

アクティブメッセージで配信する ネットワークアプリケーションフレームワークの構築と オークション事例作成

榮 純平(15597056) 元春 雄一郎(15597111) 森 健太郎(15597114)
原 田 研 究 室

1. はじめに

原田研究室ではこれまでに、インタラクティブなメッセージ (Java プログラム) をインターネットメールとして送受信するシステム JAMES[1]を開発し、様々なアプリケーションを提案してきた[2][3]。本研究では、この JAMES を改良し、リアルタイムで複数クライアント間通信を行う DJAM (Dynamic Java Active Message) を送受信できるようにした。

2. Dynamic JAM と Dynamic JAMES

すでに開発された JAM を改良し、リアルタイムで複数クライアント間通信を行えるようにした。このリアルタイム通信機能を有した JAM を DJAM (Dynamic JAM) と呼び、DJAM を送受信するシステムを Dynamic James と呼ぶ。

3. リアルタイムのメッセージ交換のセッション制御

DJAM は基発信者側 DJAM のみで実行されるサーバ部と、基発信者を含む受信者で実行されるクライアント部の処理に別れている。このため DJAM はサーバ部を起動するインタフェースとクライアント部を起動するインタフェースを持つ。

Dynamic James は、図 1 に示すように、基発信者が DJAM を配信①した直後に、その DJAM をサーバとして起動し②、RMI レジストリの起動④とサーバ部のオブジェクトの RMI レジストリへの登録を行う⑤。配信された DJAM の受信者は、DJAM をクライアントとして起動し⑦、ヘッダ部に埋め込まれた基発信者コンピュータの IP アドレスを基に、RMI の機構を利用して通信を行えるようにした。

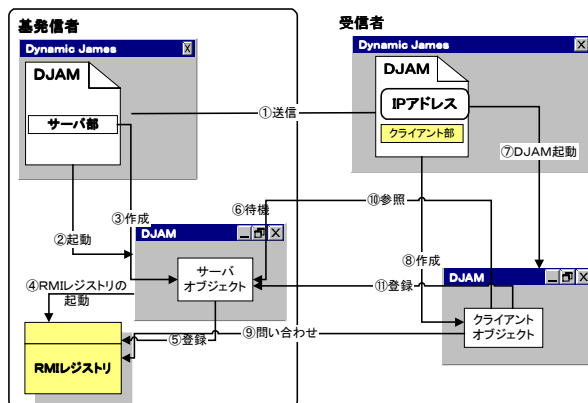


図1 リアルタイム通信モデル図

4. Dynamic JAM 作成支援フレームワーク

ある程度の JAVA プログラミングの知識があればリアルタイムでのクライアント間通信を行う動的な DJAM プログラムを作

成できるように、この DJAM プログラムのフレームワークを作成した。フレームワーク作成において、DJAM を図 3 に示したように Active Message 部 (DynamicActiveMessage)、RMI 通信部 (Talker)、クライアント処理部 (DjamClient)、サーバ処理部 (DjamServer)、ユーザ管理部 (UserControl) に分けた。

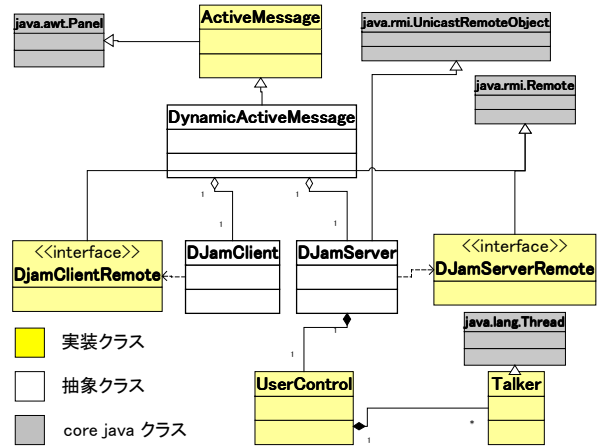


図2 フレームワーククラス図

5. まとめ

本研究によってアクティブメッセージシステム JAMES の実用性が高まり、リアルタイム通信によってアクティブメッセージの新たな応用が開けた。

6. 謝辞

本研究の一部は、(財)テレコム先端技術研究支援センターからの助成研究「アクティブメッセージ JAM のビジュアル作成環境 JamBox の開発と JAM を用いたネットワークアプリケーションの開発」の研究助成金を基に行われた。

7. 参考文献

- [1] 林哲也, 原田実, "Java を用いたアクティブメッセージ交換システムの開発", 情報処理学会論文誌, Vol.38, No.12, pp.2599-2607 (1997.12).
- [2] 鈴木亮, 原田実, "アクティブメッセージシステム JAMES へのセキュリティ実装と集信システム MCS の開発", 情報処理学会第 58 回全国大会論文集, 1p-10, 第 3 分冊, pp.361-362 (1999.3).
- [3] 原田実, 今井哲朗, 川邊建次郎, "アクティブメッセージによるドキュメントワークフローシステムの開発", 電子情報通信学会春季大会論文集, 情報・システム D-9-23, pp.147 (2000.3).