

エージェント指向言語Flageによるソフトウェア部品の再利用

5C-5

佐藤 仁孝¹ 加藤 哲男² 糸野 文洋³ 永井 保夫⁴ 本位田 真一⁵

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

本研究の目的は、環境の変化に自律的に適応するソフトウェアを構築するための枠組を提供することにある。我々は、エージェントと場という概念を持つ言語Flage(Flexible Agents)を提案し、処理系を開発してきた[1]。Flageのエージェントは、ソフトウェア部品(メソッド粒度の機能)を動的に変更可能なモバイル・オブジェクトである。場はソフトウェア部品の保管場所であり、エージェント間で共用される。そして、ネットワーク上に分散し局所的に生成・管理される。エージェントは場を移動し、場に入ると自動的にソフトウェア部品を組み込み、挙動を変更する事ができるため、変更要求に対し自律的に適応するシステムを構築する事ができる。本稿では、Flageの動的な適応機能を活用する例として、分散システム上のデータ連携処理において、データとデータ変換メソッドの不整合を自律的に解消するシステムを提案する。さらに、ソフトウェア部品の再利用における柔軟性に関し考察する。

2 メタ構造

動的にソフトウェア部品を獲得する場合、既存機能との間に不整合が生じることが懸念される。Flageは不整合の回避・検知・解消のための言語機能を備える[1]。不整合の回避・検知・解消を自律的に行なうため、エージェントはメタ構造(二層構造)を持つ。変更可能な定義をベースレベルとし、変更を制御する(適応処理を行なう)定義をメタレベルとする。メタレベルは、ベースレベル定義をテスト実行(疑似実行)できる他、場に記述された不整合の回避・検知・解消

のための情報を利用し[2]、場の移動を制御することができる。その際、エージェント間での情報交換を行ない協調作業による問題解決を図ることも可能である。メタレベルで、適応のための処理を記述するため、ベースレベルでの、ソフトウェア部品の動的かつ柔軟な再利用が可能な枠組であると考えられる。

3 柔軟なデータ連携

応用例としてイントラネットにおける情報配信サービスを考える。(図1)。汎用的なサービスを提供するためには、受信側クライアントの多様性に対処する必要がある。計算機・OSの種類だけでなく、CPUパワー、ネットワーク帯域幅、利用メモリ、ユーザインタフェースの表現力、配信情報を二次利用するアプリケーションなどの多様性が考えられるが、Flageエージェントはクライアントの状況に応じた情報を提供することができる。

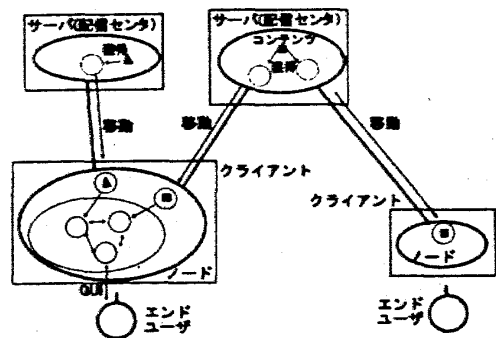


図 1: エージェントによる情報配信

配信エージェントはネットワーク上を移動しクライアントの場に到着すると、クライアント側の既存のエージェント(利用側エージェント)と相互作用を行うことで、クライアントに依存する処理(パーソナライズされた情報フィルタリングなど)を行う。

ここで、配信エージェントは到着時に初めて、利用側エージェントが受け入れ可能なデータ形式を知る場合を想定する。Javaなどは、ネットワーク経由でソフトウェア部品(プラグイン)をダウンロードして対応するが、配信サーバ側にプラグインのリポジトリが整備されている必要がある。クライアント側の要求形式が

Reuse of Software Components in Agent-Oriented Language Flage
Hiroataka Sato, Tetsuo Kato, Fumihiko Kumeno,
Yasuo Nagai, Shinichi Honiden

¹(株)アイネスより出向中
²(株)富士通FIPより出向中
³(株)三菱総合研究所より出向中
⁴(株)東芝 研究開発センター S & S 研究所より出向中
⁵現 (株)東芝 研究開発センター S & S 研究所

多様であったり、配信データの内部形式のバージョンアップが頻繁であったりすると、リポジトリの管理は面倒なものとなる。われわれの提案する情報配信システムは、配信エージェント自らが、ネットワーク上を移動し、要求と合致するソフトウェア部品を入手することで適応する。新たに獲得しようとするソフトウェア部品とエージェントの内部定義との間に不整合が生ずる場合でも、ダウンロードする方式に比べ、状況に応じた適応処理を試みることができる。

3.1 エージェントの適応動作

整合するデータ変換メソッドを獲得し、処理要求に適応する動作は、以下の手順に従う。

(1) 配信エージェントの到着

デフォルトの連携メソッドを獲得する。

(2) 変更要求の検知

配信エージェントは利用側エージェントから、データ出力の要求を受ける。メソッド名(インタフェース)が未知であることから、配信エージェントは変更要求を検知する。

(3) メソッドの探索

メタがベースの実行をサスペンドし、適応行動を開始する。要求を受けたメソッド名(インタフェース)を軸に探索するが、探索に必要な情報は場や他のエージェントから動的に入手できる。

(4) 獲得メソッドのテスト

場間のリンク情報により動的に移動先を決定し、移動を行なう。移動先でメソッドを獲得しテスト実行を行う。テスト実行が成功すれば探索を終了する(→(6))。失敗した場合はベースレベル定義を復旧し、他の探索手段を試す。エージェントはリンクをたどり探索を続けることは可能であるが、活動中の他エージェントから支援を受ける方法も考えられる。(→(5))。

(5) メソッド探索に他のエージェントが支援する。

同じ内部データ形式を持ち、求める変換メソッドを持ったエージェントがネットワーク上のどこかで活動中であると仮定する。こうした他のエージェントと協調する方法である。探索エージェントは自分の内部データを協調エージェントに転送し、テスト実行を依頼することができる(図2)。これはある特定範囲内へのマルチキャストで依頼する。依頼された側はメタレベルの制御下でテスト実行を行う。依頼されたテスト実行が成功したエージェントを同じデータ形式を持つと判断する。そのエージェントの変換メソッドが存在する場を通知してもらい、メソッドを

獲得する。再度テスト実行を行い成功後、本実行を行なう。

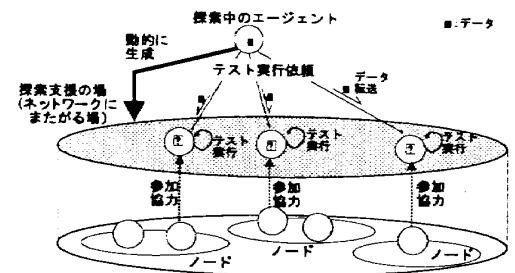


図2: エージェント協調によるメソッド探索

(6) 要求された処理の遂行

ベース実行をレジュームし、処理を続行する。

3.2 考察

Flageエージェントは移動先ノードに関する知識がなくても、最小限必要な機能のみ保持して移動を行い、移動先での連携に必要な機能を自律的に獲得する。モバイルエージェントのプログラム負荷を軽減するのに有効である。

適応行動が完了したエージェントが正当な部品場に公開できる。これは、同じ内部構造を持つエージェントが再利用することを容易にする。

集中的なりポジトリを仮定せず、メタレベル処理を利用してデータ形式とデータ操作メソッドの不整合を自律的に解消するため、再利用を見越した作り込みを減らす事ができる。これはリポジトリの利用を否定するものではない。補完する役割を担うものになると考える。

4 展望

平成9年12月からFlage処理系のプログラムを無償公開している[3]。実証試験を通し汎用性を評価することが課題である。

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うけて実施したものである。

参考文献

- [1] Kumeno, F. et al.: *Evolutional Agents: Field-Oriented Programming Language, Flage*, Proc. of ASPEC'95, 1995.
- [2] Kumeno, F. et al.: *A Framework for Adaptive Software by Agent and Thesaurus*, Proc. of SEKE'97, 1997.
- [3] 情報処理振興事業協会: *Flage*ホームページ, <http://www.ipa.go.jp/NEWSOFT/public/Flage/>, 1998.