

3W-7

時間制御を取入れた  
システム論理モデル作成法

橋本恵二

富士通株式会社 ソフトウェア開発企画本部

1. 序

システムやソフトウェアを開発するに先立って行われる要求分析、要求定義の重要性が認識されて久しい。これまで要求を記述するための形式的なモデルと技法が種々提案されている<sup>1)</sup>。

特にオフィスシステムの分析モデルとして、データフロー図 (DFD) がよく知られている<sup>2) 3)</sup>。DFDによる分析はまず、オフィスにおける帳票の動きや作業手順などをデータの流れ、変換活動、及びファイルという概念を用いて論理的に記述する。その上に計算機による処理部分を切り出し、要求仕様として用いる。

しかし、上の方法ではどのようにDFDを記述することが論理的であるかが不明確である。またその記述の自由度が大きいために、書かれたものを検証することが難しいという問題がある。この点を改善するために、システムの実現手段を“完全”と仮定してDFDを記述するという提案がある<sup>4)</sup>。

本論文では、このような研究の延長として、DFDにタイミングを取り入れたモデルと分析方法を提案する。事象とタイミングを手掛りに業務を分析すると、論理的なDFDの形態を規定することができる。それによって従来のDFDよりも記述の自由度が制限され、検証の可能性がより高められる。

次に、実際の大規模なシステムを分析する機械的な方法を提示する。そこでは、対象システムを複数のローカルシステムに分割し、段階的な論理化を積み上げることによってシステム全体のモデルに至る方法を示す。

2. オフィス情報処理のモデル

実際のオフィス業務で行われる情報処理活動は、多くの人間が関与する複雑なプロセスから成っている。複雑な例外処理や2重3重のチェック機構が組み込まれているのが普通である。

システムの論理化を行う目的は、このような複雑なシステムから、単純で整合性のある構造を見出すことにある。現実のシステムは、個々の活動が人や機械などの具体的な“もの”によって担われているため、その技術

的不完全からくる要素がシステムに付加され、それによって単純な構造がカムフラージュされている。そのような物理的性質を排除する必要がある。

システムの論理的な記述を得るために、次の2つの仮定をDFDに付加する：

① システムの実現手段の完全性

考えているシステム内部の機能を完全な計算機システムが担うものと仮定してシステムを記述する。具体的には、処理時間を無限小、ファイル容量を無限大とし、システム内部での一切の処理は誤りが無いものとする。

② イベント駆動型

オフィスの情報処理は、多数の人間による非同期並行処理系と考えられる。このことを陽に表現するには、事象とタイミングを手掛りにモデルを作成することは有効である。

このような仮定を行うと、システムの一般的な形態は図1のように「環境-タスク-ファイル」という3層構造に限定される。

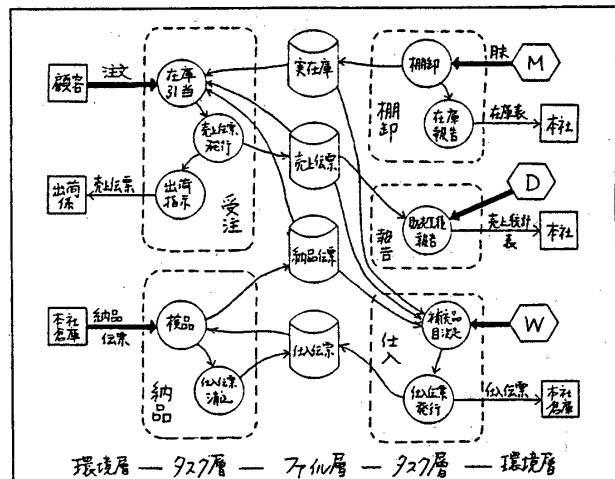


図1. オフィス情報処理の「環境-タスク-ファイル」構造

→は事象、→はデータフローを表す。○は時計であり、定期事象の発生周期と略号で表わしている。例えばD=daily, W=weeklyなどである。破線で囲まれたタスクは、固有の事象と起動タイミングを持つ。

システムを起動する事象は環境からのデータ入力（外部事象）と時間制御を表わす事象（定期事象）から成る。タスクとは、1個の事象によって起動される一連の活動をまとめたものである。事象を最初に閏知する活動、その後引き続いて実行することが論理的に可能な活動をひとまとめにしてタスクと呼ぶ。つまりタスクは1つの事象を閏知すれば、必要な応答を瞬時に完了し、再び次の事象を待つ状態に戻る。また、タスクはそれぞれ環境から規定される固有のタイミングを持つために、それらの間の情報のやりとりは、必然的に、データを保管するファイルを通じて行われる。

このように“理想的”な極限でモデルを作成すると、論理性の高いシステムのモデルが得られる。現行システムに付加されている処理誤りに備えた活動などは、環境からの要請とは無関係なため、今のモデルを作成するうちに自然とモデルから除去することができる。

### 3. システムの分析法

現実の業務を分析する場合、大きなシステム全体を理解しなければならないような方法では、実際に分析を行うことが不可能になる。特に複数の業務にまたがるシステムを考えると、その全体を詳細に把握できる人間はいないのが普通である。このような場合にも耐えられる分析法を考える必要がある。

ここでは、対象システムを既存の組織や場所によって小さなローカルシステムに分割し、それぞれ独立に論理化を行うことによってシステム全体の論理モデルに至る方法を提案する。

#### ①ローカルシステムの論理化

ローカルシステムは、対象システムにとっての環境以外に、他のローカルシステムとも相互作用を行う。ここでの分析では、他のローカルシステムを内部環境とみなして分析する（図2）。

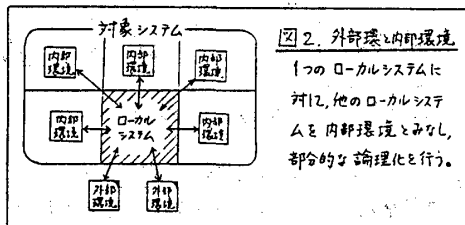


図2. 外部環境と内部環境  
一つのローカルシステムに対し、他のローカルシステムを内部環境とみなし、部分的な論理化を行う。

分析を行うには、まず事象を発見し、それによって起動される活動の連鎖反応をタスクという形にまとめる。次にタスクに属さない活動をモデルから除去する。タスクに属さない活動は、処理の誤りに備えた活動や冗長なデータを作成する物理的な活動であるからである。

#### ②ローカルシステムの結合とタスクの縮約

これまで設定されていたタスクは、内部環境からの制約で小さく設定されている。この制約が取り除かれると

事象による連鎖反応がより広範囲に拡大され、それまで別に設定されていたタスクがより少数のタスクに縮約される。また、ローカルな視点では必要であった活動が、よりグローバルな視点からは物理的な活動であることが発見され、消滅するタスクもある。図3に、タスクの縮約の典型的な例を示す。縮約の形態にはいくつかのパターンがあり、筆者の経験ではほとんど機械的なモデルの操作で行うことができる。

この方法によると、原理的にはどんな大きなシステムの分析も可能である。

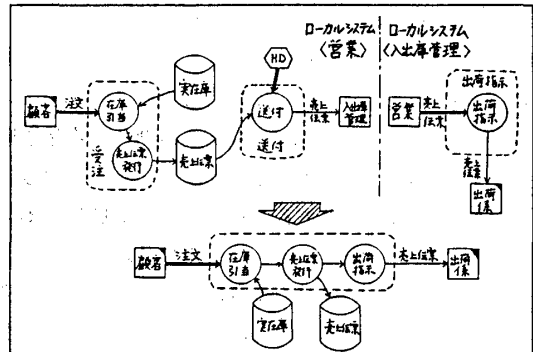


図3. タスクの縮約

ローカルシステム“営業”の“送付”というタスクは、入出庫管理という内部環境からの要請で成立していた。両者と結合させて制約が排除されると、半日毎の伝票送付は物理的な活動であることが分かる。これにより、タスクは縮約される。

### 4. 結語

オフィスシステムをモデル化するのに、システム内部の実現手段の完全性とタイミングを手掛りに分析を行うと、従来のDFDに比べて論理性の概念が明確になる。また記述の自由度が強く制限されるために、一定品質の分析が誰にでも行える。さらに、ここでのモデルは実時間システムの実現モデルとしても、我々のイメージに近いものである。

大きなシステムの論理化をローカルな論理化の積み上げによって、かなり機械的に行える点も、このモデルの特徴になっている。これは環境から規定されるタスクという明確な概念を用いていることによる。

計算機システムに対する要求仕様もこのモデルを用いて記述することができる。その場合、タスク毎に計算機処理部分を分離し、人間系を改めて環境とみなすことによってモデルの再編成を行う。

#### 参考文献

- 1) 例えば、野木兼六：要求定義技術の動向、情報処理、Vol. 20, No. 6, 487-494 (1979)；要求定義技術の最近の動向、情報処理、Vol. 27, No. 1, 21-30 (1986)。
- 2) DeMarco, T. : Structured Analysis and System Specification, Prentice-Hall, 1979.
- 3) Gane, C. and Sarson, T. : Structured Systems Analysis: Tools and Techniques, Prentice-Hall, 1979.
- 4) McMenamine, S. M. and Palmer, J. F. : Essential Systems Analysis, Yourdon Press, 1984.