t_j との接続を表し、 $[d_c]_j > 0$ のとき、 t_j から p_c へ $[d_c]_j$ 本のアークを付加し、 $[d_c]_j < 0$ のとき、 p_c から t_j へ $|[d_c]_j|$ 本のアークを付加する。 $[d_c]_j = 0$ のときは、 p_c と t_j の間にはアークは付加しない。

例として、共有ファイルにアクセスする書き込み/読出しプロセス [3], [4] に本章で述べた制御方法を適用する。書き込み/読出しプロセスのペトリネットモデルを図8に示す。ここで、各トランジションの意味は以下のとおりである。

- t_1 : 書き込みプロセスの書き込み開始
- t_2 : 書き込みプロセスの書き込み終了
- t_3 : 読出しプロセスの読出し開始
- t_4 : 読出しプロセスの読出し終了

つまり、プレース p_2 に存在するトークン数は書き込みを実行しているプロセスの数を、 p_3 に存在するトークン数は読出しを実行しているプロセスの数をそれぞれ表している。

ここでは、書き込みプロセス、読出しプロセスの数はそれぞれ2とし、二つの読出しプロセスは同時に読出しを実行することができるが、二つの書き込みプロセスが同時に書き込みを実行することは禁止する。また、読出しと書き込みが同時に実行されることも禁止する。これらの制約はマーキング M に関して、 $2M(p_2) + M(p_3) \leq 2$ 、つまり

$$[0 \ 2 \ 1 \ 0] \begin{bmatrix} M(p_1) \\ M(p_2) \\ M(p_3) \\ M(p_4) \end{bmatrix} \leq 2 \quad (4)$$

と (1) 式の形式で表現することができる。初期マーキングを $M_0(p_1) = M_0(p_4) = 2$, $M_0(p_2) = M_0(p_3) = 0$

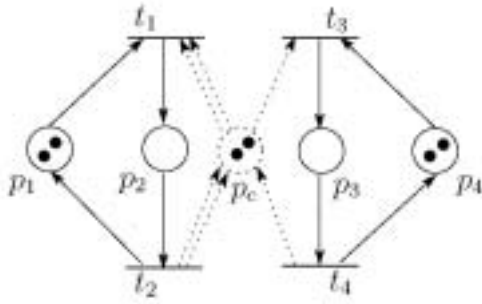


図9：制御プレースを付加したペトリネットモデル

とすると、 $2M_0(p_2) + M_0(p_3) = 0 \leq 2$ となり、(4) 式の制約が初期マーキング M_0 で満足されることがわかる。

図8のペトリネットの接続行列は

$$D_{PN} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

となる。(2) 式と (3) 式より、

$$d_c = [-2 \quad 2 \quad -1 \quad 1] \\ m_{c0} = 2$$

を得る。この結果に従い、制御プレース p_c 、および p_c とトランジションの間のアークを付加したペトリネットは図9のようになる。このペトリネットでは、制御プレース p_c により、もとの図8のペトリネットのトランジションの発火が制御される、と解釈することができる。

制御プレース p_c が付加されたペトリネットにおいて、制御プレースのトークン数を m_c と書くと、

$$w^T M + m_c = b \tag{5}$$

が成立する。 $m_c \geq 0$ であるから、(1) 式の制約が満足されることになる。図9のペトリネットにおいては、 $2M(p_2) + M(p_3) + m_c = 2$ となり、(4) 式の制約が満たされる。

5 おわりに

本稿では、事象の生起によってその状態が遷移するような離散事象システムの図的な数学モデルとしてペトリネットを紹介した。ペトリネットはほぼ確立された理論体系を有しており、日本語の教科書的な本も数冊出版されている [5], [3], [6], [4]。本稿では触れることができなかったペトリネットの性質、解析手法、その応用などに関しては、これら教科書等を参照されたい。

参考文献

- [1] C. G. Cassandras and S. Lafortune: Introduction to Discrete Event Systems, Kluwer Academic Publishers (1999)
- [2] K. Yamalidou, J. Moody, M. Lemmon, and P. Antsaklis: Feedback control of Petri nets based on place invariants, Automatica, vol. 32, no. 1, pp. 15 – 28 (1996)
- [3] 熊谷, 薦田: ペトリネットによる離散事象システム論, コロナ社 (1995)
- [4] 青山, 内平, 平石: ペトリネットの理論と実践, システム制御情報学会編, 朝倉書店 (1995)
- [5] 村田: ペトリネットの解析と応用, 近代科学社 (1992)
- [6] 奥川: ペトリネットの基礎, 共立出版 (1995)