

No. 746

ビジネスプロセスのペトリネットモデル

by

佐藤 亮

September 1997



# ビジネスプロセスのペトリネットモデル

筑波大学社会工学系

佐藤 亮

## abstract

Business transaction systems have discrete event nature, and they are usually described by data flow diagrams. In order to calculate and design dynamic properties such as the cycle time of a business system, business transaction Petri nets are defined as a specific class of timed Petri nets that are simulated by the business transaction systems corresponding to business transaction Petri nets. The result will provide a basis of a novel design method for the performance of business systems.

## Key words

business transaction system, business transaction Petri net, discrete event system, homomorphism, business process

## 1 はじめに

業務取引システム-Business Transaction System-は Sato and Praehofer (1997) によって提唱されたビジネスプロセスの離散事象モデルであり、同時に、広く情報システム方法論に用いられているデータフローダイアグラムによってその静的構造がモデル化される (Sato, 1997)。業務取引システムは Zeigler (1976, 1984) の DEVS - discrete event system specification - モデルを利用している。一方、ペトリネットも離散事象システムのモデルとして広く用いられていて、時間変化を扱う方法として時間ペトリネットも提案されている (村田, 1992)。

この論文では、カンバンシステムや MRP (Material Requirements Plannig) のような管理システムを組み込んだビジネスプロセスを、業務取引システムと時間ペトリネットによってモデル化できることを示す。業務取引システムを単純化した構造をもつ時間ペトリネットは、業務取引ペトリネットと呼ばれる。

業務取引ペトリネットによって、ビジネスプロセスのサイクルタイムなどの時間生産性(田中, 1993)の計算方法を業務取引システムにも利用できることが可能になる。

## 2 業務取引ペトリネット

業務取引システムの構成要素は、データフローダイアグラムで記述できるものである。以下に定める業務取引ペトリネットは、データフローダイアグラムに対応するペトリネットである。

一般的なペトリネットは次のように定められる。

### 定義1 ペトリネット(村田, 1992)

ペトリネットは組  $PN = (P, T, F, W, M_0)$  である, ここで

$P = \{p_1, p_2, p_3, \dots, p_n\}$  プレースの有限集合

$T = \{t_1, t_2, \dots, t_n\}$  トランジションの有限集合

$F \subseteq (P \times T) \cup (T \times P)$  アーク(矢印)の集合

$W : F \rightarrow \{1, 2, 3, \dots\}$  アークの重み

$M_0 : P \rightarrow \{0, 1, 2, \dots\}$  初期マーキング

$P \cap T = \phi$

特定のマーキング(トークンの配置)がされていないペトリネット構造(グラフ)は  $N$  で表わされ, 初期マーキング  $M_0$  が規定されているものは  $(N, M_0)$  で表わす。

ペトリネットの図的表現には図1のシンボルが使われる。プレースは円で, トランジションは線分で, アークはプレースとトランジションを接続する矢印で描かれる。

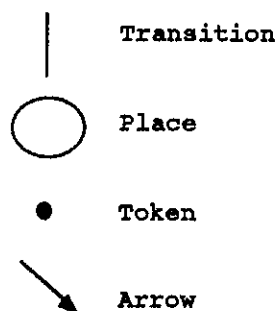


図1 Petri net の構成要素

図2は、ある瞬間の1台の機械による加工のペトリネットモデルである。アークの重みはアークの近くに数字で書かれる。重みが1の場合には省略する。図でトークンは加工待ちの材料に対応したり、また、機械が稼働中か空きかを示すために使われている。

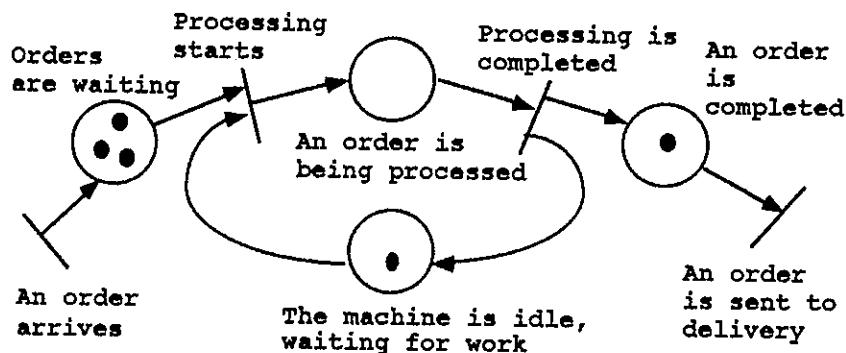


図2 機械加工のペトリネット

マーキングはトークンの各プレースへの分布（配置）である。プレースの集合から非負整数への写像と考えることができる。任意のマーキング $M$ も $M: P \rightarrow \{0, 1, 2, \dots\}$ なる関数と考えることができる。トランジションの発火で、トークンが重みの数だけ移動する。プレースからトランジションへ向かう矢印あるとき、そのプレースをそのトランジションの入力プレースという。またトランジションからプレースへの矢印があれば出力プレースという。

トランジション発火ルールは次の3条件である。

- (1) トランジション $t$ の各々の入力プレース $p$ に、 $w(p, t)$ 個以上のトークンがあれば、 $t$ は発火可能であるという。 $w(p, t)$ は $p$ から $t$ へのアークの重みである。
- (2) 発火可能なトランジションは、発火しても発火しなくてもよい。
- (3)  $t$ が発火すると、 $t$ の入力プレースすべてからそれぞれの重み $w(p, t)$ の分だけのトークンが取り去られる。 $t$ の出力プレースそれぞれには、重み $w(t, p)$ に応じたトークンが増える。 $w(t, p)$ とは、 $t$ から $p$ への重みである。

ペトリネットではトランジションの発火についての時間的な情報は組み込まれていない。プレースがトークンを保持する時間が定まっていて、トークンがそのプレースに入ってきてからその保持時間が経過すれば、トークンがそのプ

レースを入力とするトランジションによって利用可能になるようなペトリネットを時間ペトリネット(timed Petri net)と呼ぶ。本論文で扱う時間ペトリネットは、Baccelliら(1992)と同様に、トランジションのすべての入力レースのトークンが利用可能になったら直ちに発火するものとする。

時間ペトリネットのサブクラスとして、業務取引ペトリネット(business transaction Petri net: BTS-PN)を次のように定める。

## 定義2 業務取引ペトリネット(BTS-PN)

あるペトリネット構造が業務取引ペトリネットであるとは、ペトリネットが次の5つの条件を満たすことである。

- (1)  $| \rightarrow \bigcirc \rightarrow |$  という2つのトランジションとひとつのレースの結合部分を「活動部分」と呼ぶ。中のレースをbusyレースと呼ぶ。活動部分の入力とはbusyレースへの入力トランジションであり、出力も同様に定義される。活動部分のbusyレースへの入力アークと出力アークはそれぞれ一つしかない。活動部分の構成要素のアークの重みは1とする。
- (2) BTS-PNの各活動部分以外は、レースとアークからなり、結合部分と呼ばれる。結合部分は、 $\rightarrow \bigcirc \rightarrow$  のように入力アークと出力アークをそれぞれ一つだけ持つレースからなる。各アークの重みは1以上である。
- (3) ペトリネットグラフの任意のパスにおいて、活動部分と結合部分は交互に現われる。つまりBTS-PNは交代性をもつ。
- (4) 活動部分への入力アークは、活動部分の入力トランジションにだけ入れる。(出力アークは入力からも出力からも出る。)
- (5) BTS-PNの活動部分のレースのトークン保持時間はゼロでない。結合部分のレースは保持時間0である。

レース  $p$  へ入るアークの重みを  $w_{pi}$  と書き、出るアークの重みを  $w_{po}$  と書くことにする。したがって活動部分のbusyレース  $q$  にとっては  $w_{qi} = w_{qo} = 1$  である。

## 3 業務取引ペトリネットの状態遷移

BTS-PNのペトリネット構造を任意にひとつとる。これを  $J = \langle P, T, F \rangle$  とする。Jのすべてのトランジションと活動部分レースには予め番号がつけられている。  $n_A$  を活動部分の個数とすると、まず、これらに番号がつけられており、それに対応して、入力と出力トランジションが  $t_k^i, t_k^o$  というように入出力を区

別してしかも番号づけられている。つまり、 $T = \{t_k^!, t_k^\circ \mid k \in nA\}$ , ただし、 $nA = \{1, 2, \dots, nA\}$ である。

プレースの数を  $n$  とする。プレースは結合部分のプレースと、活動部分の busy プレースとがある。前者を PC とし、後者を PA とする。PA の要素は  $q_a$  と書くことにする。

関数  $lag: N^n \times PA \rightarrow R^\infty$  は、busy プレースがトークンを持ったときの保持時間をあたえる。(したがって、時不変の timed PN を考えている。)

$$lag(\mu, q_a) = \begin{cases} lag_a, & \text{if } \mu(q_a) = 1; \\ \infty, & \text{その他のとき.} \end{cases}$$

$R$  を実数の集合とする。 $N$  を 0 を含む自然数集合とする。 $N^n$  によってマーキングの集合を表わす。BTS-PN である  $J$  をシミュレートする BTS を構成していく。ここでシミュレートとは、つぎの可換図を満たすものである。

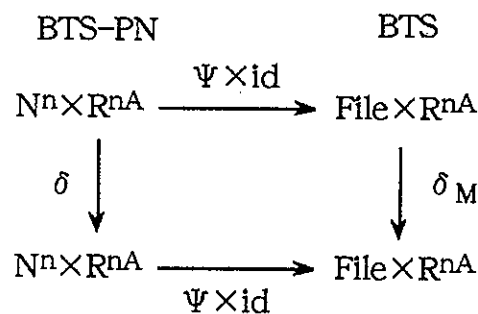


図3 可換図

$e = (e_1, e_2, \dots, e_{nA}) \in R^{nA}$  は各 busy プレースが現在のトークンを持ったときからの経過時間である。トークンを持たないときはその経過時間はゼロとする。

$\Psi: N^n \rightarrow \text{File}$  は、いわばマーキングの状況をファイルシステムで自然に表現するものである。

まず BTS-PN の状態遷移  $\delta: N^n \times R^{nA} \rightarrow N^n \times R^{nA}$  を定める。

$dueP: N^n \times R^{nA} \rightarrow P(T)$

$dueP(\mu, e) = \{t_{11}^\circ, t_{12}^\circ, \dots, t_{1m}^\circ\} \iff$  任意の  $k, 1 \leq k \leq m$ , について

$$(1) I(t_{1k}^\circ) = \{q_{1k}\}, \text{ つまり } \begin{array}{c} q_{1k} \quad t_{1k}^\circ \\ \circ \rightarrow | \end{array}$$

$$(2) \infty \neq \text{lag}(\mu, q_{ik}) - e_{ik} = \min\{i \in P_A\} \{\text{lag}(\mu, i) - e_i\}$$

つまり  $\text{dueP}$  は満期のもっとも近い将来の終了トランジションを示す関数である。

経過時間を進める(time advance)ための関数を

$$\text{taP}(\mu, e) = \min\{i \in P_A\} \{\text{lag}(\mu, i) - e_i \mid \infty \neq \text{lag}(\mu, q_i) - e_i\} \text{ と定める.}$$

これらの終了トランジションを活動部分の順番で、ひとつづつトークンを  $q_a$  から出力ブレースへ移動し、その後、入力開始可能かをしらべて、可能なら入力側のブレースから活動部分ブレースへトークンを移動する。つまり以下のよう定義する；

$\mu_0 = \mu$  とし、各  $j, 1 \leq j \leq m$ , について

$$\mu_j(p) =$$

$$\mu_{j-1}(p) - w_{p0}, \quad \text{if } p = q_{i_j} \text{ (つまり, } I(t_{i_j}^0) = \{q_{i_j}\});$$

$$\mu_{j-1}(p) + w_{p1}, \quad \text{if } p \in O(t_{i_j}^0);$$

$$\mu_{j-1}(p) \text{ 不変, } \text{その他のとき.}$$

$\delta_1 : N^n \times R^{nA} \rightarrow N^n \times R^{nA}$  を以下で定める。

$\delta_1(\mu, e) = (\mu^1, e^1)$  とおくとき、 $\mu^1 = \mu_m$  と定める。また、 $e^1$  は次とする：

$$e^1(q_a) =$$

$$e_a + \text{taP}(\mu, e), \quad \text{if } a \notin \text{dueP}(\mu, e);$$

$$0, \text{ その他のとき.}$$

この定義によって満期となったためにトークンが出力された活動部分は経過時間が 0 に設定され、また、満期でなかったところは今回の時間進め分を経過時間に加える。

次に、開始可能な入力トランジションがあればそれらを発火する操作として  $\delta_2$  を定める。

まず  $\mu$  によって決まる発火可能な入力トランジションを集め  $\text{st}(\mu)$  とする：

$$\text{st}(\mu) = \{t_{k_1}^1, t_{k_2}^1, \dots, t_{k_r}^1\} \longleftrightarrow \text{任意の } h, 1 \leq h \leq r, \text{ について}$$

$$(1) O(t_{k_h}^1) = \{q_{k_h}\}, \text{ つまり } \begin{array}{c} t_{k_h}^1 \quad q_{k_h} \\ | \rightarrow \circ \end{array}$$

かつ(2)  $\mu(q_{k_h}) = 0$ ;  $q_{k_h}$  にトークンがない；

かつ(3) 任意の  $p$  について  $[p \in I(t_{k_h}^1) \rightarrow \mu(p) \geq w_{p1}]$ .

このとき、 $\mu'_0 = \mu$  とし、各  $j, 1 \leq j \leq r$ , について

$$\mu'_j(p) =$$

$$\mu'_{j-1}(p) - w_{p0}, \quad \text{if } p \in I(t_{k_j}^1);$$

$$\mu'_{j-1}(p) + w_{p1}, \quad \text{if } p \in O(t_{k_j}^1);$$

$$\mu'_{j-1}(p) \text{ 不変, } \text{その他のとき.}$$

関数  $g$  によって、 $\mu$  から  $\mu'_r$  を与えることを表現する。つまり  $g(\mu) = \mu'_r$  である。



マーキングの最終的な更新結果を与える関数を  $h$  であらわすとき、

$$h(\mu) = \begin{cases} \mu, & \text{if } \text{st}(\mu) = \Lambda; \\ h(g(\mu)), & \text{if } \text{st}(\mu) \neq \Lambda \end{cases}$$

と定める。あとで示すように関数  $h$  は well-defined である。

$\delta_2 : N^n \times R^{nA} \rightarrow N^n \times R^{nA}$  を以下で定める。

$$\delta_2(\mu, e) = (h(\mu), e^2),$$

ただし、 $e^2(q_a) =$

$$\begin{cases} 0, & \text{if } O(t_a^!) = \{q_a\} \text{ かつ } t_a^! \in \cup_{p=0, \infty} \text{st}(g^p(\mu)) \\ \text{不変,} & \text{その他のとき.} \end{cases}$$

とする。

$\delta(\mu, e) = \delta_2(\delta_1(\mu, e))$  と定める。

### 命題 1

関数  $h(\mu) =$

$$h(\mu) = \begin{cases} \mu, & \text{if } \text{st}(\mu) = \Lambda; \\ h(g(\mu)), & \text{if } \text{st}(\mu) \neq \Lambda \end{cases}$$

は well-defined である。

なぜなら、 $\text{st}(\mu) \neq \Lambda$  とすると、 $\text{st}(\mu)$  の入力トランジションに対応する活動部分は、トークンを取り組むために、もはやその時刻には開始することはできない。活動部分の数は有限個だから、かならず有限回の、たとえば  $p$  回の、 $g$  の適用によって  $\text{st}(g^p(\mu)) = \Lambda$  となる。

## 4 業務取引ペトリネットをシミュレートする業務取引システム

次に BTS-PN に対応する業務取引システム(Business Transaction System: BTS)を定める。以下で定義するような、BTS-PN から作られる BTS の例は 5 節にある。

BTS を定めるには、

$$G_a = \langle S_a \times R^\infty, \delta_a, ta_a \rangle,$$

$$fFA: \text{File} \rightarrow P(A)$$

$$fFval: \text{File} \times P(A) \rightarrow \text{File}$$

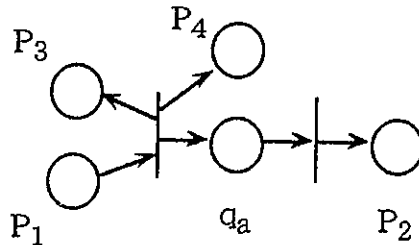
を定めればよい(Sato and Praehofer, 1997)。

各 busy プレースに対して内部活動を対応させる。つまり、 $P_A$  と  $\{G_a \mid a \in nA\}$  は 1 対 1 に対応させる。

まずファイルシステムを定め、次にトークン配置をファイルシステムとして表わす対応である  $\Psi: N^n \rightarrow \text{File}$  を定義する。  $KS=P$  とする。  $KS$  の要素である任意の管理対象  $p$  についての属性は、  $p \notin P_A$  について  $A(\text{FT}(p)) = \{\# \text{FT}(p), \text{個数}\}$  と定め、また、  $p \in P_A$  について  $A(\text{FT}(p)) = \{\# \text{FT}(p), \text{個数}, \text{終了可否}\}$  とする。  $\# \text{FT}(p)$  は主キーである。 トークンが色付きでない場合（トークンが無差別）には、この表にはいつも一行のデータしか存在せず、個数属性の値だけが変わることになる。これで  $\text{File}$  が定まった。なお、記述の簡単のために、マーキング  $\mu$  に対するファイル内容関数を  $f_\mu$  と表わし、  $p \in P_A$  における個数を  $f_\mu(p)$  と表わす。つまり、常に  $f_\mu(p) = \mu(p)$  である。  $p \in P_A$  については  $f_\mu(p) = (\mu(p), [p, \text{終了可否}])$  の属性値であり、たとえば  $(2, \text{no})$  とか  $(1, \text{yes})$  という値をとる。  $\text{yes}$  と  $\text{no}$  という値は、その処理を終了できる場合には  $\text{yes}$  となるという意味をもつ。  $\mu(p)$  を  $f_\mu(p)$  .個 と冗長に書くこともある。また、  $f_\mu(p)$  .終了可否 を  $f_\mu(p)$  .終 と省略して書く。

この対応を  $\Psi(\mu) = f_\mu$  と表わす。ただし、各  $p \in P_A$  については  $f_\mu(p)$  .終 =  $\text{yes}$  とする。

$f_{AK}$  と  $f_{KA}$  については、たとえば、



というとき  $p_1 \in f_{KA}(a)$ ,  $q_a \in f_{KA}(a)$ ,  $q_a \in f_{AK}(a)$ ,  $p_2 \in f_{AK}^F(a)$ ,  $p_3, p_4 \in f_{AK}^S(a)$  と定める。つまり  $p_3, p_4$  は活動部分の入力トランザクションからの出力アークであり、また、  $p_2$  は活動部分の出力トランザクションからの出力アークである。  $f_{AK}(a) = \{q_a\} \cup f_{AK}^S(a) \cup f_{AK}^F(a)$  と定める。他の場合も同様である。

$$S = \{(f_\mu, e_1, e_2, \dots, e_{n_A}) \mid f_\mu \in \text{File}, (e_1, e_2, \dots, e_{n_A}) \in R^{n_A}\}$$

$$S_a = \text{File} \mid f_{AK}(a) \cup f_{KA}(a)$$

$ta_a(f_\mu^a) = \text{lag}(f_\mu^a, F^{-1}(a))$ ; ただし  $f_\mu^a$  は  $f_\mu^a = f_\mu^a$  であるような任意。いまは、色付きでなはないトークンだけを考えるから、トークンがあればファイルによらず各活動ごとに定まった値にする。  $\text{lag}$  の定義からトークンがなければ満期時間は無限大となる。

$ta_a$  の定義は well-defined である。

$$\text{fFA}(f_\mu) \ni a \iff (1) (\forall p)(p \in f_{KA}(a) \setminus \{q_a\} \rightarrow f_\mu(p) \geq \text{wpl}), \text{ かつ}$$

$$(2) f_\mu(q_a) \cdot \text{個} = 0$$

上の(1)は入力ブレース (busy ブレースを除く) に十分にトークンがあること,  
(2)は busy でないことを意味している。

当該の BTS の内部活動  $a$  が  $f_\mu^a$  で「開始可能」であるとは,

あるトークン分布  $f'_\mu$  があって,  $f_\mu^a = f'_\mu |_{f_{KA}(a)}$  かつ  $a \in f_{FA}(f'_\mu)$   
であることと定める. この定義は well-defined である. なぜなら  $f_{FA}$  の定義に  
おいては,  $p \in f_{KA}(a)$  であるような  $p$  についての値  $f'_\mu(p)$  だけしか用いないから.

$\delta_a$  を定めるために3つの関数  $\delta_{aI}$ ,  $\delta_{aQ}$ ,  $\delta_{aO}$  を次のように定める.

$\delta_{aI} : \text{File} |_{f_{AK}(a) \cup f_{KA}(a)} \rightarrow \text{File} |_{f_{AK}(a) \cup f_{KA}(a)}$

$\delta_{aI}(f_\mu^a)(p) =$   
 $f_\mu^a(p) - w_{pl}$ , if  $p \in f_{KA}(a) \setminus \{q_a\}$ , かつ  $a$  が  $f_\mu^a$  で開始可能;  
不変, その他のとき.

$\delta_{aQ} : \text{File} |_{f_{AK}(a) \cup f_{KA}(a)} \rightarrow \text{File} |_{f_{AK}(a) \cup f_{KA}(a)}$

$\delta_{aQ}(f_\mu^a)(p) =$   
1, if  $p = q_a$ , かつ  $a$  が  $f_\mu^a$  で開始可能;  
0, if  $p = q_a$ , かつ  $a$  が  $f_\mu^a(q_a) = 1$ ;  
不変, その他のとき.

$\delta_{aO} : \text{File} |_{f_{AK}(a) \cup f_{KA}(a)} \rightarrow \text{File} |_{f_{AK}(a) \cup f_{KA}(a)}$

$\delta_{aO}(f_\mu^a)(p) =$   
 $f_\mu^a(p) + w_{po}$ , if  $p \in f_{AK}(a) \setminus \{q_a\}$ , かつ  $a$  が  $f_\mu^a(q_a) = 1$ ;  
不変, その他のとき.

$\delta_a = \delta_{aI} \cdot \delta_{aQ} \cdot \delta_{aO}$  と定める.

$f_{FVal} : \text{File} \times P(A) \rightarrow \text{File}$  を定める.

$\delta_{start} : \text{File} \rightarrow \text{File}$  と,  $\delta_{finsh} : \text{File} \times P(A) \rightarrow \text{File}$  を定める.

$f_{FA}(f_\mu) = \{a_{i1}, a_{i2}, \dots, a_{ir}\}$  とするとき, 各  $j, 1 \leq j \leq r$ , について逐次計算する.

$f^0 = f_\mu$ ,

$f^j(p) =$

$f^{j-1}(p) - w_{po}$ , if  $q_{ij} \neq p \in f_{KA}(a_{ij})$ ;  
 $(f^{j-1}(p).個 + w_{pl}, no)$ , if  $q_{ij} = p$ ;  
 $f^{j-1}(p) + w_{pl}$ , if  $q_{ij} \neq p$  かつ  $p \in f_{AK}^S(a_{ij})$ ;  
 $f^{j-1}(p)$  不変, その他のとき.

関数  $g_F$  を  $g_F(f_\mu) = f^r$  で定める.

$\delta_{start}(f_\mu) =$

$f_\mu$ , if  $f_{FA}(f_\mu) = \Lambda$ .

$$\delta_{\text{start}}(g_F(f_\mu)), \text{ if } f_{\text{FA}}(f_\mu) \neq \Lambda.$$

と定める。これは well-defined である。

$$\delta_{\text{finish}}(f_\mu, \Lambda) = f_\mu,$$

$$\delta_{\text{finish}}(f_\mu, \{a_{i_1}, a_{i_2}, \dots, a_{i_m}\}) = f^m, \text{ ただし各 } j, 1 \leq j \leq m \text{ について}$$

$$f^0 = f_\mu,$$

$$f^j(p) =$$

$$(f^{j-1}(p).個 - w_{po}, \text{yes}), \text{ if } p = q_j \text{ かつ } f^{j-1}(q_j).終 = \text{yes};$$

$$f^{j-1}(p) + w_{pi}, \text{ if } q_j \neq p \in f_{\text{AK}^F}(a_{i_j}) \text{ かつ } f^{j-1}(q_j).終 = \text{yes};$$

$$(f^{j-1}(p).個, \text{yes}), \text{ if } p = q_j \text{ かつ } f^{j-1}(p).終 = \text{no};$$

$$f^{j-1}(p) \text{ 不変}, \text{ その他のとき.}$$

任意の  $\{a_{i_1}, a_{i_2}, \dots, a_{i_r}\}$  に対して,

$$f_{\text{FVal}}(f_\mu, \{a_{i_1}, a_{i_2}, \dots, a_{i_r}\}) = \delta_{\text{finish}}(\delta_{\text{start}}(f_\mu), \{a_{i_1}, a_{i_2}, \dots, a_{i_r}\})$$

と定める。

BTS の  $\delta_M$  は、上の関数を用いて、次のように定義されている (Sato and Praehofer, 1997) .

$$ta: \text{File} \times \mathbb{R} \rightarrow \mathbb{R}^\infty$$

$$ta(f_\mu, e_1, e_2, \dots, e_{n_A}) = \min_{a \in A} \{ ta_a(f_\mu^a) - e_a \}$$

まず関数 due を次で定義する :

$$\text{due}(f_\mu, e_1, e_2, \dots, e_{n_A}) = \{a_{i_1}, a_{i_2}, \dots, a_{i_m}\} \longleftrightarrow$$

$$\text{任意の } k, 1 \leq k \leq m, \text{ について } ta(f_\mu, e_1, e_2, \dots, e_{n_A}) = ta_{a_k}(f_\mu^{a_k}) - e_k$$

$$\delta_M(f_\mu, e_1, e_2, \dots, e_{n_A}) =$$

$$(ff[f_{\text{FVal}}(f_\mu, \text{due}(f_\mu, e_1, e_2, \dots, e_{n_A}))], e'_1, e'_2, \dots, e'_{n_A}),$$

ただし,  $e'_i =$

$$0, \text{ if } i \in \text{due}(s) \cup \bigcup_{k=0}^{\infty} f_{\text{FA}}[f_{\text{Dispatch}}^k(f_{\text{FileValue}}(f, \text{due}(s)))];$$

$$e_i - ta(f_\mu, e_1, e_2, \dots, e_{n_A}), \text{ その他のとき.}$$

以上によって, BTS-PN に対応する BTS が定まった. 以下で可換性の例を示す.

## 5 業務取引ペトリネットと業務取引システムの可換性

図4の BTS-PN を例題として考える. このマーキングを  $\mu$  とする. アークに数値が書かれているのは重みであるが, 重み1は省略されている. 一般的な可換性の証明は付録に示されている.

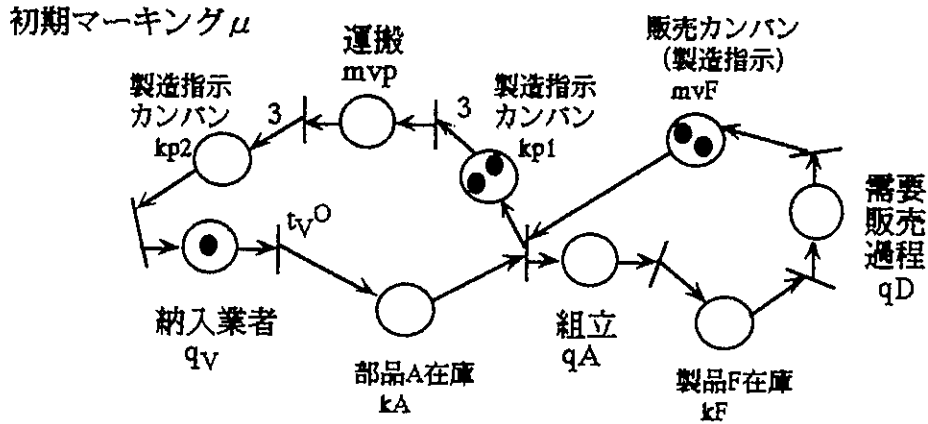


図4 カンバンシステムの業務取引ペトリネットと初期マーキング

初期条件として

- (1) すぐに開始できる活動部分  $q_i$  はない ;
- (2) 任意の  $q \in P_A$  について  $\text{lag}(\mu, q) - e_q \geq 0$ .

このときに  $\delta$  による状態遷移を計算する.

(PN1)  $\text{due}P(\mu, e) = \{t_V^0\}$ . なぜなら, 他のどれも発火可能ではない.

(PN 2)  $\delta_1$  の定義によって,  $\delta_1(\mu, e) = (\mu^1, e^1)$  とするとき, 図5の  $\mu^1$  となる.

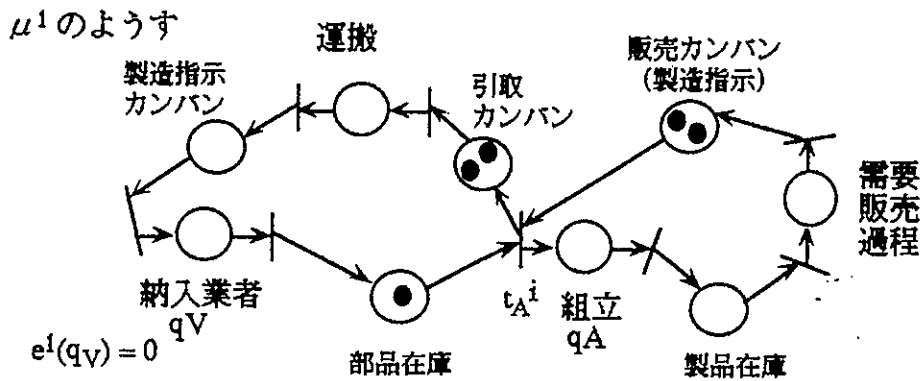


図5 納入の終了

(PN 3)  $\text{st}(\mu^1) = \{t_A^1\}$  である.

(PN 4) 次に,  $\delta_2(\mu^1, e^1) = (\mu^2, e^2)$  を求める. まず  $g(\mu^1)$  は図6のようになる.

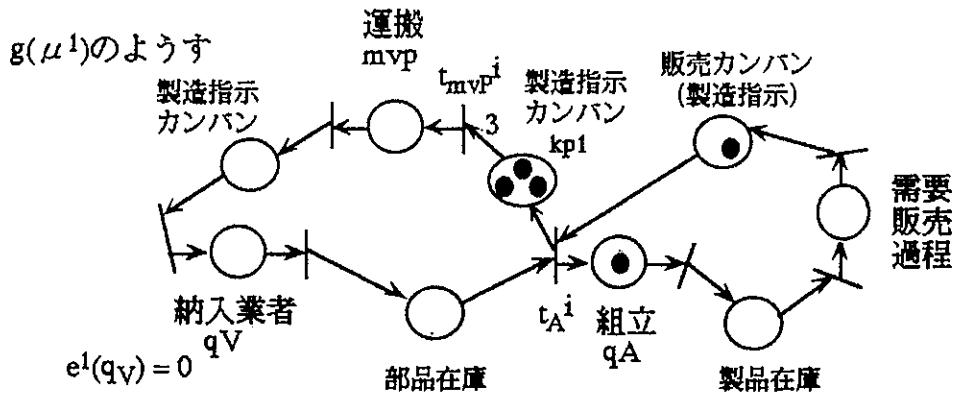


図6 組み立ての開始

すると  $st(g(\mu^1)) = \{t_{mvp}^1\}$  である。だから  $g(g(\mu^1))$  は次の図7のようになる。

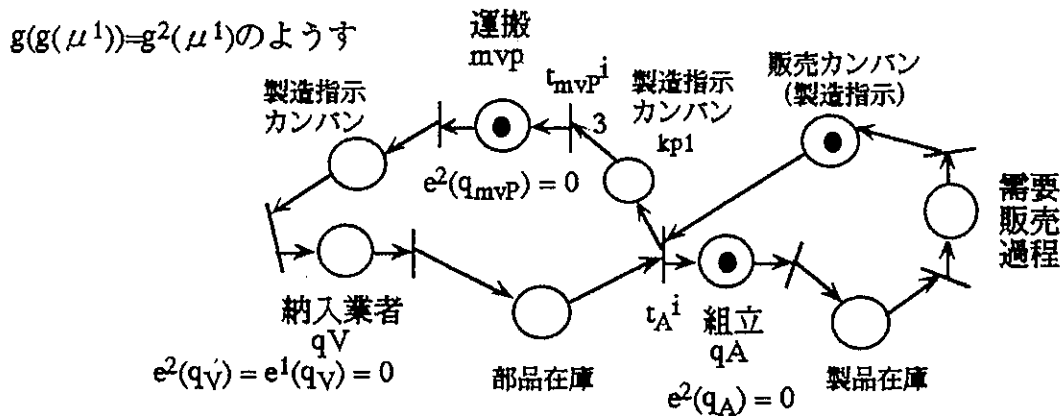


図7 カンバンの運搬

すると、 $st(g(g(\mu^1))) = \Lambda$  だから、 $\mu^2 = g(g(\mu^1))$  となる。

$\cup_{p=0, \infty} st(g^p(\mu)) = st(\mu^1) \cup st(g(\mu^1)) = \{t_A^1, t_{mvp}^1\}$  だから、 $e^1(q_V) = 0$  に加えて新たに  $e^2(q_A) = e^2(q_{mvp}) = 0$  となる。

次に図4の BTS-PN に対応する BTS と、それらの状態遷移を調べる。  
 (BTS1) まず、図4の初期マーキングに対応する初期状態  $f_\mu$  をもつ BTS は図8になる。図で、たとえば、 $kA(F)$  は  $kA$  が  $f_{AK}^F(V)$  の要素であることを示し、また、 $kp1(S)$  は  $kp1$  が  $f_{AK}^S(A)$  の要素であることを示している。

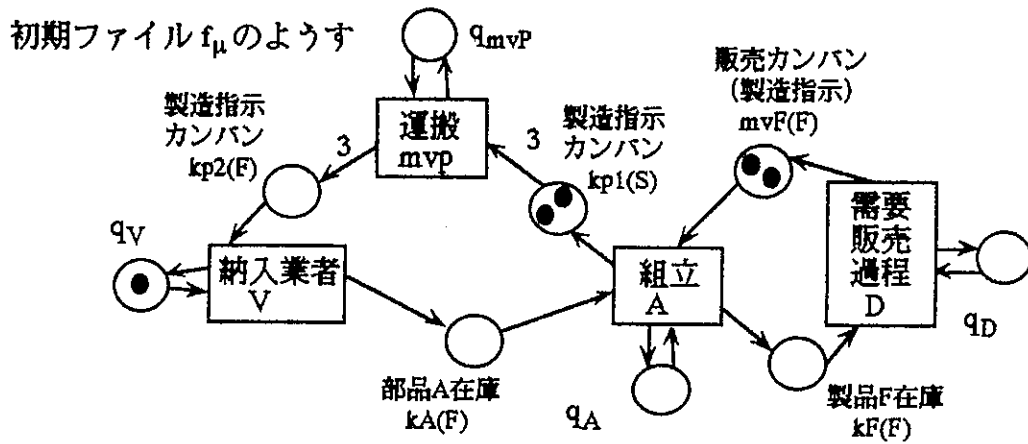


図8 カンバンシステムの業務取引システム

図8で、 $f_\mu(q_v)$ . 終も  $f_\mu(q_A)$ . 終も他も yes の値をもつ。  
 (BTS 2)  $\text{due}(f_\mu, e) = \{V\}$ . また  $f_{FA}(f_\mu) = \Lambda$  (空).  
 (BTS 3) したがって、 $f_{Fval}(f_\mu, \text{due}(f_\mu, e)) = \delta_{\text{finish}}(f_\mu, \{V\}) = f_1$ . ただし、  
 $f_1(k_A) = 1, f_1(q_v) = (0, \text{yes}), f_1(\text{その他}) = f_\mu(\text{その他})$  で不変. (図9)

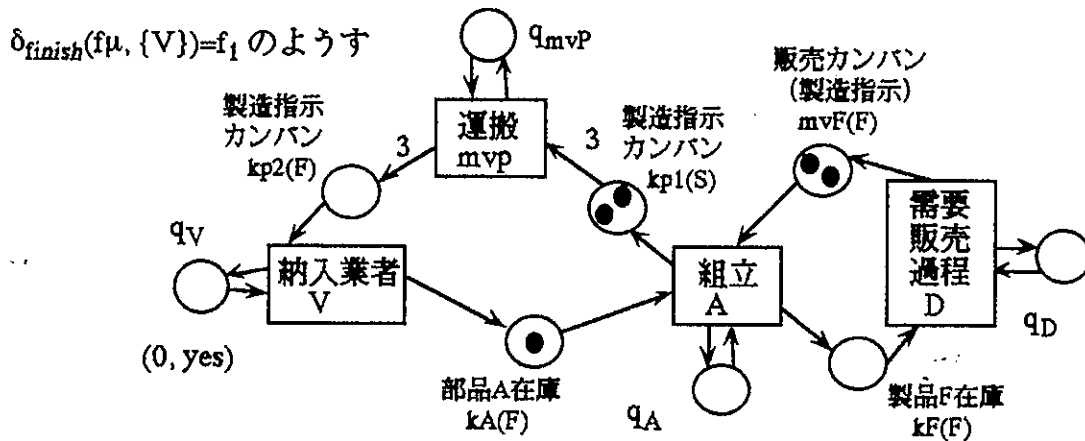


図9 納入の終了

したがって、 $f_F[f_{Fval}(f_\mu, \text{due}(f_\mu, e))] = f_F(f_1)$ .

(BTS 4)  $f_F(f_1)$  を計算する.  $f_{FA}(f_1) = \{A\}$  だから、 $f_F$  の定義によって  
 $f_F(f_1) = f_F(f_{Fval}(f_1, \{A\}))$   
 $= f_F(\delta_{\text{finish}}(\delta_{\text{start}}(f_1), \{A\}))$   
 $\delta_{\text{start}}(f_1)$  を計算するために、 $g_F(f_1) = f_2$  を求める.

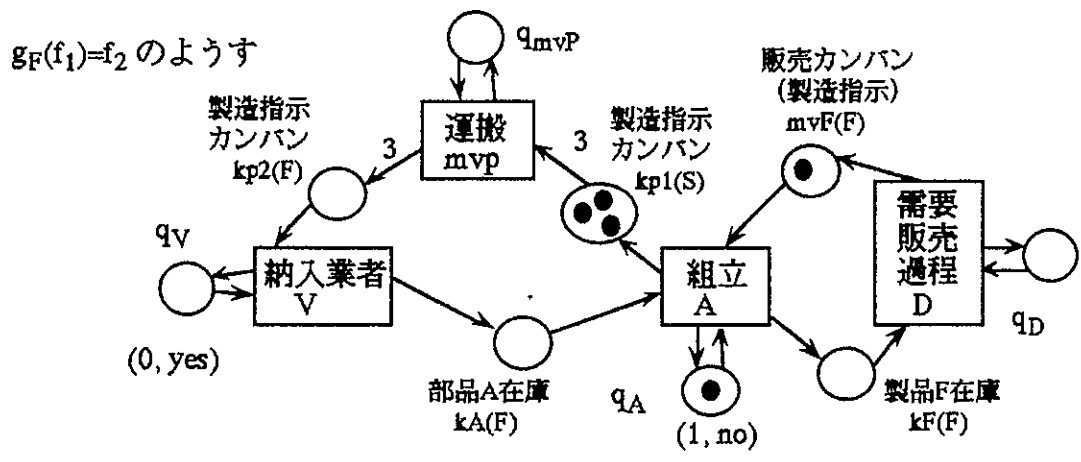


図 1 0 組み立ての開始

$f_{FA}(f_2) = \{mvp\}$  だから  $g_F(g_F(f_1)) = f_3$  を求める。

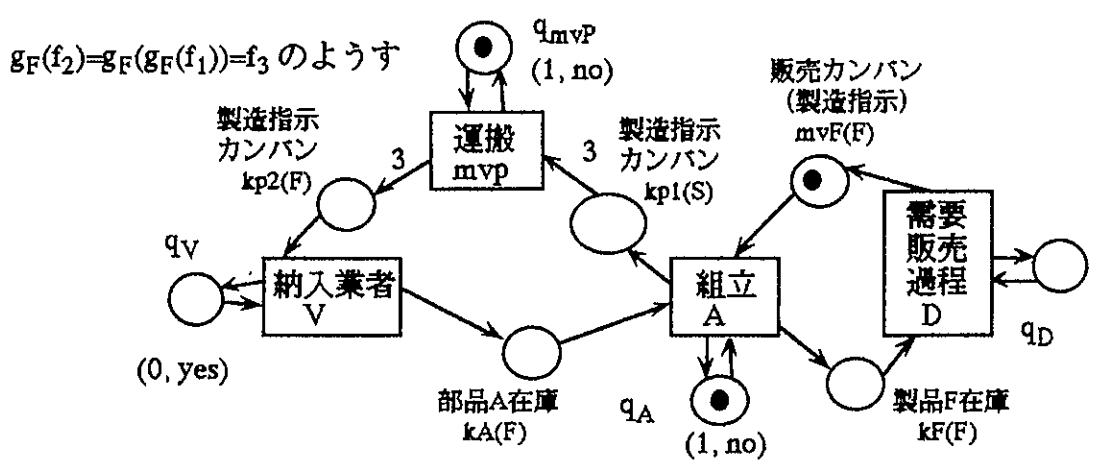


図 1 1 カンバンの運搬

$f_{FA}(f_3) = \Lambda$  だから  $\delta_{start}(f_1) = f_3$  となった。

(BTS 5)  $\{A, mvp\} = due(s) \cup \bigcup_{k=0}^{\infty} f_{FA}[f_{Dispatch}^k(f_{FileValue}(f, due(s)))]$ , (ただし,  $s=(f_{\mu}, e)$ ) なので,  $q_A, q_{mvp}$  の経過時間が 0 となる。

以上のことから, 可換性は成立する。つまり, 図 7 の BTS-PN を  $\Psi$  と  $id$  でマップすると図 11 のファイルシステムと経過時間ゼロが, 確かに得られる。図 3 が可換になるという意味で, BTS-PN を BTS で“シミュレートできる”こと



が例示された。

## 6 MRP

MRPやカンバンは情報フィードバックによるビジネスシステムのクローズドシステム化の一形態である。図8からカンバンによるコントロールを取り除くと各々図12のようになる。この生産物流を制御する方法は一つではなく、基本的には消費した量を前にフィードバックする仕組みがあればよい。たとえばMRPでは図13のようにして情報センターである計算機に実績情報を収集し、計画との差異をもとに再計画して生産指示を出す、というフィードバックを行う。

この場合も業務取引ペトリネットと業務取引システムの状態遷移の可換性を利用して、時間的変化のスピードを、Baccelliら(1992)の時間ペトリネット計算法で求めることができる。

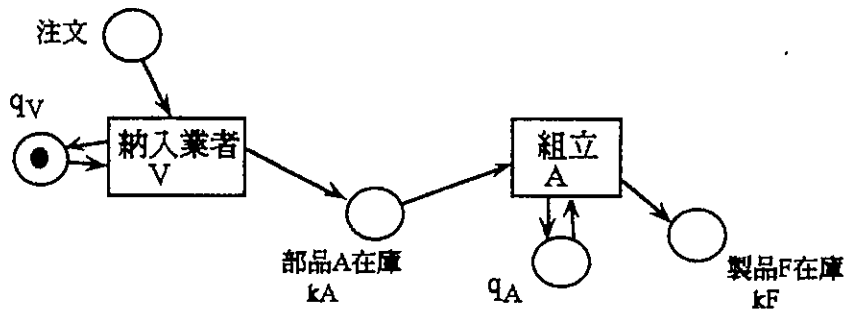


図12 制御システムなしの業務取引システム

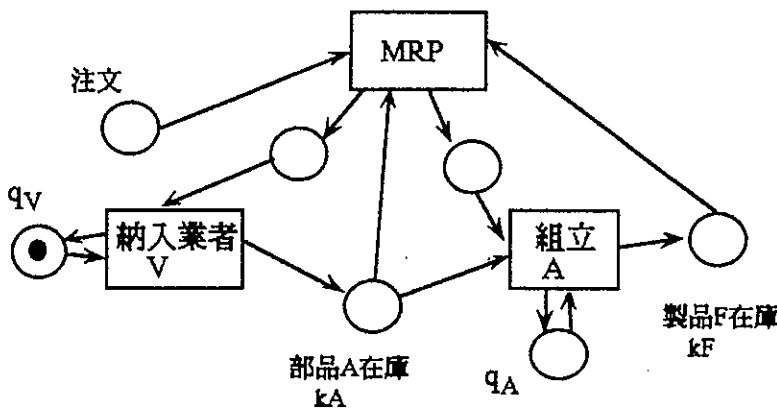


図13 MRP制御システムを持つ業務取引システム

## 7 結論

業務取引システムを業務取引ペトリネットに関係づける方法を示した。

ビジネスプロセスの時間生産性についての設計法は、情報システム方法論の歴史においてこれまでほとんど取り扱われてこなかった領域である。その理由は、重要性は認識されていたが、ビジネスプロセスについての計算できるほどに形式化されていて、かつ、情報システム方法論と整合性のあるモデルが存在しなかったからである。本論文の結果によって、時間ペトリネットに対して知られている Max-Plus 代数 (Baccelli ら, 1992) という時間特性の計算方法を業務取引システムにも利用できることになった。これはさらに、業務取引システムによってジャストインタイム管理システムというビジネスプロセスの時間生産性の設計方法の可能性を開いた。

## 参考文献

- [1] Baccelli, F.L., Cohen, G., Olsder, G.J., and Quadrat, JP: Synchronization and Linearity — An algebra for discrete event systems, John Wiley, 1992.
- [2] 門田安弘：新トヨタシステム, 講談社, 1991.
- [3] 村田忠夫：ペトリネットの解析と応用, 近代科学社 1992.
- [4] Sato, R. and Praehofer H. "A discrete event model of business system - A Systems Theoretic Foundation for Information Systems Analysis: Part 1", IEEE Transactions on Systems, Man, and Cybernetics, 27-1, pp.1-10, 1997.
- [5] Sato, R.: Meaning of Dataflow Diagram and Entity Life History - A Systems Theoretic Foundation for Information Systems Analysis: Part 2, IEEE Transactions on Systems, Man, and Cybernetics, 27-1, pp.11-22, 1997.
- [6] 田中一成：「時間生産性」をどう高めるか, 東洋経済, 1993.
- [7] Zeigler, B. P. Theory of modelling and simulation. John Wiley, 1976.
- [8] Zeigler, B. P. Multifaceted modeling and discrete event simulation. Academic, 1984.

## 付録 可換性の証明

本論文で示したような, BTS-PN とそれから作られる業務取引システムの状態遷移の可換性 (互換性) が一般的に成立することを以下で証明する.

BTS-PN を所与とする.  $\Psi$  と id でマップして, 対応する BTS を作ったとする. 記号は本文の説明で用いたものを使う. このとき以下の順序で可換性が示される.

### 命題 2

$$\text{st}(\mu) = \{tk_1^1, tk_2^1, \dots, tk_r^1\} \longleftrightarrow \text{fFA}(f_\mu) = \{ak_1, ak_2, \dots, ak_r\}, \text{ただし, } ak_v = F(O(tk_v^1)).$$

(証明) 定義より明らか.

Q.E.D.

### 命題 3

$$\text{taP}(\mu, e) = \text{ta}(f_\mu, e)$$

(証明) 任意の  $p \in P_A$  について  $\text{lag}(\mu, p) = \text{ta}_{F(p)}(f_\mu^{F(p)})$  より明らか. Q.E.D.

### 命題 4

$$\text{dueP}(\mu, e) = \{tl_1^0, tl_2^0, \dots, tl_m^0\} \longleftrightarrow \text{due}(f_\mu, e) = \{a_1, a_2, \dots, a_m\}, \text{ただし, } a_v = F(O(tl_v^0)).$$

(証明)

各々の定義と, 命題 2 のことから.

Q.E.D.

### 命題 5

$\delta_1(\mu, e) = (\mu^1, e^1)$  とするとき,

$$\Psi(\mu^1) = f_{\mu^1} = \delta_{\text{finish}}(f_\mu, \text{due}(f_\mu, e))$$

(証明)

$\Psi(\mu) = f_\mu$  と表わすとき, 各  $p \in P_A$  について  $f_\mu(p).終 = \text{yes}$  であるから,  $\delta_1$  と  $\delta_{\text{finish}}$  の定義と, 命題 3 が成立することから, 成立する. Q.E.D.

### 命題 6

$\delta_2(\mu, e) = (\mu^2, e^2)$  とするとき,

$$\Psi(\mu^2) = \delta_{\text{finish}}(\delta_{\text{start}}(f_\mu), \text{fFA}(f_\mu))$$

(証明)

命題 1 によって  $\text{st}(\mu) = \{tk_1^1, tk_2^1, \dots, tk_r^1\} \longleftrightarrow \text{fFA}(f_\mu) = \{ak_1, ak_2, \dots, ak_r\}$  が成立している.  $\text{st}(\mu)$  の各トランジションに対して  $\delta_2$  によるプレースカ

らのトークン減少操作に対応して、 $\delta_{start}$  は個数を減少する。そのとき  $f_{FA}(f_\mu)$  の各活動の終了可否の欄は no という値になる。そして、その後の  $\delta_{finish}$  は単に、その no を yes に変更するだけである。したがって命題は成立する。 Q.E.D.

### 命題 7

次の可換図は可換となる。

$$\begin{array}{ccc}
 \text{BTS-PN} & & \text{BTS} \\
 N^n \times R^{nA} & \xrightarrow{\Psi \times \text{id}} & \text{File} \times R^{nA} \\
 \delta \downarrow & & \downarrow \delta_M \\
 N^n \times R^{nA} & \xrightarrow{\Psi \times \text{id}} & \text{File} \times R^{nA}
 \end{array}$$

(証明)

- 1) 初期状態条件を満たすような任意の  $(\mu, e)$  を  $N^n \times R^{nA}$  からとる。
- 2)  $\delta(\mu, e) = \delta_2(\delta_1(\mu, e))$  である。  $\delta_1(\mu, e) = (\mu^1, e^1)$  とすると、命題 4 より

$$\Psi(\mu^1) = \delta_{finish}(f_\mu, \text{due}(f_\mu, e))$$

である。  $\delta_2(\mu^1, e^1) = (\mu^2, e^2)$  とすると命題 5 から、

$$\begin{aligned}
 \Psi(\mu^2) &= \delta_{finish}(\delta_{start}[\delta_{finish}(f_\mu, \text{due}(f_\mu, e))], f_{FA}[\delta_{finish}(f_\mu, \text{due}(f_\mu, e))]) \\
 &= \delta_{finish}(\delta_{start}[\Psi(\mu^1)], f_{FA}[\Psi(\mu^1)])
 \end{aligned}$$

- 3) 一方、

$$f'_\mu = f_F[f_{FileValue}(f_\mu, \text{due}(f_\mu, e))] = f_F(\delta_{finish}[\delta_{start}(f_\mu), \text{due}(f_\mu, e)])$$

初期状態表現により  $f_{FA}(f_\mu) = \Lambda$  だから、 $\delta_{start}(f_\mu) = f_\mu$  である。したがって

$$f'_\mu = f_F(\delta_{finish}[f_\mu, \text{due}(f_\mu, e)]) = f_F(\Psi(\mu^1)).$$

- 4) 次に、 $f_{FA}(f_\mu) \neq \Lambda$  とすると、

$$\begin{aligned}
 f'_\mu &= f_F[f_{FileValue}(\Psi(\mu^1), f_{FA}(\Psi(\mu^1)))] \\
 &= f_F(\delta_{finish}[\delta_{start}(\Psi(\mu^1)), f_{FA}(\Psi(\mu^1))]).
 \end{aligned}$$

ここで、 $f_{FA}(\delta_{finish}[\delta_{start}(\Psi(\mu^1)), f_{FA}(\Psi(\mu^1))]) = \Lambda$  となる。なぜなら、まず  $\delta_{start}$  によってすべての開始可能な活動が開始され、終了可否欄が no となり、その後の  $\delta_{finish}$  によって変化するのは no であった値が yes に変わるだけであるから。したがって

$$f'_\mu = \delta_{finish}[\delta_{start}(\Psi(\mu^1)), f_{FA}(\Psi(\mu^1))] = \Psi(\mu^2). \mu \text{ については可換となる.}$$

- 5) また、 $f_{FA}(f_\mu) = \Lambda$  の場合には、命題 1 より  $\text{st}(\mu^1) = \Lambda$ . よって、 $\mu^1 = \mu^2$  だから、 $f'_\mu = \delta_{finish}[f_\mu, \text{due}(f_\mu, e)] = \Psi(\mu^1) = \Psi(\mu^2)$  より、 $\mu$  については可換となる。

6) これらの $\mu$ についての可換性から,  $e_i$ が0になるのは終了と開始だがそれらは, 命題1と3から可換性がわかる. また, 終了も開始もしないものは, 時間が進むだけだが, 命題2より同じ経過時間となる. Q.E.D.

