

分散システムにおける協調動作の設計手法

3R-1

前川佳春, 野田夏子, 池田健次郎, 岸知二
 NEC マイコンソフト開発環境研究所

1 はじめに

分散システムが様々な領域で重要になっている。そのベースとしてORBのような分散オブジェクト基盤が注目/利用されている。しかしながらオブジェクトの配置などに起因する性能問題を考慮しながら、オブジェクトの協調動作を設計することは必ずしも容易ではない。本稿ではそうした問題を改善するための開発アプローチや支援環境について提案する。

2 開発アプローチのねらい

分散システムの開発においては並行に動作する分散オブジェクトの協調動作を設計する必要がある。そのためには論理的なふるまいだけでなく、物理的なオブジェクトの配置や用いる基盤の特性などに起因する性能問題に対する考慮が必要になる。しかしながらこれらを同時に設計することは困難である。

本アプローチでは、これらの問題を分離して設計することでこれを解決しようとしている。

- 論理的なふるまいについては、ある程度抽象度の高いレベルでシミュレーションを行って確認する。
- 性能については、実証的な確認が必要なので、実際にコードを生成して実行確認する。

しかしながら、一般的にシミュレーションのためのモデル構築コストは大きくなりがちである。そこで、本アプローチでは設計モデルからシミュレーションと実際のコードを生成することでこの問題に対処している。

“Design Method of Collective Behavior for Distributed System”, Yoshiharu MAEGAWA, Natsuko NODA, Kenjiro IKEDA and Tomoji KISHI, NEC Corporation

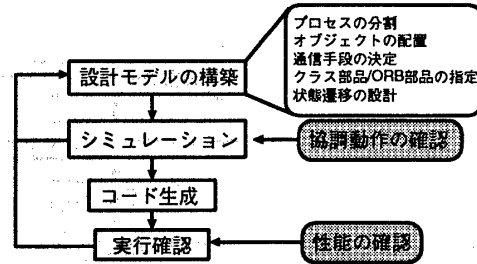


図1: 分散システムの協調動作設計

3 協調動作の設計

3.1 設計モデルの構築

設計モデルは、拡張オブジェクト図とステート図により構築する。

● 拡張オブジェクト図 [1]

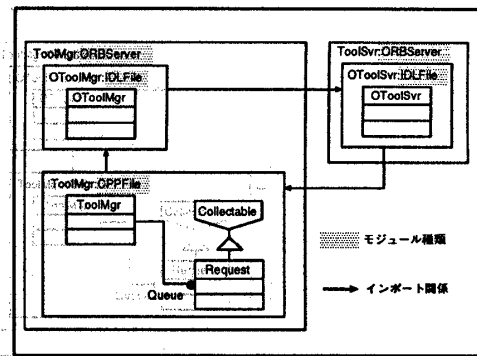


図2: 拡張オブジェクト図の記述例

図2において、プロセスの指定はプロセスモジュールで、ORBオブジェクトの指定はIDLファイルモジュールを設計モデル上で指定することで実装上の決定を行っている [2]。また、ファイルモジュール間の矢印線によりモジュールのインポート関係を指定している。

● ステート図

ステート図のガードやアクション部分はマクロ的な記述と C++ のコードにより記述する (図 3)。ORB オブジェクトのメソッド呼出しなど基本的な処理についてマクロ記述を用意している。例えば ORB オブジェクトのメソッド呼出しは、“SEND メソッド名 TO ORB インスタンス名” という形式で記述する。

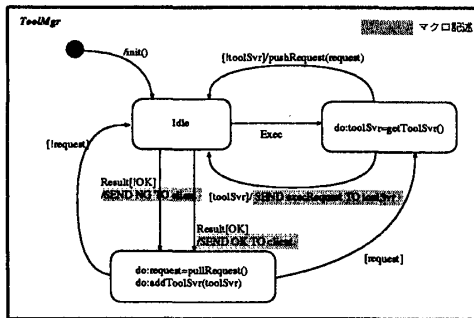


図 3: ステート図の記述例

3.2 シミュレーション

設計モデルからシミュレータ用コードを生成して、シミュレーションを行う。シミュレーションでは、設計モデル上で指定したプロセスを並行動作の単位とみなし、抽象度の高いモデルとして解釈して実行し、論理的な協調動作を確認する。本シミュレータでは、

- コード生成と同様のモデルを利用し、シミュレーション用のモデルはあえてつくらない。
- シミュレーション時に確認したい動作シナリオやインスタンスの初期設定などを定義ファイルで指定する。

ことにより、確認したいポイントに関わる動作モデルを短時間に構築して動作を確認することができる。また、シミュレーション結果を事象トレース図として表示することで、オブジェクトの状態遷移やオブジェクト間のメッセージ送受信をより視覚的に確認することができる。尚、シミュレーションと生成コードでの基本概念の対応は表 1 のようになる。

表 1: 基本概念の対応関係

設計モデル上の概念	シミュレーション	生成コード
プロセス	並行動作単位	並行プロセス
ORB オブジェクト	オブジェクト	ORB オブジェクト
同期通信	同期型シグナル送受信	同期型 ORB メソッド呼出し
非同期通信	非同期型シグナル送受信	非同期型 ORB メソッド呼出し

3.3 コード生成・実行確認

設計モデル上の C++ ファイルモジュールからは C++ クラス定義と C++ 実装スタブコード、IDL ファイルモジュールからは IDL 定義と IDL 実装スタブコード、ステート図から状態遷移実装コード、モジュールの包含関係およびインポート関係からは Makefile がそれぞれ生成される。必要に応じて生成されたスタブコードにユーザコードを追加する。ユーザコードは、コードの再生成を行っても自動的に生成コードにマージされる。

生成されたコードを用いて分散環境上で実際の動作や性能の確認を行う。性能が出ない場合は、設計モデル上でオブジェクトの配置を変更後、コードを再生成して再度確認するというサイクルを繰り返す。これにより、適切な設計モデルを作成する。

4 おわりに

プロトタイプ開発を通して、設計モデルのシミュレーションや実行可能コードの生成により ORB を用いた分散システムにおける協調動作の設計支援ができることを確認した。

今後の課題としては、シミュレーションのためのモデル構築コストの改善や本アプローチの開発プロセスでの位置付けの明確化が挙げられる。

参考文献

- [1] 野田夏子, 他: オブジェクト指向設計を支援する設計モデルからのコード生成機能. ソフトウェア工学研究会, (1996).
- [2] 岸知二, 他: オブジェクト指向設計のためのソフトウェアアーキテクチャモデル. FOSE'95, (1995).