

1S-3 制御用ソフトウェア開発実行環境 Objectpowerにおけるオブジェクト指向技術の適用

大柴正清 安藤博文 石川健一 金子淳一
富士ファコム制御（株）

1. はじめに

制御用ソフトウェアObjectpowerは、アプリケーションプログラムを開発・実行するための分散オブジェクト環境である。

Objectpowerは、開発環境と実行環境から構成されている。本稿では、その中の実行環境において、制御用ソフトウェアに要求される高速応答性を考慮したオブジェクト指向技術の適用について述べる。

2. 概要

Objectpowerでは、アプリケーションプログラムと、関連するデータをカプセル化したものをオブジェクトとして管理する。オブジェクトを構成するメソッド（Objectpowerではサービスと呼ぶ）は、メッセージ式によるサービス要求によって起動することができる。

Objectpowerでは、更にアプリケーションプログラムの常駐機能や実行優先管理、および実行多重制御方式による負荷分散などを付加することによって、高速応答性を実現している。

3. オブジェクトの管理

オブジェクトは3種類のプロセスから構成される。

(a) オブジェクト初期処理プロセス

(b) オブジェクト終了処理プロセス

(c) サービスプロセス

オブジェクト初期処理プロセスはオブジェクトの起動時に初期サービスを実行するプロセスであり、オブジェクト終了処理プロセスはオブジェクトの終了時に終了サービスを実行するプロセスである。また、サービスプロセスは他のオブジェクトからの要求を受けてサービスを実行するプロセスである。1つのオブジェクトは複数のサービスプロセスから構成される。サービスプロセスの生成は次の生成属性に従う。

- ・先行生成属性：オブジェクトの起動時に、あらかじめプロセスを生成
- ・動的生成属性：他のオブジェクトからの要求時にプロセスを生成

高速な応答が要求されるサービスには先行生成属性を指定し、また、通常の運用ではあまり要求がなく、高速な応答が要求されないサービスには動的生成属性を指定することによって、システム運用中のプロセス数を減らすことができる。

4. サービスの管理

(1) サービスの実行タイプ

サービスの利用用途に応じて、3種類の実行タイプを選択することができる。（表1）

(a) 関数モード

関数資格で実行されるサービスであり、サービスの要求元のプロセス配下で実行される。

(a) プロセスモード

サービスの要求元とは別の要求先のプロセス配下で実行される。プロセスモードの場合処理結果を要求元に返すか否かにより、同期

Introducing object-oriented technology by control software development executive environment "Objectpower"

Masakiyo Ohshiba, Hirofumi Ando,
ken'ichi Isikawa, Jyun'ichi Kaneko
Fujifacom Corporation 1-2-2, Manpukuji,
Asao-ku, Kanagawa 215, Japan

タイプと非同期タイプの2種類がある。

表1 サービス実行タイプ

実行タイプ		利用用途
関数モード (同期タイプ)		プロセス切り換えのオーバーヘッドがなく、高速な応答を期待するサービス
プロセスモード	同期タイプ	サービス要求の受信前に事前処理を必要とし、要求元でサービス完了を確認する必要があるサービス
	非同期タイプ	要求元でサービスの完了を待ち合わせる必要のないサービス

(2) サービスの実行優先管理

オブジェクトの定義時に、サービス単位に実行優先レベルを指定することができる。実行可能なサービスが複数ある場合、高レベルのサービスは優先的に実行される。実行優先レベルを指定するスケジューリングクラスには、リアルタイムクラスとタイムシェアリングクラスがある。

(3) サービス実行グループ

オブジェクト内のプロセスモードのサービスをまとめてサービス実行グループを定義することができる。これによりオブジェクト内の複数サービスを逐次処理するか、並行に処理するかの実行制御が実現できる。すなわち、サービス実行グループとして定義されたサービスは同じサービスプロセス配下で実行される。そのためそれらのサービスに対して連続して要求してもシーケンシャルにしか実行されない。この機能を活用することにより、逐次処理が必要なアプリケーションプログラムの処理を軽減することができる。

(4) サービスの実行多重制御

プロセスモードのサービスでは、サービス実行グループにサービスの実行多重度を指定することができる。実行多重度を指定したサービス実行グループに要求が集中した場合、サービス

は指定の多重度数だけ並行処理が実現できる。

(5) サービスのメモリ常駐

オブジェクト定義時にサービスに対してメモリ常駐を指定することができる。指定されたサービスは物理メモリに常駐される。実行優先管理と組み合わせることで、より高速にサービスを実行することができる。

5. オブジェクト間のメッセージ通信

他のオブジェクトへの処理要求はメッセージ式によって行う。このメッセージ式はアクセス名とサービス名から構成される。アクセス名は要求先であるサーバオブジェクトを指定するための論理的な名前であり、オブジェクト間の関連を定義時にサーバとして想定するオブジェクトを識別するために割り付ける。

このアクセス名と、実体となるサーバオブジェクトとの対応付けは、アプリケーションプログラムの記述から切り離されている。この結果アプリケーションプログラムとサーバオブジェクトとの結合度は疎となり、オブジェクトの可搬性・再利用性の向上を図ることができる。

また、オブジェクト間の関連を定義することによって、メッセージ式によって要求を送信するためのクライアントスタブと、受信した要求を実行するためのサーバスケルトンが生成される。オブジェクト間の通信は、このクライアントスタブとサーバスケルトンを使って、サービスの属性やオブジェクトの場所に関係なく、同形式のメッセージ式でサービスの実行を要求することができる。

6. おわりに

Objectpower では、分散オブジェクト環境の考え方を取り入れ、更にサービスに属性を付加することで、産業制御分野で必要とされる高速応答性を持った実行環境を実現している。今後は、分散オブジェクト環境の標準化動向を踏まえ、レベルアップしていく予定である。