

通信網ワイド分散システムにおける高信頼オブジェクトの生成

40-2

武本 充治

NTT ネットワークサービスシステム研究所

1 はじめに

並行オブジェクト指向モデルに基づく通信網ワイド分散処理プラットフォーム [1] の検討を行なっている。本プラットフォームの適用対象は通信システムであり、高性能と同時に高信頼性も要求される。アプリケーションの動作は並行オブジェクトの動作で記述される。この並行オブジェクトを必要に応じて複製し、稼働及び予備系として多重化することにより、冗長な多重化を含まないシステムの高信頼化ができる [2]。本稿ではその複製をプラットフォームで管理する機構について述べる。また、アプリケーションが外界に作用する、いわば目的とする機能とそれ以外の付加される機能の分離とその考えに基づいたコード生成法を提案する。

2 プラットフォームでの管理機構

2.1 オブジェクトの複製

動作実体であるオブジェクトが目的の動作を続けることができなくなった場合に、オブジェクトの複製で実現される予備系に切替えることで高信頼動作を実現する。オブジェクト単位で予備系を実現する方式は、予備系の構成をノード単位とする方式よりも、不要な冗長構成がシステム中に含まれないという利点を持つ。

高信頼動作を実現している時に、動作主体のオブジェクトが複製されていることは隠蔽されなければならない。また、高信頼ともに高性能も要求される。これらの要求条件を満たす複製管理機構が必要になる。

2.2 複製管理

アプリケーションオブジェクトの高信頼化を目的とするために、アプリケーションの故障の影響が及ばないプラットフォーム内に複製管理機構をおく。また、高信頼化の実現が分散透過性を損なうことがあってはならないので、高信頼化に関する情報は複製管理機構で管理する。さらに、通信網ワイド分散システムにおいては、複製管理機構が取扱う情報は、各ノードに分散され管理される。

2.3 高信頼オブジェクト

高信頼動作を行なう仮想的なオブジェクトを高信頼オブジェクト、目的とする機能の実際の処理を行なうオ

ブジェクトを複製オブジェクト、プラットフォーム内でノード毎に置かれる複製管理機構オブジェクトを複製管理サーバオブジェクトと呼ぶ。ノードに分散された複製管理サーバオブジェクトはお互いに周期的に通信を行ない、ノード自身の動作とノード上で動作している複製オブジェクトの状態を監視する。あるノードやオブジェクトの動作不良を検出すれば、複製管理サーバオブジェクトは、ノード再起動や複製切替の処理を行なう。また、高信頼オブジェクト宛メッセージの複製オブジェクト宛への変換など、あるノードでの高信頼オブジェクトの外部との作用を管理する機能を持つ。

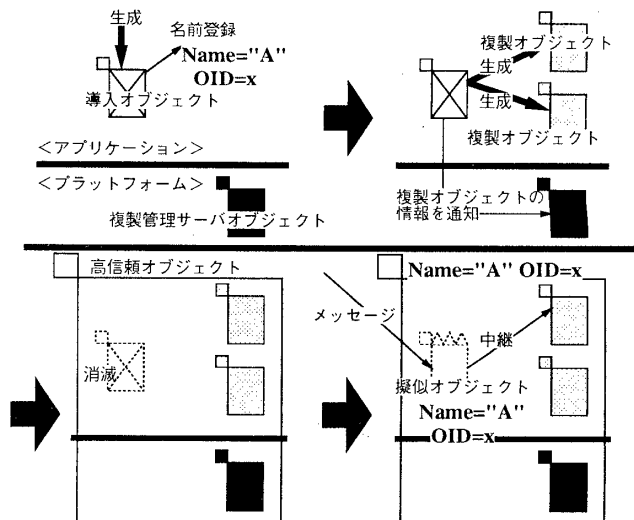


図 1: 高信頼オブジェクト

複製オブジェクトと複製管理サーバオブジェクトが協調する。そのために、高信頼オブジェクト生成時に、複製オブジェクトを生成し、その複製オブジェクトに関する情報を複製管理サーバオブジェクトに通知し、自身が消えるというオブジェクトを用いる。このオブジェクトを導入オブジェクトと呼ぶ。高信頼オブジェクトに対して、目的とする機能を実現する元のオブジェクトを基本オブジェクトと呼ぶ。

高信頼オブジェクトは実際には1つのオブジェクトではない。その動作は、外部から導入オブジェクトを目指してメッセージが送信されたり、導入オブジェクトが種々の処理を行なったりしているように観測される。この状態は、消滅したはずの導入オブジェクトがアプリケーション空間に引き続いて存在し、複製オブジェクトを管理しているようにとらえることが可能である。この場合、プラットフォームの複製管理サーバオブジェクトの作用により、導入オブジェクトに替って、アプリケー

ション空間に存在しているようにとらえられるオブジェクトを擬似オブジェクトと呼ぶことにする。以上の構成を図1に示す。

3 高信頼オブジェクトソースコードの生成法

3.1 基本方式

高信頼化実現のための複製オブジェクトは基本オブジェクトと比較すると、目的とする機能部分は性能劣化を除いては同じである。しかし、複製オブジェクトには高信頼動作のための機構が追加されているところが異なる。

複製オブジェクトの高信頼動作機構のソースコード片は同一の雛型から生成できる。したがって、複製オブジェクトのソースコードは、その雛型と基本オブジェクトのソースコードから自動生成することが可能である。また、導入オブジェクトのソースコードも基本オブジェクトの構造を解析することで自動生成することが可能である。管理機構をアプリケーション空間におく方式で提案・実装された言語処理系 [2] と同様に言語処理系で自動生成できる。

基本オブジェクトのソースコードに高信頼化マークを付加し、それをプリプロセッサで処理することにより、導入オブジェクトと複製オブジェクトのコードを得る。これはアプリケーションプログラマが実装するのは目的とする機能部分であり、プリプロセッサが高信頼に関する付加的な機能を自動生成するという、機能分離の考えに基づいている。この流れを図2に示す。

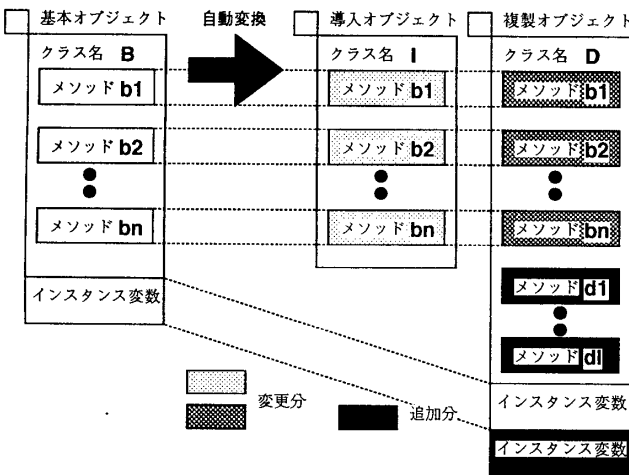


図 2: 高信頼オブジェクトコード生成

3.2 導入オブジェクト

導入オブジェクト中のメソッドは、起動されると、図1に示した動作を行なう。図2で、導入オブジェクトが基本オブジェクトの持つすべてのメソッドの名前と同じ名前を持つメソッドを持っているのは、基本オブジェクト中のどのメソッドが最初に起動されるかオブジェクトを生成する側には一般には決定できないからである。どのメソッドが最初に起動されるかを言語処理系に与え

ることによって、導入オブジェクトのコードを小さくすることも可能である。この最適化を施せば、高信頼オブジェクト生成時のコストが小さくなる。

3.3 複製オブジェクト - 内部状態管理 -

複製オブジェクト中の機能で、基本オブジェクトと大きく異なるのは、内部状態維持の動作である。オブジェクトの内部状態はインスタンス変数で実現されており、インスタンス変数が変更される時に、オブジェクトの内部状態が変化する。複製オブジェクトが動作しなくなった場合に対処するために、高信頼オブジェクトの内部状態維持はプラットフォーム内の複製管理サーバオブジェクトに通知することで実現する。

この通知の方式として、(1) インスタンス変数領域へのアクセスで自動的にサーバオブジェクトと通信する機能をプラットフォームが提供する、(2) ソースコード上でインスタンス変数をアクセスするコードの直後でサーバオブジェクトと通信するコードを生成する、の2方式がある。(2) はソースコードを並行オブジェクト指向言語 ACOOL [3] で記述すると、図3の下線部に当たるコードを言語処理系により追加することを意味する。

```

MY.x := y ; /* インスタンス変数への代入 */
Server <: msg(MY.x) ; /* サーバへ通知 */
    
```

図 3: ソースコード上での内部状態処理

(1) と (2) では、その柔軟性から、初期段階としては (2) を採用する。(2) では、複製管理サーバオブジェクトへの通知のタイミングを変化させることが容易であり、それを使った最適化方式の検討も行なえる。また、オブジェクトクラス毎にその最適化方式を適用することも容易である。

4 まとめ

通信網ワイド分散システムにおいて、高信頼オブジェクトの構造と、オブジェクトの複製をプラットフォームで管理する方法について述べた。また、高信頼オブジェクトの生成法について検討し、導入オブジェクトを用いる方法を提案した。さらに、高信頼オブジェクト生成に関して、導入オブジェクトと複製オブジェクトのソースコード生成法について述べた。これらにより、高信頼オブジェクトの効率の良い実行環境を提供できる。今後は、言語処理系の実装と複製管理サーバオブジェクトとプラットフォーム内メッセージ管理機構の実装を行ない、本方式が有効であることを示す。

参考文献

[1] 久保田 稔, 丸山 勝己, “通信網のための分散処理プラットフォーム,” 信学論 B-I, no. 5, pp. 301-309, Mar. 1996.
 [2] 武本 充治, “通信網ワイド分散オブジェクト指向システムにおける高信頼化,” 信学技報, SSE95-177, 1996.
 [3] 丸山 勝己, “分散並行オブジェクト指向言語 ACOOL の実装,” 情処学論, pp. 207-217, 1995.