

## オブジェクト指向フレームワークによる広域監視制御システムの開発

4C-5

上野浩一郎\* 大塚義浩\* 秋富利伸\*\* 佐藤弘幸\*\* 金枝上敦史\*

\*三菱電機（株）情報技術総合研究所 \*\*同 電力・産業システム事業所

### 1. はじめに

広域監視制御システムは、中央センターから広域ネットワークを介して、広いエリアに分散した機器群を監視制御するシステムであり、水道/ビル/道路など様々な分野で使用されている。筆者達は、このシステム開発における生産性向上を図るべく、広域監視制御共通プラットフォーム(以下共通プラットフォームと略)の開発と実工事適用を進めてきた。共通プラットフォームは、水道やビルと言った各分野のシステム仕様を標準化し、その標準仕様を実装した共通ソフトウェア群から構成されており、各工事開発で共通的に再利用されるソフトウェア基盤である。図1に共通プラットフォームのソフトウェア構成の概略を示す[1]。本稿では、この共通プラットフォームに対するオブジェクト指向フレームワーク(以下フレームワークと略)技術の適用について報告する。

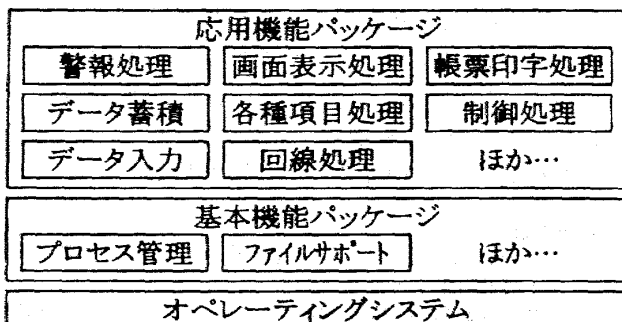


図1 共通プラットフォームのソフトウェア構成

### 2. フレームワーク開発の方針と狙い

共通プラットフォームを構成する各種ソフトウェア群は、(1)多くの適用実績があり、既に現場検証された資産である。(2)種類が多数あり、ソースコード量が膨大である。そこで今回の開発では、既存のソフトウェア資産を活用しつつ、一部についてフレームワーク化の方針を取ることにした。

#### 2.1 狙い(1) 応用機能パッケージの適用性向上

Developing Wide-area supervisory and control systems using object oriented framework  
Koichiro UENO, Yoshihiro OHTSUKA, Toshinobu AKITOMI, Hiroyuki SATO, Atsushi KANAEGAMI  
\* INFORMATION TECHNOLOGY R & D CENTER,  
\*\* ENERGY & INDUSTRIAL SYSTEMS CENTER  
MITSUBISHI ELECTRIC CORPORATION

共通プラットフォームを各工事に適用する際、次の3種類の作業がある。(1)パラメータ値設定[システムの振舞を定義している各種パラメータ群を各工事仕様に合わせて値設定する。]、(2)アドオン実装[工事毎に異なるソフトウェアを追加開発する。例えば、監視制御画面や帳票がこれに該当する。]、(3)ソースコード改造[共通プラットフォームのソースコード自体に変更を加える。]

パラメータ値設定やアドオン実装は、ビジュアルな支援ツール(エンジニアリングツールや画面/帳票ビルダ等)で生産性を向上させている。しかし、パラメータ値設定やアドオン実装だけで仕様を満たせない場合はソースコード改造を実施せざるえず、その主対象は応用機能パッケージである。従来のソースコード改造では、一か所の変更により影響範囲が連鎖的に拡大し、それに伴い試験対象も膨らみ、これが生産性向上を阻害してきた。そこでフレームワーク技術により、応用機能パッケージを柔軟に適用可能にすることが第1の狙いである。

#### 2.2 狙い(2) 応用機能パッケージ間でソフトウェア構造とソースコードを共有

応用機能パッケージ群は各々の実行内容は当然異なっているが、その一方で、システムを構成するパッケージとして例えば以下のような類似したロジックが存在している。

例1: 各パッケージ間ではシステムとして協調動作するために、イベントの受発信を繰り返している。各パッケージ毎に扱うイベント群は異なるが、イベント処理するロジックは類似している。

例2: 各パッケージは監視制御データ値やパラメータ値にアクセスするのに、基本機能パッケージ「ファイルサポート」が提供するAPI(Application Programming Interface)を呼出している。これらのAPI呼出のロジックは類似している。

例3: 各パッケージでは定周期で機能を実行するために、前回の機能実行時刻と現在時刻を比較する処理を繰り返している。各パッケージや機能毎に定周期のタイミングは異なるが、タイミング判定のロジックは類似している。

しかし従来は、応用機能パッケージ間で類似ロジックが重複して各々実装されていた。そこでフレームワークにより、応用機能パッケージ間でソフトウェア構造とそのソースコードを共有することが第2の狙いである。

### 3. フレームワークの開発とその適用

#### 3.1 応用機能テンプレート

2.2節「狙い(2)」に対応して、応用機能パッケージ間で共通するソフトウェア構造を「応用機能テンプレート」としてフレームワーク化した。応用機能テンプレートは、応用機能パッケージを構成する各プロセスのひな型となるものである。図2のカテゴリ1にそのクラス例を示す。2.2節の例1に対しては、「プロセス」クラスと「ハンドラ」クラスにイベントドリブンな基本メカニズムを組込んでいる。例2に対しては、基本機能パッケージが提供するプリミティブなファイルアクセス用API関数を「ファイル」クラスとしてラッピングしている。例3に対しては、タイミング判定のロジックを「タイムスイッチ」クラスにカプセル化している。

#### 3.2 演算項目処理への適用

上述した応用機能テンプレートの適用対象例として、応用機能パッケージ「各種項目処理」内の演算項目処理プロセスを取り上げる。演算項目とは機器群の監視データ値を加工(アナログ積算や四則演算など)して求めるデータであり、演算項目処理とは、この演算項目の値を周期的に生成するものである。

図2のカテゴリ2にそのクラス例を示す。演算項目処理プロセスの実装は、応用機能テンプレートからクラスを派生して、派生した子クラスに対して演算項目処理特有な仕様のみを上書するだけである。他のプロセスの実装でも同様に応用機能テンプレートを適用する事により、2.2節「狙い(2)」を達成できる。

#### 3.3 演算項目処理の拡張例

図2のカテゴリ2では、現在の共通プラットフォーム

が定義している演算種別(アナログ積算や四則演算など)に対応する「演算項目」クラスが既に提供されている。この演算項目処理に、例えば累計雨量(雨量カウンタの差分が0より大きい間、その差分を累積した値)を算出する仕様を追加したいとする。この場合、2.1節で述べたパラメータ値設定やアドオン実装だけでは対応できず、演算項目処理自体を改造する必要が有る。このためには、図2のカテゴリ3に示すように「演算項目(累計雨量)」クラスを追加し、このクラス内に追加仕様を差分実装する。この際、既存のカテゴリ2に対するクラスとは独立しているため、2.1節「狙い(1)」を達成できる。

#### 4. まとめ

今後、応用機能パッケージを応用機能テンプレート上に段階的に移行することにより、既存のソフトウェア資産と補完しつつ、応用機能パッケージの適用性向上を徐々に拡大できる。また、基本機能パッケージのAPIをラッピングしている応用機能テンプレート上に応用機能パッケージを組立てる事により、基本機能パッケージを交換しても応用機能テンプレートに対応させるだけで、広域監視制御の本質である応用機能のソフトウェアはそのまま流用できる。

その一方、本稿の取組は現状資産の改善活動であるため、今後の分散オブジェクト技術等によるシステム構築にも抜本的に取組む事が課題である。

#### 参考文献

- [1] 内藤,他:「配水管理システム新シリーズ」、三菱電機技報・Vol.71・No.11・1997.
- [2] 金枝上,他:「広域監視制御システムフレームワーク開発」、情報処理学会第55回全国大会論文集(1),pp1-376.

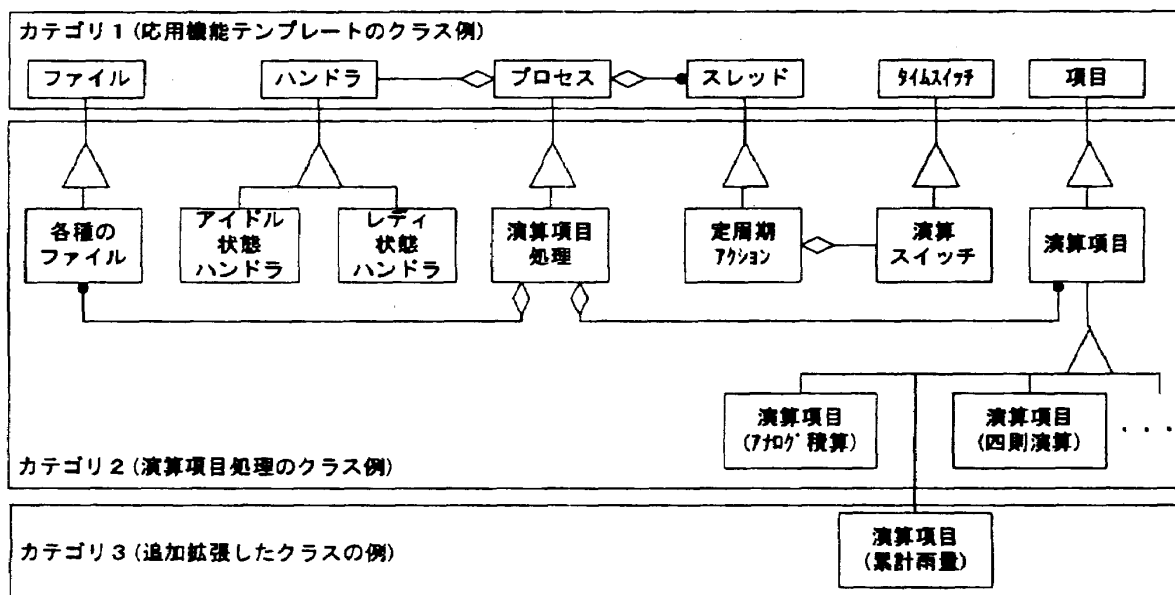


図2 応用機能パッケージのクラス図例