

エージェントを用いた非同期メッセージングシステムの設計と実装

4J-5

関場 治朗[†] 北方 元^{*} 菅沼 拓夫^{*} 木下 哲男^{*} 白鳥 則郎^{*}

[†]東北大学工学部, ^{*}東北大学電気通信研究所 / 情報科学研究科

1 はじめに

インターネットに見られる通信ネットワークの普及により、非同期メッセージングツールとして電子メールは一般に広く利用されてきている。我々は、ユーザーが慣れ親しんだ電子メールのメッセージ機能を最大限に利用しつつ、より高度な非同期型グループウェアを実現することを目標に、エージェント指向コンピューティングを応用したエージェント指向非同期メッセージングシステムを提案している [1]。本稿では、エージェント指向コンピューティングのプラットフォームとして ADIPS フレームワーク [2] を用いた、システムの設計と実装について述べる。

2 エージェント指向非同期メッセージング

2.1 概要

現状の非同期メッセージングにおける問題点として、以下の点が挙げられる。

1. 非同期メッセージングのための専用システムを必要とし、既存のユーザー作業環境との親和性が確保できない。
2. インターネットを介した広域分散環境における利用が困難。
3. 非同期性を十分考慮したメッセージング処理機能が不十分。

以上の問題点を解決するために、エージェント指向の概念を取り入れた、エージェント指向非同期メッセージングシステムを提案する。本システムはエージェントの持つ特性である 1) スケーラブルなシステム構成能力、2) ゴール指向の知的処理能力、3) タスクの永続的処理能力等により、これらの問題点を解決する。

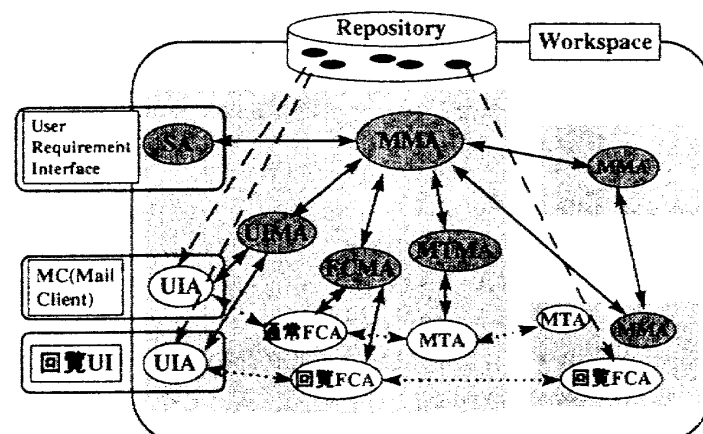
2.2 ADIPS フレームワーク

本研究ではエージェント指向コンピューティングのプラットフォームとして ADIPS フレームワーク (Agent-based Distributed Information Processing System) [2] を用いる。ADIPS は先に述べたエージェントの特性を実現する以下の特徴をもつ、(1) 利用者駆動、イベント駆動で自律的にシステムが構成される、(2) エージェントは各ドメインのシステム設計者・運用技術者の専門知識を利用する、(3) エージェント化により既存プロセスの系統的な再利用ができる。

3 ADIPS に基づく非同期メッセージングシステムの設計

3.1 全体構成

本システムのエージェント構成を図 1 に示す。それぞれのエージェントの機能は次の様に定義される



個人環境 常駐型エージェント 非常駐型エージェント
 ◀ 組織構成に関するメッセージ ▶ コンテンツ、コントロールの流れ

図 1: フレームワーク

SA (Secretary Agent):

ユーザーの要求を受け取る。また、ユーザーのプレファレンスを管理する。

MMA (Messaging Manager Agent):

個人環境上のすべてのメッセージングに関するエージェントの組織を管理する。また、他の個人環境上の MMA と組織管理に関する協調を行う。

UIA (User Interface Agent):

メッセージングの各機能をユーザーへ提供するインターフェース (既存の電子メールクライアントを含む) を制御する。

UIMA (User Interface Manager Agent):

すべての UIA を管理する。

FCA (Flow Control Agent):

メッセージのフローコントロールを制御する。メッセージの回覧を制御する回覧 FCA などがある。

FCMA (Flow Control Manager Agent):

すべての FCA を管理する。

MTA (Message Transfer Agent):

メールの内容の転送、キューイング等の処理を行う。

MTMA (Message Transfer Manager Agent):

MTA を管理する。

これらのエージェント群が協調的に動作することにより、利用者が通常利用している電子メールクライアントを使用した、より高度な非同期メッセージング機能をエージェントが提供することが可能となる。

Design of Asynchronous Communication using Agents

Jiro Sekiba[†], Gen Kitakata^{*}, Takuo Suganuma^{*}, Tetsuo Kinoshita and Norio Shiratori^{*}

[†] Dept. of Technology, Tohoku Univ. ^{*} Research Institute of Electrical Communication/Graduate School of Information Sciences, Tohoku Univ.

3.2 システムの動作例

次に本システム上でのエージェントの動作例を、メッセージング機能の一つであるメッセージの回覧要求が発行された場合の例で説明する。MC(MailClient)は回覧機能を持っていないとする。ここで、 \overrightarrow{TA} , \overleftarrow{ASK} , \overrightarrow{TELL} はADIPSにより定義されたタスク要求、情報要求、情報提示の各フォーマティブを示す。

1. 要求の獲得

ユーザーの要求をSAが獲得する。

2. MMAへの要求伝搬

SA \overrightarrow{TA} MMA

利用者要求及びユーザーのプレファレンスをMMAに伝える。

3. タスク分割

- MMA \overrightarrow{TA} UIMA
- MMA \overrightarrow{TA} FCMA
- MMA \overrightarrow{TA} MTMA

これを受けてMMAはタスク分割を行い、それぞれのManagerAgentにタスクアナウンスを行う。

4. タスク分割

- UIMA \overrightarrow{TA} Repository \Rightarrow MC用UIA
回覧用UIA
- FCMA \overrightarrow{TA} Repository \Rightarrow 通常FCA
回覧FCA
- MTMA \overrightarrow{TA} Repository \Rightarrow MTA

次に、それぞれのManagerAgentはRepository内のクラスエージェントへタスクアナウンスする。ここで、回覧用UIA、回覧FCAが動的にインスタンスエートされることにより、利用者は必要な機能を必要な時に取り出し、利用することができる。もし、MailClientが回覧機能を持っていた場合は、回覧UIはインスタンスエートされること無くMailClientの回覧機能を使うことができる。

5. FCAに関する情報の問い合わせ

MTA \overleftarrow{ASK} MMA

6. 回覧メッセージの送信先及びコンテンツ入力

- ユーザー \rightarrow 回覧用UIA \overrightarrow{TELL} 回覧FCA
- ユーザー \rightarrow MC用UIA \overrightarrow{TELL} 通常FCA
- ユーザー \rightarrow MC用UIA \overrightarrow{TELL} MTA

7. 通常メッセージの送信先をMTAに通知

通常FCA \overrightarrow{TELL} MTA

8. 送信相手先の回覧FCAの生成

回覧FCA \overrightarrow{TELL} MMA \overrightarrow{TELL} 送信相手MMA \overrightarrow{TA}
送信相手FCMA \overrightarrow{TA} Repository \Rightarrow 送信相手回覧FCA

回覧の宛先を聞いたMMAは送信相手先のMMAに回覧FCAのインスタンスエーションを要求する。

9. 送信相手先とネゴシエーション

回覧FCA \overleftarrow{ASK} 送信相手回覧FCA

回覧FCA どうして現在回覧メールを送るべきかどうかを、ネゴシエーションする。もし、相手が長期不在といった場合は別の回覧相手に先に回すなどの処理を行い、可能な限り効率的な回覧を実現する。

10. 回覧メッセージの送信先をMTAに通知

回覧FCA \overrightarrow{TELL} MTA

11. 送信相手MTAの生成

MTA \overrightarrow{TELL} MMA \overrightarrow{TELL} 送信相手MMA \overrightarrow{TA} 送信相手MTMA \overrightarrow{TA} Repository \Rightarrow 送信相手MTA
ここまで、MTAは全ての宛先を知っているの、送信相手先のMMAにMTAのインスタンスエーションを要求。

12. コンテンツ送信

MTA \rightarrow 送信相手MTA

全ての送信相手に送るまで11,12を繰り返す。

13. 終了通知

MTA \overrightarrow{TELL} MMA

送り終わったらMTAがMMAに終了通知を出す。

14. 組織解放

4 システムの試作

3.1節の設計に基づきエージェント指向非同期通信システムを実装した。エージェント指向コンピューティングのプラットフォームとしてはADIPSフレームワークを用い、ベースプロセスはJAVAで実装している。PC上で本システムの試行実験を行った結果、3.2節で示した動作例のように利用者の要求に応じた機能を提供するエージェントがインスタンスエートされ、その電子メールクライアントが保持していないメッセージング機能を利用者に提供することが出来る事が確認できた。

5 おわりに

本稿ではADIPSフレームワークを用いたエージェント指向非同期通信システムの設計について述べた。また、この設計に基づきエージェント指向非同期通信システムを試作した。システムの試作から、エージェント利用により、既存の電子メールクライアントを使用した、より高度な非同期メッセージングが実現可能であることが示せた。今後の課題として、フローコントロールエージェントの追加により、より高度なメッセージングを行えるよう、拡張することが考えられる。

参考文献

- [1] 青木 哲二： エージェントを用いた非同期メッセージングシステムの提案、情報処理学会 第56回 全国大会講演論文集 (1998).
- [2] 藤田茂, 他： 分散処理システムのエージェント指向アーキテクチャ、情報処理学会論文誌, Vol.37, No.5, pp.840-852 (1996).