

ネットワークを介して筆談を行うコミュニケーションツールの提案

縄手 雅彦[†] 佐々木 武^{††} 吉山 竜也^{††} 畑本 和徳^{††} 福間 慎治[†]
本多 茂男[†]

[†] 島根大学総合理工学部, 松江市

^{††} (株) ワコムアイティ, 松江市

E-mail: [†]{nawate,fukuma,honda}@ecs.shimane-u.ac.jp, ^{††}sasaki@wacom-it.co.jp

あらまし インターネットを経由して複数のクライアント上でリアルタイムに筆談するコミュニケーションツールの試作を行った. 通信プロトコルとして HTTP を使用し, 手書き情報を時系列でベクトル化して情報の共有を行うシステムとした. 受信者側は書き順等そのまま表示できることから, 正確な意志伝達を可能にするチャット形式の筆談を行うことが可能になった.

キーワード 筆談, リアルタイム, クライアントサーバシステム, コミュニケーションツール

A communication tool by writing conversation via network

Masahiko NAWATE[†], Takeshi SASAKI^{††}, Tatsuya YOSHIYAMA^{††}, Kazunori HATAMOTO^{††},
Shinji FUKUMA[†], and Shigeo HONDA[†]

[†] Department of Electronic and Control Systems Engineering, Shimane University, 1060 Nishi-Kawatsu,
Matsue 690-8504 Japan

^{††} Wacom IT Co. Ltd., 43 Hokuryo, Matsue 690-0816 Japan

E-mail: [†]{nawate,fukuma,honda}@ecs.shimane-u.ac.jp, ^{††}sasaki@wacom-it.co.jp

Abstract We have made a prototype system of communication that can provide real-time handwriting conversation between plural clients via the internet. The system uses HTTP for communication and the data are vectorized in time-sequence, resulting in having information commonly between clients. Even stroke orders of written data are reproduced in the clients' windows, so writing conversation with accurate notification of writer's will becomes realized.

Key words writing conversation, real time, client server system, communication tool

1. はじめに

高速インターネット網の整備は CATV や xDSL の普及により急速に進んでいる. 今後は光ケーブルによる接続の増加も予想され, 通信形態や接続プロトコルなど次世代の基盤技術も盛んに議論されている [1]. また, 現実に官民の協力により地域へのブロードバンド展開も進展している [2]. さらに, 「ホームネットワーク」という言葉も定着し [3], 家庭や SOHO 事業所等まで高速ネットワークの普及が進み, 日常生活においてネットワークに接続したパソコンが傍らに存在するという形態が今後ますます増加すると予想される.

そのようなネットワークの普及にともない多様な情報や知識を入手し, リアルタイムで共有や発信できるアプリケーションの必要性がでてきた. 情報のリアルタイム共有法としてテレ

ビ会議とチャットが現状では挙げられる. テレビ会議は遠隔地にいる相手と顔を見ながらリアルタイムで話ができ, 最近では遠隔教育などにも利用されている. しかし, 専用の機材が必要であったり利用する場所が限られるなどの問題がある. 一方, チャットはインターネット上でユーザ同士が画面上で文字を使ってリアルタイムに会話する方式であり, パソコンとネットワーク環境さえあればどこでも手軽に利用できる. しかし, キーボードの入力技術が無い人には参加は難しい.

手軽なネットワーク越しのコミュニケーションツールの開発は以前からなされてきていたが, 例えばマイクロソフトが提供している Netmeeting システム [4] は一つのソリューションといえる. Netmeeting ではテレビカメラやマイクを接続することにより安価なテレビ会議システムの構築が可能で, Windows クライアントのペイントツールなどアプリケーションを使用し

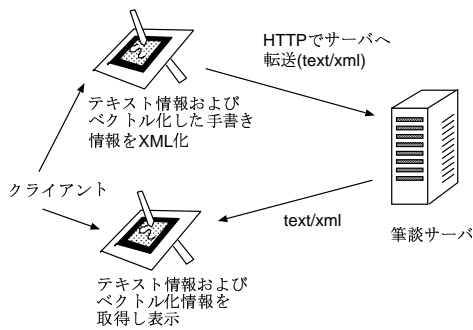


図 1 システム構成

て作成したデータを「ホワイトボード」と呼ばれる疑似掲示板にコピーペーストすると、相手先に情報が表示され、ファイル転送も可能となっている。テキストベースの意志伝達はキーボード入力からのチャットで実現されている。

一方、キー入力という点で見ると、携帯電話や PDA の普及により、テンキーを利用して文字入力を高速で行う方式や入力先読み機能などを取り込んだ製品も登場してきた [5] が、パソコンの入力についてはいまだにキーボードが主流である。そこで、本研究ではネットワークを介してのテキストおよび手書き情報の混在した筆談システムの可能性について実際にサーバクライアント型のツールを試作して検討した。既存のチャットとは異なりキーボードから入力したテキスト情報だけでなくペンタブレット等で入力した手書き情報を自由に混在させることが可能であることが特徴であり、筆順も含めた手書き情報により、意志伝達の正確性に与える寄与について検討した。

2. システム構成

筆談システムの構成を図 1 に示す。クライアントとしてタブレットを接続した Windows パソコンを使用した。タブレットを接続せずに通常のマウスだけでも操作は可能であるが、手書きによる詳細な文字や画像の描画を試みるためにここではタブレットを利用している。

サーバには Microsoft Internet Information Service (IIS) 4.0 もしくは 5.0 が稼働する Windows サーバを用いた。サーバとクライアントは TCP/IP, HTTP の一般的なプロトコルで通信を行うため、ネットワークとしては IP 接続されていることが必要である。伝送するデータは XML 形式を採用した。

3. クライアントアプリケーション

3.1 画面構成と機能概要

3.1.1 ログイン画面

筆談サーバへのログインを行う画面を図 2 に示す。画面内のウィンドウに筆談サーバ一覧を表示するが、サーバ一覧はローカルの PC の設定ファイルから取得する。筆談サーバにログインするには一覧からサーバを選択しユーザ名を入力した状態で「ログイン」ボタンをクリックすることにより行う。サーバへのログインが許可されると筆談サーバ受信用のダウンストリームコネクションを形成する。ログイン処理が終了すると筆談メイン画面を表示しログイン画面を閉じる。



図 2 ログイン画面

3.1.2 筆談メイン画面

筆談を行うメイン画面を図 3 に示す。筆談エリアに筆談内容の書き込みおよび受信したデータを表示し筆談を行う。表示されている筆談内容に上書きする形で書き込みと表示を行う。書き込まれた内容はクライアントで設定したデータ送信タイミングでサーバに送信され別の各クライアントに配信される。他のユーザからの筆談データを受信した場合は受信した順に表示する。自分で作成したデータも他のユーザと同様に受信した時点で表示を行う。入力は、ペンやマウスからの手書き入力と、貼付によるテキスト入力の二通りで行う。

テキスト入力は「テキスト貼付け」ボタンをクリックし、入力モードを筆談からテキスト貼付に変更して行う。テキストボックス内にテキスト入力後、筆談エリアをクリックすることによりクリックした位置にテキストが貼付けられる。なお、貼付けるテキストのフォントはテキストフォント設定部分で選択したスタイルを使用する。貼付後は入力モードを自動で筆談に戻す。

「切断」ボタンのクリックによりサーバからログアウトし、データ受信用のダウンストリームコネクションを切断する。ログアウト後は切断ボタンおよび筆談エリア、その他の各ボタンを使用不可に設定し、接続ボタンを使用可に設定する。なお、筆談サーバとの接続が保たれた状態で画面を閉じる場合には、切断ボタンのクリック時と同様の切断処理を行ってから画面を閉じプログラムを終了する。

「接続」ボタンのクリックにより図 2 に示すログイン画面をモーダルで表示する。ログインした場合にサーバ名、ユーザ名をタイトルバーに表示する。また、ログイン後、接続ボタンを使用不可に設定し、筆談エリアおよび各ボタンを使用可に設定する。また、筆談エリアはクリアされる。

送信ボタンのクリックにより未送信のデータをサーバに送信する。未送信データが無い場合にはこの操作は無視される。通信状態は画面上部の表示エリアに表示される。

筆談サーバに接続しているユーザ名一覧は筆談参加者リストとして画面に表示される。表示するタイミングはログインおよびログアウトデータを受信した時点でユーザリストを取得して行う。

筆談エリアのデータが増えてくると情報を見辛くなるので画面クリア機能を用意した。「クリア」ボタンのクリックにより画面クリア要求データを送信する。送信後、クリアボタンを使用



図 3 筆談メイン画面

不可に設定する。クリア要求データを送ったクライアントが自分で要求データを受け取った時点から一定時間経過した段階で画面クリア実行データを送信する。待ち時間はクライアントごとに設定する。

また、クリア要求を行ったクライアント以外のクライアントがクリア要求データを受信した場合にはこの時点から設定ファイルで設定した時間だけ「クリア拒否」ボタンを使用可に設定する。クリア拒否ボタンのクリックにより画面クリア要求キャンセルデータを送信する。送信後はクリア拒否ボタンを使用不可にする。

クリア要求を行ったクライアントが画面クリアキャンセルデータを受信するとクリア要求からのクリア関連処理を初期化し再びクリアボタンを使用可に設定する。また、画面クリア実行データを受信すると筆談エリアの画面をクリアしクリアボタンを使用可にする。

クリアキャンセルおよびクリア実行のデータは先に届いたデータを優先する。クリア実行データの発行までの待ち時間およびクリア拒否可能時間は設定ファイルから取得する。

筆談作業における一連の処理の流れを図 4 に示す。個々の処理については次節以降で紹介する。

3.2 筆談データ

筆談データは MSXML ライブラリから提供される DOM の機能を使用し作成する。タブレットを使用した場合のペンの操作はタブレットドライバによりマウスイベントに変換されるのでマウスイベントを捕らえることによりペンの動作を検知する。図 5 に時間に沿った処理の流れを示す。

未送信データの無い状態で手書きもしくはテキストの貼付に関わらず筆談エリアへのマウスダウンイベントが発生した時間を描画開始時間とする。筆談データ (XML) を初期化し、<DataRoot> をルート要素とする XML を作成し、クライア

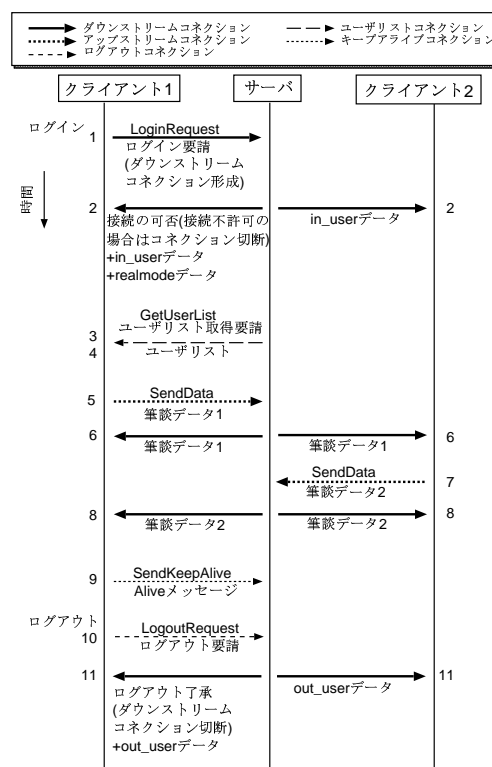


図 4 筆談作業の流れ

ント情報に関する <Information> ノードを追加する。送信処理が実行されるまで、手書きおよびテキスト貼付の各操作終了ごとに <Data> ノードを追加する。この <Data> ノードの作成処理は手書きと貼付の操作終了ごとに行う。

手書き入力の場合の処理を図 6(a) に示す。手書きの場合、ストロークごとに <Data> ノードを作成する。ストローク開始から終了までマウスイベント (ダウン, ムーブ, アップ) ごとにマウスの位置座標 (x, y) と描画開始からの経過時間を取得し

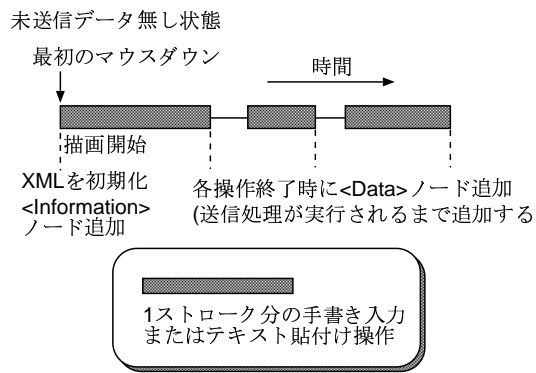


図 5 筆談データ作成の概要

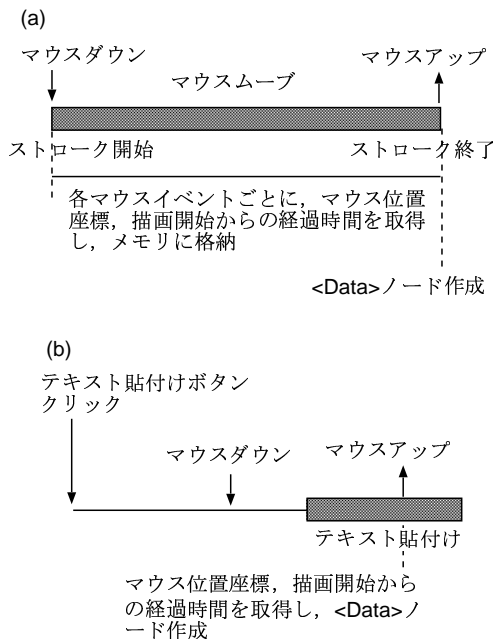


図 6 手書き入力時 (a) およびテキスト貼付け時 (b) のデータ生成模式図。

メモリに格納する。ストローク終了時にメモリを参照し、「筆談データ構成」に基づき <Data> ノードを作成する。

一方図 6(b) に示すテキスト貼付の場合は、マウスアップイベント時にテキスト貼付を実行し、マウス位置座標および描画開始からの経過時間を取得し、「筆談データ構成」に基づき <Data> ノードを作成する。

3.3 筆談データの送信

筆談データの送信処理は送信用のスレッド内で処理する。スレッドは送信処理ごとに作成し、送信処理終了時に終了させる。筆談データの送信処理は、次に示す条件で実行する。

(1) 送信ボタンをクリックしたとき

送信ボタンをクリックした時点で未送信のデータがあれば送信する。

(2) ストローク数が「送信が実行されるストローク数」に達したとき

ストローク数は描画開始時からの合計とし、送信後に 0 にリセットする。テキスト貼付データは 1 回の貼付分を 1 ストロークとする。「送信が実行されるストローク数」は設定ファイルか

ら取得する。

(3) 入力操作の無い時間が「送信が実行される無操作時間」に達したとき

入力操作終了時 (マウスアップ時) からの経過時間を入力操作の無い時間とする。「送信が実行される無操作時間」は設定ファイルから取得する。

(4) 描画以外のデータを送信するとき

画面クリア関連データを送信する際には、このデータを筆談データに追加し、直ちに送信する。

これら一連の送信処理に関して、現在の設定では指定したバイト数である 64 バイトに達するまで受信内容が取得されないため、全ての送信データの末尾に 64 バイト分のスペースを付加して送信を行うこととしている。

3.4 筆談データの受信と描画

筆談データの受信にはログイン時に形成したダウンストリームコネクションを使用する。データの受信と描画処理は専用のスレッド内で行う。このスレッドはログイン時に作成し、ログアウト時に終了させる。また、処理は筆談データの <DataRoot> 要素を単位とし、受信、描画の順に交互に行う。

データの受信は 64 バイト単位で行う。受信したデータから <DataRoot> 開始、終了タグを検出し、<DataRoot> 要素を抽出する。抽出した <DataRoot> 内に含まれる <Information> と <Data> の要素を MSXML ライブラリから提供される DOM の機能を使用し解析する。各 <Data> 要素のデータの種別に応じた処理を行い、<DataRoot> 要素に含まれる全ての <Data> 要素の処理が終了すると再びデータの受信処理に移る。

<Data> 要素のデータの種別が手書きデータおよびテキスト貼付データの場合には描画処理を行う。<Data> 要素に含まれるペン情報、フォント情報を適用し描画開始からの経過時間に従って描画を行う。ただし、ログイン以前のデータに関しては経過時間を無視してスキップモードで描画する。

4. サーバ側アプリケーション

筆談用の通信プロトコルは HTTP 上に実装されており、クライアントからサーバへのデータは HTTP の GET または POST メソッドとして送信され、WWW サーバ上で動作する CGI として実装されているサーバアプリケーションを起動する。サーバアプリケーションは受け取った HTTP リクエストの内容に応じて必要な処理を行う。一方、サーバからクライアントへのデータ送信は、ログイン処理時にクライアント側から張られた TCP コネクションを利用して行われる。

以上の処理を全てのクライアントに対して行うことにより、複数のクライアント間での手書きデータの共有を実現する。

4.1 使用ファイル

データファイルとして「手書きデータ蓄積ファイル」と「オンラインユーザ管理ファイル」の二つを使用する。「手書きデータ蓄積ファイル」は各クライアントから送られてきた手書きデータを蓄積するファイルであり、最初のユーザがログインしたときに生成され、最後のユーザがログアウトしたときに消去される。クライアントから送信された手書きデータをそのまま保存

表 1 オンラインユーザ管理ファイルの内容

[UniqueID]		
Name	文字列	クライアントから送られてきたユーザ名
LoginTime	文字列	ログインした日付と時刻 yyyy/mm/dd hh:mm:ss
IPAddress	文字列	クライアントの接続元の IP アドレス. CGI の環境変数 REMOTE_ADDR の値をそのまま格納する.
LastKeepAliveTime	文字列	最後にクライアントから KeepAlive メッセージが送られてきた時刻 yyyy/mm/dd hh:mm:ss

することとし、データ区切りには^Z(キャラクタコード 0x1A)を使用する。

「オンラインユーザ管理ファイル」はログインしているユーザの一覧を管理する。形式は Windows の INI ファイル形式とし、1 ユーザごとに一つのセクションで管理する。セクション名はログイン時にユーザに割り当てたユニーク ID を使用する。表 1 にファイルが扱うデータの形式を示す。

4.2 処理内容の概要

HTTP メッセージに対する処理を表 2 に示す。

表 2 HTTP メッセージに対する処理

メソッド	コマンド名	処理内容
GET	Login	ログイン処理。クライアントにはログインの可否を返す。ログイン可の場合、ログイン処理のためにクライアント側から張られたコネクションはログアウト処理が行われるまで維持し、ダウンストリームコネクションとして使用する。
	logout	ログアウト処理。筆談システムの使用終了が通知される。サーバはダウンストリームコネクションを切断する。
	userlist	ユーザリスト送信要求。ログイン中のユーザのリストを返信する。
	keepalive	KeepAlive メッセージ。クライアントから約 5 分おきに送られてくるクライアントが接続していることを通知する信号。ユーザリストの LastKeepAlive 項目に受信時間を書き込み、クライアントには通常の HTTP ヘッダを返す。
POST	(なし)	データアップロード処理。1 ストローク分の手書きデータがクライアント側からサーバに送信される。サーバは送られてきた XML データを接続中の全てのダウンストリームコネクションに対して送信する。

クライアントから筆談データが送信されると、サーバ側アプリケーションのプロセスが保持している全てのダウンストリームコネクションに対してそのデータを送信する。プロセス間でのデータの受け渡しには手書きデータ蓄積ファイルが使用される。ファイルの読み込みにおいて、前回読み込みが完了した位置からファイルの末尾までを読み込みバッファに追加する。ファイルは 1 メッセージごとに^Z で区切られているので、読み込み完了後はバッファの一番最初に現れる^Z の直前までの文字列を抽出し、その部分はバッファから削除する。

ファイル末尾まで読んだにも関わらず^Z が現れなかった場合にはファイルが更新されるまで最大 10 秒待つ。(Win32API の FindFirstChangeNotification を使用してディレクトリ更新の通知を受けることにより実現している。) 待っても^Z が現れない場合には空文字列としてそのまま次の処理へ進む。

5. 筆談データの形式

データ形式は XML 形式に準拠している。(詳細は付録を参照。) <DataRoot> ノードを通信の単位データとし、要素にはクライアント情報、手書きデータ、テキストデータおよびその他のデータを持つ。クライアント情報は <Information> ノードに記し、<DataRoot> ノードに 1 ノード必要とする。ただし、サーバ側で作成するデータには必要ない。その他のデータはそれぞれ <Data> ノードに記し、<Kind> ノードの値によりデータの内容を区別する。手書きデータは 1 ストローク分の情報を 1 <Data> ノードとし、筆跡データとしてサンプリング点に応じた座標データを持つ。手書きデータとテキストデータを要素とする <Data> ノードは必要な数だけ <DataRoot> ノードに追加できる。

6. 実験結果と考察

以上のような筆談システムを試作し、複数のクライアントによる試用を行った。まず、筆談データを自動送信するストローク数を 3, 5, 7 の 3 種類で実験したところ、ストローク数が 3 と 5 の場合は手書き情報が自然なタイミングで送信されたが、7 では相手側への表示の遅れが体感的に大きくなり、不自然な印象を与えた。ストローク数の設定は個人差を考慮しても 5 以下に設定するのが望ましい。

蓄積データが存在するときに自動送信が行われる時間設定に関しては、2, 5, 10 秒の場合について検討した。2 秒に設定した場合には手書き情報を書き終わると同時くらいの送信タイミングとして感じられた。5 秒と少し長く設定した場合でもコミュニケーションにおける支障はほとんど感じられなかったが、10 秒では送信がかなり遅れる印象が強く、スムーズな筆談とはならなかった。よって、送出時間の設定は 5 秒程度がそれ以内が望ましい。

クリア要求に対する猶予時間は 10, 15, 20 秒として実験した。10 秒の時間があれば、要求が届いてからボタンを押すまで余裕はあるのだが、自分も書き込みをしている場合にはギリギリのタイミングになることが多かった。15 秒以上であれば、拒否を行うのに十分であった。

表 3 筆談時の送信データ量

CPU	データ量 (byte)		B/A
	A. 一括送信	B.1 画ごとに送信	
AMD-K6 350MHz	14738	20910	1.42
PentiumIII 500MHz	14960	20059	1.34
Crusoe 600MHz	11621	15704	1.35
PentiumIII 850MHz	7621	11145	1.46

以上のような検証により送信設定パラメータを適切に設定したところ、手書きと文字情報の混在したチャットにより通常のテキストのみによるチャットに比較すると説明する際の表現力において大きな優位性が見られ、システムの可能性が証明された。しかしながら、いくつかの問題点も顕在化してきたのでそれについて検討を加える必要がある。

限られたスペースに手書き情報が繰り返し入力されると、だれがどこに書いているのかわからなくなる問題が生じた。また、チャットとは異なり会話を時系列で追うことができないので、会話の順序についてはわかりづらくなる。また、キーボード操作になれた人から見れば手書きは手間と時間がかかるので、手書きだけで筆談を行うことは不便を感じる結果となった。IP 電話を別途設置し音声と筆談とで会話を行う実験も行ったところ、筆談だけの場合と比較してより円滑なコミュニケーションが可能であったので、コミュニケーションツールとして見た場合の補助手段についてはさらに検討が必要かも知れない。

通信プロトコルに HTTP を使用しているので、NAT によるアドレス変換が行われている環境でも使用可能であることは利点であった。しかし、筆談が行われている時間は HTTP のコネクション(ダウンストリーム)を張ったままであるので、HTTP プロキシサーバを経由した場合には問題となることがあるかも知れない。より汎用性を高めるためには、クライアントからサーバに対して一定間隔ごとにポーリングするなどの措置が必要であろう。

データ形式に XML を使用したことは、(1)Windows が提供している XML パーサを流用できる、(2)送受信データの変更に柔軟に対応できる、という利点があった。しかし、数値をテキスト形式で表現していることや XML のタグ分だけデータ量が膨れる、という問題点もある。実際に同時に複数のユーザが手書き入力を始めると多少反応が悪くなるという結果も生じた。そこで、実際に送信されるデータの大きさを調べた結果を表 3 に示す。データは 160 × 160 ピクセルの大きさの「正」の字を 10 回入力しその平均値を取ったものである。「一括送信」は正の字を 5 画すべて入力後にサーバに送信したもので、「1 画ごとに送信」は 1 画ずつサーバに送信した場合の 5 画分のデータ量である。また、データ量はタブレット上の座標を取得する回数が時間に依存するため、1 画を 1 秒で入力するように統一してある。データ量はサーバで稼働している IIS のログから PUT メソッドによりクライアントから送信されたデータ部分から取得している。

CPU の異なる 4 種類のパソコンを利用して計測したが、表 3 に示すようにクロック周波数との相関はない。また、1 画ずつ

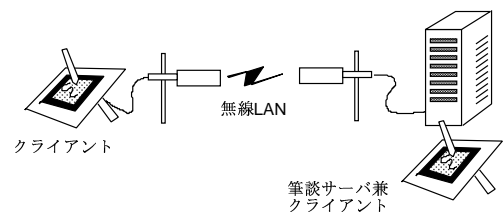


図 7 無線 LAN による実験構成

送信した場合には XML のタグ部分がオーバーヘッドとなって加わるために、一括送信に比べて 40% 程度データ量が増えている。このように、細かく送信すると XML タグの情報量が問題になってくる可能性がある。また、160 ピクセル角の漢字情報で 7-15kbyte 程度のデータ量となることはネットワーク速度によっては筆談に影響を受ける可能性があるかも知れない。

そこで、上記の試用は基本的には 100Base-TX の LAN 環境の中で行ったが、試験的に 11Mbps の速度を持つ無線 LAN 環境でも行った。図 7 に示されるような構成で対一の接続で試験を行った。この場合には一方がサーバとクライアントをかねている環境である。このような速度環境においても、100Mbps の LAN 環境と同程度の使用感を得ることはできた。

7. ま と め

キーボード入力が苦手な人でも手書き入力によりネットワークを介した筆談形式でリアルタイムにチャットが可能システムを提案した。データに XML 形式を使用し、通信に HTTP を使用することで手軽にクライアント/サーバ形式のコミュニケーションツールを実現できた。

何らかの意志を伝達する際の入力手段としては、現状ではキーボードがもっとも効率がよいことは否定できない。しかし、取っつきやすさや感情表現という面においては手書きが優れている。手書き情報を筆順も含めてやり取りすることが実現できたことにより、今後のコミュニケーションツールの開発に新しい展開が期待される。

謝辞

筆談システムの実験に関して久木崎英章、森本大資、福田耕大の各氏に感謝する。この研究は平成 13 年度島根県創造技術研究開発補助金により行われた。

文 献

- [1] 江崎浩, “インターネットの将来像”, 信学誌, Vol.85, No.2, pp.88-93, 2002.
- [2] 木下康昭, 阪本泰男, 佐藤千明, 青木敏, 野本真一, 篠原弘道, “ブロードバンドインターネットの動向と課題”, 信学誌, Vol.85, No.3, pp.204-213, 2002.
- [3] 玉木規夫, “ホームネットワーク”, 信学誌, Vol.84, No.4, pp.258-260, 2001.
- [4] www.microsoft.com/japan/windows/NetMeeting/default.htm
- [5] 入鹿山剛堂, “ケータイ文字入力の現状と将来”, 信学誌, Vol.84, No.11, pp.819-827, 2001.