

2G-7

共有ウィンドウのアーキテクチャに関する考察*

桑名栄二 小村誠一 柴崎雅史†

NTT ソフトウェア研究所‡

1 はじめに

ソフトの開発形態に目を向けると開発拠点の地方分散、ソフトの分割開発が行なわれている。またソフト開発作業自身に目を向けると、開発作業は多くのフェーズで複数の担当者によるリアルタイム的な協調作業(関係作業:例えば設計作業、レビュー作業、調整作業など)という性格を持つ。

このような、地域的に開発対象物、開発担当者、開発環境が分散し、しかもその作業自身は実時間で協調的に遂行される環境を支援する概念としてCSCWの共有ウィンドウがある。最近、共有ウィンドウシステムとして、種々のシステムが試作され、そのアーキテクチャ、ユーザインタフェース条件、システム構築方法などが提案されている[1,2,3,4,5]。しかし、ユーザ個別の環境と共有ウィンドウ環境の制御方式、共有ウィンドウとその上で動作するソフト開発支援ツールとのインタフェースなど、明らかになっていない点が多い。

本稿では、共有ウィンドウシステム(シングルユーザ用のツールを共有するウィンドウシステム)の試用から得た問題点をもとに、ユーザの個別環境を実現する共有ウィンドウのシステムアーキテクチャについて提案する。

2 共有ウィンドウシステム

本節では、最近のCSCWなどで発表されている共有ウィンドウシステムアーキテクチャ(以後アーキテクチャと省略する)、ユーザインタフェースモデル、システムの持つべき機能を簡単に説明する。

2.1 システムアーキテクチャ

Lauwers/Lantz[1]は共有ウィンドウシステムのアーキテクチャとして、ユーザインタフェース部分を担当するウィンドウシステムとそのウィンドウ上で動作するツールとの仲介部としてConference Agent(以後CAと呼ぶ)を示している。このCAはツールからの出力情報をmultiplexし、また各ユーザのウィンドウからの入力情報をdemultiplexすることで、共有ウィンドウを実現する[図-1]。実際のインプリメント時には、(1)Canonical centralized方式(以後集中方式と呼ぶ)と、(2)Canonical fully replicated方式(以後分散方式と呼ぶ)がある。(1)の方式は実現が楽であるが、(2)の方式に比べて性能や融通性において問題がある[1]。

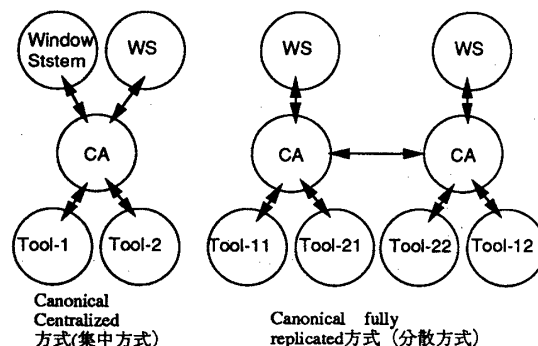
2.2 共有ウィンドウの備えるべき機能

共有ウィンドウの備えるべき機能としては、ツールとウィンドウ間のインタフェース管理機能、ユーザインタフェース機能がある[1,3,4]。以下に、その機能をまとめる。

- 共有環境管理機能(ツールとウィンドウシステム間の入出力情報のMultiplex/Demultiplex機能、個人環境と共有

環境の自由な切替え機能、自由な共有環境の開設/消滅機能、任意の時点での共有環境参加機能、フレキシビリティのあるツールアクセス権設定機能、会議の実際の会話とのベース同期機能、会話のログ機能)、

- 個人環境管理機能、
- 共有情報表示機能(WYSIWIS(What You See Is What I See)の原則、共有物の指示機能)、
- その他(会議参加者間の社会的な構造が反映されていること、簡単に利用できること)。



[図-1]共有ウィンドウアーキテクチャ

3 共有ウィンドウの試用

3.1 試用

筆者らはCanonical centralized方式のmc/mcx[5]を用いて、共有ウィンドウを試用した。試用に当たっては、mc環境でのemacsを利用した協調的なテキスト編集作業、mcx環境では、Xウィンドウ上のTeXのプレビューアを利用したドキュメント説明作業を採用した。

3.2 問題点

試用から以下の問題点が判明した。

- ユーザ環境の個別化:

個人環境管理に関係し、システムアーキテクチャに大きな影響を及ぼす問題として、個人環境のカスタマイジング問題がある。これは、Lantzらのモデル[1]では扱っていない問題である。通常のツールはシングルユーザ用に作られており(Collaboration Transparency)、ツール毎のユーザ個別部もシングルユーザ用になっている。例えば、emacs環境に日本語環境を付加し利用した場合、日本語辞書、日本語変換方式はツールにくくりつけとなり、共有ウィンドウを利用するユーザは他人の環境を利用することになる。この問題は集中方式、分散方式に無関係に存在する問題である。その他、ファイルアクセス権と共有ウィンドウに対するアクセス権の整合性問題などがある。

*Case Study of Shared Window System Architecture

†Eiji KUWANA, Seiichi KOMURA, Masashi SHIBASAKI

‡NTT Software Laboratories

● ツールとウィンドウ間の関係制御:

ユーザ毎に個別に起動したツールに付随するウィンドウを後に共有ウィンドウとして用いる場合、集中型の CA だと、ツールとウィンドウの関係を再設定しなければならない。

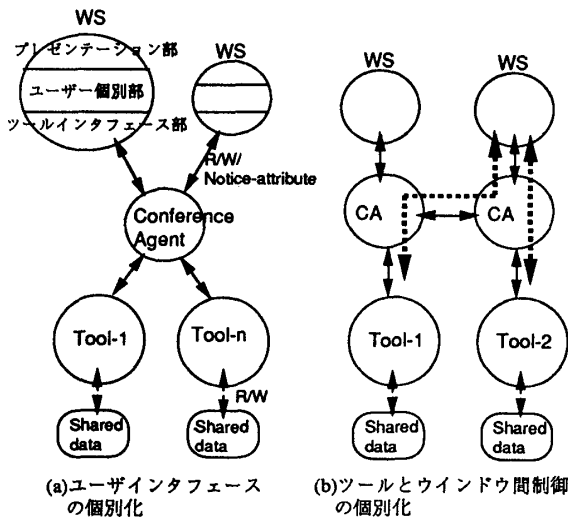
4 システムアーキテクチャの拡張

● ユーザ環境の個別化:

前節で示したユーザ個別部の問題点を解決するためには、ユーザ個別部をアプリケーションの層から分離し、ウィンドウシステムなどのユーザインタフェース部に位置付けるシステムアーキテクチャの基本的な拡張案が考えられる。

一方、UIMS (User Interface Management System) の世界では、IFIP / EG の Seeheim モデルがある。このモデルはユーザインタフェース構築のためのパラダイムとして、ユーザインタフェースをプレゼンテーション部、ダイアログ部、ツールインタフェース部の3層に分割するモデルである。

この Seeheim モデルと基本的な拡張案を、Lantz らの集中方式に適用すると図-2(a) のようになる。本モデルにおいては、ユーザインタフェース部の中を Seeheim モデルをもとに詳細化し、ダイアログ部でユーザ個別環境を実現する。このモデルの具体的な例として、共有ウィンドウ上でエディタを起動し、日本語辞書部/変換部をフロントエンドプロセッサ (例えば ATOK や X ウィンドウの kinput など) で行なう環境が当てはまる。しかし、実際のツールを本拡張モデルで動作させるには、ツールを本拡張モデルに従って改造する必要がある。



[図-2]共有ウィンドウアーキテクチャの拡張

● ツールとウィンドウ間の関係制御:

分散型の CA を用いると、ユーザ単位で起動した CA 間の通信を利用することにより、ウィンドウの共有化が図られる (図-2(b))。従って、ツールとウィンドウの関係を再設定する必要はなくなる。但し、CA では、CA 自身に対する命令とツールに対する命令を識別する機能が必要である。

● ファイルアクセス権と共有ウィンドウアクセス権の整合問題:

通常、共有ウィンドウに対するアクセス権 (Read / Write / Notice-attribute) は CA が担当している。またファイルに対するアクセス権は OS が管理しており、ツールからのファイル Open 時にチェックされる。共有ウィンドウにおける共有の意味づけにより、この共有ウィンドウとファイルに対するアクセス権が矛盾を起す場合が存在する。共有環境とは「最初にシステムを起動した人間とその分身からなる環境である」と考えた場合は、CA は共有環境 (ユーザインタフェース部) とツールに対するアクセス権を制御し、OS がツールとそのツールの対象とするファイルのアクセス権を管理するとみなすことができ、アクセス権に対する矛盾は発生しない。

一方、「異なるユーザが協調して作業する環境である」と考えた場合は、CA で管理するアクセス権とファイルに対するアクセス権の整合性をチェックする機構が必要となってくる。

5 おわりに

本稿では、次の二点を明らかにした。

1. シングルユーザツールを共有するタイプの共有ウィンドウシステムにおいては、例えば辞書などのユーザ個別環境の実現が必要であり、ユーザ個別環境を提供する Seeheim モデルを共有ウィンドウシステムのユーザインタフェース部に適用すれば解決すること。
2. ツールとウィンドウ間の関係制御、例えばユーザが個別に起動したツールとウィンドウを後に共有化する場合、Conference Agent (ウィンドウ共有化機構) を集中型ではなくユーザ個別の分散型にする必要があり、そうすることで関係制御のユーザ個別環境を提供できること。

[謝辞] 本研究にあたり多大な御助言、御指導して頂きました NTT ソフトウェア研究所 細谷僚一部長、神谷グループリーダー、ならびに実験に協力して頂いた坂本泰久社員、吉田善幸社員、吉田勝彦社員、有吉史子社員に記して感謝いたします。

参考文献

- [1] J.C.Lauwers, K.A.Lantz: *Collaboration Awareness in support of Collaborative Transparency: Requirements for The Next Generation of Shared Window Systems*, Proc. of ACM CHI'90 (1990)
- [2] T. Crowley et al.: *MMConf: An Infrastructure of building shared multimedia applications*, Proc. of CSCW'90 (1990)
- [3] M.Elwart-Keys, et al.: *User Interface Requirements For Face To Face Groupware*, Proc. of ACM CHI'90 (1990)
- [4] 荒川、柴崎: ウィンドウ共有管理システム XMULTI, 情報処理学会第 40 回全国大会 (1990)
- [5] 二上、坂本: 分散開発環境における共同作業用ツールの試作, 情報処理学会第 41 回全国大会 (1990)
- [6] M.Stefik et al.: *WISIWIS Revised: Early Experiences with Multiuser Interfaces*, ACM Trans. on Office Info. Systems, Vol.5, No.2(1987)
- [7] S.J.Gibbs: *LIZA: An Extensible Groupware Toolkit*, Proc. of ACM CHI'89 (1989)