OE2-1

インタネットアプリケーション開発のためのエージェント型開発ツール
Agent-based Tool for Developing Internet Applications

菅原研次†，藤井良知††，原英樹†　　杉山達彦,麻生紀雄†††
Kenji Sugawara, Yositomo Fujii, Hideki Hara Tatsuhiko Sugiyama and Norio Asou

1．まえがき

インターネットの普及により，ビジネス活動から日常生活までのさまざまな社会活動を支援するための分散アプリケーションの開発が必要となっている．このようなソフトウェアに対する要求は高度化する一方であり，たとえばデジタルデバイドを解消するための知的インタフェース機能，利用者のシステム利用に関する知識を補い，労力を軽減するためのエージェント機能，情報の流れを効率化する知的情報フィルタ機能などさまざまな機能の実現が求められている．このような分散システムを効率的に実現する開発パラダイムとしてオブジェクト指向方法論が提案され，さまざまな開発環境の実用化が進められている．しかしながら，前述の分散システムに求められている，知性，自律性，柔軟性，安全性などの機能の実現を目指す上で，オブジェクト指向パラダイムだけでは十分ではなく，新たにエージェント指向パラダイムの確立を目指した研究が進められ，新しいモデルの提案が行われてきた．しかしながら，これらのモデルを用いたアプリケーションの開発は期待しているほど進展しておらず，エージェント指向プログラミングの普及に向けたモデルと言語の改善が必要となっている．

本発表では，エージェント指向プログラミングの普及を目指して，初学者の教育用言語として開発した言語ＴＡＦと，その技法をベースにした，インタネットアプリケーション開発のためのエージェント指向プログラミング言語Eagleのデモンストレーションを行う．このデモンストレーションは，東北大学木下研究室と共同で研究を進めているエージェントフレームワーク開発研究の一部である．

2．エージェント指向プログラミング教育用言語 TAF

初学者が，エージェント指向プログラミングの実習を行うことを目的としたツール TAF(Traininng tool for Agent Framework)を Java 言語を用いて開発した．PC/WS 上で簡単に利用でき，大学学部あるいは大学院の演習授業で使っている．TAF は以下の指針で設計された．

(1)エージェントモデルとしては単純な構造にする．
(2)協調プロトコルなどエージェント間プロトコルの実装が簡単にできる．
(3)エージェントのライフサイクルなど，エージェントシステムの開発法を理解するための概念が簡単に修得できる．
(4)エージェントの動作が視覚的に把握できる．
(5)メンタルモデルなど，知的機能の実現法が修得できる．

この，条件を満足するエージェントプログラミング言語の形態として，図１に示すようなプロダクションモデルを採用した．この理由として，(a)エージェント間プロトコルを状態遷移機械モデルを利用して設計することを前提にして，これを実装するときに，ルール型言語が比較的容易にプログラムできること，(b)メンタルモデルなどの実装が比較的簡単に行え，知的エージェントの設計から実装までが半期の実習科目で修得できること，などが挙げられる．

また，リポジトリをベースにした簡単なエージェントシステム開発フレームワークが提供されており，再利用をベースとした開発支援環境を利用する手法と概念を修得できる．初学者は，既存エージェントを利用しながら，(1)エージェントの協調モデル設計(2)エージェントのプログラミング(3)エージェントデバグ作業(4)エージェントシステムデバグ作業(5)エージェントシステムメンテナンス作業，の実習を進めることができる．

3．エージェント指向分散アプリケーション開発用言語 EAGLE

3.1　言語の設計指針

EAGLE(Extensible AGent Language and Execution platform)は，TAF と同じエージェントモデルと言語を採用し，実用的なインターネットアプリケーションソフトウェアをエージェント指向モデルを利用して開発することを目指して設計を行った．現在，プロトタイプを作成し，エージェント指向システム開発の普及用言語としての十分な能力があるかどうかの実験を行っている．この実験から，不足している機能や性能の洗い出しを行い，改良を進めていく予定である．

Eagle は分散環境で動作する Java プラットフォームの上でマルチエージェントシステムを実装するための言語であり，以下の特徴を持つ．

(1) ルール形式の言語で協調知識を記述する
(2)プラットフォーム（コンピュータ，ネットワーク）を制御管理する機能を記述する
(3)知識のモジュール化機能があり，状況に合わせて知識モジュールの切り替え制御が可能
(4) 知識モジュールを動的にエージェントに取り込むことができる
(5)インターネット上で動作する移動エージェント機能を持ち，名前の一意性が確保され，ファイアウォールを越えてメッセージ交換が可能である．
(6)マルチエージェントシステム内の協調プロトコルと，マルチエージェントシステム間の協調プロトコルがサポートされている．

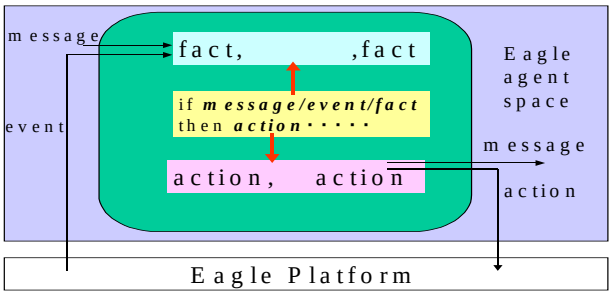

図1．TAF

†千葉工業大学情報科学部　††千葉工業大学情報工学専攻
†††㈱トレジャーコミュニケーションズ

3.2 動作環境

Eagleを用いて開発する対象は，利用者あるいは他のシステムにサービスを提供するマルチエージェントシステムであり，これをEAS(Eagle Agent System)と呼ぶことにする．設計指針(6)はEAS内協調プロトコルとEAS間協調プロトコルがEASの動作環境で提供されていることを目的としている． EASは，Javaで実装されるEagle Virtual Machineの上で動作する．EAS は，利用者や他のシステムにサービスを提供することを目的とするエージェントシステムであり，そのためにEAS同士は協調や交渉を行うことが必要になる．すなわち，インターネット上のEASの全体は人間のように社会を構成し，この中でさまざまな活動を行う．このようなEAS社会をEAST(EAS Town)と呼ぶことにする．インターネットに分散するEVMの全体は図２のようにEASTを動作させるためのインフラストラクチャであり，これにEASTを支援するためのサーバを加えたものをEAST実行環境と呼ぶことにする．

サーバとして，(1) ネームサーバ(2)コミュニケーションサーバ(3)知識モジュールサーバが提供される．

3.3 エージェントの構造

エージェントは図３に示すように，３つのモジュールから構成される．通信モジュール CM はメッセージの送受信と，イベントの受信を行い，これらをバッファに蓄積し．知識モジュールKMで処理するためのエージェントメッセージ形式とファクト形式を内部データ形式に変換する．KM は Eagle を用いてプログラムされたエージェントの知識を蓄積し，外部のメッセージやイベントをファクトとして入力し，これをルールで記述された知識を用いて処理し，エージェントの動作を決定するためのモジュールである．AM は，決定されたアクションを実行するプログラムを蓄積し実行するモジュールである．

3.4 エージェントの知識のモジュール化

Eagleで記述されたプログラムの実行において，プロダクションサイクルが処理速度のボトルネックになるため，この高速化に工夫が必要になる．プロダクションサイクルに時間がかかる大きな原因に，ルールの数とファクトの数が増えたときの条件部のユニフィケーション処理がある．このボトルネックを解消するために，ユニフィケーションの対象を最低限必要なルールの集合と最低限のファクトの集合に限定して，プロダクションサイクルを適用する事が考えられる．このために，ルールセットという概念を導入する．図４に示すように，エージェントプログラムagpは，ルールセットフォルダーRSFとWMフォルダーWMFから成る．

エージェントプログラムagpは，エージェントが起動されるとKMに読み込まれる．ルールインタプリタは，ルールセットのポインタとWMのポインタを有しており，このポインタで指示されているものを，それぞれアクティブルールセット，アクティブWMと呼ぶ．ルールインタプリタはアクティブルールセットとアクティブWMに対してプロダクションサイクルを適用する．アクティブルールセット，アクティブの切り替えは，アクションによるポインタ操作により行う．

エージェントは使う可能性のあるすべてのルールセットを常時保持しておく必要はない．状況に応じて必要なルールセットを動的にインクルードする機能があれば，ルールセットを読み込むルールを記述しておくことにより，プログラム記述の簡素化を図ることができる．また，共通に使われる知識は，ライブラリとして各エージェントで共有することがエージェントシステムの効率的な実現に有効である．さらに，経験のエージェント間共有のためなどの操作を行うことができる．

4. まとめ

分散エージェントシステム開発用ツールの評価実験として，エージェント型のコミュニティ支援システムを開発中である．この評価実験を通して，言語の改良を進めていきたい．

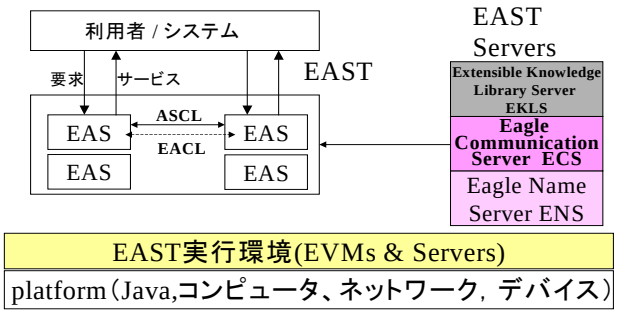

図 2. EAST 実行環境

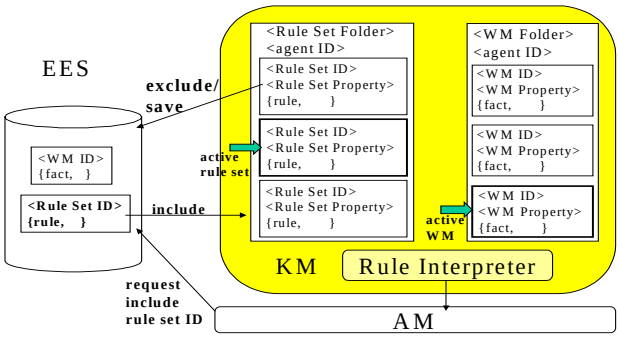

図 3. エージェントの構造

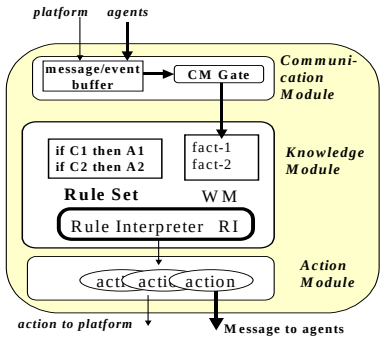

図 4. 知識のモジュール化と動的インクルード