

オブジェクト指向設計によるシステム開発

System Development Applying Object-Oriented Design

加納 治英
KANOU Haruhide

高橋 徹
TAKAHASHI Toru

将来の航空管制方式の確立を目的とした、次世代飛行場管制シミュレータ装置を開発した。オブジェクト指向^(注1)分析、Java^(注2)言語によるプログラミング、分散オブジェクト技術、XMLによるデータ定義の標準化などのオブジェクト指向技術を全面的に取り入れた。ソフトウェアのフレームワーク化、クライアントプログラムの部品化、仲介アプリケーションソフトウェア(以下、アプリケーションと略記)アーキテクチャなどにより、拡張性、再利用性の高いソフトウェアが構築でき、開発効率も30%以上向上した。

We have developed a next-generation air traffic control simulation workstation with the aim of establishing a future air traffic control system. Object-oriented technologies were fully employed such as object-oriented analysis, programming in Java language, distributed-object technology, and standardization of data definition by XML.

Highly expandable and reusable software can be constructed by a framework architecture, component programming, mediation application software architecture, etc., thereby improving development efficiency by more than 30%.

1 まえがき

オブジェクト指向設計が、ソフトウェアの生産性を向上させる技術として期待されて久しい。再利用性の向上や仕様変更への対応の容易さなどの利点が見込まれるため、この間、多くの派生技術や開発手法が提唱され、ソフトウェア技術の重要な柱の一つになっている。

航空管制分野でも、将来に向けてより電子化されたシステムの研究が進んでおり、拡張性や再利用性の向上など柔軟なソフトウェアの開発が不可欠になっている。また、この分野で新規開発を行う場合、航空管制官の評価を受ける必要があり、その際にはさまざまな仕様変更や機能追加が求められる。

次世代型飛行場管制支援装置は、特に電子化による管制業務効率の向上が見込まれる飛行場管制に的を絞って、航空管制装置の開発検討を支援するものである。次世代型飛行場管制支援装置の開発にあたり、先に述べた機能変更や追加の要求に柔軟に対応できることを期待し、設計から開発・製造まで一貫してオブジェクト指向技術を適用したので、以下にシステムの概要、およびオブジェクト指向での開発方法、その効果について述べる。

2 システム概要

次世代飛行場管制支援装置(図1)は、将来の航空管制環境を想定し、空港における管制業務を支援するシステムがどうあるべきかを研究するための実験用システムである。空港における管制業務には、航空機の離着陸の許可を行う

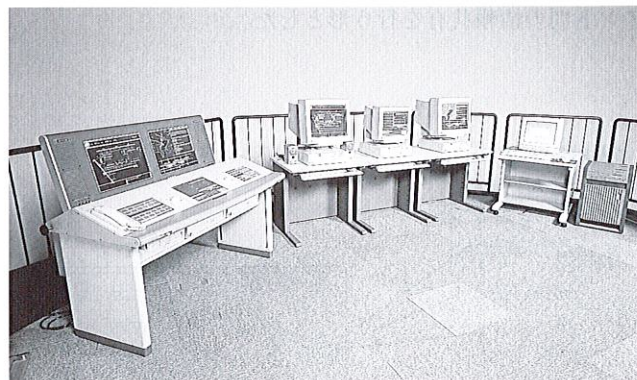


図1. 次世代飛行場管制装置 管制機器を一体化したコンパクトな設計となっている。

Air traffic control workstation

飛行場管制(LC)と、空港内で航空機を誘導する地上管制(GC)がある。現状の空港で行われている航空管制は、人間系によるものであるが、次世代の飛行場管制では、空港内の安全性と業務の効率化を図ることを目的として、コンピュータ処理をベースにしたシステムが検討されている。このシステムは、その研究のための実験用装置であり、以下の装置で構成される。

- (1) シミュレータサーバ 管制対象航空機を生成する航空管制用ソフトウェアが動作し、UNIX^(注3)サーバ

(注1) 関連するデータとそのデータを扱う手順の集合とを含んだソフトウェアパッケージをオブジェクトと呼び、このように定義されたオブジェクトをベースに分析、設計、開発を進めていく、ソフトウェアの範例の一つをオブジェクト指向と呼ぶ。

(注2) Javaは、米国Sun Microsystems社の商標。

で構成される。

- (2) 管制用クライアント シミュレータで生成される航空機の管制を行うためのグラフィカルユーザーインタフェース(GUI)部品から構成される。飛行場管制卓、地上管制卓はそれぞれ3台のパソコン(PC)で構成される。

3 適用技術

システム開発においては、オブジェクト指向技術を全面的に適用した。各適用技術について以下に記述する。

3.1 オブジェクト指向分析

システム全体の分析、モデル化をオブジェクト指向で行なった。これにはOMG(Object Management Group)で標準化が進められているUML(Unified Modeling Language)を用いた。この分析により設計したシステムアーキテクチャ(注4)を図2に示す。サーバとクライアント間の通信には、仲介を行うアプリケーション(メディエータ)を設け、サーバとクライアント間の情報交換、および3台のクライアント間の情報共有を行う形とした。

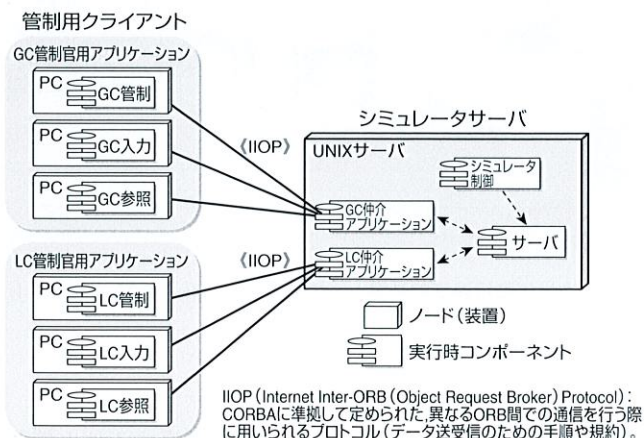


図2. オブジェクト指向分析によるソフトウェアアーキテクチャ
シミュレータ制御を行うサーバと、管制用アプリケーションを仲介アプリケーションが接続する構造とする。

Software architecture by object-oriented analysis

3.2 Javaによるプログラミング

Javaはオブジェクト指向言語およびその実行環境であり、プラットフォームに依存しないソフトウェアの開発が可能である。今回は、以下の理由でJavaを選択した。

- (1) 環境の異なるサーバとクライアント間で実装環境を統一する。これにより開発効率を向上させる。
(2) 将来の実機における実行環境の変化を、Javaに吸収させる。ソフトウェアを変更することなく実機が動

作する。

3.3 分散オブジェクト

分散オブジェクト技術とは、異なる実行環境やプロセス上に存在するオブジェクト間での通信を実現するための技術である。この開発では、分散オブジェクト技術であるCORBA(Common Object Request Broker Architecture)を採用した。通常、ネットワークに分散された異機種・異言語のアプリケーション間の通信処理には、複雑なコードが必要になる。CORBAは、アプリケーション間のインタフェース定義(オブジェクトの型、属性、サポートするメソッドなど)を基に、その通信処理に必要なコードを自動生成する。そのため他システムとの接続は、アプリケーション間のインタフェースを定義するだけでよい。

3.4 XMLによるシナリオ定義

XML(eXtensible Markup Language)は、拡張することが可能なマーク付け(タグ付き)言語である。このXMLをシミュレーションのシナリオ定義用言語として採用した。シミュレータサーバは、XMLで記述されたシナリオを解釈する機能をもつ。XMLは将来データ管理の標準言語として位置付けられることが期待されており、今後のシステム拡張に対する柔軟な対応が可能となる。

4 オブジェクト指向技術適用の効果

今回適用したオブジェクト指向技術の効果について、①ソフトウェア再利用性の向上、②開発効率の向上、③変更の容易さ、拡張性の向上、という3点から検証する。

4.1 フレームワーク化による再利用性の向上

フレームワークとは、ソフトウェアの基本的、汎(はん)用的な機能および構造を提供するソフトウェア部品群である。クラスライブラリと役割が似ているが、次の点で異なる。

- (1) 利用はクラスの単位ではなく、ある機能および構造を提供するようなクラス群の単位である。
(2) クラスライブラリは、それを利用する側からの一方的な制御であるが、フレームワークの場合はその逆もある。

サーバアプリケーションで、航空機の動作を模擬するシミュレータ制御と、エラー処理手続きをフレームワークとして実装した。このフレームワークは、他のシミュレーションシステムでも再利用可能である。

4.2 コンポーネント化による再利用性の向上

コンポーネントとは、ある機能を提供するソフトウェア

(注3) UNIXは、The Open Groupの米国およびその他の国における登録商標。

(注4) アーキテクチャとは、ハードウェアおよびソフトウェアを含めた、コンピュータ全体に関する基本的な設計思想。

のブラックボックス部品である。その部品のインタフェースだけが公開されており、部品の内部構造を知らなくても利用できるという特長をもつ。対象アプリケーション用のコンポーネントを作成し、それらを組み合わせることで、仕様の変更や追加に伴うコードの修正が局所化される効果が生まれる。クライアントアプリケーション(図3)は、空港面地図、メッセージ表示領域、時計、気象などのGUIコンポーネントの組合せとして開発した。

上記4.1、4.2節の方法により、図4に示すように、実装したソフトウェアコードのうち、全体の約30%が再利用可能なコードとなった。

4.3 開発効率の向上

全Javaソースコードのうち、CORBA準拠ミドルウェアにより自動生成されたクライアント/サーバ間の通信ソフトウェアの割合は約30%である。通信処理コードに関

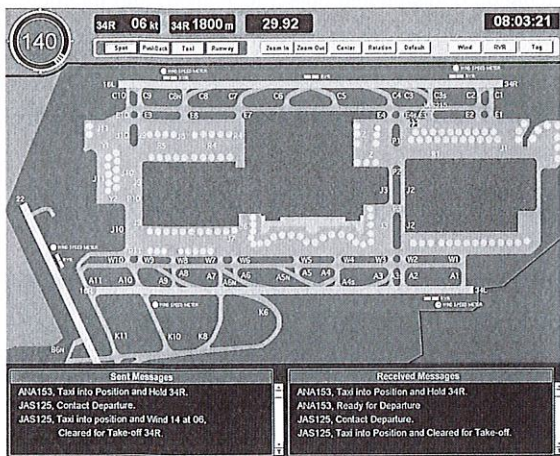


図3. クライアントアプリケーションのGUI 飛行場地図、時刻、天候などのGUIがすべて部品化され、完全に独立して使用できる。

Graphical user interface of air traffic control workstation

しては、ほとんどコードを開発することがなくなり、開発効率が大幅に向上した。

4.4 変更の容易性

オブジェクト間の関係について、従来方式とこの開発方式の比較を図5に示す。メディエータを介するアーキテクチャとしたことにより、サーバとクライアント間、およびクライアント間に存在するオブジェクトの依存関係が、メディエータに吸収され最小限になる。そのため、オブジェクトの変更による、他のオブジェクトへの影響が少なくなった。

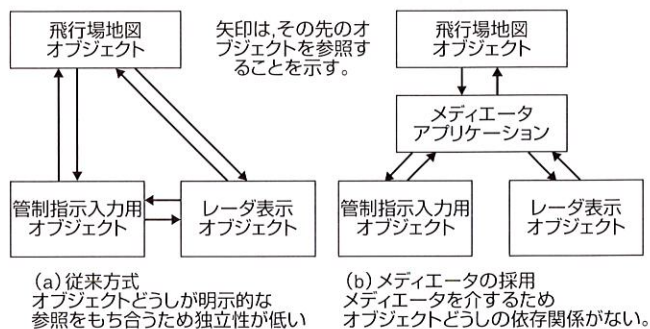


図5. メディエータアーキテクチャの概要 コード変更時、従来に比べメディエータ部分が、他のソフトウェアへの影響を吸収する。

Outline of mediator architecture

5 あとがき

オブジェクト指向技術の適用により、再利用性に優れたソフトウェアの開発が可能になった。また、約半年という短期間で開発を終了することができた。

今後は、飛行場管制分野だけでなく、広く航空管制分野へのオブジェクト指向技術の適用を図り、より効率的なシステムの開発が可能となるよう努力を続けていきたい。

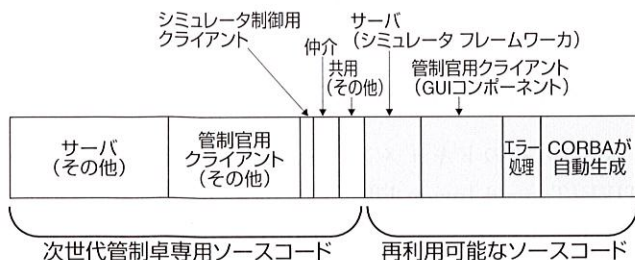


図4. 再利用可能なソフトウェアコードの比率 フレームワーク化とコンポーネント化により、約30%のコードが再利用できる。

Ratio of reusable software code



加納 治英 KANO Haruhide

情報・社会システム社 小向工場 電波応用システム技術
部主務。

航空管制システム機器の設計・開発に従事。

Komukai Operations



高橋 徹 TAKAHASHI Toru

情報・社会システム社 小向工場 電波応用設計部。

航空管制システム機器の設計・開発に従事。

Komukai Operations