

緩やかな分散オブジェクト連携モデル: Field-Reactor Model

1H-5

-局所パラメータとオブジェクト連携パターン-

岩尾 忠重 岡田 誠 高田 裕志

富士通研究所 ネットメディア研究センター

iwao@flab.fujitsu.co.jp

1. はじめに

インターネットの普及やモバイルユーザの増加により、分散オブジェクトにおける実行環境自体は多様化し、また動的に変化するような状況が今後ますます増えると予想される。しかし、分散オブジェクト間の連携をいかに動的に構築していくかという問題は未だ解決されていない。オブジェクト同士が相互に作用しながら動的かつ柔軟に連携するための緩やかな連携基盤とその連携の仕組みが必要である。分散オブジェクトの動的連携の構築は、密なオブジェクト連携で行うことは非常に困難であり、単純にアプリケーションをコンポーネント化していくだけでは解決できない。

そこで本論文では、分散オブジェクトの緩やかな連携の基盤としてのField-Reactor ModelとそのOpen性について述べる。さらに、このモデルに契約ネットの概念を導入することにより、1つの局所パラメータの操作でオブジェクト間全体の連携パターンを動的に変化させることが可能であることを報告する。

2. Field-Reactor Model

Field-Reactor Modelは、メッセージとメッセージ反応パターンにより分散オブジェクト間の緩やかな連携をサポートするためのオブジェクトコミュニケーションモデル[1]である。このModelはFieldと呼ばれるメッセージ通信基盤とメッセージ反応パターンテーブルを持つReactorオブジェクトによって構成される。

Field-Reactorの特徴を以下に示す。

1. ReactorはFieldを流れるメッセージを感知する能力を持つ。
2. Fieldを流れるメッセージは、そのFieldにいるReactor全員に到達する。
3. FieldとReactorは独立しており、お互いに生成消滅に影響しない。

Fieldは概念的なメッセージ通信基盤である。Fieldは、音や光の場合と同様に、基本的に常時感知可能であると同時に、Field上のメッセージを周辺に伝達するメディアとしての特性を持つものである。またFieldはさまざまな物理的なメディアとその性質を含めて統合的に扱うことを可能としている。

ReactorはField上のメッセージを感知できるオブジェクトであり、独立かつ自律的に動作するためのメッセージ反応パターンテーブルを持つ。我々はこれをReactorテーブルと呼んでいる。Fieldを流れるメッセージは、Tuple形式をとり、Reactorテーブルには、メッセージのTuple形式に対するマッチング

パターンとそのメッセージによって発火される関数がペアで書かれている。

Reactorは、オブジェクト単位でReactorテーブルを書き換えることができる。これは同じメッセージを受信した場合、同一クラスであってもインスタンスにより、異なる動作を許すものである。

Field-Reactorの持つOpen性

FieldとReactorを同時に利用することで、以下の4つのOpen性を得ることができる。

- A) オブジェクトの種類Open性
- B) パラメータ出現のOpen性
- C) オブジェクトの独立処理性のOpen性
- D) オブジェクトの独立存在のOpen性

A)-C)については、述べている[2]。(D)の独立存在のOpen性は、それぞれのReactorがFieldにより隔離されていることにより、同一Field内の他のReactorの生成・消滅による影響は全く受けないというOpen性である。

これらのOpen性から、(A)Field-Reactorはどのようなオブジェクトでも、(D)いつでも自由に参加、不参加でき、他のオブジェクトには参加・不参加による影響を与えず、(B)メッセージパターンマッチによる柔軟性と(C)他のオブジェクトから独立したメッセージの反応による連携を支えるプラットフォームとすることができる。これらの特性は、柔軟性、スケーラビリティ、ロバスト性を生み出し、このモデル上で構築されたシステムは柔軟性などの特性を得ることができる。

3. Field-Reactorによる契約ネット

Field-Reactor Modelは、プラットフォームとしては十分に柔軟で拡張性にも優れているが、それぞれのオブジェクトの連携方法については、フレームワークを与える必要がある。

分散問題解決の1つの手法として、契約ネットがある。契約ネット[3]は、タスクの提示、入札、落札の決定、落札の通知という手順をとる。この契約ネットでは、提示されたタスクに対して、どのサービスプロセスが処理を行うかは、予めプログラムされているわけではない。タスクが発生した時点で、そのタスクを実行するサービスプロセスを広く公募し、動的に決定する。

動的に連携の相手を決定するポリシーは、緩やかな連携を行う上で非常に重要な要素である。オブジェクトの実行環境やそのときのネットワークや他のオブジェクトの状況というのは時々刻々と変化している。このような環境下では、密な連携のように最初から通信相手や種類を特定しては、連携を行うことは難しく、予め非常に慎重な設計を要する。うまく設計されたとしても、設計時の予測の範囲を超えて連携することはない。

タスクが生じたときに動的に連携相手を決定する手法は、タスクを要求した時点で実行可能なオブ

ジェクトが自己申告するので、実行環境、ネットワークやオブジェクトの状況に応じた連携を可能とする。Field-Reactor Model は前述したように Open な環境を提供する。そのため Field-Reactor Model で契約ネットを用いて動的に連携相手を決定する場合、設計時に予期していないオブジェクトとの連携も可能となる。また、(D)の独立存在の Open 性からサービスオブジェクトを生成した直後に連携がなされたり、何らかの理由により消滅したとしてもシステム全体への影響はない。

Field-Reactor Model では、Field に対する Tuple メッセージ形式とそのメッセージパターンによる反応により処理が進む。したがって、特定のタスクやサービスでなくても同じメッセージの連鎖の利用が可能である。Field-Reactor Model に契約ネットの概念を導入する場合、図 1 のように複数のタスク生成オブジェクト、複数のサービスオブジェクト、調停オブジェクトが Field 上にあるという構成になる。タスク生成オブジェクトやサービスオブジェクトというのは、メッセージの内容によって決まるので、あるオブジェクトはタスク生成オブジェクトとサービスオブジェクトに同時なりうる。

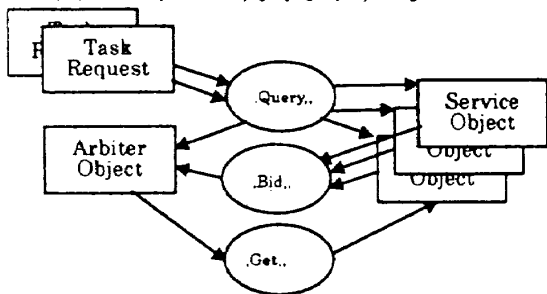


図 1 Field-Reactor による契約ネット構成図

表 1 メッセージタイプ

メッセージパターン	説明
<TaskID, "Query", Service type>	タスクの提示
<ServiceID, "Bid", TaskID, weigh>	入札
<ServiceID, "Get", TaskID>	落札

表 2 調停モジュール Reactor テーブル

T1	T2	T3	Function	備考
*	Query	*	Queue	内部キュー登録
*	Bid	*	Decision	落札者の決定

表 3 サービスモジュール Reactor テーブル

T1	T2	T3	Function	備考
*	Query	*	Bid	入札
ServiceID	Get	*	Serve	サービス実行

Field-Reactor における契約ネットの動作の概要を説明する。

1. タスク生成オブジェクトはタスク要求 ID とサービス種類付きのメッセージを Field へ流す。
- 2.a. 調停オブジェクトは、タスク要求メッセージを内部キューに保存する。(表 2)
- 2.b. 各々のサービスオブジェクトは、自オブジェクトを示すサービス ID と自オブジェクトの内部パラメータに応じた重みを負荷した入札メッセージを、Field に送出する。(表 3)
3. 調停モジュールは、入札メッセージに対して反応し、内部キューを照合し、調停モジュールの決定ポリシーによって、落札サービスオブジェクトを決定し、調停オブジェクトは落札メッセージをタ

ク ID とサービス ID を付けて送信する。

4. サービス ID を持つサービスモジュールが落札メッセージに反応し、サービスを提供する。

4. オブジェクト内部パラメータと連携パターンの変化

負荷の分散を行う際には、負荷を入札メッセージの重みとし、決定ポリシーを入札の重みの最小値とすることによって、負荷の分散・平均化を行うことができる。また、入札メッセージの重みを落札回数などにし、決定ポリシーを入札の重みの最大値とすることにより、負荷を集中させることもできる。

つまり、システム全体の動作は、決定ポリシーを入札時の重みの最小値とすると、分散・平均化となり、最大値とすると、集中化させることができる。これらを組み合わせることは容易であり、また、決定ポリシー自体をプログラム化しシステム全体の状態の制御も可能である。Field-Reactor Model は、オブジェクトの種類や数に対して Open な系であるため、契約ネットとの組み合わせにおける決定ポリシーによる動作変化は、すなわち Open 系の制御に他ならない。システムの状態空間を全てのサービスオブジェクトの状態とし、入力をタスクの要求とする状態遷移である。

このように考えると分散、集中、プログラム化は以下のようにみなすことができる。

1. 分散・平均化は、サービスオブジェクトの状態を平均化・均一化する方向で遷移させている。
2. 集中は、Open な状態の中の選択された状態を遷移することになる。各サービスオブジェクトが落札回数を入札の重みとした場合、状態遷移の学習になる。つまり、Open な系での経験的に状態遷移の学習を行うことが可能となる。
3. プログラムにより状態の遷移制御は、ある程度状態が分かっており、意図的に遷移をさせたい場合に適する。設計時予期していない状態にある場合でも、システムとして機能させることができる。

さらに言えば、Field-Reactor Model の持つ Open 性から調停オブジェクトは任意のサービス要求メッセージに対し反応するため、サービスの種類に応じて調停オブジェクトを変える必要はない。また、サービスオブジェクトを動的に投入した瞬間から連携の対象となる。さらに、この調停モジュールを動的に任意に投入できるため、システムを動作させたまま動的に、システム全体の動きを変化させることも可能である。

5. まとめ

本論文では、緩やかな分散オブジェクト連携のための基盤として Field-Reactor Model について説明し、その連携の方法として、契約ネットを導入し、局所パラメータによる連携パターンの制御について述べた。

参考文献

[1] 岩尾忠重, 岡田誠, 竹林知善: 第55回情処全国大会, 1996, 3T-1, pp659-660
 [2] 岩尾忠重, 岡田誠, 竹林知善: 情処研報1998 DPS86-3, pp13-18
 [3] R.G.Smith: IEEE Trans. Comput., Vol. C-29, No. 12, pp. 1104- 1113, 1980