

監視制御システムにおけるプログラミングレス開発の支援

A Support Tool for Non-programming Development of Supervisory Control System

甲斐 啓文†
Hirofumi Kai

津高 新一郎†
Shinichiro Tsudaka

中田 秀男†
Hideo Nakata

山内 健司†
Kenji Yamauchi

1. はじめに

監視制御システムの分野では、近年の技術の進歩に伴い、大規模かつ複雑なシステムへの要求が高まる一方で、短工期かつ高品質なシステムの開発が求められている。ソフトウェアの再利用性を高め、システム開発を効率化する技術の一つにフレームワーク技術がある[1]。我々は、定義ファイルの作成のみでシステムの構築を可能とする独自の監視制御システム向けフレームワークを開発し、開発コストの低減化を実現した。しかしながら、定義ファイルの作成には、一定期間の訓練を要する、タイプミスが混入する可能性がある、システムの規模に比例して作成・管理すべき定義ファイルの数が増加する、といった問題がある。そのため、定義ファイルの容易かつ効率的な作成を支援する環境が必要になる。そこで、本研究では、上記定義ファイルの作成・編集を支援するツールの開発を行うことで、開発コストの更なる低減化を図っている。ツールは、近年、拡張可能な統合開発環境として注目されている Eclipse[2]のプラグイン[3]として開発した。本稿では、本監視制御フレームワークを利用したプログラミングレス開発を支援するために Eclipse プラグインとして開発したツールについて述べる。

2. プログラミングレス開発を実現する監視制御フレームワーク

我々が開発した監視制御フレームワークの概要を図1に示す。本フレームワークは、サーバ側プログラムの開発環境であるアプリケーションフレームワークと、マンマシン側プログラムの開発環境である HMI (Human Machine Interface) フレームワークからなる。本フレームワークでは、あらかじめ用意された定義ファイルの内容に従って実行時にソフトウェアコンポーネントをインスタンス化することで、監視制御機能を実現する。そのため、プログラミングを行うことなく、定義ファイルの作成のみでシステムを構築することができる。本フレームワークで作成すべき定義ファイルは、以下の3つである。

- 処理内容定義ファイル
アプリケーションフレームワーク上で監視制御プログラムの生成に用いられる。監視制御システムに必要な処理のレポートを、あらかじめ再利用可能なソフトウェアコンポーネントとして用意し、それらの組み合わせを定義することにより、各種機能プログラムを開発することができる。
- 操作画面定義ファイル
HMI フレームワーク上で操作画面プログラムの生成に用いられる。ラベルやボタン、テーブルといった GUI コンポーネントの組み合わせ、及びイベントハンドラ

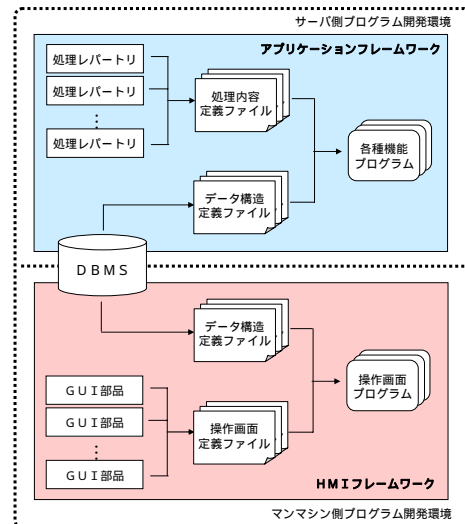


図1 監視制御フレームワークの概要

を定義することにより、マンマシン側の操作画面プログラムを開発することができる。

- データ構造定義ファイル
アプリケーションフレームワーク、HMIフレームワークの両方で用いられ、サーバ側の機能プログラム及びマンマシン側の操作画面プログラムがアクセスするデータ構造(ビュー)を定義する。テーブルの結合を定義ファイル内に記述することで、複数のテーブルに跨ったアクセスを簡略化することができる。

3. 課題

上記監視制御フレームワークの利用により、高度なプログラミングの技術が不要となるため、低コストでのシステム構築が可能となる。しかしながら、汎用のエディタを用いて定義ファイルを作成する場合、以下のような課題がある。

- 作成ノウハウの習得が必要
上記の各定義ファイルの作成には、監視制御システム設計に関する知識と、定義ファイルの書式・文法の理解が不可欠である。そのため、それらのノウハウの習得に一定の期間を要する。
- タイプミスの混入
汎用のエディタによって手入力で作成する場合、タイプミスが混入する可能性がある。
- システム規模に比例して定義ファイルの数が増加
開発する機能や操作画面毎に定義ファイルを作成する必要があるため、システムの規模が大きくなると作成すべき定義ファイルの数も増加する。

4. プログラミングレス開発の支援

上記で述べた各定義ファイルの中でも、特に複雑な形式で記述されるデータ構造定義ファイルについて、専用の作成・編集ツール JoinEditor を Eclipse プラグインとして開発した。Eclipse は、拡張可能なアーキテクチャを備えており、プラグインを追加することで Eclipse 環境および Eclipse 上の他のツールとシームレスに統合されるツールを比較的容易に開発することができる。そこで今回、将来的な開発環境整備を視野に入れ、Eclipse プラグインとしての実装形態を選択した。

4.1 JoinEditor

JoinEditor は、Eclipse エディタの拡張ポイント (org.eclipse.ui.editors) を拡張することにより作成した、データ構造定義ファイル専用のエディタプラグインである。Eclipse ワークベンチ上でデータ構造定義ファイルをオープンすることにより起動する (図 2)。JoinEditor は、以下の機能を備えている。

マルチページ構成エディタ

データ構造定義ファイルは、図 3 に示すように複数のセクションが列挙されたフラットな形式で記述される。そこで JoinEditor では、セクション毎に GUI ベースの編集画面を割り当て、タブによるマルチページ構成とすることで、ファイルの内容を構造化し、データ構造定義ファイルの複雑な記述フォーマットを意識しない直感的な編集作業を支援する。

編集支援機能

データ構造定義ファイルでは、テーブル名やカラム名など、設定する値の範囲があらかじめ決まっている設定項目が存在する。JoinEditor では、そのような項目についてコンボボックス等を利用した選択式の入力フォームを提供する。また、テーブル名が決まれば対応するカラム名が決まるように、ある編集項目に設定した値に依存して他の編集項目に設定可能な値の範囲が限定される場合、その依存関係を GUI コンポーネントに反映する。例えば、図 4 に示すように、データベース上の任意のテーブル名及びカラム名を指定する入力フォームの場合、左のコンボボックスから任意のテーブル名を選択すると (図 4 の)、右のコンボボックスに、選択したテーブル名に対応するカラム名のリストが設定される (図 4 の)。利用者は、リストの中から要求に応じてカラムを選択する (図 4 の)。以上

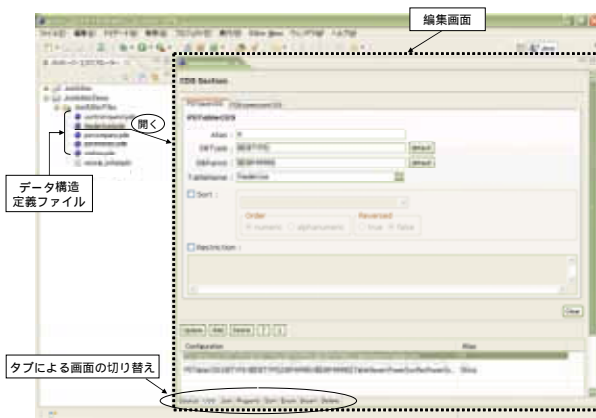


図 2 JoinEditor の起動画面

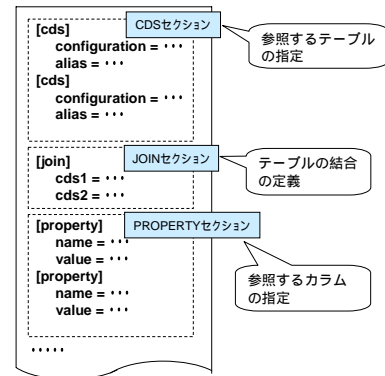


図 3 データ構造定義ファイル

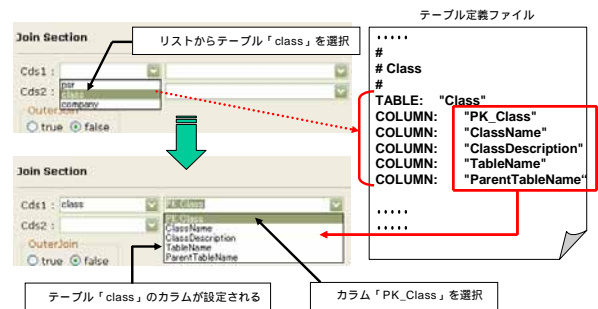


図 4 編集支援機能

の機能により、タイプミスの防止、編集の効率化を実現している。

新規作成ウィザード

データ構造定義ファイル専用の新規作成ウィザードを備える。

環境設定ダイアログ

データ構造定義ファイルの格納場所、及びデータ構造定義ファイルの編集項目に設定する初期値といった環境変数を設定するための、環境設定ダイアログを備える。

以上の機能により、データ構造定義ファイルの作成ノウハウの習得時間を短縮し、タイプミス混入の確率を減らすことができる。また、作成・管理すべきファイル数が増大した場合でも、効率的な作成・管理が可能となる。

5. おわりに

本稿では、当社で開発した監視制御フレームワークの容易かつ効率的な利用を支援するツールの開発について述べた。今後は、編集画面上に説明用のツールチップを表示する、作成したデータ構造定義ファイルのテスト環境を構築するなど、JoinEditor の機能強化を行うと共に、他の定義ファイル (処理内容定義ファイル、操作画面定義ファイル) についても、同様に作成・編集用の Eclipse プラグインを開発し、本フレームワークの利用を総合的に支援するための統合開発環境を整備する予定である。

参考文献

- [1] Ralph E Johnson, 中村宏明 他 (1999), 『パターンとフレームワーク』, 共立出版
- [2] Eclipse.org, <http://www.eclipse.org/>
- [3] Elic Clayberg, Dan Rubel (2006), "Eclipse Building Commercial-Quality Plug-ins Second Edition", Addison-Wesley