並行システムモデル NeO とその解析†

猪股俊光†† 片野田守人††* 小野木克明†† 西村義行††

並行システムの制御の問題は，次の３つの問題を含むと考えられる：(1)システムの各所に存在する非決定性を明らかにする．(2)与えられた制御方策（例えばシステムを周期的に動作させる等）に対して，各所の非決定性の解消が受ける制約について解析する．(3)非決定性を解消した各状況に対して，システムの挙動を詳細に評価し，適当な解消法の１つを選択する．従来の研究については，各種の離散系シミュレーション言語は上記の(3)に対して，各種の並行システム記述言語は(1)，(3)に対して，ペトリネットは(1)，(2)に対して優る．本報では，上記(1)，(2)，(3)の問題に一貫して対処することができるような，１つの並行システムモデル NeO を提案する．NeO は，ペトリネットとオブジェクト指向を融合したモデルであり，上記(1)，(2)，(3)の問題に次のように対処する．(1)非決定性を，メッセージ受理可能なレシーバ（オブジェクトの手続きに対応する）と受理可能なメッセージの選択で表現する．(2)各レシーバの相対メッセージ受理率なる概念を導入し，それらが満たすべき関係を導出する．(3) NeO に基づいて事象スケジューリング方式のシミュレータを構成する（ただし，本報ではその可能性を示すにとどめる）．

1. はじめに

コンピュータシステム，OA システムなど多くの情報システムは，複数個のプロセスが互いに通信等を行いながら進行する並行システムとしてとらえられる．

一般にシステムの制御の問題には，

(1) 対象とするシステムにどのような運用の多様性がありうるかということを考える問題

(2) ありうると思われるものが与えられた制御方策（例えばシステムを周期的に動作させる等）に対して実現可能かどうかを解析する問題

(3) 実現可能な各案に対してシステムの動作を評価し適当な案の１つを選択する問題

等があるが，並行システムの制御の問題は，それがもつ非決定性とその解消の問題に帰着することができ，上記(1)，(2)，(3)はそれぞれ次の(1)′，(2)′，(3)′のようにとらえることができるものと思われる．

(1)′ システムにおける非決定性が，システムのどの部分でどのように生ずるかを考える問題．

(2)′ 与えられた制御方策に対して，各部分における非決定性の解消を他の部分と独立に行うことができるかどうか，できないとすればどの程度の自由度で行うことができるか等を解析する問題．

(3)′ 非決定性が解消された各状況に対して，システムの動作を詳細に評価し，適当な非決定性解消案の１つを選択する問題．

したがって，並行システムの制御の諸問題を考察するためのモデルとしては，非決定性を明確な形で含み，上記(1)′，(2)′，(3)′に対する考察を一貫して行うことができるものが望ましいが，このようなモデルはまだ開発されていない．

従来，多くの離散系シミュレーション言語[1)] (GPSS, SIMSCRIPT など）が開発され，(3)′の問題の解決のために用いられてきたが，これらは，非決定性を陽に含まないので問題(1)′，(2)′の解決に使用することはできない（これらの言語が，問題(1)′，(2)′を認識することなしに，極端な場合には，存在するかもしれない非決定性の解消をこれらの言語の処理系にまかせて使用されている場合も多いようである．それで十分な場合もあるかもしれないが）．

上記(1)′，(3)′の解決を目指すものとして，CSP[2)]，CCS[3)] などの並行システム記述言語があるが，(2)′の問題が陽に扱われていないので，これらのモデルに基づくこの種の解析は困難ではないかと思われる．

上記(1)′，(2)′の解決を目指すものとして，ペトリネット[4)]がある．ペトリネットはプレース，トランジションの２種類のノードからなる２部有向グラフ，マーキング（プレース上のトークンの配置）およびマーキングの遷移則よりなり，これらに基づいて，システムの状態，状態遷移の非決定性，可達性，活性，

† Development and Analysis of a Concurren System Model NeO by TOSHIMITSU INOMATA, MORITO KATANODA, KATSUAKI ONOGI and YOSHIYUKI NISHIMURA (Department of Production Systems Engineering, Faculty of Engineering, Toyohashi University of Technology).

†† 豊橋技術科学大学工学部生産システム工学系

* 現在 東レ・ダウコーニング・シリコーン(株) Dow Corning Toray Silicone Co., Ltd.

有界性などの概念が定義されるとともに，これらに対する解析手法の開発も進められている．しかしながら，ペトリネットは，並行システムのモデルとして表現力に乏しく（表現力と解析性はしばしば相反するので止むを得ない面もあるが）上記（3）′ の意味のシステムの評価に用いるのは困難と思われる．

本報では，上記の（1）′，（2）′，（3）′ に対する考察を一貫して行うための1つの並行システムモデル NeO を提案する．ただし本報では（1）′，（2）′ に重点をおいて述べ，（3）′ については可能性を指摘するにとどめ，詳細は別報とする．

NeO は並行システムのモデリングに有効とされるオブジェクト指向[5]とペトリネットを融合したものである．すなわち NeO は，オブジェクト（内部状態と受理メッセージに基づく内部状態の変更・送信メッセージの決定を行う手続きよりなる）とキュー（メッセージを蓄えるバッファ）の2種のノードよりなる2部有向グラフ，および遷移則よりなる．各プロセスの挙動は各オブジェクトとして独立に定義され，相互作用はメッセージの送受のみによって行われる．また，ペトリネットの場合と同様にシステムの状態はマーキングとしてとらえられ，状態遷移の非決定性はマーキングの遷移則の中で明確に定義される．さらに，可達性，活性，有界性などの概念が導入されるとともに，システム内の各所で生ずる非決定性の解消相互の関係もメッセージ受理の間の関係としてとらえることとなる．

以下，2章で NeO の定義を行い，3章でそれによる並行システムのモデリングについて述べる．4章では並行システムの解析と制御に有効と思われる諸性質を NeO に対して導入しそれに基づく解析法について述べる．5章ではペトリネットとの関係および NeO によるシステム挙動の評価の可能性について論ずる．

2. 並行システムモデル NeO

2.1 NeO の定義

この節では本報で提案する並行システムモデル NeO について述べる．

NeO は，基本的にはオブジェクトの集まりとそれらの間のメッセージパッシングよりなる．メッセージは順序づけられた定数の組であり，メッセージパッシングは次のような形で行われる：

（1） 各オブジェクトで作られるメッセージは，オブジェクトとは別に設けられるメッセージキューの1つにいったん送られ，それを他のオブジェクトが受け取るという形で行われる（このとき各キューに蓄えられているメッセージの組をそのキューのキューマーキングと呼ぶ）．

（2） 各オブジェクトで作られるメッセージがどのキューに送られるかということ，および各オブジェクトがどのキューからメッセージを受け取ることができるかということは，あらかじめ定めておくものとする（以下，これらをオブジェクト間の結合関係と呼ぶ）．

各オブジェクトは，内部状態と複数個のレシーバ，複数個のセンダよりなる．内部状態は，各オブジェクトに1つずつ設けられたメモリに蓄えられる複数個の変数（内部変数と呼ぶ）によって表されるものとする．内部変数の種類はオブジェクトごとに定められており，各変数には定数が割り当てられる．メモリに蓄えられる各内部変数に対する定数の割り当てをそのメモリのメモリマーキングと呼ぶ．また，内部変数とそれがとる値の組をセルと呼ぶ．

各レシーバは，複数個のポートをもち，ある条件（メッセージ受理条件）が満たされるときこれらのポートを通してメッセージを受け取ることになるが，この条件の一部を担うものとして，各ポートにはメッセージに応じて真偽が定まる述語（ポート条件）が1つずつ割り当てられる．また各レシーバは，メッセージを受理するとき，メモリの内容を参照・変更するが，レシーバごとに対象とする内部変数の組（対象内部変数集合）が定められ，さらにこの集合の要素ごとに，メッセージ受理条件の一部として用いられる述語（内部変数条件）が割り当てられる．また各レシーバには，受理したメッセージとメモリの内容に応じて対象内部変数集合の内容を変更するときに用いられる関数（状態遷移関数）が割り当てられる．

各センダにはメモリの内容に応じて真偽が定まる述語（メッセージ生成条件）と送信メッセージを1つ定める関数（メッセージ生成関数）が割り当てられる．

オブジェクト間の結合関係は，各センダからのメッセージがどのキューに送られるかを指定する関係と各ポートがどのキューのメッセージを受理の対象とするかを指定する関数によって定まるものとする．ただし，同一のレシーバに属する異なるポートが同じキューからメッセージを受理することはないものとする．オブジェクト O のセンダ s からのメッセージがキュー q に送られるとき，q は s（または O）の出力

キューであるといい，オブジェクトOに属するレシーバrのポートpがキューqのメッセージを受け取る可能性があるとき，qはp（またはr，またはO）の入力キューであるという．キューは複数個のオブジェクトの出力キューとなることもあり，複数個のオブジェクトの入力キューとなることもある．

以上のように定められたオブジェクトの結合関係と，現在のキューマーキング，メモリマーキングに対して，あるレシーバが次の条件(i)，(ii)（メッセージ受理条件）を満たすとき，そのレシーバはメッセージ受理可能であるという．

(i) そのレシーバがもつすべてのポートについて次のことが成り立つ：各ポートの入力キューにそのポートのポート条件を満たすメッセージが少なくとも1つ存在する．

(ii) そのレシーバの対象内部変数集合のすべての要素が，それぞれ内部変数条件を満足する．

NeO はシステムの構造をオブジェクト間の結合関係で表し，システムの状態をキューマーキングとメモリマーキング（以下，両者を合せて単にマーキングと呼ぶ）で表し，状態変化を次のような規則（遷移則）に従うマーキングの変化で表そうとするものである：マーキングの変化は，次の(1)，(2)が繰り返される形で行われる．

(1) 受理可能なレシーバのうち1つが選択される（受理可能なレシーバが存在しないとき，NeO は現在のマーキングのもとでデッドロックの状態にあるという）．

(2) 選択されたレシーバを含むオブジェクトが次の一連の動作を行う．

(i) そのレシーバの各ポートの入力キューからそのポートのポート条件を満足するメッセージを1つ選んで取り出す．

(ii) そのレシーバに割り当てられた状態遷移関数に従ってメモリマーキングを変更する．

(iii) メッセージ生成条件が真であるすべてのセンダについて，メッセージ生成関数に従ってメッセージを1つ生成し，それをそのセンダのすべての出力キューに送る．

ここで(1)におけるレシーバの選択の任意性と(2)(i)におけるメッセージの選択の任意性に基づいてシステム動作の非決定性が生ずることとなる．このように NeO は並行システムがもつ非決定性をこのような形に残したままでモデル化するものである．

図1に，NeO の形式的な定義を示す．次節以下，用語，記号はこれに従うものとする．

2.2 NeO の図的表現

NeO の図的表現を**図2**に示す．楕円で囲まれた有向グラフは1つのオブジェクトを表し，○はキューqを表す．個々のオブジェクトは◎で表される1つのメモリm_e，▮で表される複数個のレシーバr，▯で表される複数個のセンダsおよびそれらを結ぶアークとオブジェクト間の結合関係を表すアークから構成される．ノードn_iからノードn_jに至るアークを$\overline{n_in_j}$で表し，条件，関数のインデックスのうちアークから明らかであるものについては省略する．$\overline{qr}$，$\overline{m_er}$，$\overline{rm_e}$に付けられたラベルはそれぞれポート条件$\alpha_r[p]$，内部変数条件β_r，状態遷移関数γ_rを，$\overline{m_es}$，$\overline{sq}$に付けられたラベルはそれぞれメッセージ生成条件$\beta[s]$，メッセージ生成関数$\gamma[s]$を表す．以下，モデルの記述はこの表現に従うものとする．

<u>NeO</u> NeO N は次の6項組である：

$N=(\{O^i\}i\in I,\ Q,\ \bigcup_{i\in I} C^i,\ \{F^i\}i\in I,\ M,\ K)$

(1) I：オブジェクトのインデックスの有限集合

(2) $\{O^i\}i\in I$：次のように定義されるオブジェクト O^i の集合

$O^i=(X^i, R^i, S^i, \beta^i, \gamma^i)$

(a) X^i：オブジェクト O^i の内部状態を表す変数（内部変数）の有限集合，
ただし，$X^i\cap X^{i'}=\phi\ (i\neq i')$

(b) R^i：次のように定義されるオブジェクト O^i に含まれるレシーバの有限集合
$R^i\ni r=(X_r, P_r, \alpha_r, \beta_r, \gamma_r)$，ただし，$R^i\cap R^{i'}=\phi\ (i\neq i')$

i) $X_r\subseteq X^i$：r で参照・変更される内部変数の集合（対象内部変数集合）

ii) P_r：レシーバrがもつポートの有限集合，
ただし，$P_r\cap P_r'=\phi$ ($r, r'\in R^i$，$r\neq r'$ または $r\in R^i$，$r'\in R^{i'}$，$i\neq i'$)

iii) α_r：$P_r\ni p\mapsto\alpha_r[p]\in[M\to\{真, 偽\}]$
各ポートにメッセージに関する条件（ポート条件）を割り当てる関数

iv) β_r：$X_r\ni x\mapsto\beta_r[x]\in[K\to\{真, 偽\}]$
rが対象とする各対象内部変数に，その値に関する条件（内部状態条件）を割り当てる関数

v) γ_r: $M^{|P_r|}\times K^{|X_r|}\rightarrow[X_r\rightarrow K]$: 状態遷移関数

(c) S^i: オブジェクト O^i のセンダの有限集合，ただし，$S^i\cap S^{i'}=\phi\ (i\neq i')$

(d) β^i: $S^i\ni s\mapsto\beta^i[s]\in K^{|X^i|}\rightarrow\{$真, 偽$\}]$
各センダにメモリの内容に関する条件（メッセージ生成条件）を割り当てる関数

(e) γ^i: $S^i\ni s\mapsto\gamma^i[s]\in[K^{|X^i|}\rightarrow M]$
各センダにメモリの内容に応じてメッセージを１つ作り出す関数（メッセージ生成関数）を割り当てる関数

(3) Q: キューの有限集合

(4) $\bigcup_{i\in I} C^i\subseteq S^i\times Q$: オブジェクトとキューの送信側の結合関係

(5) $\{F^i\}i\in I$: $F^i=\bigcup_{r\in R^i} f_r$,
f_r: $P_r\rightarrow Q$ はオブジェクトとキューの受信側の結合関係を表す単射

(6) M: 順序づけられた有限個の定数の組の集合（その要素をメッセージと呼ぶ）

(7) K: 定数の集合

<u>マーキング</u>

μ: $Q\rightarrow\mathcal{M}(M)$: キューマーキング

λ_r: $X_r\rightarrow K$: レシーバ r の対象内部変数集合 X_r に定数を割り当てる関数

λ^i: $X^i\rightarrow K$: オブジェクト O^i の内部変数集合に定数を割り当てる関数 $(X^i=\bigcup_{r\in R^i} X_r)$
この関数をオブジェクト O^i のメモリマーキングと呼び，$x\in X^i$ と $\lambda^i(x)$ の組をセルと呼ぶ

次のように定義される λ: $X\rightarrow K$ を NeO のメモリマーキングと呼ぶ $(X=\bigcup_{i\in I} X^i)$

$\lambda(x)=\lambda^i(x),\ \ x\in X^i$

$\delta=(\mu,\lambda)$: NeO のマーキング

<u>諸定義</u>

(1) $R=\{r\mid i\in I,\ r\in R^i\}$: NeO に含まれるレシーバの集合

(2) $P=\bigcup_{i\in I}(\bigcup_{r\in R^i} P_r)$: NeO に含まれるポートの集合

(3) $S=\bigcup_{i\in I} S^i$: NeO に含まれるセンダの集合

(4) レシーバ r に対応するインデックス i を $i(r)$ とおく.

(5) ポート p が属するレシーバを $r(p)$ とおく.

(6) f: $P\ni p\mapsto f_{r(p)}[p]\in Q$ と定義する.

(7) $p\in P$ に対して次の集合を定義する.
$T_p=\{m\in M\mid m\in\mu(f(p)),\ \alpha_{r(p)}[p](m)=$真$\}$

(8) $r\in R$ が次の条件を満足するとき r は受理可能であるという.
i) すべての $p\in P_r$ に対して $T_p\neq\phi$
ii) すべての $x\in X_r$ に対して $\beta_r[x](\lambda^{i(r)}(x))=$真

(9) 受理可能なレシーバ全体の集合を受理可能集合という.

(10) $r\in R$ に対して ${}^{\bullet}r=\{f(p)\mid p\in P_r\}$

(11) $s\in S$ に対しては $s^{\bullet}=\{q\in Q\mid(s,q)\in C^{i(s)}\}$，$i(s)$ はセンダ s に対するインデックス i

<u>NeO の遷移則</u>　次の(1)，(2)を繰り返す.

(1) 受理可能集合の要素の１つを選択し r とする.

(2) r に対して次の i)，ii)，iii) を行う.
i) すべての $p\in P_r$ について $f(p)$ より T_p の要素の１つを除く.
除いたものを $(m_p)_{p\in P_r}$ とおく.
ii) $\lambda_r\leftarrow\gamma_r((m_p)_{p\in P_r},\ (\lambda_r(x))_{x\in X_r})$
X_r の内容を書き換える.
iii) $\beta^{i(r)}[s]((\lambda^{i(r)}(x))_{x\in X^{i(r)}})$ が真となるすべての $s\in S^{i(r)}$ について次の(a)(b)を行う.
(a) $m=\gamma^{i(r)}[s]((\lambda^{i(r)}(x))_{x\in X^{i(r)}})$ とおく.
(b) すべての $q\in Q$, $(s,q)\in C^{i(s)}$ について
$\mu(q)\leftarrow\mu(q)\cup\{m\}$

上の(2)をレシーバ r によるメッセージの受理と呼ぶ.

<u>記号について</u>

(1) $|A|$: 集合 A の基数

(2) $[A\rightarrow B]$: A を定義域とし B を終集合とする関数の全体

(3) $\mathcal{M}(M)$: 集合 M の多重集合の全体

図 1 NeO の定義
Fig. 1 Formal definition of the NeO.

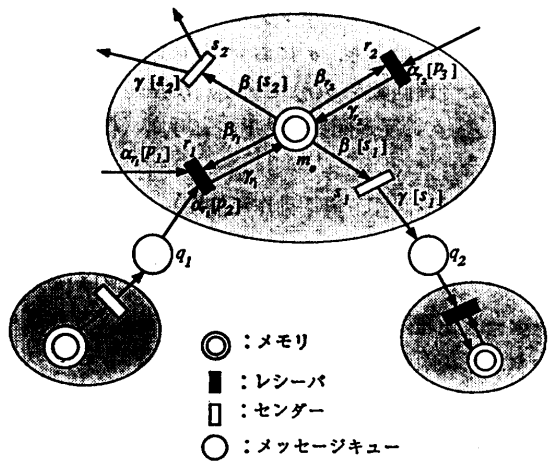


図 2 NeO の図的表現
Fig. 2 Pictorial representation of the NeO.

3. NeO による並行システムのモデリング

3.1 同期・競合・同時送信

並行システムの特徴はプロセスの間の同期，競合，あるプロセスから他の複数個のプロセスへの同時送信が生ずることである．以下，これらの動作の NeO によるモデリングについて述べる．

（1） 同 期

図 3（a）は，キュー q_1 に $\alpha_r[p_1]$ を満たすメッセージが届き，q_2 に $\alpha_r[p_2]$ を満たすメッセージが届くという 2 つの条件が同時に満たされるときにのみオブジェクト O のレシーバ r がメッセージ受理可能となることを示す．すなわち，r からのプロセスは，s_1 を含むプロセスからのメッセージと s_2 を含むプロセスからのメッセージが r で同期がとられたとき開始する．

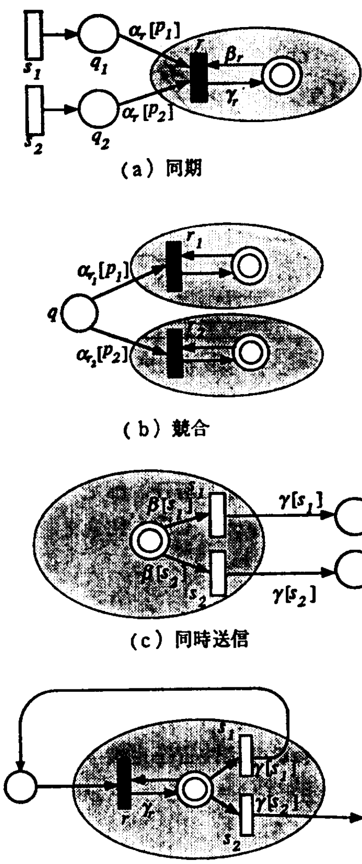
（2） 競 合

図 3（b）は r_1 を含むプロセスと r_2 を含むプロセスが競合する例である．メッセージ m は条件 $\alpha_{r1}[p_1]$, $\alpha_{r2}[p_2]$ のいずれをも満たすものとする．m が q に到着すると r_1, r_2 の両者が受理可能となる．しかし，いずれか一方のプロセスのみが開始することとなる．NeO ではどちらが開始されるかは定めない．

（3） 同 時 送 信

図 3（c）のように容易に表現できる．$\beta[s_1]$, $\beta[s_2]$ が同時に満足されると $\gamma[s_1]$, $\gamma[s_2]$ に従ってメッセージがそれぞれ生成され，q_1, q_2 に同時に送られる．

待ち行列システム等のモデリングにあたっては次々とあるものを生成するオブジェクトがしばしば必要となる．図 3（d）に自然数を次々と生成するオブジェクトの例を示す．



$\gamma_r\ (m \in M, n \in K): X_r \ni x \mapsto n+1 \in K$，$K$ は自然数の集合，$X_r = \{x\}$
$\gamma[s_1]: K \ni n \mapsto m \in M$
$\gamma[s_2]: K \ni n \mapsto n \in M$

（d） 自然数の列の生成

図 3 並行システムに特徴的なプロセス間の関係
Fig. 3 Typical relations of processes in concurrent systems.

3.2 モデリングの例

1 台の CPU と 3 台の IO 装置 IO_1, IO_2, IO_3（IO_1 と IO_3 は同等の処理能力をもつ）からなるあるコンピュータシステムを NeO を使ってモデリングした例を**図 4**（ただし α, β, γ の表示は省略）に示す．CPU の前のキュー q_1 に到着した 3 種類のジョブはその種類に従って定まる演算処理を受けたのち，ジョブの種類ごとに指定されたキューに送られる．IO_1 は r_1 または r_2 を終えて q_2 に蓄えられているジョブの中から任意に 1 つを選び出しそれに処理 r_4 を行い再び CPU に送り出す．IO_2 は r_2 を終えたジョブと r_3 を終えたジョブがそれぞれ 1 つ以上そろったとき処理 r_5 を行い新しいジョブを生成したのちそれを CPU に送り出す．また，IO_3 は r_3 を終えて q_4 にあるジョブに処理 r_6 を行いそれを CPU に送り出す．r_5 は r_2 を終

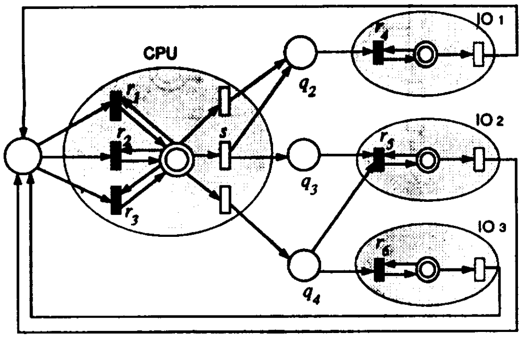


図 4　コンピュータシステムの NeO による表現
Fig. 4　A NeO for a computer system.

えたジョブと r_3 を終えたジョブがそろってはじめて開始できる同期的な処理であり，r_5 と r_6 は競合関係にある処理である．また，CPU に含まれるセンダ s は q_2, q_3 に同時送信を行う．

4. NeO の解析

4.1 NeO の諸性質とその解析

この節では，NeO について，並行システムの解析・設計に有用と思われる諸性質を定義するとともに，NeO がその諸性質をもつかどうかを判定するための条件について述べる．

【定義 1】(可達性)　N を NeO とする．

(1)　N のマーキング δ_0 のもとで，レシーバ r がメッセージ受理可能であって，r による受理によってマーキングが δ に遷移するとき，条件

$$\delta_0[r>\delta$$

が成立するという．

(2)　マーキング δ_0 とレシーバの列 $\sigma=(r_1,\cdots,r_k)$ に対して，条件

$$\delta_0[r_1>\delta_1\cdots\delta_{k-1}[r_k>\delta_k$$

が成立するとき，σ は δ_0 からのメッセージ受理の列であるといい，δ_k は δ_0 から σ によって可達であるといって

$$\delta_0[\sigma>\delta_k$$

で表す．

レシーバの無限列 $\sigma=(r_1,r_2,\cdots)$ は，“任意の自然数 k に対して部分列 $(r_1,\cdots,r_k)$ がメッセージ受理の列である” とき，メッセージ受理の列であるという．

(3)　マーキング δ_0, δ に対して，δ_0 からのメッセージ受理の列 σ (空列も含む) が存在して，$\delta_0[\sigma>\delta$ であるとき，δ は δ_0 から可達であるといい，δ_0 から可達なマーキング全体の集合を δ_0 の可達集合と呼んで $[\delta_0>$ で表す．

(4)　σ をあるマーキングからのメッセージ受理の列とする．このとき，σ におけるレシーバ r のメッセージの受理回数を $\bar{\sigma}(r)$ で表し，R 型のベクトル $\bar{\sigma}: R\to\boldsymbol{N}$ を σ の受理回数ベクトルと呼ぶ ($\boldsymbol{N}$ は非負整数の集合)．■

【定義 2】(活性，周期動作可能性)　N を NeO とする．

(1)　N に対して，マーキング δ_0 とそれからのメッセージ受理の列 σ が存在して，σ のもとですべてのレシーバが限りなく多数回メッセージ受理を行うとき，N は活性であるという．

(2)　N に対して，マーキング δ_0 とそれからのメッセージ受理の列 σ が存在して，σ のもとですべてのレシーバが少なくとも 1 回はメッセージの受理を行ってもとのマーキング δ に戻るとき，N は周期動作可能であるという．またこのとき，ベクトル $\bar{\sigma}=(\bar{\sigma}(r_1),\cdots,\bar{\sigma}(r_{|R|}))$ の成分の比 $\bar{\sigma}(r_1):\cdots:\bar{\sigma}(r_{|R|})$ を σ の相対メッセージ受理率と呼ぶ．■

この定義より，NeO N が活性ならば，適当な初期マーキングを選ぶことによってデッドロック状態とならないように，しかも N に含まれるすべてのオブジェクトのすべての手続き (レシーバに対応) を有効に用いて N (あるいは N でモデル化されるシステム) を運用できることがわかる．また，明らかに N が周期動作可能ならば N は活性である．

【定義 3】(数マーキング)　N を NeO とし $\delta=(\mu,\lambda)$ を N のマーキングとする．このとき，各 $q\in Q$ に q に含まれるメッセージの数を対応させる写像を数マーキングと呼び，$\bar{\mu}$ で表す：

$$\bar{\mu}: Q\ni q\mapsto|\mu(q)|\in\boldsymbol{N}$$ ■

【定義 4】(有界，保存的)　N を NeO とする．

(1)　N の任意のマーキング δ_0 と任意のマーキング $\delta=(\mu,\lambda)\in[\delta_0>$ に対して $\bar{\mu}$ が有界 ($n\in\boldsymbol{N}$ が存在して任意の $q\in Q$ に対して $\bar{\mu}(q)\langle n)$ であるとき，N は有界であるという．

(2)　N に対して $\eta>0$ なる Q 型のベクトル $\eta(\eta: Q\to\boldsymbol{N})$ が存在して，任意のマーキング δ_0 と任意のマーキング $\delta=(\mu,\lambda)\in[\delta_0>$ に対して

$$\eta^{\mathrm{T}}\bar{\mu}=\eta^{\mathrm{T}}\bar{\mu}_0$$

が成立するとき，NeO は保存的であるという．■

N が保存的であるとは，N に含まれるメッセージの数の各 $q(\in Q)$ に重みを付けた加重和が一定であることを表す．また，定義より明らかに，N が保存的ならば N は有界である．

NeO が定義1，2および4で定義した諸性質をもつための条件について考察するため，次の仮定 H が成立する NeO N に対して，N のメッセージパッシング行列なる概念を導入する．

(H) N のレシーバ r がメッセージを受理するとき，r が属するオブジェクトのすべてのセンダ s について，s から $q\in s^{\cdot}$ に送られるメッセージの数（$=w(r\,;\,s,q)$ とおく）がメモリの内容によらない． ■

【定義5】(メッセージパッシング行列，インバリアント) N を (H) を満たす NeO とし，N のレシーバ r がメッセージを受理するときに $q\in{}^{\cdot}r$ から取り出されるメッセージ数（これはメモリの内容に依存しない）を $v(q,r)$ とする．

(1) $Q\times R$ 型の行列 $B: Q\times R\ni(q,r)\mapsto B(q,r)\in \boldsymbol{N}$ を次のように定義し，メッセージパッシング行列と呼ぶ．ただし，

$$B(q,r)=\begin{cases}-v(q,r) & (q\in{}^{\cdot}r-\bigcup_{s\in S^{i(r)}}s^{\cdot})\\ \sum_{s\in S^{i(r)}}w(r\,;\,s,q) & (q\in\bigcup_{s\in S^{i(r)}}s^{\cdot}-{}^{\cdot}r)\\ -v(q,r)+\sum_{s\in S^{i(r)}}w(r\,;\,s,q) & (q\in{}^{\cdot}r\cap(\bigcup_{s\in S^{i(r)}}s^{\cdot}))\\ 0 & (\text{その他})\end{cases}$$

(以下，メッセージパッシング行列 B をもつ NeO N を $N(B)$ とも書くこととする)

(2) $N(B)$ に対して，条件

$$Bx=0$$

を満たす R 型のベクトル $x(\neq 0)$ を N の R-インバリアントと呼び，条件

$$y^{\mathrm{T}}B=0$$

を満たす Q 型のベクトル $y(\neq 0)$ を N の Q-インバリアントと呼ぶ． ■

B および $\bar{\sigma}$ の定義より明らかに次の命題1が成立する．

【命題1】 $N(B)$ に対して $\delta_0[\sigma\rangle\delta$ であるとき，

$$\bar{\mu}=\bar{\mu}_0+B\bar{\sigma}$$ ■

この命題と Q-インバリアントの定義より，明らかに次の定理が成立する．

【定理1】 δ を $N(B)$ の δ_0 から可達なマーキングとする．このとき，

(1) 任意の Q-インバリアント y に対して，

$$y^{\mathrm{T}}\bar{\mu}=y^{\mathrm{T}}\bar{\mu}_0$$

(2) $Bx=\bar{\mu}-\bar{\mu}_0$ を満たす $x\geqq 0$ が存在する． ■

【定理2】 $N(B)$ に対して，次の(1)，(2)についてそれぞれ(a)⇒(b)が成立する．

(1) (a) N は活性である．

(b) $Bx\geqq 0$ となる $x>0$ が存在する．

(2) (a) N は周期動作可能である．

(b) $x>0$ なる R-インバリアントが存在する．

証明 (1) 対偶法による：$Bx\geqq 0$ となる $x>0$ が存在しないならば，付録に示すように，$B^{\mathrm{T}}\eta\lneqq 0$ を満たす $\eta\geqq 0$ が存在する．ここで $\eta^{\mathrm{T}}B=(k_1,\cdots,k_n)$ とおき，集合 $R=\{r_1,\cdots,r_n\}$ を次のように2つに分割する：$R_-=\{r_j\,|\,k_j<0\}$, $R_0=\{r_j\,|\,k_j=0\}$

このとき，

$r\in R_0$ に対して，$\delta[r>\delta'$ ならば，$\eta^{\mathrm{T}}\bar{\mu}=\eta^{\mathrm{T}}\bar{\mu}'$

$r\in R_-$ に対して，$\delta[r>\delta'$ ならば，$\eta^{\mathrm{T}}\bar{\mu}>\eta^{\mathrm{T}}\bar{\mu}'$

であることに注意する．ここで N が δ_0 とともに活性であって，$\delta_1,\delta_2,\delta_3,\cdots$ を R_- に属するレシーバがメッセージ受理を行った直後のマーキングの列とすると，

$$\eta^{\mathrm{T}}\bar{\mu}_0>\eta^{\mathrm{T}}\bar{\mu}_1>\eta^{\mathrm{T}}\bar{\mu}_2>\eta^{\mathrm{T}}\bar{\mu}_3>\cdots$$

すなわち，正の数の列が限りなく単調減少することとなって，矛盾である．よって，仮定のもとで，$N(B)$ は活性ではない．

(2) 周期動作可能ならば，δ と $\bar{\sigma}>0$ なる σ が存在して $\delta[\sigma>\delta$. 命題1より $B\bar{\sigma}=0$. ■

定理2(2)の証明から，N がメッセージ受理の列 σ によって周期動作可能ならば，σ の相対メッセージ受理率が R-インバリアントの1つであることが必要であることがわかる．

【定理3】 $N(B)$ に対して，次の(1)，(2)についてそれぞれ(a)⇒(b)が成立する．

(1) (a) $B^{\mathrm{T}}y\leqq 0$ となる $y>0$ が存在する．

(b) N は有界である．

(2) (a) $B^{\mathrm{T}}y=0$ となる $y>0$ が存在する．

(b) N は保存的である．

証明 定義3と命題1より明らかである． ■

4.2 解析例

以下に図4に示したコンピュータシステムモデルの解析例を示す．このモデルのメッセージパッシング行列 B_1 は，

$$B_1=\begin{bmatrix}-1 & -1 & -1 & 1 & 1 & 1\\ 1 & 1 & 0 & -1 & 0 & 0\\ 0 & 1 & 0 & 0 & -1 & 0\\ 0 & 0 & 1 & 0 & -1 & -1\end{bmatrix}$$

であり，$N(B_1)$ の R-インバリアントは，

$$x=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\1&1&0\\0&1&0\\0&-1&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}$$

となる．これより6個の相対メッセージ受理率のうち3個は自由に選べるが，他はそれらに伴って定まってしまうことがわかる．さらに，$x_2=x_3$ ならば，$x_6=0$ となり，定理2より $N(B_1)$ は周期動作可能でないことがわかる．$x_6=0$ は r_2 に到着するジョブ数と r_3 に到着するジョブ数が等しいならば，IO_3 はいかなるジョブも処理しないことを表す．

ここで各IO装置で処理されるジョブ数の均等化を図るため，図4において IO_1 に送り出されていた r_2 を終えたジョブを**図5**のように q_4 に送り出しそれを IO_3 で処理することを試みる．このときモデルのメッセージパッシング行列 B_2 は，

$$B_2=\begin{bmatrix}-1&-1&-1&1&1&1\\1&0&0&-1&0&0\\0&1&0&0&-1&0\\0&1&1&0&-1&-1\end{bmatrix}$$

となり，$N(B_2)$ の R-インバリアントは，

$$x=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}$$

となる．したがって，$x_1=x_2=x_3$ のとき，$x_4=x_5=x_6$ となり，処理 r_1, r_2, r_3 で行われるジョブ数が等しいとき各IO装置において処理されるジョブ数も等しくなる．

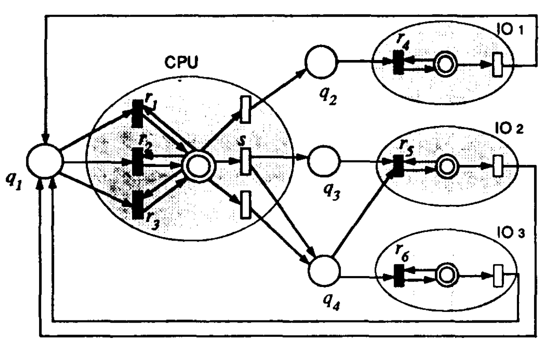


図5 変更されたコンピュータシステムに対するNeO
Fig. 5 The NeO for the modified computer system.

5. 考　　察

5.1 NeO とペトリネットとの関係

次のような性質(1)，(2)をもつ特別なNeOを考え，キューの集合をプレースの集合と考えるとそれは完全にペトリネットに一致する．

(1) ただ1種のメッセージをもつ：$M=\{m\}$．

(2) 各オブジェクトが次の性質をもつ．

① 内部状態をもたない（内部変数条件，メッセージ生成条件は常に真となるものとする）．

② ただ1つのレシーバとただ1つのセンダをもつ（合せて1つのトランジションと考える）．

③ ポート条件は m に対して常に真となる．

④ メッセージ生成関数によって常にメッセージ m が生成される．

前章の定理2，定理3は，それぞれ，活性，周期動作可能性に対しては必要条件であり，有界性，保存性に対しては十分条件であるが，上のように特殊化されたNeO（ペトリネット）に対しては必要十分条件となることが知られている[6]．

5.2 NeO によるシステムの評価について

離散系のシステム評価を行う1つの方法は，次のものの組によってシステムを記述し，事象スケジューリング方式によってシミュレータを構成する事である[7]．

(a) システムの状態から生起可能な状態事象の組を定める写像

(b) システムの状態から時間事象の組を定める写像

(c) 各事象の生起に伴う状態遷移を定める写像

NeOに対して，各レシーバのメッセージ受理を状態事象とすると，遷移則のうちの現在のマーキングから受理可能なメッセージの組を定める部分が(a)に対応することになる．

ここで送信メッセージがセンダを出てキューへ届くまでの時間を考慮に入れることとし，メッセージのキューへの到着を時間事象と考える．こうすると(b)に対応する写像，(c)に対応する写像は次のように自然に定まる．

(b)：メッセージ生成関数 γ^i

(c)：状態遷移関数 γ_r とキューに届いたメッセージをキューに加える部分

以上のように送信メッセージの移動の部分に時間（乱数でもよい）を加えるのみでシミュレーション方式を定めることができる．前章での解析結果は，もちろんこのシミュレーション結果にも当てはまる．

6. おわりに

本報では，並行システムのための，特にそれの制御問題を考察するためのモデルとして，ペトリネットとオブジェクト指向を融合した１つのモデルを定義し，それに対して可達性，活性，周期動作可能性，有界性，保存性なる概念を導入するとともに，システムがこのような性質をもつための条件について考察した．また，システムが周期動作を行うとき，各レシーバがどのような割合でメッセージを受理することができるかということを解析する手法を示した．しかし，本報での条件はいずれも必要条件または十分条件であり，これが必要十分条件となるようなクラスの設定あるいは必要十分条件の開発が今後の課題である．

本報で提案した NeO は１つのオブジェクト指向並行プログラミング言語[8]とみることができる．しかし，NeO ではそれの制御構造の解析という面を重視し，多くのオブジェクト指向言語にみられる動的なメッセージパッシング，動的なオブジェクトの生成は考慮しなかった．しかしながら，これらの導入も今後の課題であると思われる．

本報ではまた，メッセージ送信に時間を導入するだけで NeO に基づく１つのシミュレーション方式を定めることができることを示した．現在，これに基づくシミュレータを中心とした並行システム解析設計支援システム Cosmos[9] の開発を進めているが，これについては別に報告したいと思っている．

参考文献

1) 山本，浦：離散型シミュレーション言語の現状と将来の展望(１)(２)，情報処理，Vol. 22, No. 9, pp. 836-845; No. 11, pp. 1012-1023 (1981).

2) Hore, C. A. R.: Communicating Sequential Processes, *Comm. ACM*, Vol. 21, No. 8, pp. 666-677 (1978).

3) Milner, R.: A Calculus of Communicating Systems, *Lecture Notes in Computer Science 92*, Springer-Verlag (1980).

4) Peterson, J. L.: *Petri Net Theory and Modeling of Systems*, Prentice-Hall (1981).

5) 米沢：オブジェクト指向計算の現状と展望，情報処理，Vol. 29, No. 4, pp. 290-294 (1988).

6) Murata, T.: Petri Nets and Their Applications, 計測と制御，Vol. 22, No. 3, pp. 3-11 (1983).

7) Inomata, T., Onogi, K., Nakata, Y. and Nishimura, Y.: A Rule-based Simulation System for Discrete Event Systems, *J. Chem. Eng. Jpn.*, Vol. 21, No. 5, pp. 482-489 (1988).

8) 石川，所：オブジェクト指向並行プログラミング言語，情報処理，Vol. 29, No. 4, pp. 325-333 (1988).

9) 猪股，片野田，西村：Cosmos のプログラミング環境について，情報処理学会ソフトウェア工学研究会資料，SE 60-2 (1988).

付録　$\nexists x: Bx \geqq 0,\ x > 0 \Rightarrow \exists \eta: B^{\mathrm{T}}\eta \lneq 0,\ \eta \geqq 0$ の証明

Tucker の定理（２組の不等式系(１) $Ax=0$, $x \geqq 0$, (２) $A^{\mathrm{T}}u \geqq 0$ が $A^{\mathrm{T}}u + x > 0$ を満たす解をもつ）より，x, y, η が存在して，

$$\begin{bmatrix} -B^{\mathrm{T}} \\ E \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} x \\ y \end{bmatrix} = 0 \tag{1}$$

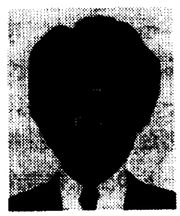
$$\begin{bmatrix} x \\ y \end{bmatrix} \geqq 0 \tag{2}$$

$$\begin{bmatrix} -B^{\mathrm{T}} \\ E \end{bmatrix} \eta \geqq 0 \tag{3}$$

$$\begin{bmatrix} -B^{\mathrm{T}} \\ E \end{bmatrix} \eta + \begin{bmatrix} x \\ y \end{bmatrix} > 0 \tag{4}$$

ただし，E は単位行列．(３)式より $B^{\mathrm{T}}\eta \leqq 0$, $\eta \geqq 0$ であるから，"$B^{\mathrm{T}}\eta \lneq 0$, $\eta \geqq 0$ が解をもたない" とすると $B^{\mathrm{T}}\eta = 0$ でなければならない．このとき(４)式より $x > 0$ で，(１)，(２)式より $Bx = y \geqq 0$ となって，仮定に反する．■

(平成元年２月 13 日受付)
(平成２年１月 16 日採録)



猪股　俊光（正会員）

1961 年生．1989 年豊橋技術科学大学大学院博士後期課程修了．同年豊橋技術科学大学助手，現在に至る．並列計算モデル，ビジュアルプログラミング環境の研究に従事．工学博士．電子情報通信学会，日本ソフトウェア科学会，人工知能学会などの会員．

片野田守人（正会員）

1962 年生．1989 年豊橋技術科学大学大学院修士課程修了．同年東レ・ダウコーニング・シリコーン(株)入社．現在は研究開発本部技術部において工場の CIM に関する研究開発に従事．

小野木克明

1951 年生．1978 年名古屋大学大学院博士課程修了．名古屋大学工学部助手，豊橋技術科学大学講師を経て 1988 年同大学助教授，現在に至る．プロセスシステム工学の研究に従事．工学博士．計測自動制御学会，システム制御情報学会，化学工学会などの会員．

西村　義行

1938 年生．1966 年東京大学大学院博士課程修了．東京大学工学部講師，名古屋大学工学部助教授を経て 1980 年豊橋技術科学大学教授，現在に至る．システム工学，情報科学の研究に従事．工学博士．計測自動制御学会，システム制御情報学会，化学工学会などの会員．