

シングルユーザアプリケーションプログラムの共有方式と そのMERMAIDへの実装

2E-7

久保 信也 阿部 豊子 前野 和俊 †丸山 修司 †今橋 泰則
 NEC C&C 研究所
 †NEC 技術情報システム開発

1 はじめに

近年、ローカルエリアネットワーク(LAN)やLAN同士が相互接続されたインターネット、あるいは専用・公衆回線を利用することによって相互に接続された計算機を利用する広域分散環境において、地理的に離れた場所にいるユーザ同士による協同作業を支援する環境(CSCW)が求められている。筆者らのグループでも、これまでに遠隔多者間でテキスト、手書き、イメージ、動画像、音声などのマルチメディアデータの共有・交換を実現するグループウェアのプラットフォーム、“MERMAID”[1]の研究・開発を行なってきた。さらに、既存のアプリケーションプログラム(AP)を共有することができるようになると、実行中のAPの出力(画面表示)を遠隔地にいる複数の協同作業ユーザが同時に得ることができるようになり、データのみを静的に共有している場合以上に密接なグループ協同作業が行なえるようになる。筆者らは、これまでに既存のAPの共有を実現するAP共有機構の実現方式の検討を行なってきた。本稿では、その実装方法の概要について述べる。

2 XウィンドウシステムとAPの共有

Xウィンドウシステム上では全てXプロトコルにもとづいて、APの入出力が行なわれる。そこで1つのAPからのリクエストを複数のXサーバに分配し、また複数のXサーバからのリプライ、エラー、イベントを要求元のクライアントに送信することによって、比較的容易にAP共有を実現することができる。

この様に1つのクライアントを複数のXサーバに接続させる方式によって、AP共有を実現しているシス

テムの1つにXTV[2]がある(Copyright ©1991 Old Dominion University. Copyright ©1991 University of North Carolina at Chapel Hill. X11R5のUser Contributed Programの1つ)。XTVは仮想サーバ方式によってAP共有を実現するシステムである(図1)。XTVプロセスがAP共有に参加する各ユーザのDISPLAY(Xサーバ)ごとに起動され、複数のXTV同士がXバケットの転送を行なうことによって、1つのAPを複数ユーザで共有できるようにしている。XTVは専用のシステムとしてXTV単独で利用され

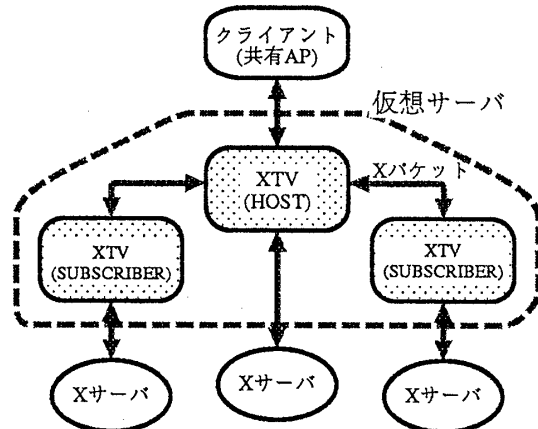


図1: XTVを利用したAP共有

るように設計・実装されているため、他のシステムから利用する、あるいは他の協同作業支援システムと協調させるような使い方はできない。

3 AP共有機構の設計

3.1 基本モデル

既存のAPを無改造で共有可能にするため、本AP共有機構もXTVと同様に、仮想サーバ方式を採用する。仮想サーバは、AP(Xウィンドウシステムのクライアントプログラム)に対して通常のXサーバと同様に振舞うため、共有実現のためにAPを変更する

An Approach of Single User Application Program Sharing and It's Implementation on MERMAID

Nobuya Kubo, Toyoko Abe, Kazutoshi Maeno, †Shuji Maruyama and †Yasunori Imahashi

C&C Research Laboratories, NEC Corp.

†NEC Scientific Information System Development, Ltd.

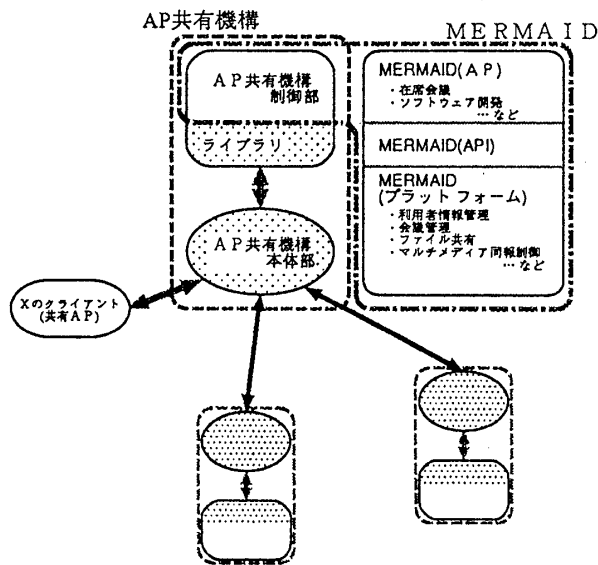


図 2: AP 共有機構のモデル

必要は全くない。AP 共有機構の実装方式について、MERMAID 自体にその機能を取り込んでしまうというアプローチも考えられるが、汎用性や拡張性を考慮し、MERMAID を利用していない時でも AP 共有を可能にするため、今回は MERMAID は AP 共有機構を制御し、共有は MERMAID の外のプロセスが行なう方式を提案した。つまり本 AP 共有機構は、X プロトコルの分配・転送などの基本機能を提供し、仮想サーバとして動作する AP 共有機構本体部と、ユーザインタフェースを提供し、本体部を制御する AP 共有機構制御部から構成されることになる (図 2)。図 2 では AP 共有機構制御部は MERMAID が提供しているが、他のシステムが提供する、あるいは共有機構の制御専用で作られたプログラムでも良い。

このアプローチを実現するためには、AP 共有機構本体部は制御部に対して自分をコントロールするための手段を提供しなければならない。そこでこの制御手段を API と定義し、ライブラリ関数として実現・提供することを決め、その設計、および実装を行なった。

3.2 ライブラリ関数の設計

まず最初に本体部が制御部に対して API として提供しなければならない機能を検討し、以下のように分類した。

- AP 共有機構本体部のプロセス自身の制御手段

AP 共有機構本体部 (制御部に対してはサーバとして動作する) プロセスの起動、終了を行なう

- AP 共有セッション参加者の管理の制御手段
AP 共有セッションに参加するために、本体部同士の接続の設定を行なう

- 共有 AP 管理の制御手段

共有する AP のプロセスの起動、および終了を行なう

AP の公開モード (AP を起動した人以外の参加者の画面にも、AP の出力を許す) \leftrightarrow 非公開モード (AP 起動者の画面にしか表示されない) の切替を行なう

- floor 管理の制御手段

データの一貫性保持、およびシステム (プロセス) の一貫性保持のために、同時には 1 ユーザしか入力が行なえないようにする

4 おわりに

今回、X ウィンドウシステム上の仮想サーバ方式による AP 共有システムをベースに、既存の AP をプログラムを無改造で共有する方法の検討を行ない、MERMAID、および他のシステムから制御することが可能な AP 共有機構のモデルを提案した。この機構の実現により、これまでスタンドアロン用に開発されてきた様々な AP を、変更することなく広域分散環境で協調的に動作させることが可能になり、CSCW の強力なツールを提供できることになる。さらに、今後は性能向上、セキュリティ強化、マルチメディアへの対応について検討を重ねていく必要がある。

参考文献

- [1] K. Watabe and S. Sakata et al. Distributed Desktop Conferencing System with Multi-user Multimedia Interface. *IEEE Journal on Selected Areas Communications*, Vol. 9(No. 4):pp. 531-539, 1991.
- [2] Hussein Abdel-Wahab and Mark Feit. XTV: A Framework for Sharing X Window Clients in Remote Synchronous Collaboration. In *Proc. of IEEE TriComm '91*, pages 159-167, April 1991.