分散オブジェクト環境を使ったアプリケーションの開発に関する考察

3R－2　－グラフィカル・ユーザ・インタフェース開発の観点から－

上田 世志　武脇 敏晃

(株) 東芝　情報・通信システム技術研究所

1　はじめに

オブジェクト指向技術を利用して、異機種分散ネットワーク環境におけるアプリケーション開発を促進する、分散オブジェクト環境を実現する製品が増えてきた。しかし、分散オブジェクト環境を用いたアプリケーションのオブジェクト指向開発についての事例は、まだ少ない。

筆者らは、グループ構成員のスケジュールをもとに会議の企画／設定を行なうスケジュール管理アプリケーションを、オブジェクト指向開発方法論 OMT 法 [1] に基づき分析／設計し [2]、コーディングを行なった。これを、クライアント／サーバ形態として実現するために分散オブジェクト環境を利用した。

本報告では、上記のアプリケーション (以後アプリと略す) の中でグラフィカル・ユーザ・インタフェース (GUI) 部分の開発に着目し、異機種分散環境での分散オブジェクト環境を用いた場合について、設計の過程で生じた要求と方針及び、コーディングの方法、考察について示す。

2　設計フェーズでの要求と方針

要求　開発のプラットフォームは、ワークステーション (WS) とパーソナルコンピュータ (PC) が混在するネットワーク環境であり、それぞれに分散オブジェクト環境を搭載している (図 1)。

GUI 開発環境については、オブジェクト指向開発との適合性を考慮し、次を要求として挙げた。

- GUI とアプリの相互間で双方向に呼び出せる方式のサポート
- アプリの開発言語との適合性
- 簡便にプロトタイピングできること

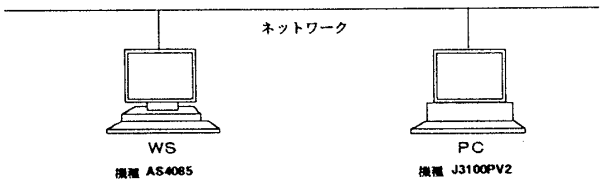

図 1: プラットフォームとネットワーク環境の構成

GUI とアプリ間の制御方法として、(1) GUI からアプリの操作を呼び出す GUI 主導の方式、(2) アプリから GUI の操作を呼び出すアプリ主導の方式、(3) 相互間で自由に呼び出す方式、の 3 つがある。

(3) は (1)、(2) と比べ、分析時の動的モデルからプログラムへの移行性が良く、またエラー処理の点でも優れている。

また、アプリ本体の開発言語として C++ を用いるため、C++ あるいは C との相性の良い GUI 開発環境が必要である。

プロトタイプを簡便に作成できることも必要である。これは、ユーザインタフェースを実際に動作させることにより、分析者がシステムに対する理解を深め、そのモデル化に対しフィードバックを得るためである。

方針　分散オブジェクト環境を使わずに、GUI のみやアプリ自体のみの単独テストによる動作確認のため、ツールとしての機能を実現する部分、分散オブジェクト環境に依存する部分、GUI の部分とし、システムをレイヤ化するアプローチをとった。図 2 で、これらをそれぞれ「ドメイン」、「インフラ依存部」、「GUI」として示す。

GUI 開発環境は、画面や操作の細部を決定するための指向錯誤を考慮し、初期はプロトタイプを簡便に行なうため Tcl/Tk [3] を用いる方針とした。

アプリケーションのモデルの概要が決定した段階で、各プラットフォームに対する GUI 開発の適合性から、PC 側は Visual C++ [1] を用いて、WS 側は Motif [2]

*A case study of development with distributed object environments – from the standpoint of a graphical user interface –, by Hiroshi UEDA and Toshiaki TAKEWAKI, Information & Communication Systems Laboratory, TOSHIBA CORPORATION

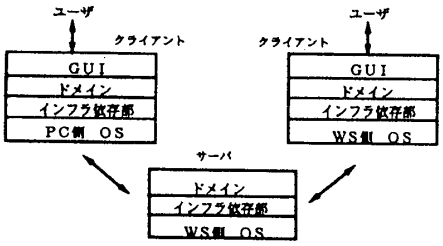

図 2: レイヤ化アプローチ

を用いて構築する方針にした。

WS側の分散オブジェクト環境は、分散オブジェクトの生成ごとにスレッドを起動する MT 方式である。このため複数スレッドを起動しないように、GUI 主導の手続き駆動的なシステムで開発することにより、MT セーフでない GUI ライブラリを利用可能とした。

3 コーディングフェーズについて

クライアントのプログラムには、Tcl/Tk の GUI 実行環境であるインタプリタを組み込んだ。さらに Tcl/Tk 用結合インタフェースとして、クライアントのインタフェースを実行するインタプリタのコマンドを組み込んだ (図 3)。

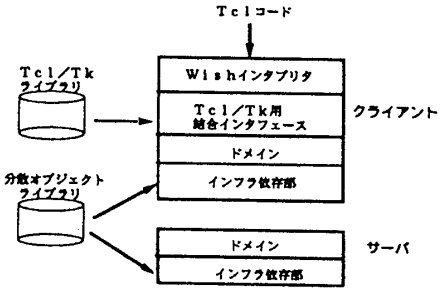

図 3: 各インタフェースレイヤの構成図

Visual C++, Motif を用いる場合は、それぞれ対応のライブラリを用いて GUI を作成し、 Tcl/Tk 用結合インタフェースを C 言語マクロによる変換を行なって、結合インタフェースを実現した。

4 考察

プロトタイピングの有効性　Tcl/Tk を用いた場合、GUI画面はプロトタイピング時に作成して、分析フェーズに利用し、制御の部分は後から作成して、アプリに利用できるため、開発効率の点で有効であった。

[1]Visual C++ は、米国 Microsoft Corporation の登録商標です。

[2]Motif は、米国 Open Software Foundation の登録商標です。

レイヤ化アプローチの有効性　レイヤ化アプローチは、各部分のみでコーディング／テストを進めることができ、GUI 開発環境依存部を明確に切り出すことも効果があった。

GUI 部分に関しては、分散オブジェクト環境への適合性については、今回のアプローチにおいては特に問題がなかったが、下記の 2 点を考慮すべきである。

MT セーフな GUI ライブラリの必要性　当初、GUI とアプリの相互間で双方向に呼び出せる方式を目指して分析をした。しかし、分散オブジェクト環境がマルチスレッドに対応している場合に、これを実現するためには、コーディングフェーズで MT セーフな GUI ライブラリが必要である。

統一的 GUI 開発環境の必要性　GUI の開発は、GUI 開発環境及び実行環境抜きにしては考えられないため、設計フェーズで GUI 環境と、前述のようなアプリの制御方法における実現方法等について要求に照らし合わせて、十分検討しておく必要がある。

分散オブジェクト環境は、マルチプラットフォームでオブジェクトを共有して実行できるが、 GUI 開発環境については特にマルチプラットフォーム性は考慮されていない。現状では GUI 開発環境の選定に当たって、労力が大きいことが分かった。

5 おわりに

本報告では、スケジュール管理アプリケーション開発におけるグラフィカル・ユーザ・インタフェース (GUI) 部分の開発に着目し、異機種分散環境での分散オブジェクト環境を用いた場合について、設計の過程で生じた要求と方針及び、コーディングの方法、考察について示した。またプロトタイピング及びレイヤ化アプローチの有効性、MT セーフな GUI ライブラリ及び統一的 GUI 開発環境の必要性について示した。

参考文献

[1] J. ランボー他著, 羽生田 栄一 監訳, オブジェクト指向方法論 OMT − モデル化と設計 − ,(株) トッパン, 1992 年

[2] 尾崎奈帆子, 「OMT 法における分析／設計フェーズに関する考察」, 情報処理学会第 51 回全国大会 4M-1

[3] 北川和裕, プログラミング言語 Tcl と Tk, pp.48-59, bit Vol.27 No.5, 1995