# 7H－1 図式的プロセス記述によるアプリケーション自動/対話実行支援システムの設計*

安間 その美　上田 賀一†
茨城大学‡

## 1 はじめに

最近では、コンピュータによる作業が増加し、作業内容も複雑化している。そのような中で、作業者が自分自身で作業内容を考えたり、作業内容を記憶していたりするのは大変困難なことである。そこで、作業者の負担を軽減し、作業の効率をあげるために、作業指針を示すプロセスを形式的に記述し、この記述に従ってプロセスを実行する試みがなされている [1][2]。

本研究では、アプリケーションの実行順序をプロセスとみなし、オブジェクト指向のクラスとインスタンスの概念を導入して、図式的なプロセス記述をすることで、プロセスの自動実行および対話実行を可能とする支援システムの構築を目指している。

本支援システムにより、アプリケーション設計者は、図式的なプロセス記述が可能になり、プロセスの理解が容易になる。従って、効率的なプロセスプログラミングが行える。さらに、クラスレベル、または、インスタンスレベルでプロセスを記述することにより、ユーザは対話的または自動的にプロセスを実行できる。

本稿では、本システムが取り入れるべき基本方針と設計について報告する。

## 2 プロセスの記述方法

本システムで扱うプロセスは、データフローに基づいてアプリケーションの実行順序を表すものである。

そこで、本システムでは、プロセスをノードとエッジの有向グラフで表現する。ノードには、永続的クラスデータ，一時的クラスデータ，クラスアプリケーション，インスタンスアプリケーション，OR分割，OR結合，AND分割，AND結合，条件分岐の9つがある。また、エッジはデータフローを表している。ただし、ここで扱うデータフローはファイル単位のものでなければならないが、データは、プロセス記述時にはインスタンスがないことが多いので、クラスで記述する。

記述されたプロセスにおいて、データとアプリケーションのノードの入出力のフローは一つでなければならない。ただし、複数の場合は分割ノードや結合ノードを用いて記述する。条件によってフロー先が異なる場合は、条件分岐のノードを用いて記述する。

また、クラスアプリケーションを用いてプロセスを記述すれば、クラスレベルの記述となり、インスタンスアプリケーションを用いて記述すれば、インスタンスレベルの記述となる。

## 3 情報の管理方式

情報の管理を容易にするために、プロセスの情報を要素とダイアグラムに分けて管理する。

要素とは、クラスデータ，インスタンスデータ，クラスアプリケーション，インスタンスアプリケーション，条件分岐のようなプロセスの構成要素のことである。要素に関しては、要素名やファイル名やデータ型などの情報を管理する。

ダイアグラムに関しては、ダイアグラム名やノードの連結情報やノードの位置などの情報を管理する。

また、記述を実行するために、ダイアグラムのノードの情報から、ノードと要素を結び付けておく。さらに、要素の情報から、クラスとインスタンスを結び付け、インスタンスと実際のファイルを結び付けておく(図1参照)。

## 4 プロセスの実行方式

本システムのプロセスには、インスタンスレベルのものとクラスレベルのものがある。

インスタンスレベルのプロセスにおける実行は、アプリケーションの実体が決まっていて、そのまま実行できるので"自動方式"となる。これにより、ユーザは入力データを投入するだけでプロセスを実行できる。

一方、一つのクラスに対して複数のインスタンスが存在する場合、クラスレベルのプロセスにおける実行は、ユーザが作業に適したインスタンスをその都度選んでいく"対話方式"の実行となる。これにより、インスタンスごとにプロセスを記述する必要がなく、プロ

*Design of application automated/interactive execution support system based on graphical process description
†Sonomi Anma, Yoshikazu Ueda
‡Ibaraki University

セスの設計・管理が容易になる。また、クラスレベルのプロセスであっても、インスタンスを指定しておけばその部分は"自動方式"となる。

さらに、一つのプロセスにおいて、クラスとインスタンスを混在させて記述することもできるので、より柔軟にプロセスを記述し実行できる。

## 5 記述・実行支援システム

本システムは以下の5つ主機能から構成される(図2参照)。

- ダイアグラムエディタ

  ユーザがプロセスを記述する際に、それらを図式的に記述・編集するためのものである。また、図式表現をテキスト表現に変換する機構を備えている。

- 要素マネージャ

  要素に関する情報を管理し、クラスとインスタンス、または、インスタンスと実際のファイルを結び付ける。また、エディタで編集が行われた時、整合性のチェックをする。

- ダイアグラムマネージャ

  ダイアグラムに関する情報を管理し、ノードと要素を結び付ける。また、エディタで編集が行われた時、整合性のチェックをする。

- 実行マネージャ

  プロセスの実行を管理し、データの投入によってプロセスを起動する。また、実行中のプロセスに関する情報が変更されたり、ユーザがプロセスの中断を要求してきたら、プロセスを中止する。さらに、プロセスの実行をシステムの子プロセスとすることで対話中にも処理が行える。

- スーパバイザ

  すべての機能を管理する。また、機能間のデータのやり取りは、スーパバイザを通して行われる。さらに、システム終了時に実行中のプロセスが存在し、ユーザの了解を得たら、プロセスを中断する。

## 6 まとめ

本稿では、クラスとインスタンスの概念を用いたプロセスの記述によるアプリケーションの自動/対話実行支援システムの基本方式と設計について述べた。本研究では、現在、支援システムの試作を行っている。

また、要素(またはファイル)間の関係や制約の記述・管理による新しい関係およびプロセスの導出についても研究を進めている。

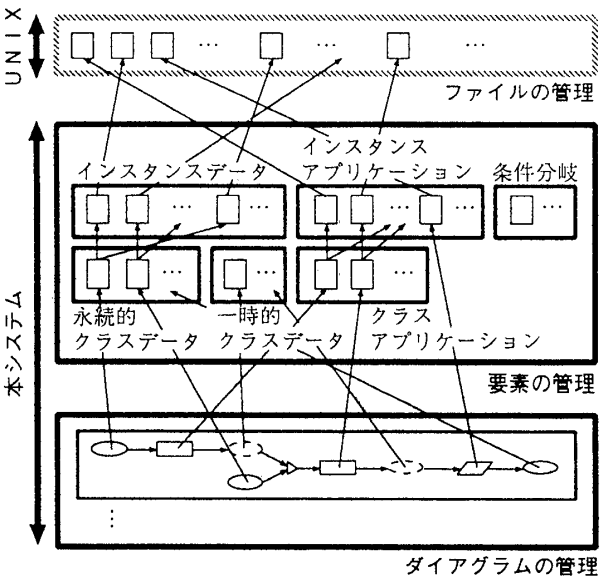


図1 : 情報管理の基本構造

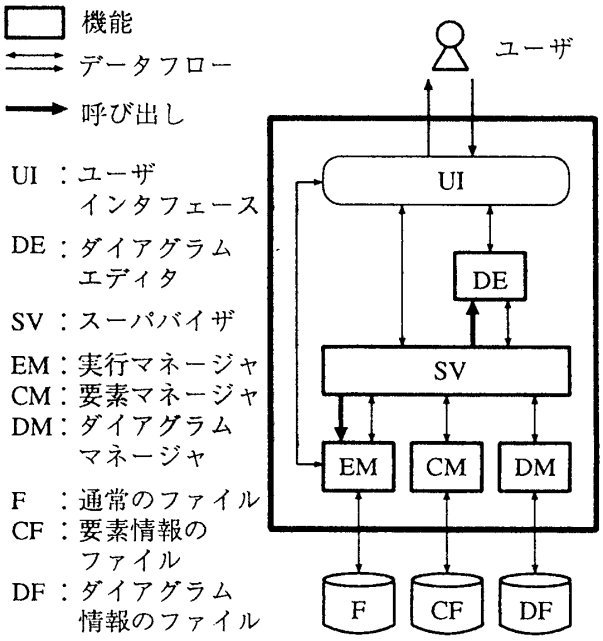


図2 : システムの基本構成

## 参考文献

[1] 荻原, 井上, 鳥居, 「図式表現を用いたソフトウェア構成・実行システムの試作」, 電子情報通信学会論文誌, Vol.J76-D-I, No.6, pp.324-326, 1993

[2] T.Shepard, S.Sibbald, C.Wortley, 「A Visual Software Language」, CACM, Vol.35, No.4, pp.37-44, 1992