

Java 搭載携帯電話における同期式共有ホワイトボード

5A-02

太田雅敏[†] 吉滝幸世[†] 川口明彦[‡] 石原進^{*} 水野忠則[†][†] 静岡大学情報学部 [‡] 静岡大学大学院情報学研究科 ^{*} 静岡大学工学部

1 はじめに

参加者が一堂に会さなくても会議が行える遠隔会議システムとして、DYNAMITE[1] や ASSIST[2] などの多くの研究が行われている。さらに、NetMeeting(Microsoft) や CUSeeMe(CUSeeMe Networks) などの会議支援ソフトも数多く市販されており、これらを活用することで、ユーザ間の物理的距離や環境の違いを意識せずに会議が行える。中でも DYNAMITE や NetMeeting では、会議参加者が共有できる仮想的なホワイトボードを実現しており、フローチャートを描いたり、プレインストーミングを行う際に効果的である。しかし、これらのシステムでは、会議参加者が自宅やオフィスのコンピュータの前になければならないという制約があり、街中を歩きながら会議に参加したり、電車での移動中に参加することができない。

そこで本研究では、普及率が高く、利用可能範囲が広く、さらに Java によるプログラムが可能な携帯電話に注目し、携帯電話を用いた仮想ホワイトボードを提案する。本システムを用いることにより、ユーザはどこにいても手軽に電子会議に参加することが可能になる。例えば、新幹線やタクシーによる移動時間と会議の時間帯が重なってしまった場合や、外出中に緊急の会議が入り、PC を利用できる環境が近くにない場合などに有効である。電子会議以外の用途として、友人に自分のいる場所までの道順をリアルタイムに提示したり、チャットのようなコミュニケーションソフトとして利用することも可能である。

本システムでは、サーバへの無駄なアクセスを減少し、ユーザが快適な描画を行えるようにするため、ユーザの描画頻度に従って、クライアントからサーバへアクセスする間隔を動的に変更する。

Masatoshi OHTA[†], Sachiyo YOSHITAKI[†],
Akihiko KAWAGUCHI[‡], Susumu ISHIHARA^{*},
Tadanori MIZUNO[†]

[†]Faculty of Information, Shizuoka University

[‡]Graduate School of Information, Shizuoka University

^{*}Faculty of Engineering, Shizuoka University

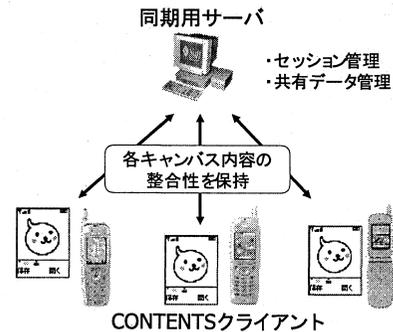


図 1: ホワイトボード同期のイメージ

2 同期式共有ホワイトボード CONTENTS

2.1 概要

携帯電話を利用したスタンドアロン型のお絵描きソフトに、CONTE(Canvas ON mobile TElephone)[3] がある。CONTE ではユーザの描いた絵をサーバ上に保存できるが、他のユーザと共同して絵を描くことはできない。

本研究で提案する CONTENTS(CONTE with Network Synching) は、CONTE の拡張バージョンであり、1つのキャンバスを複数の携帯電話で共有し、共同作業が行える同期式共有ホワイトボード・アプリケーションである。CONTENTS の各クライアントは、描画データ同期用サーバに接続し、描画イベントを示すベクタデータを送受信することにより、各ユーザのキャンバス内容を一致させる (図 1)。今回設計するプラットフォームではサーバからのプッシュが使用できないと仮定する。CONTENTS の同期用サーバは、セッション情報を管理し、ユーザがどの共有キャンバスを利用しているかというデータや、共有キャンバスの描画データを保持する。会議を開始するときは、主催者がメールなどで開始時刻を通知し、各ユーザは指定された URL にアクセスすることで会議に参加できる。

2.2 設計上の問題点

このような同期式ホワイトボードを携帯電話で実装するには、次のような技術的問題点がある。

インタフェースの問題 一般的な携帯電話では、マウスやタブレットのようなポインティング・デバイスを使用することができない。PC と比べ、ボタンの数も少ない。

ホワイトボードの整合性の保持 携帯電話網を介した通信は有線に比べて帯域が狭く、端末間の遅延が大きい。このためホワイトボードのように複数ユーザで同一のデータを共有する場合、ユーザ間で完全な整合性を保つのは困難である。各ユーザが表示するキャンパスの整合性が保てないと、あるユーザの画面では見えている図が、他のユーザの画面では異なるという不都合が起こる。

通信プロトコルの制限 携帯電話上の Java からネットワーク通信を用いる場合、様々な制限が課せられる。これはホワイトボードの同期を取る際に問題となる。例えば NTT ドコモの 503i シリーズの i アプリでは、通信プロトコルに HTTP(HTTPS) のみ利用できる。使用できるプロトコルが HTTP に限られると、共有データが更新されたという情報をサーバからクライアントへプッシュすることができない。このため、クライアントはサーバ上の共有データが更新されているかを定期的に確認する必要がある。この確認の間隔が、ホワイトボードへの描画操作頻度に対して小さいと、サーバへの無駄なアクセスが行われることになる。

2.3 描画用インタフェース

CONTENTS は CONTE と同様のインタフェースを採用している。CONTE では、数字キーがポインティング・デバイスの役割を果たし、ボタンの少なさの問題は、決定キーで機能を切り替えることにより対応する。カーソルの移動は、5 キーの周りの数字キーを押すことにより行う。CONTE は、数字キーの同時押しをサポートし、16 方向のカーソル移動を実現している。これにより、ユーザは滑らかな曲線を描画することが可能になる。カーソル移動と同じく、頻繁に行うペンの上げ下げという動作を 5 キーに配置するなど、CONTE では使用頻度に応じたキー配置となるよう考慮している。カーソルの移動を数字キーで行うのは、各社の携帯電話のインタフェースは異なるが、数字キーだけは、ほぼ共通のボタン配置を利用できるからである。

2.4 描画内用の整合性保持

キャンパス内容の整合性を保つためのアプローチとして次の 2 つの方法が考えられる。

1. 描画できるユーザ数を制限する。
2. サーバ上で管理された描画順序に従って描画する。

1 つ目の方法は、共有キャンパスに描画できるユーザを一時的に限定することで、整合性を保つ方法である。一人だけに描画を許可するならば、他のユーザがその間描画することができなくなるが、描画順序をサーバで管理する必要がなくなる。ドロー型でなく、ペイント型(上書き型)のキャンパスを用い、同じ色で描画する場合、他のユーザも同時に描画することが可能である。異なる色で、複数のユーザが同時に描画できるようにする場合は、各ユーザに一定の大きさの描画領域を割り当て、その領域が重ならないならば描画を許可するといった方法が挙げられる。しかし、この方法では、描画領域の割り当てのために、各描操作の前にサーバと通信する必要がある。

2 つ目は、あるサーバが描画順序を集中管理し、クライアントがその描画順序に従って描画する方法である。各ユーザの操作を、常にサーバで管理し、全てのクライアントで常に正確な描画順序で画面を更新する場合、ユーザの描画操作から実際に画面へ描画するまでの遅延が大きくなるため、ユーザの不快感が増すことになる。クライアントが描画動作を行った瞬間に、その画面に描画動作を反映させるならば、クライアント間の画面の整合性を保つために後から改めてサーバで管理された描画手順に従って再描画する必要がある。この方式では、完全な画面の同期までに多少時間がかかるが、ユーザ自身の描画動作がすぐに画面に反映されるため、ユーザは比較的快適に描画操作が可能である。本システムではこの方式を採用する。

2.5 データ更新手法

本システムにおいて、クライアントがサーバ上のデータの更新を確認するタイミングは 2 通りある。一つは、クライアント側の描画操作データが指定したサイズまで溜まった時(たとえばベクタデータが 10 個に達した時)で、もう一つはサーバから指定された更新チェック間隔だけ時間が経過した時である。

今回使用したプラットフォーム(NTT ドコモの 503i シリーズ)では、サーバからのプッシュができないため、クライアントがサーバ上の更新データを取得するには、定期的にサーバへアクセスする必要がある。このアクセス間隔を長くすると描画データの整合性が取りにくくなり、逆に短くすると無駄なアクセスが増加

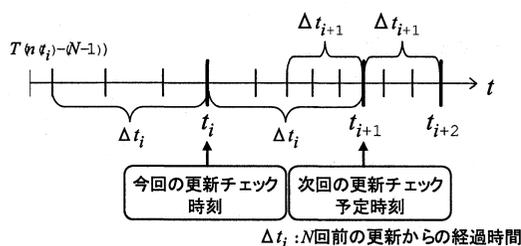


図 2: 更新頻度によるアクセス間隔の変化

する。ここでいう無駄なアクセスとは、クライアントの描画データバッファが空であり、かつサーバ上のデータが更新されていない場合のアクセスのことである。

サーバ上のデータ更新が頻繁に発生しないのであれば、更新チェック間隔を大きくすることができる。例えば、1 分間サーバ上のデータが更新されていなければ、数秒後もデータが更新されないことを予測し、レスポンスとして、クライアントへ更新チェック間隔を返す。

描画データ同期用サーバは、どのユーザがいつサーバ上のデータを更新したかという情報を保持する。あるユーザの i 回目のサーバへのアクセス時刻を t_i とし、その時サーバ上に $n(t_i)$ 個の更新時刻が保持されているとする。この更新時刻とは、ここで対象とするユーザ以外の、すべてのユーザの更新イベントがサーバで受信された時刻である。描画中の共有キャンバスへの a 回目の更新時刻を $T(a)$ とするならば、そのユーザの次のアクセス予定時刻 t_{i+1} は次の式で表される。ただし、 N は参加人数が多いほど大きな値となる整数とする。更新チェック間隔は N により左右される。

$$t_{i+1} = t_i + \Delta t_i$$

$$\Delta t_i = t_i - T(n(t_i) - (N - 1))$$

すなわち、 N 回前のサーバ上のデータ更新から、今回のアクセスまでの経過時間を計測し、その経過時間を次のアクセスまでの待機時間とする。サーバ上のデータが頻繁に更新されているならば、この待機時間は短くなることになる (図 2)。

共同作業人数が多くなると、サーバ上のデータ更新が頻繁になる。この時、 N を固定すると、 Δt_i の値は小さくなる。作業人数の増加によって N の値を増加させると、 Δt_i の値は、参加人数に依存せずに、参加者の更新頻度だけに依存するようになる。この方式で、参加者の増加によるサーバへの過剰アクセスを防ぐことができる。

3 実装

CONTENTS クライアントを i モード Java、同期用サーバを Java サブレットでそれぞれ実装している。携帯電話にはマウスカーソルが存在しないため、ペンの位置を示すカーソルの描画も独自に行っている。ペンの種類は、太さ 10 段階、色は 16 色から選択でき、単純な線の描画以外に、円を描画するためのスタンプ機能も使用できる。

同期用サーバでは描画データをベクタ形式で保存するが、クライアント側では、メモリ容量の都合上、ラスタ形式で保存している。しかし、画面の拡大表示機能を追加することを考えた場合、クライアントもベクタ形式のデータを保持することが望ましい。

実装では更新チェック間隔は 1 秒～10 秒の間で変動するとした。ただし、長時間サーバ上のデータが更新されないような場合 (例えば 3 分間更新されなかった場合) は、会議が一時中断したか終了したとみなし、画面に「ネットワークから切断した」というメッセージを表示して、サーバへの更新チェックを中断する。その後、ユーザからなんらかの入力があれば、再びサーバへの更新チェックを開始する。

NTT ドコモ 503i シリーズに実装し、2 名でホワイトボードへの同時書き込みを行ったところ、処理速度にもたつきなく同期させることができた。

4 検討課題

4.1 携帯電話以外の端末との通信

現在の実装では、ホワイトボードの同期は携帯電話同士の同期に限定している。本システムを電子会議システムの一部として組み込むには、デスクトップ PC や PDA など、様々な端末との通信を実現しなければならない。この際、携帯電話の画面サイズなどのスペックの差を吸収する必要がある。例えば、デスクトップ PC 上に表示されている大きいホワイトボードを縮小して携帯電話の画面上に表示すると、細い線や、小さな文字が判読不明になってしまう。

この問題を解決するために、デスクトップ PC 上のアプリケーションでは、携帯電話側でどのように見えているかを別画面に表示したり、ある一定の大きさ以下の文字や、線を描けなくする必要がある。携帯電話のアプリケーションでは画面を拡大したり、スクロールする機能を追加するという方法も考えられる。

4.2 途中参加するユーザの考慮

ホワイトボードの共同作業中に他のユーザが途中参加した場合、そのユーザは最初にそれまでの描画データを受信する必要がある。システムを長時間使用して

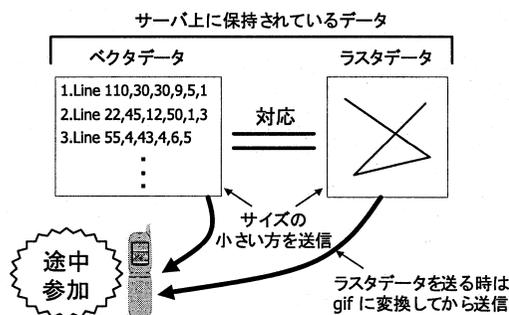


図 3: 途中参加するユーザへの描画データ配信

いと、描画データのサイズは大きなものになる。本システムの通信時の描画データを、ベクタ形式で送るのは、トラフィック量を抑えるためである。しかし、長時間共同作業が続くと、ある時点でベクタ形式のデータサイズの方が、ラスタ形式のデータサイズより大きくなる。デスクトップ PC における同期式ホワイトボードに比べ、携帯電話では、キャンパスのサイズが小さいため、データサイズの逆転が起こりやすい。

そこで、サーバではベクタデータを受信し、キャンパス操作履歴に追加すると同時に、それに対応するラスタデータも作成する。そして、途中参加のユーザには、サイズの小さい方のデータ形式で送信するようにする (図 3)。この際、ラスタデータは GIF に変換して送信する。

4.3 短距離無線通信の利用

携帯電話に Bluetooth を搭載した製品がいくつか販売されている。現行の携帯電話上の Java による通信は、Bluetooth による通信と同時に使用することはできないが、このような短距離無線通信が携帯電話の通信と同時に使用できれば、別の応用が可能になる。短距離無線で相互通信可能なユーザのグループが、遠隔地に 2 つ以上存在し、これらの複数のグループが共同でホワイトボードを使用する場合、遠隔地との同期は携帯電話網を介し、近くにいるグループでは短距離無線通信による直接相互通信によりアドホックネットワークを結成し、ローカルに同期させる使い方ができる。これにより基地局を介したデータ通信量を減らすことが可能になる。アドホックネットワークが n 人で構成されたとすると、携帯電話網へのトラフィックは $1/n$ に抑えられることになる (図 4)。この方式を使うことにより、近くの人のインターネットを利用した作業を自分の携帯電話の画面で参照したり、任意のタイミン

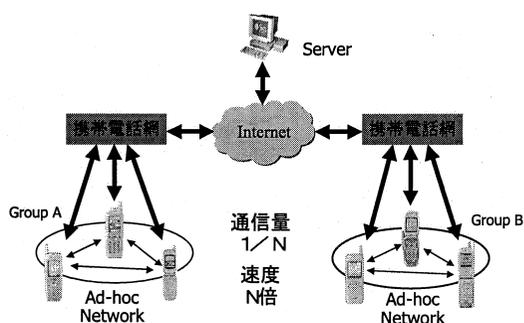


図 4: 短距離無線通信の利用

グで自分がその共同作業に加わることができる。

逆に、各アドホックネットワーク内で、各端末の持つ携帯電話通信回線を同時に使用することで、速度と信頼性の向上を図ることも可能である。[4]

5 まとめ

携帯電話を利用した電子会議システムの一部として同期型共有ホワイトボードを提案・実装した。今回、ホワイトボードの同期をとる際のトラフィックを抑えるため、クライアント側のサーバへのアクセス間隔を動的に変更する方法を提案した。今後は本システムの有効性の評価を行い、携帯電話以外の端末との通信を検討していく予定である。

参考文献

- [1] 加藤 他: 同期コミュニケーションへの途中参加を支援するダイジェスト作成・提供方式, 情処研報, 2001-GW-38-8, Vol. 38, pp. 43-48 (2001-1)
- [2] 田中 他: 同期・非同期統合型マルチメディア会議システム ASSIST におけるマルチメディア議事録の開発と評価, DICO M シンポジウム論文集, Vol. 99, No. 7, pp.79-84 (1999)
- [3] 吉滝 他: 携帯電話を利用したお絵描きツールの提案, 平成 13 年度電気系学会東海支部連合会, No. 653 (2001)
- [4] 小西 他: 通信回線共有方式を利用した高速 Web アクセスの実現, 情処研報, Vol. 2001, No. 13, pp. 17-24, 2001-MBL-16 (2001)
- [5] CLDC(Connected Limited Device Configuration) 仕様, <http://jdc.sun.co.jp/wireless/>
- [6] i モード対応 Java コンテンツ開発ガイド (詳細編) 1.1 版, <http://www.nttdocomo.co.jp/me-user/i/java/index.html>