

コミュニティ指向アプリケーション構築支援方式の提案と実装<sup>1</sup>

3 Z B - 8

樋渡 仁 関根 和寿 樋浦 裕二 打橋 知孝

NTT 情報流通プラットフォーム研究所

## 1 はじめに

近年、インターネットを代表とするネットワークの急速な発展に伴い、ネットワークを介した協調作業を支援するコミュニティ指向アプリケーションを容易に実装することが期待されている。従来、グループウェア等のコミュニティ指向アプリケーションは、個別に設計され、開発されてきた。しかし、このようにコミュニティ指向アプリケーション毎に行われる開発は煩雑であり、コミュニティ指向アプリケーション間の連携が困難だという課題があった。

本稿では、協調作業での利用を想定していない、改造が可能な既存アプリケーションに、通信機能を付加することで、コミュニティ指向アプリケーションを容易に構築する支援方式を提案する。

## 2 コミュニティ指向アプリケーション

本稿では、コミュニティ指向アプリケーションを、ネットワークを経由して、複数の利用者が情報を共有し、情報の更新が即座に反映され、利用者が他の利用者の存在を意識できるアプリケーションと定義する。

例えば、通常の WWW では、他の利用者が同じページを閲覧していたとしても、その存在を意識することはない。しかし、Gooy[1]では同じページを閲覧している利用者同士はチャットができ、他の利用者の存在を意識できる。

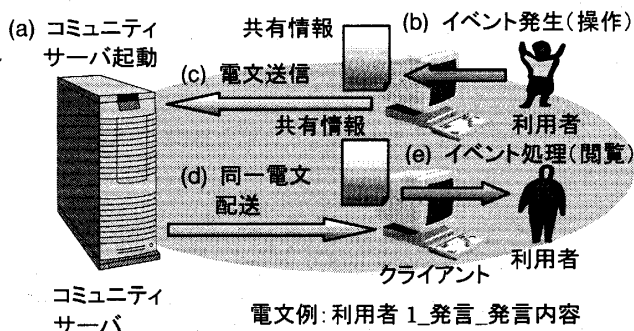


図1: 本稿で採用した情報共有方式

一方、コミュニティ指向アプリケーションを実現するためには、クライアント間で情報を共有しなければならない。本稿では、図1に示すように、各クライアントが共有情報を管理し、クライアントを経由して、利用者は、共有情報を操作したり、閲覧する方式を採用する。

当該モデルにおいて、チャットを例にして、情報が共有される様子を説明する。(a) ネットワーク上にコミュニティサーバ(例:チャットサーバ)が起動され、クライアント(例:チャットクライアント)からの接続を待機する。(b) 利用者はクライアントを操作し、(c) 共有情報を更新するイベント

(例:発言)が発生した時には、当該イベントに対応し、クライアント識別子(例:利用者名)、イベント名(例:発言)、イベント引数(発言内容)から構成される電文を生成し、コミュニティサーバに当該電文を送信する。(d) コミュニティサーバは、クライアントから電文を受信すると、何も変更せずに各クライアントに送信し、(e) 各クライアントは、電文を受信すると、電文から、クライアント識別子、イベント名、イベント引数を抽出し、当該イベントに対応する処理(例:発言ウィンドに利用者名と発言を追加)を実行する。

## 3 提案方式

提案方式が仮定するアーキテクチャを図2に示し、提案方式の詳細について述べる。

コミュニティ指向アプリケーション(例:ホワイトボード)には、既存アプリケーション(例:ドローツール)に通信機能を付加することで実装できるものが存在する。提案方式では、この特徴により、既存アプリケーションを改造して、共有すべき情報(例:描かれた文字や図形)を抽出し、共有情報に対する操作をイベント(例:文字・図形描画)およびイベント引数(例:文字・図形位置)として定義し、イベント発生時には、対応する電文(例:利用者名\_DRAWCIRCLE\_X座標\_Y座標\_半径)を送信し、電文を受信したアプリケーションは対応するイベント処理(例:ウィンドに文字・図形を描画)を実行する。また、電文の送受信および配送は、クライアントおよびサーバ管理機構方式で実行される。

## 3.1 コミュニティ指向アプリケーション構築支援方式

提案方式では、下記の手順を用いて、既存アプリケーションを改造する。

1. コミュニティ指向アプリケーションとして提供された時に利用者間で共有されるべき情報を抽出
2. 当該情報を更新するイベントおよびイベント引数を電文として設計
3. 既存アプリケーションを改造し、クライアント管理機構をインスタンス化
4. 既存アプリケーションを改造し、イベントが発生した場合には、対応する電文をコミュニティサーバに送信するように、当該クライアント管理機構に依頼
5. 既存アプリケーションを改造し、電文を受信した場合には、対応するイベント処理を呼出すように、当該クライアント管理機構に登録
6. 既存アプリケーションを改造し、イベント処理を実装

## 3.2 クライアント管理機構方式

提案方式におけるクライアント管理機構は、下記のように

<sup>1</sup> Proposal and Implementation of Method for Developing Community-Oriented Applications  
Jin Hiwatashi, Kazuhisa Sekine, Yuji Hiura and Tomotaka Uchihashi  
NTT Information Sharing Platform Laboratories

動作する。

7. クライアント管理機構は、インスタンス化されると、指定されたコミュニティサーバに、指定されたクライアント識別子で接続
8. 電文の送信を依頼されると、コミュニティサーバに電文を送信
9. 登録された電文を受信した時には、クライアント識別子、イベント名、イベント引数を抽出し、対応するイベント処理を呼出

### 3.3 サーバ管理機構方式

提案方式におけるサーバ管理機構は、下記のように動作する。

10. サーバ管理機構は、インスタンス化されると、クライアント管理機構からの接続要求を待機
11. クライアント管理機構から接続要求があると、現在接続しているクライアントとして、当該クライアント識別子で特定されるクライアントを追加
12. クライアント管理機構から電文が送信されると、現在接続している各クライアントに、同一電文を送信

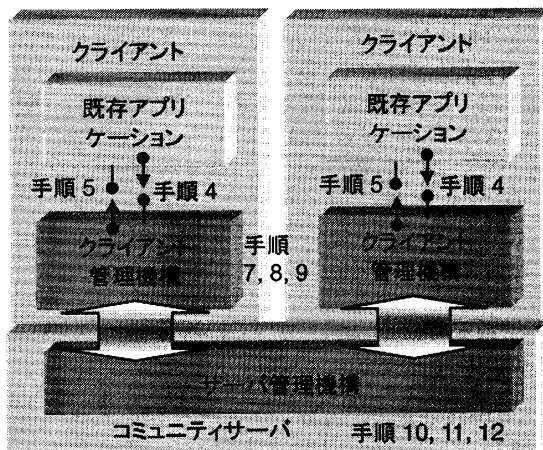


図 2: 提案方式のアーキテクチャ

## 4 実装

本稿で提案した方式において、節 3.1 に示した既存アプリケーション改造方式は、コミュニティ指向アプリケーションに依存するため、共通的にソフトウェアコンポーネント化するのは難しい。一方、節 3.2 に示したクライアント管理機構方式および節 3.3 に示したサーバ管理機構方式は、コミュニティ指向アプリケーションとは独立に機能ため、ソフトウェアコンポーネント (ActiveX) として実装し、再利用を試みた。

## 5 適用例

本稿で提案した方式を用いて、お絵描きボードおよびコミュニティ指向テレビを試作した。

### 5.1 お絵描きボード

お絵描きボードは、同じコミュニティサーバに接続しているクライアント間では、同じ画面が共有され、ある利用者がマウスで画面に線を描くと、他の利用者の画面でも反映されるというコミュニティ指向アプリケーションである。既存アプリケーションとしては、サンプルとして用いられるお絵描きソフトを用いて、改造を行なった。当該既存アプ

リケーションからは、利用者がマウスを操作することにより、ペンを画面に置いた (描画開始)、ペンを動かした (ペン移動)、ペンを画面から離れた (描画終了) というイベントを抽出した。また、当該イベントから画面上のペン座標をイベント引数として持つ電文を設計し、イベント発生時には当該電文を送信するようにした。さらに、イベント処理では、他のクライアントでのペンのイベントから、画面への反映を行なうようにした。一方、通信機能については、章 4 で実装したコンポーネントを再利用した。

### 5.2 コミュニティ指向テレビ

コミュニティ指向テレビは、同じコミュニティサーバに接続しているクライアント間では、同じ動画が再生され、動画に関してチャットできるコミュニティ指向アプリケーションである。

既存アプリケーションとしては、インターネット上の動画を再生できるプレイヤーを用いた。当該既存アプリケーションは、ソフトウェアコンポーネント (ActiveX) であったので改造はせず、ラッパー手法により対処した。イベントとしては、動画の再生を開始した (動画再生開始)、発言した (発言) というイベントを抽出した。また、動画再生開始イベントからは、イベント引数として動画が格納されている URL アドレスを持つ電文を、発言イベントからはイベント引数として発言内容を持つ電文をそれぞれ設計した。さらに、イベント処理としては、他のクライアントが動画を再生した場合、同じ動画を再生する処理、および、他の利用者が発言した場合、発言をウィンドウ上に反映する処理を行なった。

## 6 考察

章 5 で試作したコミュニティ指向アプリケーションは、性能的には実用上問題なく実装できた。また、本稿で提案した方式は、基本的なアプリケーションロジックは、既存のアプリケーションを利用し、通信機能に関しては、章 5 で実装したコンポーネントを利用できるため、開発工程を減少できた。

一方、既存アプリケーションがバイナリ配布の場合には、利用できないという課題が残ったが、既存アプリケーションとは別に、チャットやホワイトボード等のコミュニティ指向アプリケーションを用意することで対処でき、コミュニティ指向アプリケーション間の連携に関しても同様に対処できる見通しを得た。

## 7 まとめ

本稿では、既存アプリケーションからコミュニティ指向アプリケーションを構築支援する方式を提案した。共通化できる方式についてはコンポーネントとして実装した。また、いくつかのコミュニティ指向アプリケーションに適用し、簡易に開発できる上、性能的にも良好な結果を得られ、その有用性を確認した。また、コミュニティ指向アプリケーション間の連携方式に関しては、今後の課題とした。

## 参考文献

- [1] Hypernix: "Get gooey and chat over all the Web!", <http://www.getgooey.com/> (1999).