

分散型監視制御システム構築環境 (1)

1 N-1

— システム概要 —

小島泰三 野里貴仁 大崎雅代 北村操代 杉本明  
三菱電機株式会社 産業システム研究所

1 はじめに

筆者らは、電力系統や公共プラントなどの監視制御マンマシンシステムのソフトウェア開発の支援を目的とし、オブジェクト指向フレームワークを開発中である。本フレームワークは、C++ 言語によるクラスライブラリと、そのライブラリを利用したツール群から構成される。以下では、この開発中のフレームワークの概要を述べる。

2 監視制御向けオブジェクト指向フレームワーク

本フレームワークは、監視制御マンマシンシステム作成のためのアプリケーションフレームワークである。一般的なオブジェクト指向フレームワークは、汎用性の高いものが主流である。例えば、オブジェクト指向フレームワークの先駆けである Smalltalk-80 の MVC モデルは、GUI プログラム全般の作成に適した汎用的なフレームワークである。これに対し、アプリケーションフレームワークは、適用範囲を明確にすることで、より対象に即した機能を提供する。これにより、汎用フレームワークと比較すると、適用先は限定されるものの、個々の応用プログラムの開発では、より高度なソフトウェア再利用性が期待できる。従って、類似したプログラムが多数開発される大規模制御システムのソフトウェア開発に適している。

一般にオブジェクト指向フレームワークは、抽象的なクラス群と、そのオブジェクト間のプロトコルから構成される。本フレームワークの開発では、エンドユーザが柔軟にマンマシンシステムをカスタマイズできることを目的に、クラスの構成法とオブジェクト間の関係形態を設計した。本フレームワークの特長は以下のとおり。

- ミラーワールド方式と呼ぶ、監視制御対象や監視用機器の扱いに適したモデル化方式。

An object-oriented framework for distributed supervisory control systems, Taizo KOJIMA, Takahito NOZATO, Masayo OSAKI, Misayo KITAMURA, and Akira SUGIMOTO, Mitsubishi Electric Corp.

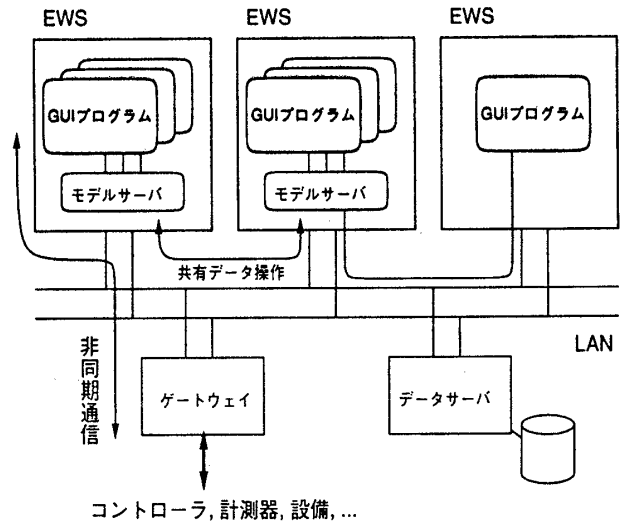


図 1: 監視制御システム構成

- 非同期メッセージ交換を基本としたオブジェクト間通信による実時間性の確保、及び、非同期メッセージ交換を用いた監視制御システム向け抽象クラス群。
- オンラインでの監視処理実行中においても、監視画面の編集が可能となる、フレームワークの基本機能として組み込まれた画面編集機能。
- GUI 部品の視覚的定義機能、及び、ユーザ入力に対するシステムの振舞いの視覚的定義機能。

なお、各項目については、別稿 [1] で述べる。

3 ソフトウェア構成

図 1 に本フレームワークによる監視制御システムの構成例を示す。図では、複数台の EWS が、ネットワーク上のゲートウェイを介して監視対象の設備と接続している。ソフトウェアは、サーバクライアント方式で構成されており、GUI プログラムは、モデルサーバと呼ばれるサーバを経由してデータ取得や制御操作を行う。耐故障性を高めるため、複数のモデルサーバを対称に配置してお

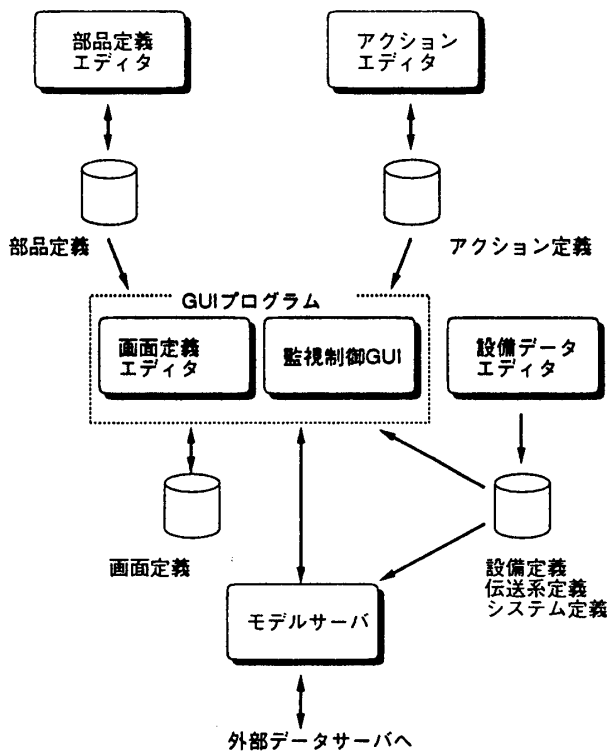


図 2: プロセス構成

り、共有データの操作においては、排他制御がモデルサーバ間で行われる。

図 2 に本フレームワークにおける代表的なプログラムとデータの関係を示す。図において、画面定義エディタは、個々の監視画面の定義を、設備データエディタは、監視対象の設備や伝送系の定義を編集するためのプログラムである。そして、部品定義エディタは、画面定義で用いられる GUI 部品の、また、アクションエディタは、ユーザ入力に対するシステムの振舞い(アクション)の視覚的な定義に用いる。

図 2 では、監視制御 GUI 以外に画面定義エディタがモデルサーバのクライアントとして位置付けられている。画面定義エディタは、単独でも動作するが、サーバと接続することにより、画面定義の編集時に、実際の設備の現在状態を示すオンラインデータを用いて、監視処理実行中の画面動作を確認することができる。なお、画面やアクションの編集機能は、本フレームワークの中の基本機能として組み込まれている。これにより、監視制御 GUI においても、監視処理と同時に監視画面の編集が可能である。

図 3 に、本フレームワークのソフトウェア階層図を示す。図のように、本フレームワークのベ-

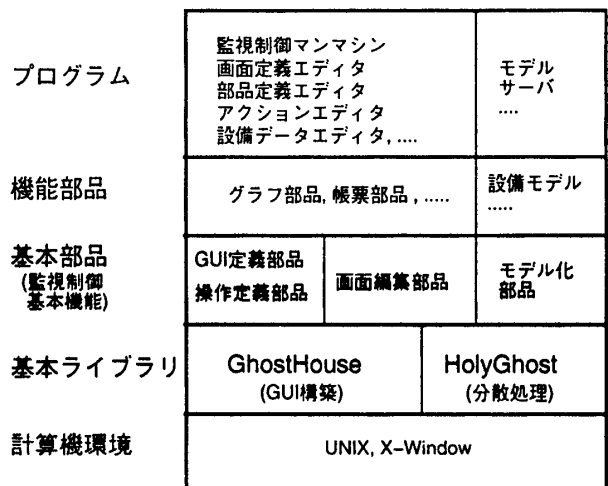


図 3: ソフトウェア階層

スには、2種類のクラスライブラリが位置しており、この上に、多種多様な部品群やツール群が実装されている。基本ライブラリのうちの1つは、従来から筆者らが開発していた GUI 構築用ライブラリ GhostHouse[2] である。そして、もう一つは、今回新たに開発している HolyGhost と呼ぶライブラリで、分散処理機能やモデル化の基本機能を提供する。これらライブラリをベースに、階層的に、GUI 編集機能や通信制御機能を提供する基本部品、トレンドグラフや帳票などの機能部品、そして、それらを用いたプログラムが実現されている。なお、現在のシステムは、UNIX ベースのものであるが、Windows-NT 等、他の計算機環境での実現についても検討を開始している。

#### 4 まとめ

本稿では、筆者らが開発中の監視制御向けオブジェクト指向フレームワークについて、そのシステム構成について述べた。なお、現在、フレームワークの開発と並行し、実際の監視制御システムの開発も実施中である。そして、この経験を元にフレームワークの洗練と拡張を行うことが今後の課題である。

#### 参考文献

- [1] 北村他, 分散型監視制御システム構築環境 (2) ~ (4), 第 52 情処全大, 1996
- [2] 北村他, 生成・カスタマイズ方式による GUI 構築手法の提案とクラスライブラリ GhostHouse による実現, 情処論文誌, Vol.36, No.4, 1995.