リアルタイム共同情報処理支援オフィスシステムにおけるアプリケーションプログラム実行制御方式

7H-5

中山 良幸　森 賢二郎　中村 史朗　山光 忠
㈱日立製作所システム開発研究所　同ソフトウェア工場

1．はじめに

従来、オフィスでは、文書作成などの個々の作業の電子化が実現されてきたが、それらの連携という点で不十分であった。我々は、複数の人間が関与する打合せや会議などのリアルタイム共同情報処理の支援を目標とするオフィスシステムWATCH／RT (Window-Associated Telecommunication Handler/Realtime) を開発している[1]。WATCH／RTの特徴は、既存のマルチメディア環境の上で、個人作業から多者が参加する共同作業までを連続的に同時に支援できることである[2]。本論文では、WATCH／RT上で、実際の作業の場を提供するアプリケーションプログラム（AP）を作動させる方式について述べる。

2．WATCH／RTの概要

2.1 システム構成

WATCH／RTは、ワークステーションと電話を利用して、人間同士のデータと音声の交換を支援する。ワークステーションは当社の2050／32であり、UNIX*ベースのOS (Operating System) を持つ。通信ネットワークとして、ISDN (Integrated Services Digital Network) 及びLAN (Local Area Network) を利用でき、OSI (Open Systems Interconnection) 応用層の機能を参考にしたプロトコルを使用している。LANを用いる場合には、RS-232Cインタフェースで電話を制御している。

2.2 ソフトウェア構成

図1に、各参加者を表現する、ワークステーション上のWATCH／RTプログラム群の構成を示す。

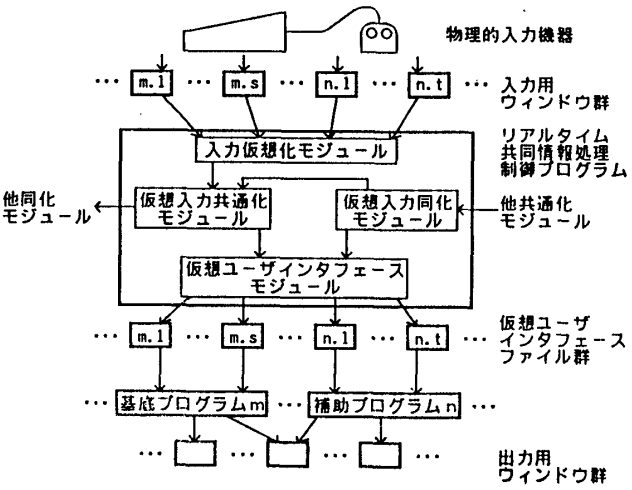

図1 WATCH／RTのソフトウェア構造

WATCH／RTは、各参加者の共同処理制御プログラムを論理的通信路で接続することにより、共同情報処理環境を構築する。制御プログラムのもとで、共同作業の実体となる既存プログラムなどの基底プログラム、及び、指示棒や手書きなど共同作業を円滑にする手段を与える補助プログラムを動かす。WATCH／RTは、分散型で、共同情報処理の基盤となる汎用のシステムである。

2.3 モジュール構成

制御プログラムは、4つのモジュールから構成される。入力仮想化モジュールは、各参加者が入力機器を用いて入力するデータを、全参加者がアクセスできる共同作業の場に送り出す。仮想入力共通化／同化モジュールの2つは、通信ネットワークを利用して共同作業の場を形成する。仮想ユーザインタフェースモジュールは、基底／補助プログラムに対し、共同作業の場へのアクセス手段を提供する。

3．共同処理制御プログラムによるAP実行方式

3.1 仮想ユーザインタフェースと基底プログラム

従来のAPは、利用者が入力機器から投入するデータを直接受け取って処理していた。ウィンドウシステムも、利用者の操作が介在しない入力は考慮していない。WATCH／RTは、ウィンドウシステムを通した入力を一旦入力仮想化モジュールが受け取り、共同作業を構成している各制御プログラムに分配する。種々の入力機器からのデータは、仮想ユーザインタフェースを通して基底プログラムに与えられる。仮想ユーザインタフェースは、ウィンドウシステムが提供する入出力制御と同等の機能を基底プログラムに提供するので、既存のソフトウェアを利用するための変更がほとんど不要である。WATCH／RTは、出力に関して、他の制御プログラムと無関係にローカルに動作するので、基底プログラムが使用するウィンドウに対して、入力先としてFIFOファイル（仮想ユーザインタフェースファイル）を、出力先としてウィンドウ自身を与える(この2つを併せて資源と呼ぶ)。

以下の説明に先だって、「基底プログラム」が意味するソフトウエアについて注意する。基底プログラムは、一般に、開始／終了部分と本体部分から構成される。本体部分は、与えられた資源を利用して制御プログラムから入力データを受け取る。それ以外の部分は開始／終了部分である。例えば、WATCH／RT対応に作成されていない既存プログラムを基底プログラムとして用いる場合、それ自身は本体部分であり、ステーション間連動を可能とする形で本体部分の起動処理などを行うプログラムが開始／終了部分である。本体部分が別プロセスの場合、仮想ユーザインタフェースモジュールの下で、実際に作動するのは本体部分の

Controlling Application Programs Running in an Office System for Realtime Cooperative Information Manipulation
Yoshiyuki NAKAYAMA[1], Kenjiroo MORI[1], Fumio NAKAMURA[1], and Tadashi YAMAMITSU[2]
[1]Systems Development Laboratory, Hitachi Ltd.　[2]Software Works, Hitachi Ltd.

*UNIXオペレーティングシステムはAT&T社ベル研究所が開発したソフトウェアであり、AT&T社がライセンスしている。

みであるが、以下では特に断わらない限り区別しない。

3.2 APによる資源要求

制御プログラムが、他の制御プログラムや基底プログラムに入力を伝えるには、基底プログラムが使っているウインドウや仮想ユーザインタフェースファイルの識別子及び状態などを知る必要がある。WATCH/RTの制御下で動作する基底プログラムは、使用する資源を、OSではなく、制御プログラムに対して要求する。この資源要求に対して制御プログラムは、OSの機能を用いて必要な資源を作成し、基底プログラムにその情報を通知する。起動方法や資源獲得方法のみを変更することにより、自然に既存APを基底プログラムとして動かすことができる。

3.3 AP/資源管理

制御プログラムは資源要求に応じて作成した資源を、次章で述べるプロトコルを利用して、基底プログラム対応に管理している。各資源は、それを利用している基底プログラムの識別子 apid と当該基底プログラムに関する資源の識別子 rscid の組で一意的に表現することができる。即ち、ある参加者が資源 (apid,rscid) から投入したデータは、他の参加者のもとにある対応する基底プログラムに対しても同一の資源 (apid,rscid) を通じて伝えられる。

4. AP実行制御プロトコル

ここでは、制御プログラムのもとで基底プログラムを適切に実行するための、両者間に規定されているプロトコルを説明する。資源要求は、本プロトコルに含まれる。プロトコルは、制御プログラムに基底プログラム実行状況を通知する役割も果たしている。各メッセージの送受信機能は関数の形で提供されているので、基底プログラムは、具体的手順を知る必要がなく、理解しやすい表現で記述できる。

(1) AP登録
基底プログラムとして動作するAPに関し、その名称等の情報を制御プログラムに通知する。制御プログラムは本AP情報を登録し、割り当てられた基底プログラム識別子 apid を返す。

(2) AP抹消
「AP登録」でなされた基底プログラムの登録を抹消し、対応する基底プログラム識別子 apid を解放する。

(3) 実行開始報告
基底プログラムの実行が開始したことを制御プログラムに通知する。

(4) 実行終了報告
基底プログラムの実行が終了したことを制御プログラムに通知する。

(5) 資源要求
制御プログラムに対して、資源の確保を要求する。制御プログラムは資源を確保し、その名称等の情報を資源識別子 rscid と共に返す。

(6) 資源返却
制御プログラムに対して、資源の返却を通知する。制御プログラムは通知された資源を破棄対応する資源識別子 rscid を解放する。

基底プログラムが apid と rscid を使えば、例えば、制御プログラム経由の相互交信などが可能である。

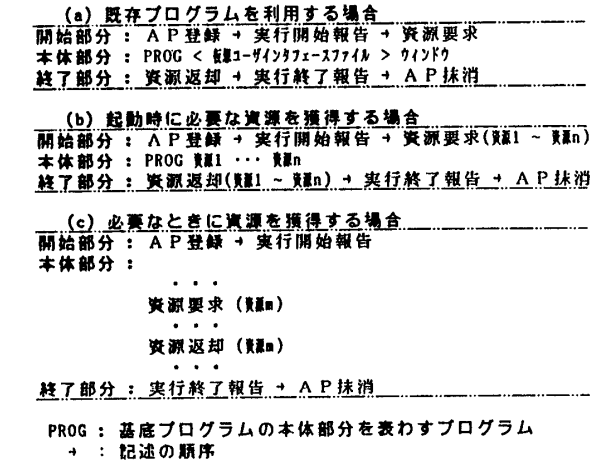

図2 基底プログラムとしてのAPの記述例

5. 基底プログラムの記述

本章では、4章で説明したプロトコルを利用して、既存のプログラムやWATCH/RT対応のプログラムを、基底プログラムとして動作させる方法について述べる。

図2は、基底プログラムの記述の概略を、開始/本体/終了部分に分けて示している。

5.1 標準入出力を利用した典型的プログラム

UNIXのもとでシェルから起動される既存のプログラムは、しばしば、標準入出力ファイルとしてウィンドウを受け取る。このようなプログラムをWATCH/RTの基底プログラム(の本体部分)として利用するには、入力データを制御プログラムから操作できるように、開始部分が資源を制御プログラムに要求して取得し、仮想ユーザインタフェースファイルを入力先、ウィンドウを出力先として指定する(図2(a))。

5.2 一括資源獲得型プログラム(WATCH/RT対応)

必要な資源を全て事前に制御プログラムに要求して確保し、これらを起動時に基底プログラムに引数として渡す。一度に資源を獲得してしまうので、迅速な資源の利用が可能である(図2(b))。

5.3 動的資源獲得型プログラム(WATCH/RT対応)

実行中に必要になったときに、基底プログラムが制御プログラムに資源を要求する。資源要求の負荷が掛るが、資源を有効に利用できる(図2(c))。

6. おわりに

共同情報処理において基底プログラムを実行する方式について、基底プログラムと制御プログラムの間のプロトコル及び基底プログラムの実際の記述を説明した。

参考文献

1) 中山 良幸、森 賢二郎: リアルタイム共同情報処理支援オフィスシステムのアーキテクチャの検討と実現例, 情処研報, Vol. 89, No. 5, 1989年1月19日.
2) 中山 良幸、森 賢二郎: 個人情報処理とリアルタイム共同情報処理を統合するオフィスシステム, 情処第38回全国大会, 1989年3月.