

## Flage アーキテクチャにおける記述モデル

4N-7

来間 啓伸<sup>1</sup>

松浦 佐江子<sup>2</sup>

本位田 真一<sup>3</sup>

情報処理振興事業協会 (IPA)  
新ソフトウェア構造化モデル研究本部

### 1. はじめに

計算機がネットワークで結ばれることにより、ソフトウェア・モジュールがネットワークを介して協調し、一つのシステムとして機能することが可能になった。このようなシステムでは、システムを構成する各要素が常に変化する可能性を持つため、環境の変化に影響されないあるいは環境の変化に柔軟に適應するソフトウェア・アーキテクチャが求められる。

Flage は協調型ソフトウェア・アーキテクチャの記述言語であり、エージェントと場の概念に基づいて、環境の変化に柔軟に適應するソフトウェアを記述する枠組みを提供する。エージェントは環境の変化に適應して機能を変える実行主体であり、場は特定の環境の下でエージェントが実行する機能とその制約を表現した非実行主体である。Flage では、エージェントは場を渡り歩くことによって動的に機能を獲得し、環境の変化に適應する。

本稿では、Flage を用いてソフトウェアを記述するための記述モデルについて述べる。

### 2. 基本構成

Flage の記述モデルの概略を図1に示す。ここで、対象の各要素と Flage の言語要素を以下のように対応付ける。

- ベースレベル  
エージェントが様々な状況の下で共通に実行する機能の記述
- メタレベル  
エージェントや場との協調に関する機能の記述
- 場  
特定の状況下でベースレベルが持つべき機能の記述

エージェントがある場の中にあるとき、ベースレベルの機能はベースレベルのメソッドと場のメソッドによって規定される。したがって、場の移動を行わない範囲では、ベースレベルの機能はメタレベルからの解釈によって決定される。メタレベルはベ-

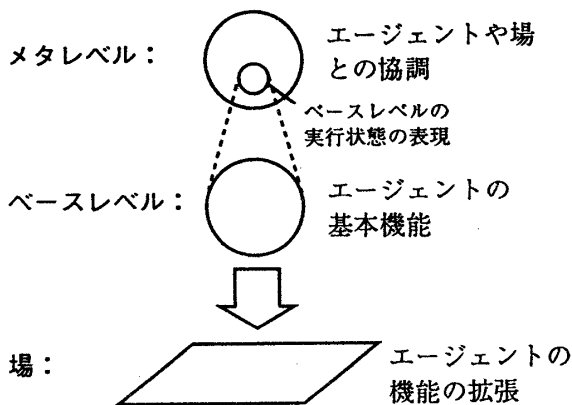


図1: 記述モデルの概略

ースレベルの実行状態の制御とベースレベルのメソッド獲得に関する記述であり、エージェントの動的な変化の側面を担う。場は、ある限られた状況のもとでエージェントに提供される機能の記述である。また、エージェントは場からメソッドを獲得するだけでなく、場にメソッドを書き込むことができる。この結果、エージェント間で場を介したメソッド伝達を行なうことができる。

エージェントが環境の変化に適應するために移動すべき場は、場の属性またはメタレベルの属性に記述するか、あるいはエージェント間のメッセージ授受によって渡すことができる。どのような場合にどのような渡し方が有効であるかについては未解決の問題も多いが、このような構造はエージェントの協調の面からも十分強力であると思われる。例えば、あるエージェントに付近の場の属性を書き換えて回らせることにより、その場に入った他のエージェントを指定した場に誘導し、誘導先の場でメソッドを獲得させることができる。

### 3. エージェントのふるまい

エージェントのふるまいは、ベースレベル、メタレベル、場の各メソッドによって規定される。これらは、以下のように特徴付けられる。

- ベースレベルのメソッド  
エージェントの基本機能の記述
- メタレベルのメソッド  
エージェントの機能の動的な変化の記述
- 場のメソッド

Field Based Modeling for Flage Architecture

Hironobu Kuruma, Saeko Matsuura and Shinichi Honiden  
Laboratory for New Software Architecture,  
Information-technology Promotion Agency (IPA)

<sup>1</sup> (株) 日立製作所より出向

<sup>2</sup> (株) 管理工学研究所より出向

<sup>3</sup> 現 (株) 東芝

特定の状況下で拡張, 制限, 共有される機能の記述

ベースレベルのメソッドと場のメソッドがエージェントの機能を規定するのに対し, メタレベルのメソッドは機能の変化を規定する。それには, 次の2つの方法がある。

1. 場の移動によるメソッド獲得
2. ベースレベルのメソッドに対する解釈の変更

場のメソッドは, エージェントのベースレベルのメソッドを拡張する一方, 既に同じメソッドがある場合にはそれを隠蔽する。また, 場のメソッドは場内のエージェント全てに対して同様に作用するので, メタレベルからの解釈の変更が加わらない限り, 共通の機能を与える。

#### 4. 記述ステップ

エージェントのふるまいの点から, Flage の記述ステップの概略を示す。

1. エージェントと場を分ける。

対象において, 自己の状態を変化させる主体をエージェントとし, エージェントのふるまいを規定する環境を場とする。エージェントのふるまいを規定する環境には例えば次のようなものがあり, 場として表現することができる。

- (a) ふるまいの対象となるデータおよびその操作
  - (b) エージェントの属性に依存した状況におけるふるまいの定義
  - (c) エージェントのふるまい間の関係 (同期関係など)
  - (d) エージェント間の交渉の定義
2. エージェントが環境によらず果たすべき機能を抽出し, ベースレベルのメソッドとして記述する。
  3. エージェントが特定の環境の下で果たすべき機能を抽出し, 場のメソッドとして記述する。

上記の各環境は, Flage では次のように表現することができる。

- (a) エージェントのふるまいとは分離可能なデータおよびその操作を場の属性・メソッドとして記述する。
- (b) エージェントの属性に対する制約を場の出入りに関する制約とし, その状況におけるエージェントのふるまいを場のメソッドとして記述する。
- (c) 場へのアクセス権を場の出入りに関する制約により定義する。

- (d) メッセージに対応するメソッドを記述する。

4. 状況の判断と機能の獲得方法を規定し, メタレベルのメソッドとして記述する。

メタレベルメソッドでは, 次のような記述を行なうことができる。

- 場への移動
- 場への出入りによるメソッド・属性の獲得
- ベースレベルメソッドの実行
- エージェントのふるまいを喚起するためのメッセージの送信
- 場への出入り制約を満たさなくなった時の続行処理

#### 5. おわりに

Flage の記述モデルについて述べた。Flage の基本的な言語要素はエージェントと場であり, 各エージェントは, ベースレベルとメタレベルの2階層の記述から構成される。本稿のモデルでは, エージェントが様々な状況下で共通に実行する機能をベースレベルに, 特定の状況下でエージェントが持つべき機能を場に, エージェントや場との協調に関わる機能をメタレベルに分離する。このように, ベースレベル, メタレベル, 場に記述を分離することにより, 一般には扱い難い動的な機能獲得を比較的容易に記述し, 変更することができる。

Flage におけるエージェントの協調には, 次の3つの形態がある。

- エージェント間のコミュニケーション
- 場からのメソッド獲得
- 場を介したメソッド伝達

本稿のモデルはこれらを記述するためのアウトラインを示すものであり, 今後は例題記述を通じて記述モデルの詳細化を進める。

#### [謝辞]

本研究は, 産業科学技術研究開発制度「新ソフトウェア構造化モデルの研究開発」の一環として情報処理振興事業協会 (IPA) が新エネルギー・産業技術総合開発機構から委託を受けて実施したものである。

#### 参考文献

- [1] 桑野ほか: Flage アーキテクチャの構想, 第51回情報処理学会全国大会 (1994)
- [2] 来間ほか: 協調型ソフトウェア・アーキテクチャに基づく開放型システムの仕様記述モデル, 情報処理学会研究報告ソフトウェア工学102, pp. 135-140 (1995)