No.464

業務システムの一般モデル

by

佐藤 亮

June 1991

業務システムの一般モデル

筑波大学　社会工学系
佐藤亮

日本語要旨

多くの情報システム方法論が提案され、計算機をベースとした経営情報システム構築に使われている。方法論による分析対象は、情報システムを含む業務システムである。方法論の成果は属人性を持つといわれているが、方法論の有効性を高めるためには、分析や設計の操作性と客観性を高める必要がある。また，経営情報システムのシステム理論を構築していく上でも業務システムのモデル化が必要である。

本論文では、定型的業務システムをトランザクション処理システムとしてとらえ、その静的モデルと動的モデルを構築する。動的モデルはファイルシステムを状態の一部として持ち、離散事象システムとしての業務システムの状態表現であることが示される。ファイルシステムがデータモデリング機能によって定式化され、有限資源条件の下で業務活動の開始を条件づけるという意味で活動の同期をとっていることが示される。このモデルは、業務システムに対する理論的基礎となる。

英語要旨

Identifiable amount of information systems methodologies has been used in building management information systems. The target systems of the analysis are business systems with information systems. Products of a methodology usually depend on the personal ability of the analysts and designers. If we have neat and manipulable theory on business systems and methodologies then we can learn and use them more effectively.

In this paper business systems are modelled as transaction processing systems. We focus on both the static and the dynamic aspect of the systems. It is turned out that a business system is a discrete event system and that the dynamic model is a state space representation of it. The file system is modelled using a data modelling facility, consists of a part of state and makes parallel processing of transactions possible under the finite resource condition. The model constructed in this paper provides a theoretical foundation of business systems.

キーワード
業務システム、データモデル、ファイルシステム、状態表現、離散事象システム、トランザクションシステム

業務システムの一般モデル

1 はじめに

情報システム方法論は経営情報学の重要なテーマのひとつである。現在、数多くの方法論が提唱され、実際に経営情報システムの構築に使われている。情報システム方法論は、業務システムの分析から始まり情報システムのインストールならびに運用に至るまでに、それぞれ独自の技法や概念やドキュメントを使用する。方法論はそれを使う分析者や設計者の個人的能力に依存するという属人性をもち、標準的なドキュメント形式を定めただけでは分析や設計の客観性を高めるには十分ではない。方法論に従った分析の成果物であるドキュメント群が情報システムを含む業務システムの設計図だとするなら、そのような設計図が表現しようとするところの業務システム自体を客観的で形式的に基礎づけることが望まれる。

方法論について、とりわけ業務システム分析についてのこのような問題状況は広く認識されている。Olle (1988) は多くの方法論を統一的に記述するための概念枠組を、実体-関係モデル（Chen, 1976; 酒井, 1987）に似たデータモデリング機能を使って記述している。どんな概念が使われるのかということと、それらの関係の有無が表の形で表されている。高原(1991)はデータ処理システム(DPS) のオートマトンモデルを定義した後、定型的業務システムで行なわれる処理は交換であるとの観点から、実体-関係モデルを使って交換のデータモデルを構築している。佐藤(1990)はある方法論のドキュメント群のデータに注目しTHデータモデル（穂鷹, 1989)によってデータベース化している。これらは方法論の静的な記述モデルを提供しているといえる。

一方で、実務における対処策としては、方法論のドキュメントの標準化やそれを踏まえたCASE (computer aided software engineering) ツールといわれる分析支援や部分的自動化のアプローチがとられている（竹下, 1990; 加藤， 1990）。

また，定型的業務システムの動的側面からのモデルとして、島田（1987, 1988）は定

型的業務のふるまいにし注目して、取引の一連の処理をワーキングデータベースとして記録して用いることを提案している。

本論文は、指向としては島田(1987, 1988)と同じように定型的業務システムの動的側面の把握に注目し、しかしシステム論の立場から、直接に業務システムの静的モデルと動的モデルを明らかにする。すなわち、定型的業務システムにおけるトランザクションやファイルの役割が定式化され、状態空間を導入することによって、離散事象システムとしての業務システムの状態表現が構成される。状態空間はファイルシステムと内部処理終了の予定時刻とから構成され、また、ファイルは各業務処理の開始を条件づけるものとして相互作用を媒介する。このモデルは、定型的業務システムひとつの理論的基礎を与えるものである。すなわち業務システムを厳密にかつ客観的に論じる概念枠組みとなる。なお、命題の証明は付録にまとめてある。

2　記法と基本概念

本節ではMesarovic and Takahara(1989)、Sato(1990) に基づき、本論文で用いる記法と基本概念を述べる。a が集合Aの要素であることを$a \in A$と表わす。AがA' の部分集合であることを$A \supseteq A'$ とかく。集合A とBの直積を$A \times B$で表わす。すなわち$A \times B = \{(a, b) \mid a \in A$ かつ$b \in B\}$。集合A とBとの関係 R は直積の部分集合である。すなわち$A \times B \supseteq R$である。A とBとの関係f がA からBへの関数であるとは、A の任意の要素に対応するBの要素必ず存在すること、及びもし$(a, b) \in f$ かつ $(a, b') \in f$ ならば$b = b'$ なることである。f をAからBへの関数、A' をAの部分集合とするとき、f のA' への制限 f' とは、$f' = \{(a, b) \mid a \in A'$ かつ$(a, b) \in f\}$である。f'を$f \mid_{A'}$と記す。集合A とBの和を$A \cup B$で表わす。すなわち$A \cup B = \{c \mid c \in A$または $c \in B\}$。集合Aのべき集合(power set) を$P(A)$ と表わす。$P(A) = \{A' \mid A \supseteq A'\}$。A からBへの関数f のAが時間を表わす集合であるとき、f を時間関数と呼ぶ。

定義 1　システム

システムとは入力集合と出力集合の関係である。特に入出力集合が時間関数のとき、

時間システムと呼ぶ。

時間関数の値の集合をアルファベットと呼ぶ。本論文では時間軸Tは非負実数R^+の区間とする。xをT上の関数つまり時間関数とする。$t \leq t'$なる任意の$t, t' \in T$について関数xを$[t, t')$に制限したものを$x_{tt'}$と書く。すなわち$t \leq s < t'$である任意のsについて$x_{tt'}(s)=x(s)$。x, x'を同じアルファベットへの任意の時間関数とする。このとき、任意の$t \in T$について新たな関数x''を次のように定義する：

$x''(t'') =$

$x(t''), t'' < t$のとき；

$x'(t''), t'' \geq t$のとき.

x''をx_{0t}とx'_tの連接(concatenation)と呼び、$x'' = x_{0t} \cdot x'_t$と表わす。

状態表現はシステムの動的側面をとらえるための概念枠組みである。

定義2　*状態表現*

時間システム$S \subseteq X \times Y$が出力アルファベットBを持つとする。

$$\phi = \{\phi_{tt'} \mid \phi_{tt'} : C \times X_{tt'} \rightarrow C \text{ and } t, t' \in T, t \leq t'\}$$

$$\mu : C \rightarrow B$$

とするとき 組$\langle\phi, \mu\rangle$がSの状態表現であるとは(i), (ii)の条件を満たすことである：

(i) 関数集合ϕが下の(a), (b)を満たす：

(a) $\phi_{tt''}(c, x_{tt''}) = \phi_{t't''}(\phi_{tt'}(c, x_{tt'}), x_{t't''})$, ただし$t \leq t'' \leq t'$かつ$x_{tt''} = x_{tt'} \cdot x_{t't''}$

(b) $\phi_{tt}(c, x_{tt}) = c$

(ii) $(x,y) \in S$であることと、ある$c \in C$があって任意の$t \in T$について

$$y(t) = \mu(\phi_{0t}(c, x_{0t}))$$

が成立することが同値である。

Cを$\langle\phi, \mu\rangle$の状態空間と呼ぶ。(i)のϕを遷移関数族、μを出力関数と呼ぶ。

状態表現は時間システムの動的ふるまいを因果的に認識するために広く用いられてい

る。Mesarovic and Takahara(1989)によれば時間システムが因果的であることと状態表現が可能であることが等価である。

遷移関数族ϕと出力関数μとの組$<\phi, \mu>$が定義する入出力システムをRes($<\phi, \mu>$)で表わす。すなわち、$(x,y) \in$ Res($<\phi, \mu>$)であることと、ある $c \in C$ があって任意の $t \in T$ について $y(t) = \mu(\phi_{0t}(c, x_{0t}))$である。$<\phi, \mu>$がSの状態表現であるときは、もちろん Res($<\phi, \mu>$)=Sである。

定義３　*離散事象システム*

時間システム$S \subseteq X \times Y$が以下の条件を満たすとき離散事象システムと呼ばれる。

1) $T=[0, T_{end})$, $T_{end} \in R^+$; 有限の時間軸：

2) 入力アルファベットはある集合のべき集合である：

3) 任意の$(x, y) \in S$ に対して次の２条件が成立する：

3-1) $event(x) = \{t \mid x(t) \neq \{\}$, ただし$\{\}$ は空集合$\}$ が有限;

3-2) $F(y) = \{[t, t') \mid t \leq t'' < t'$なる任意の t'' について$y(t) = y(t'')$ かつ $y(t) \neq y(t')\}$ が有限であり、かつ、$\cup F(y) = T$ となること。

定義４　*入力空間の時間リスト表現：L(X)*

Xを離散時間システムの入力関数としA'を入力アルファベットするとき、関数$L: X \rightarrow \{T \rightarrow \{A' \rightarrow R^+\}\}$ を次で定める。任意の $x \in X$, $t \in T$, $b \in A'$に対して

$L(x)(t)(b) =$

T_{end}, $t \leq t' < T_{end}$, なる任意のt'について$x(t')$がbを含まないとき;

$\min\{t' \mid t'>t$ なるt' について$b \in x(t')\}$, その他のとき。

$L(x)$をxの時間リスト表現と呼ぶ。

$L(x)$ も時間関数であるのでその制限を定義できる。以下では、記法の簡単のために$L(x)_{tt'}$を$Lx_{tt'}$と書く。Lは1対1対応であるので逆関数が定まる。それをL^{-1} と書く。

離散事象システム$S \subseteq X \times Y$の入力空間を$L(X)$にしたものを$L(S) \subseteq L(X) \times Y$とかき、また、

逆にL(S)の入力空間をXにしたシステムを$L^{-1}(L(S))$と書く。

Sを離散事象システムとするときL(S)の状態表現をSの離散事象状態表現と呼ぶ。離散事象システムの状態表現構築におけるこのような入力空間の変換はZeigler(1976, 1989)にもみられる。

3　データモデル

ファイルシステムのデータ構造を記述するためのデータモデリング機能としてDAE (Domain - Attribute - Entityの略)をを用いる。DAE はTHデータモデリング機能（穂鷹、1990）を簡単化したものである。

ＤＡＥデータモデリング機能は，いくつかの集合，それらの間の関数，それらが満たすべき２種類の制約からなる．順に述べる。

定義５　*データモデリングの補助集合*

ES：実体（entity）を要素とする集合で，実体集合と呼ぶ．

ETS：実体型(entity type）を要素とする集合で，実体型集合と呼ぶ．

DS：ドメインを要素とする有限集合で，ドメイン集合と呼ぶ．

KS：キー実体型(key entity type)を要素とする集合で，キー実体型集合と呼ぶ．キー実体型は管理実体型とも呼ばれる．KSはDSの部分集合．

FTS：ファイル型を要素とする有限集合で，ファイル型集合と呼ぶ．FTSはETSの部分集合．

AS：属性を要素とする集合で，属性集合と呼ぶ．

定義６　*データモデリングの補助関数*

S：DS→P(ES)

P(ES)はESの全ての部分集合を要素とする集合．

FT：KS→FTS；ファイル型関数と呼ぶ．全射的とする．すなわち、任意の$y \in FTS$に対してある$x \in KS$があって$y = FT(x)$となる。KSの要素xに対し，FT(x)をxのファイル型と呼ぶ．

A：FTS→P(AS)；属性関数と呼ぶ．$A(FT(x)) = \{a_1, \ldots, a_k\}$とするとき，各$a_i$をFT(x)の属性と呼ぶ．

dom：AS→DS；ドメイン関数と呼ぶ．dom(a)をaのドメインと呼ぶ．

LES：リスト実体を要素とする集合で，リスト実体集合と呼ぶ．

$LES = \cup \{P(S(dom(a_1)) \times \ldots \times S(dom(a_n))) \mid$ KSのある要素xについて$A(FT(x)) = \{a_1, \ldots, a_n\}\}$とする。

$File = \{f: FTS \rightarrow LES \mid$ fはファイル内容関数$\}$．ここでfがファイル内容関数であるとは，KSの任意の要素xに対して，$A(FT(x)) = \{a_1, \ldots, a_n\}$とするとき$f(FT(x))$が$S(dom(a_1)) \times \ldots \times S(dom(a_n)) \supseteq f(FT(x))$を満たすことである．ファイル型を図１のような表で表した場合、f(FT(x))はある時点におけるFT(x)の中身を表現するものである．

$V_f: f(FTS) \times AS \rightarrow S(DS)$　という関数を各ファイル内容関数fに対して次のように定義する．$A(FT(x)) = \{a_1, \ldots, a_n\}$とする．f(FT(x))の任意の要素lとA(FT(x))の任意の要素a_iに対して，$V_f(l, a_i)$はlから$S(dom(a_i))$への射影であること．この関数V_fはfが文脈から明らかなときには，単にVとかく．

なお，以下の条件が成立しているものとする；

$ETS = DS \cup FTS$(直和，$DS \cap FTS$は空集合)

$ES = ETS \cup LES \cup AS \cup S(DS)$

y			
a_1	a_2	a_3	a_4
v_{11}	v_{12}	v_{13}	v_{14}
v_{21}	v_{22}	v_{23}	v_{24}
v_{31}	v_{32}	v_{33}	v_{34}

図１′表を使ったファイルの表現

図１で、yはファイル型、つまりあるキー実体型x（KSの要素）があって$y=FT(x)$である。$A(y)=\{a_1, a_2, a_3, a_4\}$つまり、yの属性は$a_1, a_2, a_3, a_4$とする。$l_1 = (v_{11}, v_{12}, v_{13}, v_{14}), l_2 = (v_{21}, v_{22}, v_{23}, v_{24})$などとすると、$f(FT(x))=f(y)=\{l_1, l_2, l_3\}$である。また$V_f(l_1, a_2)=v_{12}$などが成立する。

ファイルシステム内のデータに関する２種類の一貫性保持条件を定める。準備として主キーを定義する。

定義７　*主キー(primary key)*

key：FTS → AS は，次の条件を満たす関数である：

任意のファイル内容関数fと，任意のz_1，$z_2 \in f(FT(x))$に対して

$[V_f(z_1, key(FT(x)))=V_f(z_2, key(FT(x)))$ならば$z_1=z_2]$．

このとき、key(FT(x))をFT(x)の主キー、あるいは単にキーと呼ぶ。

任意の$x \in KS$について$dom(key(FT(x)))=x$とする。

定義８　*参照キー(referential key, またはforeign key)*

任意のキー実体型x_1，x_2とファイル内容関数fをとる．$a \in A(FT(x_2))$が，$FT((x_1))$からのfにおける参照キー(Rキー，または外部キー)であるとは，

1) $dom(key(FT(x_1)))=dom(a)$ かつ，

2) $V_f(f(FT(x_1)), key(FT(x_1))) \supseteq V_f(f(FT(x_2)), a)$ なること。

参照キーは，ファイルの属性値間の整合性条件としての参照整合性を表現するのに使われる.

定義９　参照関係

任意のファイル内容関数ｆをとる．ｆにおける参照関係 $REFER_f \subseteq AS \times AS$ を次で定める：

$(a,b) \in REFER_f$ とはある x_1，$x_2 \in KS$ があって次の２条件を満たすこと：

1) $a \in A(FT(x_2))$，$b = key(FT(x_1))$，

2) aが $FT(x_1)$ からのｆにおける参照キーである.

定義１０　*継承整合性(inheritance)*

任意の x_1，$x_2 \in KS$ とファイル内容関数ｆをとる．$FT(x_1)$ は $FT(x_2)$ のｆにおける部分ファイル型である、または、$FT(x_2)$ は $FT(x_1)$ のｆにおける汎ファイル型である、または、ｆにおいて $FT(x_1)$ は $FT(x_2)$ の属性を継承するとは：

1) $FT(x_1)$ の主キーは $FT(x_2)$ からの参照キーである。すなわち：

任意の $f \in File$ に対して,

$$V_f(f(FT(x_2)), key(FT(x_2)) \supseteq V_f(f(FT(x_1)), key(FT(x_1)));$$

2) $FT(x_1)$ は $FT(x_2)$ のキー以外の属性を含む，つまり，$FT(x_1)$ はキー以外の属性を継承する。すなわち：

$$A(FT(x_1)) \setminus \{key(FT(x_1))\} \supseteq A(FT(x_2)) \setminus \{key(FT(x_2))\};$$

3) キー値が等しければキー値以外の継承属性値は一致する。すなわち：

任意の $a \in A(FT(x_2)) \setminus \{key(FT(x_2))\}$, $y_1 \in f(FT(x_1))$, $y_2 \in f(FT(x_2))$ に対して,

$[V_f(y_1, key(FT(x_1))) = V_f(y_2, key(FT(x_2)))$ ならば $V_f(y_1, a) = V(y_2, a)]$;

定義１１　*継承関係*

任意のファイル内容関数 f をとる.

$INHRT_f \subseteq FTS \times FTS$ を、$(z_1, z_2) \in INHRT_f$ であることは、f において z_1 は z_2 の属性を継承することであると定める。

継承整合性の第1の条件より、次は明らかである。

命題 1

任意の $x_1, x_2 \in KS$ とファイル内容関数 f に対して、

$(FT(x_1), FT(x_2)) \in INHRT_f$ ならば, $(key(FT(x_1)), key(FT(x_2))) \in REFER_f$ である。

参照関係、継承関係の例は穂鷹(1989)を参照されたい。これらはファイルが全体として保持すべきデータ間の関係を表現する。

定義１２　*ファイルの一貫性保持関係*

File 上に一貫性保持関係 $C \subseteq File \times File$ を定義する。

$(f, f') \in C$ とは、$REFER_f = REFER_{f'}$ かつ $INHRT_f = INHRT_{f'}$ なることである。

Cは任意の時点でのファイルが一貫性条件を満たしているかどうかを表現する関係である。ファイル構造とそれに対する一貫性条件を同時に指定することで、ファイルシステムを定義する。

定義１３　*ファイルシステム*

ファイルシステムとは、組$(File, REFER_0, INHRT_0)$ である。ただし、$REFER_0$, $INHRT_0$はあるひとつのファイル内容関数における参照関係と継承関係である。

本論文の以下の議論では、ひとつのファイルシステム$(File, REFER_0, INHRT_0)$ を任意に固定する。

4　業務システムの静的モデル

トランザクションを外生的なものと内生的なものに分け、前者はファイルにデータを記録し、後者はファイルからのデータを参照したりファイルを更新したりするものとしてとらえる。トランザクションを担う役割が存在する。

定義１４　*外生トランザクションと外部役割*

E：外生トランザクションを要素とする有限集合。

RE：外生トランザクションを起こす役割を要素とする有限集合。

f_{RE}：RE→E　外生トランザクションとそれを処理する役割の対応を表す関数であり、１対１対応とする。

f_{EF}：E→P(KS)　外生トランザクションが起こると、あらかじめ決められたファイルにデータが記録される。外生トランザクションが発生したときに影響するファイルを表わす関数。

定義１５　*内生終了トランザクションと内部役割*

I：内生終了トランザクションを要素とする有限集合。ファイルがある条件を満たすときに発生する処理の終了を表わす。

RI：内生終了トランザクションを起こす役割を要素とする有限集合。

f_{RI}：RI→E　内生終了トランザクションとそれを起こす役割の対応を表す関数であり、１対１対応とする。

f_{IF}：I→P(KS)　内生終了トランザクションが起こると、あらかじめ決められたファイルにデータが記録される。内生終了トランザクションが発生したときに影響するファイルを表わす関数。

RE と RI は互いに素とする。

定義１６　*内生割り当てトランザクション*

D：割り当てトランザクションの集合で有限集合。

$f_{FD}: F \to P(D)$　ファイルの状況により可能な処理の開始として割り当てトランザクションをいくつか決める。

$f_{DI}: D \to I$　割り当てトランザクションにより発生した処理の終了トランザクションを対応させる関数。

$f_{DF}: D \to P(F)$　割り当てトランザクションがデータを記録するいくつかのファイルを指定する。

割り当てトランザクションの発生はある内部処理の開始を意味し、ファイルの状況により可能な処理を開始し、対応する標準時間後の終了時に内生終了トランザクションが発生する。

E, I, D は互いに素とする。$E = \{1,2, ..., k\}$, $I = \{k+1, k+2, ..., n\}$, $D = \{n+1, ..., n+w\}$ とする。

定義１７　*業務取引システムの静的表現*

ファイルシステム、外生トランザクション、内生割り当てトランザクション、内生終了トランザクションの全体を業務取引システムの静的表現という。形式的には、

$((File, REFER_0, INHRT_0), E, RE, f_{RE}, f_{EF}, I, RI, f_{RI}, f_{IF}, D, f_{FD}, f_{DI}, f_{DF})$ という組をいう。

5　業務取引システムの動的表現

業務取引システムの入力は外生トランザクションである。外生トランザクションを外部取引とも呼ぶ。

本節では、外部取引とそれに付随した内部取引のダイナミクスを全体として記述する。外部取引の発生は離散的で、同時並行的に起こると考える。

定義１８　*時間軸*

$T = [0, T_{end}]$という有限区間を時間軸とする。ただし$T_{end} \in R^{+}$である。

定義１９　*外部トランザクション発生の特徴*

外部トランザクションの時間変化を次の特徴を持つ時間関数xの集合Xとする。

$x : T \rightarrow P(A')$とするとき、

1)アルファベットはある集合のべき集合P(A')である：

2) $event(x) = \{t \mid x(t) \neq \{\}$, ただし$\{\}$は空集合$\}$ が有限;

定義２０　*役割が持つ情報*

$AnRoleData = T \times Avail$, $Avail = \{true, false\}$

$f_{予定時刻} : RI \cup RE \rightarrow AnRoleData$

AnRoleDataはひとつの役割が持つ情報であり、役割に対応するトランザクションが発生する予定時刻を示す。役割は、対応するトランザクションを発生する時刻が有効な情報であるかを示すためにAvailという項目が付随しており、値がtrueのとき有効であり、falseのとき無効であることを表わす。システム全体では各役割に対応する情報として、$f_{予定時刻}$で表現され、ひとつの時点における役割の持つ情報を表現するが、可能な$f_{予定時刻}$全体の集合を$F_{予定時刻}$と表わす。RIとREは互いに素なので、$f_{予定時刻}$を$f_{予定時刻} \mid_{RI}$と$f_{予定時刻} \mid_{RE}$とに分割して考えることができる。$F_{I予定時刻} = \{f_{予定時刻} \mid_{RI}\}$、また、$F_{E予定時刻} = \{f_{予定時刻} \mid_{RE}\}$とかく。任意の$f_{予定時刻} \in F_{予定時刻}$は$f_{予定時刻} = f_{予定時刻} \mid_{RI} \cup f_{予定時刻} \mid_{RE}$と分割できる。

業務取引に関係するトランザクションすべてを考えて、トランザクションの処理を記述する。そのため、外生も内生も同じように扱う。

定義２１　*トランザクション処理*

外生トランザクション、内生終了トランザクションの処理を定める。

1）発生

$f_{Scan}: F_{予定時刻} \rightarrow T$, $f_{Scan}(f_{予定時刻}) = \min\{t \mid$ ある$r \in RI \cup RE$について$f_{予定時刻}(r) = (t, true)\}$

$f_{Dues}: F_{予定時刻} \rightarrow P(E \cup I)$,

$f_{Dues}(f_{予定時刻}) = \{g \mid (g = f_{RI}(r)$ or $g = f_{RE}(r))$ かつ ある$r \in RI \cup RE$について

$f_{予定時刻}(r) = (f_{Scan}(f_{予定時刻}), true)\}$

$f_{Dues}(f_{予定時刻})$は現在発生予定のトランザクションの集合である。

2）トランザクションによるファイルの更新

$f_{更新T}$： $F_{予定時刻} \times File \rightarrow File$

トランザクションにはあらかじめ同時実行可能になった場合の優先順位を付けておく。トランザクションの識別子としてつけた番号とその優先順位が同じとして一般性を失わない。$f_{Dues}(f_{予定時刻}) = \{i_1, i_2, \ldots, i_m\}$とする。優先順位は添え字のとおりとする。

$f_{更新T}(f_{予定時刻}, f) = f_m$、ただし、$f_0 = f$とし、$1 \leq j \leq m$なる$j$について

$f_j =$

$f_{FileVal}(i_j, f_{j-1})$，$(f_{j-1}, f_{FileVal}(i_j, f_{j-1})) \in C$のとき；

f_{j-1}，$(f_{j-1}, f_{FileVal}(i_j, f_{j-1})) \notin C$のとき。

また、$f_{FileVal}: (E \cup I) \times File \rightarrow File$ はファイルの更新を表わし、任意の$k \in KS$, $i \in E \cup I$, $f \in File$ に対して［ $f_{FileVal}(i, f)(k) \neq f(k) \rightarrow (k \in f_{EF}(i)$ or $k \in f_{IF}(i))$］という性質を持つ。

3）トランザクションの発生時刻の処理。

すべての外生トランザクションの次回の発生時刻を使い、$f_{予定時刻}$を書き換える。発生した外生トランザクションの次回発生時刻だけが書き換えられる。現時点における各外生トランザクションの次回発生時刻を$\beta: E \rightarrow T$とし、このようなβの集合をBとする。

$f_{再設定R}$：$F_{予定時刻} \times B \rightarrow F_{予定時刻}$

$f_{再設定R}(f_{予定時刻}, \beta) = f'_{予定時刻}$ とする；

$f'_{予定時刻}(i) =$

$f_{予定時刻}(f_{RI}^{-1}(i)),\ f_{RI}^{-1}(i) \in RI, i \notin f_{Dues}(f_{予定時刻})$ のとき;

$(t', \mathrm{false}),\ f_{RI}^{-1}(i) \in RI, i \in f_{Dues}(f_{予定時刻})$ のとき(ここで t' は任意);

$(\beta(r), \mathrm{true}), r = f_{RE}^{-1}(i) \in RE$ のとき。

定義２２　*割り当てトランザクションの処理*

現在時刻をｔとする。

１）発生

ファイルの値に依存して、f_{FD} により決められる割り当てトランザクションが発生する。

２）ファイルの更新

$f_{更新D}(f_{予定時刻}, f) = f_{n+w}$、ただし、$f_0 = f$ とするとき、$n+1 \le j \le n+w$ なる任意のjに対して

$f_j =$

$f_{DFileVal}(j, f_{j-1}),\ (f_{j-1}, f_{DFileVal}(j, f_{j-1})) \in C$ のとき;

$f_{j-1},\ (f_{j-1}, f_{DFileVal}(j, f_{j-1})) \notin C$ のとき。

また、$f_{DFileVal} : D \times File \rightarrow File$ はファイルの更新を表わし、任意の $k \in KS, j \in D, f \in File$ に対して[$f_{DFileVal}(j, f)(k) \neq f(k) \rightarrow k \in f_{DF}(j)$]という性質を持つ。

３）割り当てトランザクションによって開始された処理の終了を内生トランザクションを担う役割に設定する。

$f_{再設定D} : F_{予定時刻} \times File \rightarrow F_{予定時刻}$

$f\ (f_{予定時刻}, f) = f'_{予定時刻}$ とするとき、

$f'_{予定時刻}(r) =$

$f_{予定時刻}(f_{RE}^{-1}(e))$, 各 $e \in E$ に対し;

$f_{予定時刻}(f_{RI}^{-1}(i)), i \notin f_{DI}(f_{FD}(f))$ または $f_{FD}(f) = \{\}$ のとき;

$(t + f_{標準時間}(i), \mathrm{true}), i \in f_{DI}(f_{FD}(f))$ のとき。

ただし、$f_{標準時間}$：$I \to T$ は内生トランザクション$i \in I$を終了とする各処理の、標準処理時間である。

定義２３　*一連の処理の構成（動的表現）*

外生内生と割り当てのトランザクションの処理が時間的にどのように一貫してなされていくかを、上述の処理の結合として表現する。

１）外生、内生トランザクションの処理

$f_T : File \times F_{予定時刻} \times B \to File \times F_{予定時刻}$

任意の$(f, d, \beta) \in File \times F_{予定時刻} \times B$　に対して、

$$f_T(f, d, \beta) = (f_{更新T}(d, f), f_{再設定R}(d, \beta))$$

２）割り当てトランザクションの処理

$f_D : File \times F_{予定時刻} \to File \times F_{予定時刻}$

$f_D(f, d) =$

(f, d), $f_{FD}(f) = \{\}$のとき;

$f_D(f_{更新D}(f), f_{再設定D}(d, f))$, それ以外。

以下の議論では次の有限資源条件が満足される場合を考える。

有限資源条件：任意のfについて、$f_{FD}(f)$は有限回の展開で空集合になる。すなわち、任意の$f \in File$についてある非負整数pがあって、$f_{FD}(f_{更新D}{}^{p}(f)) = \{\}$となる。

３）状態遷移関数の定義

$\delta = f_D \cdot f_T$

$\delta^I : File \times F_{予定時刻} \times B \to File \times F_{I\,予定時刻}$

$\delta^I(f, d, \beta) = (f', d')$を$\delta(f, d, \beta) = (f', d'')$かつ$d' = d''|_{RI}$で定める。

$f_{XE} : B \to F_{E\,予定時刻}$を、各$r \in RE$について$f_{XE}(\beta)(r) = (\beta(r), true)$と定める。

$\phi_{tt'} : File \times F_{I\,予定時刻} \times L(X)_{tt'} \to File \times F_{I\,予定時刻}$を以下のように定義する：

$t<t'$なる任意のt、t'について、

$\phi_{tt'}(f, d_I, Lx_{tt'})=$

$(f_m, d_{I\,m})$, if $t_m \le t'$ and $t_{m+1} > t'$,

(f, d_I), otherwise,

ただしある正の整数 m と$1\le k \le m+1$なる各 kに対して、 $t_0 = t$, $(f_0, d_{I\,0}) = (f, d_I)$, $t_k = f_{Scan}(f_{XE}(Lx_{tt'}(t_{k-1}))\cup d_{I\,k-1})$ であり、$(f_k, d_{I\,k}) = \delta^I(f_{k-1}, f_{XE}(Lx_{tt'}(t_{k-1}))\cup d_{I\,k-1}, Lx_{tt'}(t_{k-1}))$ とする。また、任意のt に対して、 $\phi_{tt}(f, d_I, Lx_{tt}) = (f, d_I)$ と定める。$\phi = \{\phi_{tt'} \mid t, t'\in T, t\le t'\}$ とするとき、 $<\phi, \Pi_F>$ を業務取引システム （(File, $REFER_0$, $INHRT_0$) , E, RE, f_{RE}, f_{EF}, I, RI, f_{RI}, f_{IF}, D, f_{FD}, f_{DI}, f_{DF}） の動的表現（あるいは単に動的表現）呼ぶ。ただしΠ_F は $File\times F_{I\,予定時刻}$からFileへの射影で、任意の(f, d_I)に対して、$\Pi_F(f, d_I) = f$ である。

上の定義はwell-defined であることを示す。そのために、$\phi_{tt'}$の定義におけるt_k が有限個であることを示せばよい。入力空間の性質から、任意のx に対してevent(x) = $\{t_1, t_2, \ldots, t_p\}$ とかける。ただし、$t_1\le t_2\le \ldots \le t_p$とする。各$t_k \in$ event(x)において$f_{再設定R}$と$f_{再設定D}$によってRE, RI の役割に設定される。RE とRIの要素の数は n であるから、$t=0$, $t' = T_{end}$ とした場合でも高々 m＝n*p である。$0<t$、$t'<T_{end}$ の場合はそれ以下となる。

6　業務システムの構造

本節では以上で定義した動的表現の構造とその意味を調べる。はじめに$<\phi, \Pi_F>$の計算規則を導き、次にそれを用いて、ϕが遷移関数族であることを示す。

補題 2

$<\phi, \Pi_F>$ を有限資源条件を満足する業務取引システムの動的表現とする。このとき、

$\phi_{tt'}(f, d_I, Lx_{tt'}) =$

$\phi_{t''t'}(\delta^I(f, f_{XE}(Lx_{tt'}(t)) \cup d_I, Lx_{tt'}(t)), Lx_{t''t'})$, if $t''=f_{Scan}(f_{XE}(Lx_{tt'}(t)) \cup d_I) \leq t'$

(f, d_I), otherwise.

命題 3

$<\phi, \Pi_F>$ を有限資源条件を満足する業務取引システムの動的表現とする。このとき、ϕは遷移関数族である。

次の命題から、動的表現の結果として得られるシステムは離散事象システムを時間リスト表現したものであることがわかる。

命題 4

$<\phi, \Pi_F>$ を有限資源条件を満足する業務取引システムの動的表現とする。このとき、$L^{-1}(Res(<\phi, \Pi_F>))$ は離散事象システムである。

これらの命題より、主要結果の１つを得る。

定理 5

$<\phi, \Pi_F>$ を有限資源条件を満足する業務取引システムの動的表現とする。このとき、$<\phi, \Pi_F>$は$L^{-1}(Res(<\phi, \Pi_F>))$ の離散事象状態表現である。

一般システム理論(Mesarovic and Takahara, 1989) によれば、同じふるまいを生みだす状態表現は無数に存在する。業務取引システムにおけるファイルシステムの役割は、組織内の活動の連係をとってトランザクションの並行処理を可能にするということである。したがって、ファイルシステムにおけるファイルの種類や各ファイルの属性や一貫性までをも規定するものではない。この事実のゆえに、マネジメント目標に適合して業務処理のしくみとファイルシステムの構造を「うまい」方向へ絞り込んでいくためのガイド

ラインとしての情報システム分析の役割が明確になる(佐藤、1991)。

次の命題はトリビアルに成立するが情報システム方法論のドキュメント群の意味を理解する上で、概念的に重要である。

定理 6

$\langle\phi, \Pi_F\rangle$ 、$\langle\phi', \Pi'_F\rangle$ をひとつの業務取引システムの静的表現の動的表現とする。もし、それぞれにおいて時間軸、トランザクションによるファイル更新関数、割り当てトランザクションによるファイル更新関数、割り当てトランザクションによる内部処理の時間が等しければ、すなわち、$T=T'$、$f_{\text{更新D}} = f'_{\text{更新D}}$、$f_{\text{更新T}} = f_{\text{更新T}}$、$f_{\text{標準時間}} = f'_{\text{標準時間}}$が成立すれば$\langle\phi, \Pi_F\rangle = \langle\phi', \Pi'_F\rangle$である。

動的システムのふるまいを制御するためには、状態となっている情報をとらえて自在に使うことだというのが、線形システム理論(Kailath, 1980; Mesarovic and Takahara, 1989)が明らかにしたことのひとつである。それを原価管理などの業務取引システムのパフォーマンス管理に類推するとき、上記定理は業務システム設計におけるねらいどころを一般的な形式で与えている。

7　結論

経営情報学の主要な主題のひとつである経営情報システム（MIS）の理論化の試みとして、定型的業務システムの定式化を行なった。具体的には、ファイルシステムの構造を定式化した上で、業務取引システムを静的レベルと動的レベルにおいてモデル化し、それらの関係を明らかにした。その帰結として、有限性条件のもとで、業務システムの動的表現は離散事象システムとなることが示された。有限性条件は、作業を行なう人間や機械の処理能力の有限性を表していると解釈できる。

ファイルシステムの役割は、必要な形でデータを整理して記録し使うことだが、動的

側面からは、活動の開始を条件付けるという意味で活動の同期をとっていることが示された。

業務取引システムは一般的に業務システムの設計や管理を考える上での客観的基盤となるという意味で，定型的業務システムのシステムモデルとしての価値を持つ。たとえば、情報システム方法論を構成する各ステップで産み出されるドキュメントの意味づけのフレームワークとなる。さらに、高原(1986)や島田(1988)でその必要性が指摘されているような在庫管理業務や売掛買掛業務などの典型的な取引業務のモデル化を形式化して行なう上での根拠ともなる。これらは将来的研究課題である。

参考文献

Chen, P. P.(1976): The entity-relationship model - Toward a unified view of data, ACM trans. Database Systems, vol 1 (1), 9/36.

Kailath, T.(1980): Linear Systems, Prentice-Hall.

加藤英雄(1990)：実践CASE入門 - SDASによる事例 - ，共立出版.

穂鷹良介(1989)：データベースシステムとデータモデル、オーム社.

Mesarovic, M.D., Takahara, Y.(1989): Abstract systems theory, (Lecture Notes in Control and Information Science 116), Springer.

Olle, T.W.(1988): Information systems methodologies, Addison Wesley.

酒井博敬(1987)：情報資源管理の技法 -ERモデルによるデータベース設計，オーム社.

Sato, R.(1990): A formulation of a simulation modelling methodology, Discussion paper series No.437, University of Tsukuba.

佐藤亮(1990)：情報システム開発方法論のデータモデルと実現、情報処理学会研究報告90-DBS-77.

佐藤亮(1991)：情報システム分析、高原康彦・高津信三編「経営情報システム」,日刊工業新聞社.

島田清一 (1987)：中小企業における経営・情報システムの診断，日刊工業新聞社.

島田清一 (1988)：システムエンジニアのための業務分析の手法，日刊工業新聞社.

高原康彦(1986): OAシステム理論序説、オフィスオートメーション, vol 7 (3), 20/25.

高原康彦(1991)：データ処理システムの定式化と設計 - システム論からのアプローチ - 、高原康彦ほか編「システム設計の理論と実際」、近代科学社.

竹下亨(1990)：CASE概説，共立出版.

Zeigler, B.P.(1976) : Theory of modelling and simulation, John Wiley.

Zeigler, B.P.(1984): Mutifacetted modelling and discrete event simulation, Academic Press.

付録

（補題２の証明）

$\phi_{tt'}$の定義において$m\geq 1, t_m \leq t'$かつ$t_{m+1} > t'$とする。定義により、

$\phi_{tt'}(f, d_I, Lx_{tt'}) = (f_m, d_{I\,m})$である。また、$t_1 \leq t'$である。したがって、ある正の整数pがあって　$\phi_{t''t'}(f_1, d_{I\,1}, Lx_{t''t'}) = (f'_p, d'_{I\,p}), t'_p \leq t'$かつ$t'_{p+1} > t'$とかける。ただし$t'_0 = t_1$,

$(f'_0, d'_{I\,0}) = (f_1, d_{I\,1}),\ t'_k = f_{Scan}(f_{XE}(Lx_{tt'}(t'_{k-1})) \cup d'_{I\,k-1})$であり、$1 \leq k \leq p+1$なる各kに対して、$(f'_k, d'_{I\,k}) = \delta^I(f'_{k-1}, f_{XE}(Lx_{tt'}(t'_{k-1})) \cup d'_{I\,k-1}, Lx_{tt'}(t'_{k-1}))$とする。$p+1 = m$と、$1 \leq j \leq p$なる各jに対して$(f'_j, d'_{I\,j}) = (f_{j+1}, d_{I\,j+1})$と$t'_j = t_{j+1}$が成立することを示せば証明は終わる。

まず、$(f'_1, d'_{I\,1}) = (f_2, d_{I\,2})$と$t'_1 = t_2$を示す。

$t'_1 = f_{Scan}(f_{XE}(Lx_{tt'}(t'_0)) \cup d'_{I\,0}) = f_{Scan}(f_{XE}(Lx_{tt'}(t_1)) \cup d_{I\,1}) = t_2$である。

$$
\begin{aligned}
(f'_1, d'_{I\,1}) &= \delta^I(f'_0, f_{XE}(Lx_{tt'}(t'_0)) \cup d'_{I\,0}, Lx_{tt'}(t'_0)) \\
&= \delta^I(f_1, f_{XE}(Lx_{tt'}(t_1)) \cup d_{I\,k-1}, Lx_{tt'}(t_1)) \\
&= (f_2, d_{I\,2})。
\end{aligned}
$$

以下同様にして、$(f'_p, d'_{I\,p}) = (f_{m+1}, d_{I\,m+1})$、$t'_p = t_{m+1}$を得る。　　　　Q.E.D.

（命題３の証明）

$t, t' \in T, t \leq t'$, と$(f, d_I, Lx_{tt'}) \in File \times F_{I\,予定時刻} \times L(X)_{tt'}$を任意にとる。このとき$t \leq t'' \leq t'$なる任意の$t''$について

(*)　$\phi_{tt'}(f, d_I, Lx_{tt'}) = \phi_{t''t'}(\phi_{tt''}(f, d_I, Lx_{tt''}), Lx_{t''t'})$

を示せばよい。ある非負整数mがあって$1 \leq k \leq m+1$なる各kに対して$\phi_{tt'}(f, d_I, Lx_{tt'}) = (f_m, d_{I\,m})$、ただし、$t_0 = t,\ (f_0, d_{I\,0}) = (f, d_I),\ t_k = f_{Scan}(f_{XE}(Lx_{tt'}(t_{k-1})) \cup d_{I\,k-1})$、$t_m \leq t'$、$t_{m+1} > t'$であり、$(f_k, d_{I\,k}) = \delta^I(f_{k-1}, f_{XE}(Lx_{tt'}(t_{k-1})) \cup d_{I\,k-1}, Lx_{tt'}(t_{k-1}))$とする。m=1のときは明らかに(*)は成立する。以下で$1 \leq m$とする。

補題１より、$1 \leq k \leq m$なる任意のkに対して

(**)　$\phi_{t_{k-1}t'}(f_{k-1}, d_{I\,k-1}, Lx_{t_{k-1}t'})$

$$= \phi_{t_k t'}(\delta^I(f_{k-1},\ f_{XE}(Lx_{t_{k-1}t'}(t_{k-1})) \cup d_{I\,k-1},\ Lx_{t_{k-1}t'}(t_{k-1})),\ Lx_{t_k t'})$$

$$= \phi_{t_k t'}(f_k,\ d_{I\,k},\ Lx_{t_k t'})$$

が成立する。

t''を$[t, t')$ から任意にとる。まず、あるk $(0 \leq k \leq m+1)$ について$t'' = t_k$ となる場合を考える。$t'' = t_0$ のときは式(*)は成立するので、$0 \leq k \leq m$の場合を考えればよい。(*)において(**)を$k-1$ 回適用することによって

$$\phi_{tt'}(f,\ d_I,\ Lx_{tt'}) = \phi_{t_k t'}(f_k,\ d_{I\,k},\ Lx_{t_k t'})$$

となる。また、式(**)においてt' を t_k とおいて$k-1$ 回適用することによって次を得る。

$$\phi_{tt_k}(f,\ d_I,\ Lx_{tt_k}) = \phi_{t_{k-1}t_k}(f_{k-1},\ d_{I\,k-1},\ Lx_{t_{k-1}t_k})$$

ここで$Lx_{t_{k-1}t_k}(t_{k-1}) = Lx_{tt'}(t_{k-1})$であることから、

$$\phi_{t_{k-1}t_k}(f_{k-1},\ d_{I\,k-1},\ Lx_{t_{k-1}t_k}) = \phi_{t_k t_k}(f_k,\ d_{I\,k},\ Lx_{t_k t_k}) = (f_k,\ d_{I\,k})$$

を得る。以上のことから、

$$\phi_{tt'}(f,\ d_I,\ Lx_{tt'}) = \phi_{t_k t'}(f_k,\ d_{I\,k},\ Lx_{t_k t'})$$

$$= \phi_{t_k t'}(\phi_{tt_k}(f,\ d_I,\ Lx_{tt_k}),\ Lx_{t_k t'})$$

$$= \phi_{t''t'}(\ \phi_{tt''}(f,\ d_I,\ Lx_{tt''}),\ Lx_{t''t'})$$

次に、$1 \leq k \leq m+1$なるkに対して$t_{k-1} < t'' < t_k$ とする。式(**)を$k-1$ 回適用することにより、

$\phi_{tt'}(f,\ d_I,\ Lx_{tt'}) = \phi_{t_{k-1}t'}(f_{k-1},\ d_{I\,k-1},\ Lx_{t_{k-1}t'})$。また、式(**)により次を得る。

$$\phi_{tt''}(f,\ d_I,\ Lx_{tt''}) = \phi_{t_{k-1}t''}(f_{k-1},\ d_{I\,k-1},\ Lx_{t_{k-1}t''})$$

ここで$t_k > t''$ だから$\phi_{t_{k-1}t''}(f_{k-1},\ d_{I\,k-1},\ Lx_{t_{k-1}t''}) = (f_{k-1},\ d_{I\,k-1})$。

したがってもし$\phi_{t_{k-1}t'}(f_{k-1},\ d_{I\,k-1},\ Lx_{t_{k-1}t'}) = \phi_{t''t'}(f_{k-1},\ d_{I\,k-1},\ Lx_{t''t'})$ が示されれば

$$\phi_{tt'}(f,\ d_I,\ Lx_{tt'}) = \phi_{t''t'}(f_{k-1},\ d_{I\,k-1},\ Lx_{t''t'})$$

$$= \phi_{t''t'}(\phi_{tt''}(f,\ d_I,\ Lx_{tt''}),\ Lx_{t''t'})$$

となり証明は終了する。実際、$Lx_{t''t'}(t'') = Lx_{tt'}(t_{k-1})$ であることから、

$$\phi_{t''t'}(f_{k-1},\ d_{I\,k-1},\ Lx_{t''t'}) = \phi_{t_k t'}(\delta^I(f_{k-1},\ f_{XE}(Lx_{t''t'}(t'')) \cup d_{I\,k-1},\ Lx_{t''t'}(t'')),\ Lx_{t_k t'})$$

$$= \phi_{t_k t'}(\delta^I(f_{k-1},\ f_{XE}(Lx_{tt'}(t_{k-1})) \cup d_{I\,k-1},\ Lx_{tt'}(t_{k-1})),\ Lx_{t_k t'})$$

$$= \phi_{t_k t'}(f_k,\ d_{I\,k},\ Lx_{t_k t'})$$

となる。ここで式(**)より

$$\phi_{t''t'}(f_{k-1}, d_{I\,k-1}, Lx_{t''t'}) = \phi_{t_{k-1}t'}(f_{k-1}, d_{I\,k-1}, Lx_{t_{k-1}t'})$$

となり、証明は終わる。 Q.E.D.

（命題 4 の証明）

$(x, y) \in L^{-1}(Res(<\phi, \Pi_F>))$ を任意にとる。$F(y) = \{[t, t') \mid t \leq t'' < t'$ なる任意の t'' について $y(t) = y(t'')$ かつ $y(t) \neq y(t')\}$ が有限であり、かつ、$\cup F(y) = T$ となることを示せば十分である。

任意の t に対して、ある $(f, d_I) \in File \times F_{I\,予定時刻}$ があって、$y(t) = \Pi_F(\phi_{0t}(f, d_I, Lx_{0t}))$ である。

ϕ の定義より、出力 y の値が変化するのは、$t_0 = t$, $(f_0, d_{I\,0}) = (f, d_I)$ としたとき $t_k = f_{Scan}(f_{XE}(Lx_{tt'}(t_{k-1})) \cup d_{I\,k-1})$ かつ $(f_k, d_{I\,k}) = \delta^I(f_{k-1}, f_{XE}(Lx_{tt'}(t_{k-1})) \cup d_{I\,k-1}, Lx_{tt'}(t_{k-1}))$ を満たす時刻である。しかるに、この個数 k は、動的表現のwell-defined性で示したように有限個である。また、$\cup F(y) = T$ は明らかである。 Q.E.D.

（定理 6 の証明）動的表現の構成の仕方から明らか。 Q.E.D.