

## オフィス情報システムのグラフ記述の一方式†

魚 田 勝 臣††

オフィス情報システムは人間系と機械系によって構成されており、それら間における対話の非常に多いシステムである。本論文はこのようなシステムの構築、運営および維持の過程で利用されるシステムをグラフによって記述する方式の提案に関するものである。これまで、オフィス情報システムを把握する一般性のある包括的な概念は確立されていない。本論文は、それを処理手続きの面から捉え、情報システムを階層構造をもったタスクの集合、各タスクをアクティビティの順序だった集合、と考え、アクティビティの構成および進行、タスクの起動および対話、などの諸概念を明確にし、それらを表現する実用的なグラフ記述方式 (GROSS) を開発した。現状での対話は人対人が主体であるが、将来は人対機械および機械対機械の対話の比重が増大する。GROSS は人間と機械の系を統一された概念と記法で表現でき、かつそれら相互間の対話を明確に記述できるので、将来人間と機械の対話の方式に変遷があっても十分対処できる。概念と記号系は計算機やシステム理論でなく、現実のオフィス情報システムに求めたので実務家が理解しやすい。オフィス・システム業務を熟知した彼らが、システムの運用のみならず、構築および維持にも参加できるようにするため、GROSS は理論面より実用性を重視している。

## 1. ま え が き

オフィスは経営、管理および活動に関する情報を取り扱うところで、さまざまな原資の入力を、いろんな要求をもった使用者が消費するための出力へ変換するユニットである。オフィスには組織区分があり、それぞれがオフィス手続き (office procedure) を分担している。オフィスで行われる仕事には、テキストや様式の編集、記録のファイリング、会議や打合せ、情報の検証、決定、コミュニケーションなどがある。オフィス情報システムはこのような仕事を機械の援助を受けて実施するシステムで、人間系と機械系とで構成される。オフィスにおける業務量の増大と情報処理に対する要求の多様化に伴って、より高度なシステムを構築するための方法論やこれを支援するソフトウェアおよびハードウェアの開発が計算機科学の重要な課題になっている。本論文はこのような状況下にあるオフィス情報システムの構築、運営および維持の過程で利用される、システムをグラフによって記述する方式の提案である。以下、本論文では誤解の生じない限り、オフィス情報システムをたんに“情報システム”と記す。

情報システムは、本来人間系の活動をより容易にかつ効率的にするために機械系を構築し、両系の有機的な連係によって運営管理されるべきであり、対話をベースにした情報システムが確立されるようになって

きた。一方、このようなシステムの構築および維持においても、これまで計算機やシステムの専門家が中心になって進められてきた点が反省され、オフィス業務に精通した実務家 (多くは計算機やシステムに関する非専門家) の参加の下に、人間と機械の相互関連を配慮して進めるべきだという声が高まっている。

しかるに、情報システムを把握する包括的な概念はいまだ確立されていない状態<sup>1)</sup>であり、このことが上に述べた諸要求に十分応えられない理由の一つになっている。本研究では、まず情報システムを捉える概念を明示し、その上で概念に立脚したグラフ記述を提案している。情報システムは人間系と機械系とから成り立つから、このグラフ記述も両者を統一して取り扱える。

この面におけるこれまでの研究では、プログラムの作成・維持の効率向上に主眼が置かれた研究は数多いものの<sup>2)</sup>、人間系の記述に関するものは少なく、さらに両者を統一して取り扱うものは見当たらない。現実には、人間系を表すために事務工程分析図表、計算機系を表すためにランチャートや情報処理用流れ図記号が併用されており、使用者はそれぞれのグラフで独立の概念および記号系を扱うことを余儀なくさせられている。

情報システムは高度に対話的であり<sup>3)</sup>、現在は人対人の対話が主体であるが、将来は人対機械および機械対機械の対話の比重が高まる。したがって、情報システムの進歩は対話方式の変遷として捉えることもでき、人間と機械の系を統一された概念と記法で表現で

† A Graphical Representation of Office Information Systems by KATSUOMI UOTA (Systems Engineering Division, Mitsubishi Electric Corp.).

†† 三菱電機(株)電子システム部

き、かつそれら相互間の対話を明確に記述できる本方式はこれに十分対処できる。

本研究は慶応義塾大学理工学部の浦研究室で、人間を中心に置いた情報システムの確立をテーマにした諸研究の一環として研究された成果の一つで、実務家が使用する実用的なグラフ記述として開発されたものである。概念と記号系を計算機やシステム理論でなく現実のオフィス・システムに求めているので実務家にも理解しやすく、システムの構築、運用および維持への実務家の参加に道を拓くものであるといえる。ここで報告するグラフ記述を GROSS(Graphical Representation of Office Systems) と名づけた。

## 2. 情報システム概念とグラフ記述

情報システムのグラフ記述は、対象としているシステムの捉え方によって違ったものになる。この捉え方には、情報の流れに注目する方法、処理手続きに注目する方法、意思決定に注目する方法などがあるが、ここでは処理手続きに注目し、以下に示す概念で人間系と機械系とで構成される情報システム全体を捉える。

(1) 情報システムの構成：オフィス情報システムはタスク(task)の集合から構成されており、オフィスの目的はこれらのタスクの半順序(partial ordering)での実行を通じて遂行される。タスクの集合はオフィスの組織区分が担当する仕事に対応し、組織が階層構造をもっているのでタスクの集合も階層構造をもっていると考えられる。各タスクはアクティビティ(activity)の順序だった集合である。アクティビティは最小単位であり、実行するための情報をもっている<sup>1)</sup>。

(2) タスクの起動：タスクは電話、来訪などのコミュニケーションを通じた何らかの刺激によるか、時刻の到来ないし時間帯など時間的な要素によるか、あるいは別のタスクの継続として起動される。

(3) アクティビティの構成：アクティビティは処理(process)と処理の対象(operand)とから構成される。前者は後者に対する次に示す(a)~(g)の作業である。(a)準備および作成、(b)ファイリング、ファイルからの抽出およびファイルの廃棄、(c)転記および計算、(d)分類および照合、(e)検査、(f)承認または決定、(g)オフィス内またはオフィス間のコミュニケーション。

検査と承認の違いは、前者が機械化の対象になるのに対し、後者は署名(signature)または押印などによ

る責任を伴う人間の仕事であることである。

オペランドは情報であり、保存情報(file)と伝達情報(text)から構成される。前者は様式の集合、台帳、計算機のファイルなどであり、タスク内だけで通用する一時ファイル(temporary file)とタスク間で通用する永久ファイル(permanent file)とからなり、それぞれ記録(record)の集合である。テキストは画面、様式(document)、メモ、電話、音声などであり、メッセージ(message)の集合である。オフィスには稟議、メイリング、同報通信など種々の活動があるが、これらは上に述べたアクティビティの組合せで表現される。

(4) アクティビティの遷移：個々の処理はそれに関連するオペランドが利用可能な状態になったときに開始され、その終了によって次のアクティビティを活動可能な状態にする。

一つのアクティビティの終了に伴って、複数のアクティビティの集合を同時に並行して進行させること(fork)がある。また、そうして並行した複数のアクティビティの終了に伴って、一つのアクティビティを活動可能な状態にすること(join)がある。フォークは複数の人間による並行活動や計算機によるマルチタスクに相当し、後者はそれらの完了の同期をとることに相当する。

(5) 人間系と機械系：情報システムは人間同士、機械同士および人間と機械との間で対話を行いながら進行する。情報システムが果たすべき役割を人間と機械とで分担し、その接点で対話が行われる。役割分担の境界は機械化の進展に伴って変化する。

以上に示した情報システム概念は、事務工程分析図表や情報処理用流れ図記号の根本になる概念を包含しており、かつ ICN<sup>4)</sup>、Petri-Net<sup>5)</sup>、SREM<sup>6)</sup>、のモデルの意図するところも、アクティビティ自体およびアクティビティと情報の遷移、タスクの階層構造、コミュニケーションおよび起動などの点で概念をあわせもっており、情報システム概念の網羅性において問題は無い。

次にグラフの用途について述べる。

グラフ記述は情報システムの構築、運用および維持の過程で、それらに参与する人間の便宜のために使われるドキュメントの一種である。グラフ記述が使われる過程と使用目的は次に示すとおり多岐にわたる。

(a) システムの分析：グラフはシステム全体の現状を記述し、理解し、問題点を抽出するのに使われる。

表 1 GROSS の記号系  
Table 1 GROSS symbols.

表現の内容		記号	備考
処理ノード	一般		
	決定		承認および決定を表す
処理の流れを示すアーク			矢の先は流れの方向を示す
ANDノード			AND条件によるアクティビティの遷移
テキスト・ノード			
ファイル・ノード	一時ファイル		同一タスク内で有効
	永久ファイル		タスク間で有効
情報アーク			矢の先は情報の方向を示す
情報の遷移を示すアーク			テキスト・ノードまたはファイル・ノードを結合
時間ノード			時刻、時間など
終了			
結合子			
省略		...	

(b)設計および具体化：グラフによって処理方式を記述し具体化する。計算機の場合にはコーディング作業を支援する。この過程では記述内容を見直して誤りの混入を防ぐ役割もグラフが荷う。(c)運用：グラフはシステムの運用方法を理解させる手段として使われる。(d)維持：現状のシステムを再認識し問題点を抽出し、改造を検討し全体との整合性を考える。

グラフ記述がこれらの使用目的に適合するためには、次に示す4点の要求事項を満足する必要がある。(R1) 情報システム全体を記述の対象とすること、(R2) 簡単に記述できること、(R3) 記述されたものが理解しやすいこと、(R4) 正確に記述できること。このうちのR1は人間系と機械系とで構成されているオフィス情報システムの特性上当然の要求であるにもかかわらず、これまで軽視されていた項目である。

### 3. グラフ記述 GROSS

2章で展開した情報システム概念をグラフによって記述する方式を述べる。GROSSの記号系の全体を表1に示す。

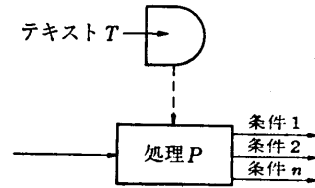


図 1 条件分岐の表現  
Fig. 1 Representation of conditional branch.

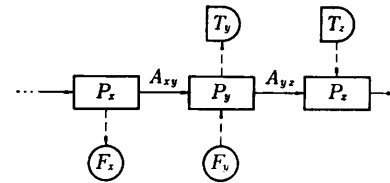


図 2 アクティビティの遷移についての規則  
Fig. 2 Transition rule of activities.

(1) アクティビティの表現：処理および処理の対象(永久ファイル、一時ファイルおよびテキスト)をそれぞれ表1に示したノードで表現する。処理ノードを実線のアークで結んで処理の順序を表す。処理ノードに向かうアークを入り側のアーク、処理ノードから出て行くアークを出側のアークという。処理ノードには点線のアークで処理の対象を関連づける。アークやノードには自然言語による説明文を付加できる。

処理ノードはすべて長方形で統一されており、作業の種別は説明文により表現する。分岐は出側の複数のアークに分岐条件を記載して表現する(図1)。処理ノードの出側として複数のアークが記載されておれば、道が分岐しているのに対応させて、制御の分岐であることは容易に理解できる。分岐に菱形などの特別の記号を用いるのは、実務家にとって覚えなければならない概念を増やすので得策でない。

(2) アクティビティの遷移：図2において処理ノード  $P_y$  は直前の処理ノード  $P_x$  が完了した時点で活動可能状態となり、処理の対象 ( $T_y$  および  $F_y$ ) が利用可能になった時点で処理が開始され、所定の活動が終了すると完了状態となり、直後の処理ノード  $P_z$  を活動可能状態にする。なお、以上の説明で直前/直後の意味はグラフ上の論理的な位置に依存する。

入り側の複数のアークおよびアークの交差は OR 条件であり、AND 条件は AND ノードで表す。入り側にアークがないときは任意の時点で開始可能である。

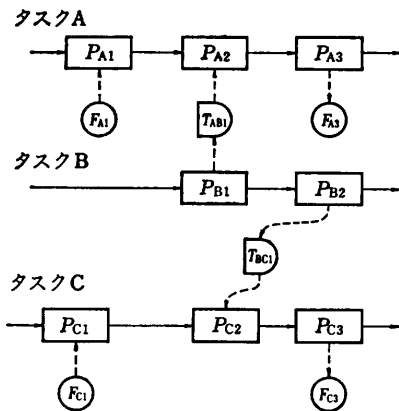


図3 タスクの基本的な記述方式  
Fig. 3 Basic representation of tasks.

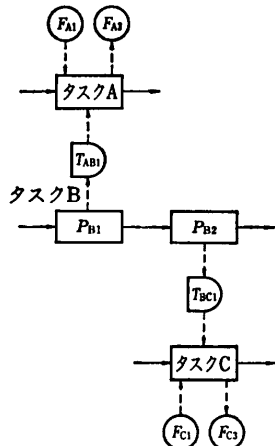


図4 タスクの要約表現  
Fig. 4 Summarized representation of tasks.

フォークは処理ノードに同一の分岐条件を併記した複数の出側アークにより表現する。また、ジョインは処理ノードまたは AND ノードの入り側へアークを連結して、それぞれ、OR 条件および AND 条件を表す。

(3) 情報の遷移の表現：同一のファイルまたはテキストを一点鎖線のアークで結んで、情報の遷移を表すことができる (図6)。

(4) タスクおよびタスク間のコミュニケーションの表現：タスクはタスクを構成するアクティビティの順序だった集合として表現し、テキストで結びつけてタスク間のコミュニケーションを表現する。この記述方式を図3に示す。

(5) タスクの階層構造の表現：タスクおよびタスクを構成するアクティビティは意味のある任意の仕方

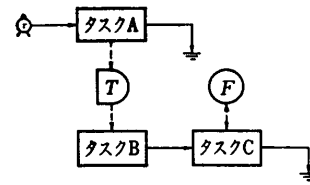


図5 タスクの起動の表現  
Fig. 5 Representation of task activation.

でまとめて表1の記号で表現しなおせる。図4は図3に関して行った例である。この操作は任意の段階で可能で、これを通じてタスクの階層構造を表現できる。

GROSS において処理ノードを長方形に統一し、分岐や分類などに独自の記号を割り付けないのは、階層構造に表現した場合のあいまいさを避ける意義がある。たとえば、計算、分岐および分類に個別の記号を与えたとすれば、この3者で構成される処理ノードはどの記号で表すのが妥当かわからない。

(6) タスクの起動：図5はタスクの起動の例を示したものである。タスクAは時間 $t$ により、タスクBはタスクAからのテキスト $T$ を通じた刺激により、タスクCはタスクBからの継続により、それぞれ起動されることを示す。

(7) 人間系と機械系：人間系と機械系は表1に示した統一された記号系で表現する。ただし、“承認または決定”は人間系にのみ許された処理であり特別な意味をもつので独得の記号で表される。

以上に記述した GROSS の記述方式は、2章で記述した情報システム概念を表現できる。

GROSS では、情報システム内の定型処理を中心に扱う。ただし、定型処理ということについての一般に認められる明白な定義はない。そこでよくいわれるように、予測可能性 (処理方法、手順、入出力イメージなどが前もって把握できること)、繰返し性 (処理が繰り返されること。繰返しの周期は一定していないものも多い)、および連続性 (処理が連続的な流れに沿って行われること)、をあわせもつことを定型的と考える。GROSS は、タイマによる起動などの記述能力を応用することにより、連続性のない処理の一部も取り扱うことができるといえる。ただし、予測困難な処理については、記述できないし、繰返し性のないものは記述の意義が少ない。

#### 4. GROSS によるホテル・システムの記述

3章において定義した GROSS の記述方式によっ

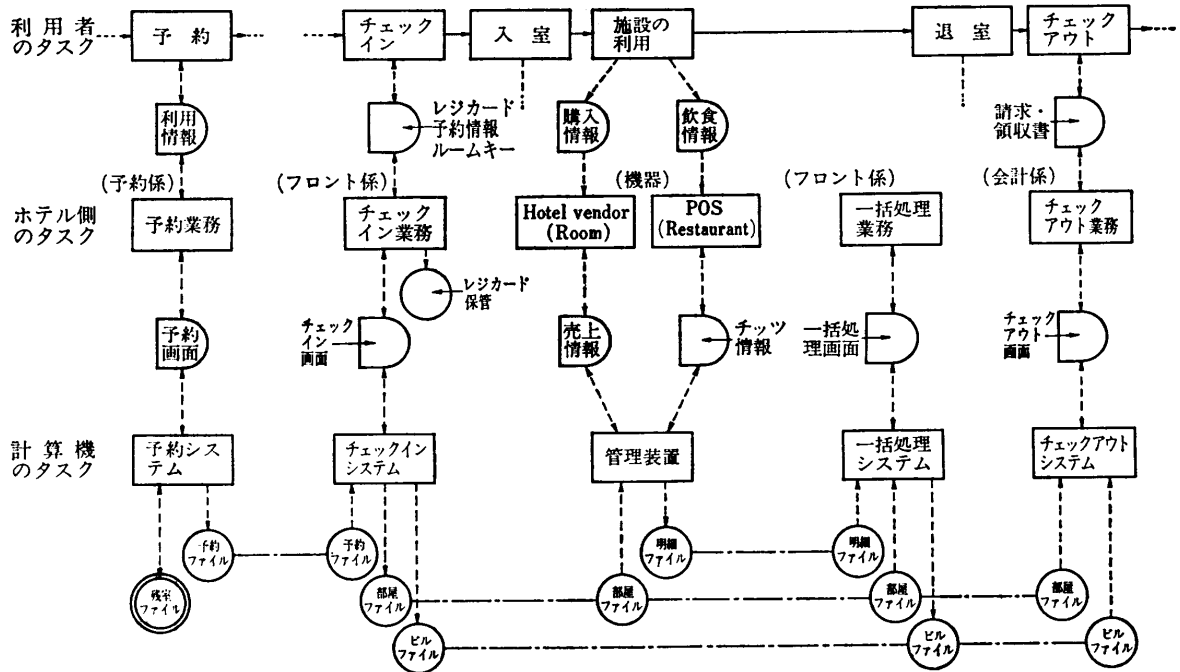


図 6 ホテルの実時間システムの GROSS による記述  
 Fig. 6 GROSS representation of a real-time hotel system.

てホテル・システムを記述し、GROSS の記述能力を例証する。ホテル・システムを選択した理由を以下に示す。(a)フロントや問合せなどの実時間処理と統計計算などのバッチ処理が含まれたシステムである。(b)人と人、とくに不特定多数の人との間での干渉が多く、対話的な処理が効果的である。(c)人間と機械との相互作用 (interaction) が多い。(d)機械と機械との相互作用も含まれる。たとえば、フロントの端末機とレストランの POS (Point Of Sales) 端末や部屋のベンディング・マシンとの結合など。(e)小規模ながらホテル全体を対象とした情報システムが実現されつつあり、人間を中心に置いたシステムの将来の問題点を指摘できる可能性がある。

この章では、GROSS によって、ホテル・システムの主要部分、その中のチェックイン・システムの詳細、およびチェックインの中の予約者照合部分のアクティビティ表現、をそれぞれ記述する。

(1) 実時間処理システムの GROSS による記述

ホテル・システムは、予約、フロント、会計、レストランの POS 端末、電話、ベンディング・マシンなどの営業システムと、経理、売掛金、用度などの管理システムとに大別される。前者はホテル要員、利用者および機械の間で対話しながら 24 時間運営される実

時間処理システム、後者は一定の周期で集計や統計などの計算を行う一括処理、がそれぞれ主体のシステムである。

図 6 は実時間処理に関する情報システム全体を GROSS によって記述したものであり、営業システムの全貌、予約、利用者の到着から退館までが示されている。この図ではおおむね上に人間系、下に計算機システムが記述されており、各タスクおよび情報の相互関連が示されている。ホテル側のタスクの“機器”の個所にはベンディング・マシンや自動精算用の POS 端末によるタスクが人間系と同一の記述方法で記述されている。この部分はホテル要員の作業を機械化したものであるが、GROSS では処理内容が同じなので同一の記号で表される。

計算機のタスクのファイルについて、鎖線のアークが同一のファイルを結合し、ファイルの遷移を表している。人間系についても同様のことが行える。

図 6 における予約業務、予約システムなどの各タスクは独立に動きテキスト情報によって関連づけられている。このことは関係者が各システム間の相互関係の理解を通じて運営が可能であることを示している。たとえば、入室後の利用者から施設が利用できない旨の連絡を受けた部屋係は、図 6 のグラフから、ベンディン

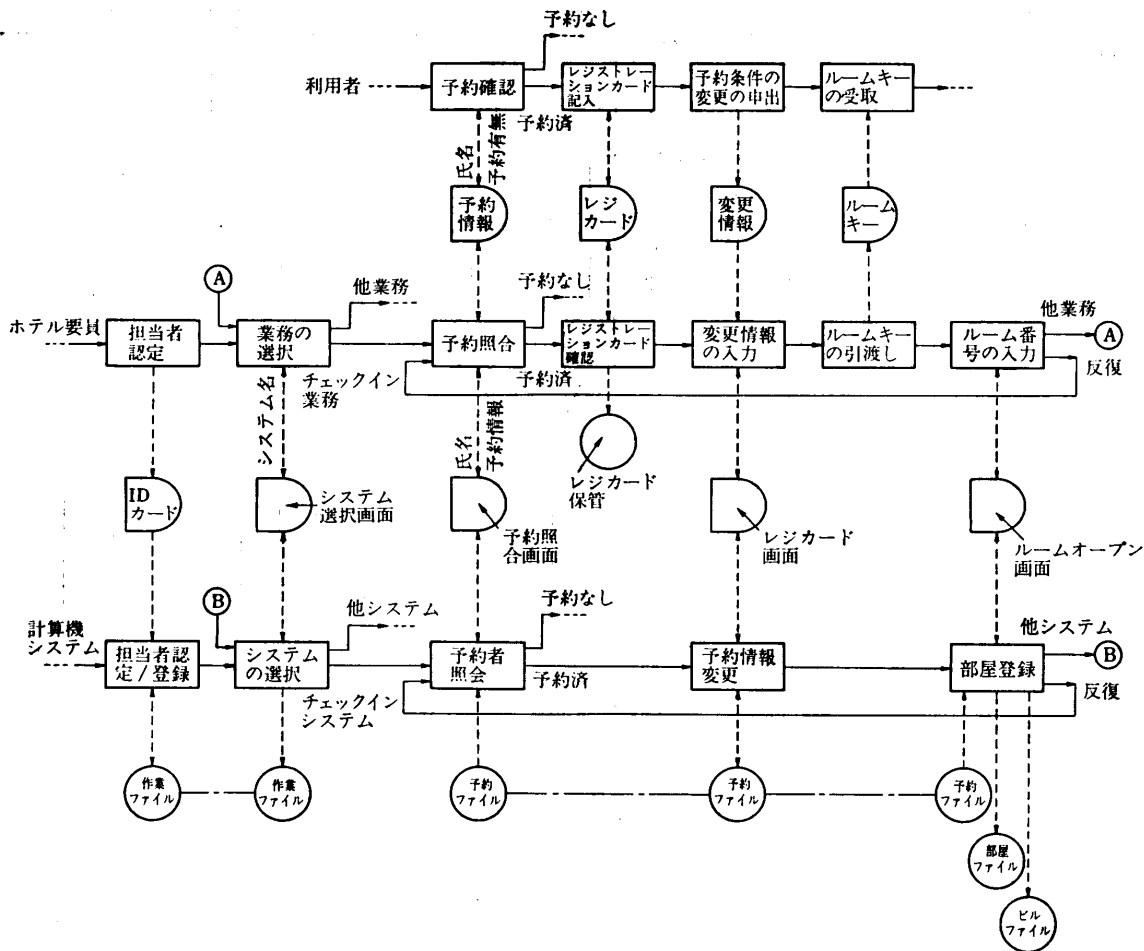


図7 チェックイン・システムの GROSS による記述  
Fig. 7 GROSS representation of checkin system.

グ・マシンや管理装置の故障のほかに、フロント係が部屋ファイルへの利用者の登録を怠っているのが原因かもしれないと推察できる。このような類推は、現在広く利用されている人間系の記述（事務工程分析図など）と機械系の記述（ランチャートや流れ図など）を別々に表現する方式では不可能であり、両者を併記することによって初めて可能となる。

(2) チェックイン・システムの詳細

図7は図6におけるチェックインに関連するタスクを、システムの選択動作（左端の2列）とともに示したものである。GROSSはプログラムの内容のみならず、その選択手順および背景についても記述できる。第3列以降はチェックインに関する詳細が示されている。図7には省略されているが、“予約なし”の分岐についても、その時点での部屋の利用率を予約ファイルから算出して諾否を決めるなど、予測可能な処理に

ついては記述できる。ただし、判断がフロントのその時点での裁量に委ねられているような分岐に対しては、明確に書くことはできない。なお、ルームキーは入退のチェック、ホテル内での本人の認証などの情報システムの一翼を荷うので明示されている。

(3) アクティビティによる記述

図7は点線のアークに複数の情報の名称が記載されている、などにより、さらに詳細なグラフを作成できる。図8は図7の予約照合に関連する部分をアクティビティによって記述したものである。図8では人間系、計算機系ともアクティビティによって克明に記述されており、処理ノードの実行順序も明白である。

図6→図7→図8は情報システムの階層構造を表現しており、GROSSにより構造的な分析、設計、プログラミング、運営が可能となることがわかる。各水準のグラフにおいて処理ノードに出入りするアークの関

係は対応が必ずしも明確でないが、それぞれのアークに同一の記号をもった結合子を付加することによって明示できる。

なお、この例では示さなかったが、裏方で実施される一括処理の表現能力も十分もっている。この場合、テキスト部分が少なく、ファイルとのやりとりを中心にしたグラフとなる。

## 5. 考 察

グラフ記述の“理解しやすさ”などの特徴点に関する、実験などを通じた客観的な評価は困難とされている。なぜなら、使用者の主観をまったく排除して評価を収集することがむずかしいためである。そこで、一般的には基本概念や方式上の特徴点の比較によって評価がなされている<sup>2)</sup>。GROSS についても、これに従い、以下に示した長所および短所を得た。一方、計算機やシステムに関する非専門家に実務を GROSS で記述してもらい、評価を聴取した。その結果、特徴点のうち (a), (b), (c), (d), および (f) の長所, (a) および (c) の短所が指摘された。したがって、以下に示す GROSS の評価は、まったく客観的なものであるとは主張できないものの、実務家による試使用結果に基づく、一応の裏付けがあるものである。

### (1) GROSS の長所

長所として次の事項があげられる。

(a) 人間系と機械系とから構成されている情報システム全体を、両者の関連をも含め統一された概念と記号系で表すことができる。このことを通じて、それぞれの系を、相手方の理解に立脚して考察することができ、システムの構築、運用および維持における配慮が行き届く。

(b) 概念と記号系は人間による現実のオフィス情報システムに求めた。また、理解の容易な人間系との対応で、それまで経験の少ない機械系のグラフ記述の学習をすすめることができるので、実務家によるシステム構築、運用および維持への道を拓く。

(c) 従来から各種のグラフの併用によって記述されていた内容は、GROSS によってシステムの全過程を通じて統一的に表現されるので、利用者はグラフの記述法を使い分ける必要がない。

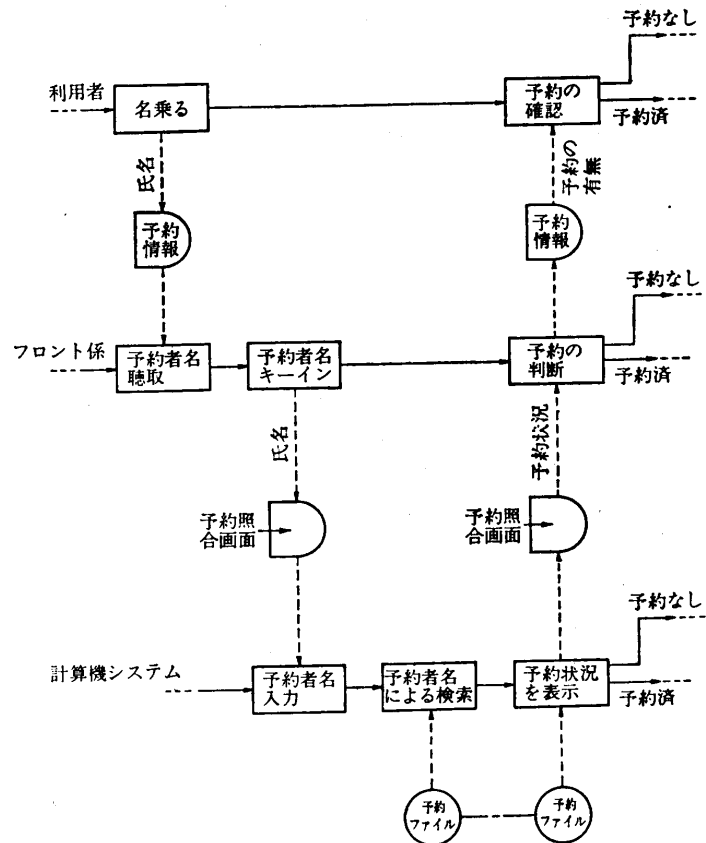


図 8 アクティビティによる記述例  
Fig. 8 Representation example using activities.

(d) 情報システムの階層構造を表現できるので構造化されたシステム構築が可能となる。また、不必要な詳細を隠蔽できるので、利用者グループごとに必要な情報のみを提供して理解しやすくすることも容易である。

(e) 情報の遷移を明確にすることができ、タスクの相互関連がより明白になる。

(f) GROSS のグラフは理解しやすいので説明書の補助手段としても使うことができる。

### (2) GROSS の限界

(a) あいまいさを避け得ない自然言語による説明文を導入しているのでグラフ自体の形式化の厳密性は追求できない。

(b) 情報システム（ただし予測困難な処理を除く）に使途が限定されており汎用性はない。

(c) 階層構造の表現においてアークの対応関係を明確にするために、結合子を導入すると簡潔さを損う。

GROSS は実務家の参加の下に情報システムの諸活動を実施するために研究されたものであるから、(a)

および(b)はそれと両立しにくい。また、完全性や正当性の追求は記述言語の傾分としている。汎用性の点については、情報システムの表現に徹しているのではあるが、それだけでなく、CAIなどのように対話を主体としたシステムには十分応用することができる。

### (3) これまでの方式との比較

これまでの研究では、ここに報告したような人間による情報システムに概念と記述法を求め、人間系と機械系を統一的に取り扱うグラフ記述方式の提案は見当たらない。GROSSで提案した方式は、アクティビティ自体およびアクティビティと情報の遷移、タスクの階層構造、コミュニケーションおよび起動などの表現力において、これまで提案されてきた各種方式の意図するところは情報システムに必要な事項に関して実現している。

現在広く応用されている方式は用途に応じてグラフ記述を使い分ける方法であり、すでに記述した欠点を有する。一方、わが国において広く用いられている方式ではないが、いくつかの方式との比較をしておく。

BDL<sup>7)</sup>はGROSSと同じように概念をオフィスに求め実務家による理解も比較的容易で支援系も考慮されているが、意図するところはプログラミングのための手法であり応用範囲が限られている。SADT<sup>8)</sup>は応用範囲は広いとされているが、データと作業の関連が陰にしか記述できないので実務家になじみにくく、概念も複雑で駆使するまでに相当な訓練が必要である<sup>9)</sup>。SREM<sup>6)</sup>を構成するR-Netは形式性に優れ、支援系をもつ有効なグラフであるが、特殊な実時間向きの記述法であり、Petri-Net<sup>5)</sup>やICN<sup>4)</sup>もシステムを理論的に取り扱うための方式であり、ともにオフィスの実務家が駆使しにくい面をもっている。

以上に記述したとおり、GROSSは人間系と機械系を統一的に取り扱う実用的な記述法として独得である。

## 6. むすび

オフィス情報システムの構築、運用および維持の過程で使う、グラフ記述GROSSを提案した。GROSSは、情報システムを捉える概念を明示し、その概念に立脚して展開されたもので、システムを構成する人間系および機械系、両者の関係を統一された方法で表現できる。GROSSでは、概念と記述法を人間による情報システムに求めており、実務家に理解しやすく、情報システムの諸活動への実務家の参加へ道を拓くもの

である。ここに述べたGROSSの理解のしやすさなどの長所は、実務家による実験を通じて確かめられたが、今後さらに、現実のシステムへの大規模な適用と、それを通じた評価と改良が必要である。また、GROSSの入力、蓄積および出力などの機械処理の研究、およびGROSSと事務用簡易言語<sup>10)~12)</sup>のような高位の記述言語とを結びつけたシステム開発ツールの研究を進めており、これらについては稿を改めたい。

謝辞 終りにあたり、本研究の機会を与えられ終始ご指導とご討論を賜わった、慶応義塾大学理工学部、浦昭二教授、同大駒誠一助教授、同永田守男講師、いつもご指導を賜わる三菱電機(株)電子システム部、河野隆一郎長および同社情報電子研究所、溝口徹夫博士に深甚の謝意を表す。

## 参 考 文 献

- 1) Ellis, C. A. et al.: Office Modeling, Workshop Report, in Naffah, N. (ed.), *Integrated Office Systems—Burotics*, pp. 269-275, North-Holland, Amsterdam (1980).
- 2) 佐藤匠正: プログラミング用ドキュメンテーション, 情報処理, Vol. 22, No. 5, pp. 383-389 (1981).
- 3) Ellis, C. A. and Nutt, G. J.: Office Information Systems and Computer Science, *Comput. Surv.*, Vol. 12, No. 1, pp. 27-60 (1980).
- 4) Ellis, C. A. et al.: Office Streamlining, in Naffah, N. (ed.), *Integrated Office Systems*, pp. 111-125, North-Holland, Amsterdam (1980).
- 5) Peterson, J. L.: Petri Nets, *Comput. Surv.*, Vol. 9, No. 3, pp. 223-252 (1977).
- 6) Alford, M. W. and Burns, I. F.: R-Nets: A Graph Model for Real Time Software Requirements, MRI Conf. Softw. Eng., N. Y. (1976).
- 7) Hammer, M. et al.: A Very High Level Programming Language for Data Processing Applications, *Comm. ACM*, Vol. 20, No. 11, pp. 832-840 (1977).
- 8) Ross, D. T.: Structured Analysis (SA): A Language for Communicating Ideas, *IEEE Trans. Softw. Eng.*, Vol. SE-3, No. 1, pp. 16-34 (1977).
- 9) 北川博之, 国井利泰: ソフトウェア仕様技術の比較的评价, ソフトウェア工学, pp. 120-151, 共立出版, 東京 (1978).
- 10) 魚田勝臣他: インライン処理指向のプログラミング言語, 情報処理学会論文誌, Vol. 21, No. 6, pp. 433-441 (1980).
- 11) 魚田勝臣: 事務処理用簡易言語, 情報処理, Vol. 22, No. 6, pp. 545-549 (1981).
- 12) Kondow, S. et al.: Screen Oriented Design and Implementation of Information System, Poster Session, Proc. of 6th International Conf. on Software Engineering, pp. 85-86 (1982).

(昭和58年5月12日受付)

(昭和58年9月13日採録)