



## 要 旨

近年PC上のコミュニケーションツールとして、チャットやインスタントメッセージが普及した。チャットは大人数で会話することに長けている一方で、インスタントメッセージは関心のある人のプレゼンス情報を詳しく表示できるという長所を持つ。

我々は、チャットとインスタントメッセージングサービスを拡張できる可能性について考察した。その拡張手法として、相手の様子とその周囲を見ながらコミュニケーションを行う方法、複数の話題を扱う方法、発言の雰囲気を補う方法を挙げ、これらを実現することを本研究の目標と定めた。

そして目標の実現のために、ライブカメラを用いたコミュニケーションツールを考案、実装した。本システムにおいて、発言内容は、ライブカメラの画像上にフキダシとして表示される。フキダシの形や表示位置を考慮することで、ユーザは相手の様子を見ながら、直感的に会話をすることができるようになった。加えて、過去にあった会話についても、当時の画像とフキダシを用いることで臨場感のある表現で振り返ることが可能になった。

# 目次

第1章	序論	1
1.1	研究の背景	1
1.2	主な同期式コミュニケーションツール	1
1.2.1	チャット	1
1.2.2	インスタントメッセージ	1
1.3	同期式コミュニケーションツールの可能性	2
1.3.1	プレゼンス情報の表現手法	2
1.3.2	発言の聞こえる範囲の調整手法	2
1.3.3	発言の表示手法	2
1.3.4	発言の雰囲気表現手法	2
1.4	本研究の目的	2
1.5	本論文の構成	2
第2章	本研究の目標	3
2.1	相手やその周囲の様子を見ながら会話ができる	3
2.2	複数の話題の理解を助ける	3
2.3	発言の雰囲気表現する	3
第3章	フキダシチャットシステムの提案	4
3.1	ライブカメラの利用	4
3.2	フキダシの表示	5
3.2.1	フキダシの表示位置と表示方法	6
3.2.2	複数の話題に対する方法	7
3.3	発言の雰囲気表現	8
3.3.1	フキダシの形	8
3.3.2	フキダシの色	9
3.4	チャットログの再現	9
第4章	フキダシチャットシステムの機能	10
4.1	起動	10
4.1.1	画像の更新	10
4.1.2	ログイン	10
4.2	発言方法	11
4.2.1	発言の入力と送信	11
4.2.2	フキダシの表示	11
4.3	フキダシ発生位置変更方法	12
4.4	発言の雰囲気表現する方法	13
4.5	ログ閲覧の方法	14
第5章	フキダシチャットシステムの実装	17
5.1	ハードウェアの構成	17
5.1.1	ライブカメラ	17
5.1.2	サーバーPC	17
5.1.3	クライアントPC	18

5.2	共有フォルダの利用 .....	2 0
5.2.1	ログファイル.....	2 0
5.2.2	同期の方法.....	2 1
5.3	サーバー P C の仕組み.....	2 1
5.4	クライアント P C のソフトウェア.....	2 1
5.4.1	主なクラス.....	2 1
5.4.2	処理の流れ.....	2 2
第 6 章	関連研究.....	2 4
第 7 章	結論.....	2 5
	謝辞	
	参考文献	

## 目次

図 3.1 : 部屋に設置されたカメラ .....	4
図 3.2 : ライブカメラの画像の例 .....	5
図 3.3 : フキダシの部品の名称.....	6
図 3.4 : 発言の順序がわからないフキダシ .....	6
図 3.5 : 発言の順序がわかるフキダシ .....	7
図 3.6 : 図 3.5 から時間が経った様子.....	7
図 3.7 : 2つの話題が並行して行われる例 .....	8
図 3.8 : フキダシの位置を変更することで、2つの話題がわかりやすくなる様子 .....	8
図 3.9 : フキダシの種類 .....	9
図 4.1 : ログインの様子 .....	10
図 4.2 : 発言内容がフキダシとして画面上に現れた様子 .....	11
図 4.3 : 図 4.2 のときから時間が経った様子 .....	12
図 4.4 : フキダシの縦座標を位置任意に変更した例 .....	13
図 4.5 : 様々なフキダシで会話をする様子 .....	14
図 4.6 : フキダシが画面外に流れてしまった様子.....	15
図 4.7 : スライダの操作によって過去のチャットの様子が表示された様子.....	16
図 5.1 : サーバー P C の大まかな処理の流れ.....	17
図 5.2 : クライアント P C の大まかな処理の流れ.....	18
図 5.3 : クライアント P C の発言時の処理 .....	19
図 5.4 : クライアント P C のログ閲覧時の処理 .....	20

# 表目次

表 4.1 : フキダシの形に対応するコマンドと文字列 .....	1 4
表 5.1 : 画像更新時のログファイルの内容 .....	2 1

# 第1章 序論

## 1.1 研究の背景

パーソナルコンピュータ（PC）とインターネットが普及し、それらを利用したコミュニケーションが盛んに行われるようになった。これにより、PCを用いた作業の最中でも、席を立たずに人とコミュニケーションが行えるようになった。

PCを用いた既存のコミュニケーションツールは、同期式と非同期式のものに大きく二分される。同期式のコミュニケーションツールは、主なものにチャットやインスタントメッセージなどある。一方、非同期式のコミュニケーションツールには、掲示板や電子メールなどが例として挙げることができる。

同期式コミュニケーションは、複数の人が同時刻に集まり、リアルタイムで会話を進めるものである。同期式コミュニケーションの多くは、現在誰が会話に参加をしているのかを表示するプレゼンス情報というものを提供している。

非同期式コミュニケーションは、会話の参加者が同じ時刻に集う必要がない。必要に応じてメッセージを送っておき、後でその反応を確認することでコミュニケーションを行うツールである。

## 1.2 主な同期式コミュニケーションツール

本研究では、同期式コミュニケーションツールに着目する。

### 1.2.1 チャット

チャットは、インターネット上で発達したコミュニケーション手法である。チャットをするには、会話をするメンバーがひとつの場所（サーバー）、つまりチャットルームに明示的に接続する。複数人数が同時に同じチャットルームで会話を行うことができる点が、チャットの持つ大きな利点である。

チャットルームへの入退室ログは、今そこに誰が入室しているのかをユーザに知らせるので、チャットにおけるプレゼンス情報であるといえる。

### 1.2.2 インスタントメッセージ

インスタントメッセージは、チャットのようにある場所に明示的に接続する必要がなく、同じソフトを利用している人にメッセージを送ることによって相手側の会話用ウィンドウが起動される仕組みである。複数人数で会話を行うこともできるが、多くの場合は1対1の会話である。

あらかじめ登録しておいたメンバーのプレゼンス情報を表示できることは、インスタントメッセージングサービスの最大の特長である。インスタントメッセージングサービスにおけるプレゼンス情報とは、たとえば「オンライン」か「オフライン」か、もしくは「取り込み中」など、相手の“今”の状態である。相手の様子が確認できるため、ユーザが自分の用件の緊急度と相手の状態を比較して、メッセージを送るかどうかが判断することができる[1]。

## 1.3 同期式コミュニケーションツールの可能性

### 1.3.1 プレゼンス情報の表現手法

会話をする際に、自分が誰と話しているのかが無意識にわかることは、コミュニケーションをする際の重要な情報となる。相手の様子を見ながらコミュニケーションができるような仕組みを取り入れることは、プレゼンス情報を拡張するひとつの方法である。

また、会話の相手だけでなく、その周囲の状況をプレゼンス情報として提供することで、コミュニケーションがより豊かになる可能性も考えられる。

### 1.3.2 発言の聞こえる範囲の調整手法

通常は、指定した人物にしか届かないインスタントメッセージを、チャットの発言のように仲間全体で共有できるようにすることもひとつの方法である。仲間全体が会話内容を把握することで、コミュニケーションの可能性が広がることが考えられる。

### 1.3.3 発言の表示手法

大人数でチャットを行う場合、議論が複数の話題へ分離することがしばしばある。このとき画面上には別々の話題に関する発言が交互に表示されることとなり、ユーザは話の流れを追うことに苦勞する場合がある。これは、発言が一行、つまり一次的に並ぶことによる問題で、発言を二次元的に表示することで解決できることが考えられる。

### 1.3.4 発言の雰囲気表現手法

チャットなどテキスト主体のコミュニケーションをする場合、ユーザは発言の内容を強調したり和らげたりするために、記号や顔文字を発言のテキストに含ませる場合がある。

テキストに加えて、より直観的な表現を用いることによって、ユーザが発言の雰囲気を把握しやすくなることも考えられる。

## 1.4 本研究の目的

本研究では、インスタントメッセージングサービスにおけるプレゼンス情報の提示手法を参考に、周囲の情報を提供する方法を提案する。また、その情報を用いる場合に適した発言表示の方法を取り入れたチャットツールを開発する。

## 1.5 本論文の構成

第2章では、本研究で開発するシステムの目標を示す。第3章では、目標を達成するための基本的なアイデアを述べ、第4章ではそのシステムの主な機能を挙げる。第5章でそのシステムの実装と、システムの処理の流れについて説明する。第6章では関連研究について触れ、第7章で結論を述べる。

## 第2章 本研究の目標

本研究では、大勢で会話をすることができるチャットを基礎として、1.3 で述べた同期式コミュニケーションツールの可能性を考慮に入れたコミュニケーションツールを作成することを目標とする。目標を達成するためには、以下に挙げる機能を実装することが必要である。

### 2.1 相手やその周囲の様子を見ながら会話ができる

ユーザに直接人と話すときのような自然で臨場感のあるコミュニケーションの感覚を与えるために、相手の様子とその周囲の様子が見える機能を実装する。このとき、プレゼンス情報が表示される部分とメッセージが表示される部分を分離しておらず、ユーザを会話に集中させる方法を目指す。

また、過去に行われたチャットの内容を、当時の様子を見ながら振り返ることのできる機能を実装する。

### 2.2 複数の話題の理解を助ける

複数の話題が同時に行われる場合においても、ユーザが話題を判断することに煩わされないように、発言に二次元的な情報を付加することのできる機能を実装する。

### 2.3 発言の雰囲気表現する

ユーザが発言の雰囲気を補うために記号や顔文字を使用した際、ユーザが直感的にその雰囲気を把握できるように、発言内容に画像や色などの情報を付加する機能を実装する。

## 第3章 フキダシチャットシステムの提案

第2章で述べた目標を達成するために、ライブカメラで撮影された部屋の画像と、マンガで用いられるフキダシを利用するチャットシステムを作成する。

ここで、本研究で提案するシステムは、オフィス内の仲間内で使用することを想定したチャットシステムとする。会話が行われる範囲は、同室にいる人物同士であると仮定する。

実際に、同じ部屋にいる人同士でも、直接会話をするのではなく、インスタントメッセージを用いてコミュニケーションをとることがある。PCで作業をしながらコミュニケーションを取ることができる点が大きな理由である。また、声を出して会話をすると、周りにはいる作業中の人の気を散らせてしまう懸念があるため、このような方法をとることもある。

### 3.1 ライブカメラの利用

図3.1は、部屋の隅に設置されたライブカメラの写真である。ライブカメラの画像は、遠隔地間のコミュニケーションを図る方法としてよく使われている。ライブカメラは、その場所の状況をそのまま映像として提供するので、プレゼンス情報として利用することができる。



図 3.1 : 部屋に設置されたカメラ

ライブカメラの画像を用いると、部屋にいるのかどうかや、PCの前に座っているかどうかを判断することができる。図3.2は、図3.1で示したライブカメラで撮影した画像であり、手前の島に3人、奥の島に2人がデスクに着席していることがわかる。



図 3.2 : ライブカメラの画像の例

## 3.2 フキダシの表示

プレゼンス情報としてのライブカメラの画像を見ながらコミュニケーションをすることを可能にするために、マンガで用いられるフキダシを利用する。

フキダシは、発言内容を含む部分と発言者が誰であるかを示す部分からなる。前者をフキダシのセリフ部分、後者をフキダシのリンク部分と呼ぶことにする(図 3.3)。フキダシがセリフ部分のほかにリンク部分を持つことで、多人数で会話をする場合においてもどの発言が誰のものなのかについて、ユーザが理解することを助ける。

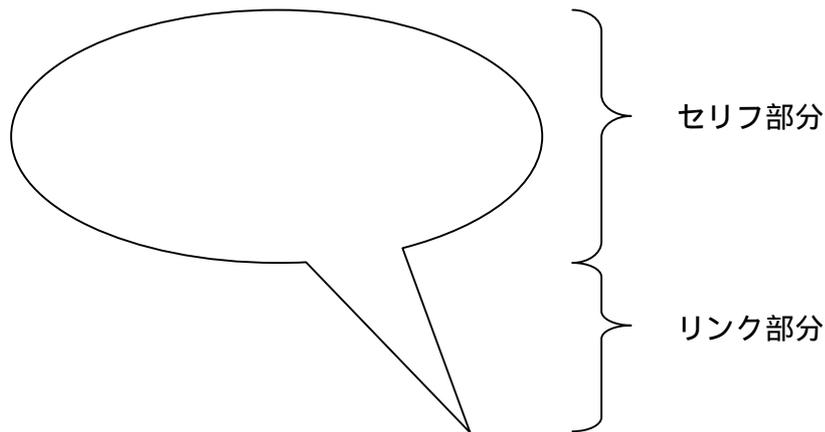


図 3.3 : フキダシの部品の名称

### 3.2.1 フキダシの表示位置と表示方法

ライブカメラ画像上に映し出される人物の付近にフキダシを表示することが、もっとも自然な表示位置であろう。しかし、この方法ではチャットにおいて重要な情報である発言の順番がわからなくなってしまう。図 3.4 示す様子は2人の会話のイメージであるが、どちらの発言が先であるか不明である。

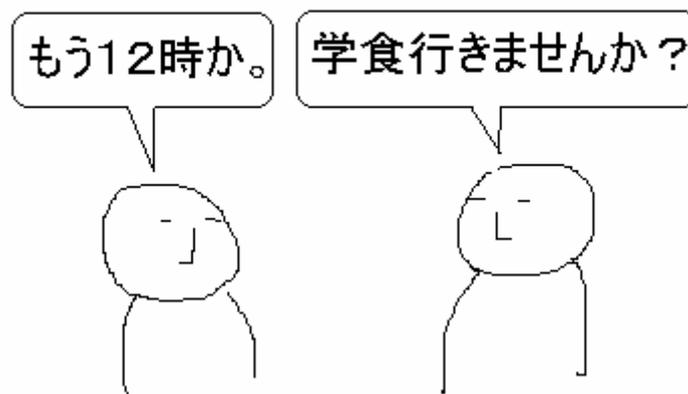


図 3.4 : 発言の順序がわからないフキダシ

このことを解決するために、フキダシの位置を人物の上に固定するのではなく、画面上をフキダシが右から左に流れるような方法を取り入れる。これにより、画面上に複数のフキダシが表示されている場合でもどの発言が新しいものなのかがわかるようになる。この方法では人物の近くにフキダシがあるとは限らないが、フキダシはリンク部分を保持しているため、どのフキダシが誰の発言であるか混乱することはない。図 3.5 にはフキダシが2つ表示されているが、左に表示されている「学食行かない?」の発言の方が先に行われたことがわかる。また、図 3.5 から時間が経った様子を、図 3.6 に示す。

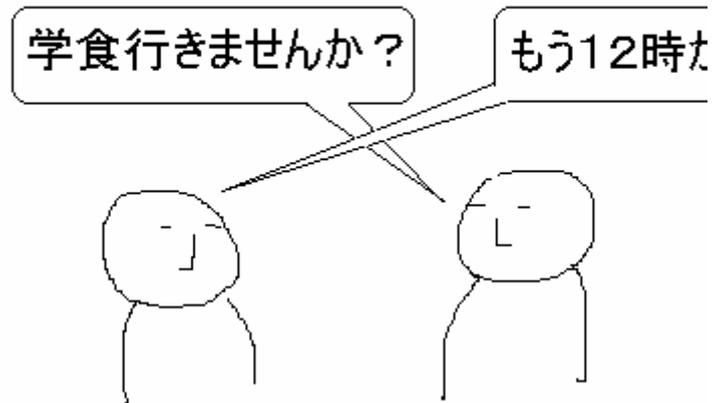


図 3.5 : 発言の順序がわかるフキダシ

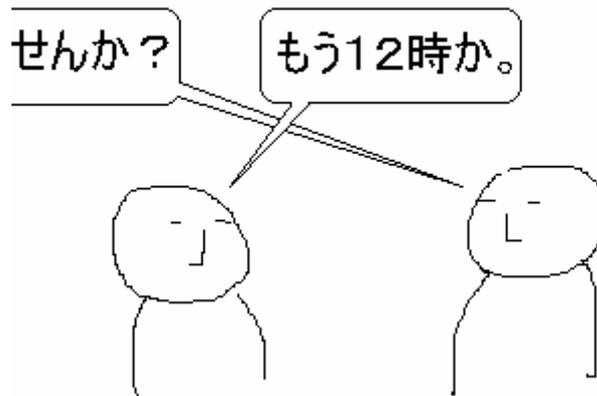


図 3.6 : 図 3.5 から時間が経った様子

### 3.2.2 複数の話題に対する方法

図 3.7 は、4 つの発言が画面に表示されるイメージを表したものである。このように複数の話題が並行して一度に行われると、どのフキダシが何の話題についてのものなのか、すぐにわからない。

たとえば、D の卒業論文の進度が気になる C は、D が返答するまで A と B の発言内容を確認する必要がある。

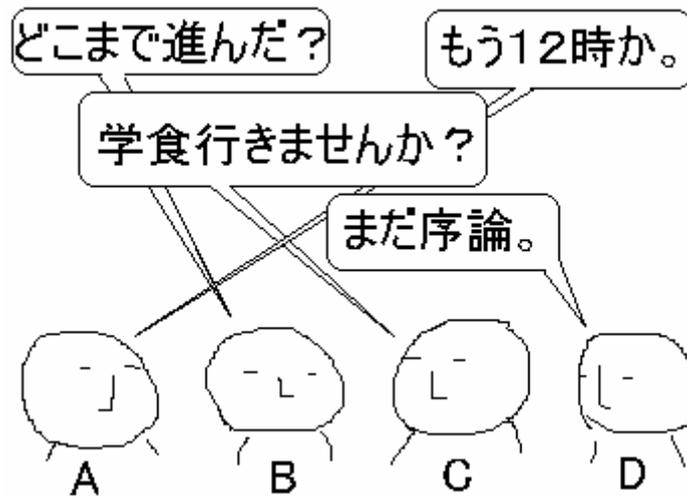


図 3.7：2つの話題が並行して行われる例

この問題は、フキダシの縦座標を任意に変更できるようにすることで解決できる。

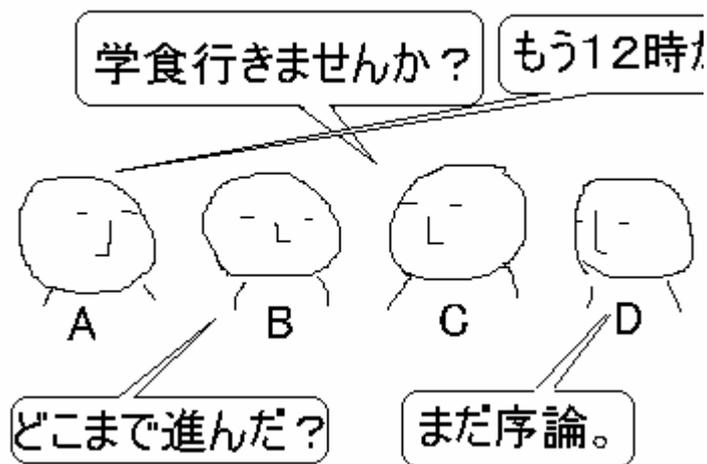


図 3.8：フキダシの位置を変更することで、2つの話題がわかりやすくなる様子

図 3.8 のように、Dの卒業論文の進度が気になるCは、AとBの発言をあまり気にせず、Dの返答を待つことができる。

### 3.3 発言の雰囲気表現

このシステムでは、フキダシの種類や色を変えることによって、チャットでは伝わりにくい発言の雰囲気を表現する。

#### 3.3.1 フキダシの形

マンガで使用されるフキダシには、発言者と発言内容のほかにも、発言者の声の大きさや発言の雰囲気を示す情報を含む。通常は図 3.9 のAのような形のフキダシが使われることが

多い。図 3.9 の B で示すものは大声や声を張り上げるようなセリフに使われる。図 3.9 の C はたとえばささやき声や独り言などに使われる。

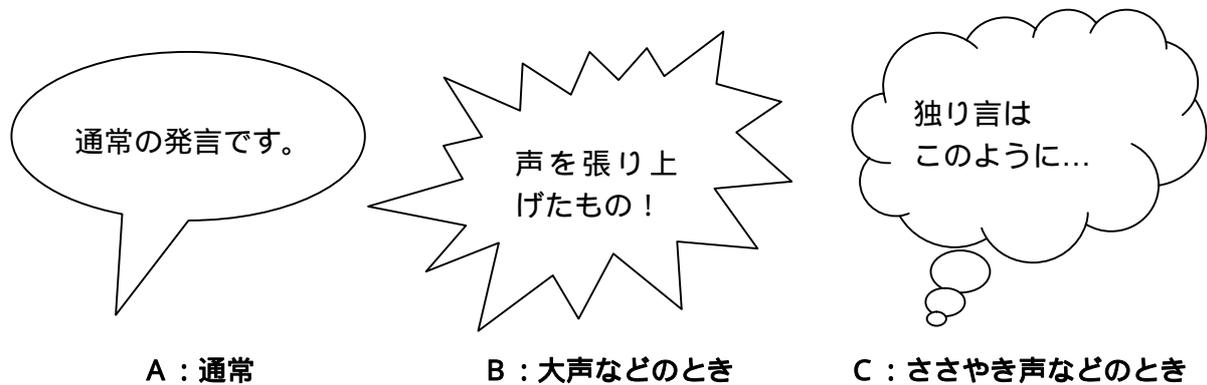


図 3.9 : フキダシの種類

本システムでは、発言内容に応じてフキダシの形を変化させることで、発言の雰囲気の理解を支援する。

### 3.3.2 フキダシの色

3.3.1 で挙げた雰囲気の情報に補うため、発言内容に応じてフキダシの色を変化させる機能も実装する。

## 3.4 チャットログの再現

3.2.1 で挙げた方法では、古い発言は図 3.6 で示したように画面外に流れてしまう。本研究で開発するシステムでは、画面外に流れてしまったログを、スライダを用いて閲覧できるようにする。

システムにライブカメラの画像とチャットの発言を保存しておき、過去の部屋の様子と発言を対にして確認することができるような機能を実装する。

## 第4章 フキダシチャットシステムの機能

この章では、フキダシチャットシステムの主な機能を紹介する。

### 4.1 起動

#### 4.1.1 画像の更新

ユーザがシステムを起動すると、画面上に部屋の画像が表示され、以後一定時間ごとに画像が更新される。ユーザはこの更新される画像を見ながら会話をする。

#### 4.1.2 ログイン

ログイン操作は、画像に映っているユーザの顔の部分をクリックすることであり、その座標がフキダシのリンクの先端となる。図 4.1 は、右手前にいる人物がログインをしている様子を示したものである。



図 4.1 : ログインの様子

## 4.2 発言方法

### 4.2.1 発言の入力と送信

画面下のテキストフィールドに発言内容を入力し、テキストフィールド右の send ボタンをクリックすることで、発言することができる。テキストフィールド内にカーソルがある場合にエンターキーを押下することでも、ユーザは発言をすることができる。

### 4.2.2 フキダシの表示

発言が完了すると、テキストフィールドがクリアされ、図 4.2 のように発言内容がフキダシとなって画面右上に出現する。



図 4.2 : 発言内容がフキダシとして画面上に現れた様子

画面の右上に表示されたフキダシは、時間とともに画面左方向へ流れて行く。画面上に複数のフキダシが表示されている場合、一番右にあるものがもっとも新しい発言である。

図 4.3 は、図 4.2 に表示された発言に対しての返答があった図で、「学食行かない?」という内容のフキダシが図 4.2 のときと比べて左方向へ流れていることがわかる。



図 4.3 : 図 4.2 のときから時間が経った様子

### 4.3 フキダシ発生位置変更方法

フキダシは通常画面上部に現れるが、複数の話題が並行して行われる場合、異なる話題が一行に表示されてしまう。そのためユーザは、話の流れがわからなくなってしまう。このようなときは、テキストフィールドに“toggle\_y”と打ち込むことによって、以後のフキダシを画面下部に流すことができる。図 4.3 において、人物 A と B が発言した内容は通常の場合に出現しているが、テキストフィールドに C、D のフキダシは、画面下部に現れている。



図 4.4 : フキダシの縦座標を位置任意に変更した例

## 4.4 発言の雰囲気表現する方法

発言に「！」や「…」など、発言の雰囲気を表す文字列を含ませることで、フキダシの色や形を変化させることができる。

また、テキストフィールドに“f\_bikkuri”または“f\_sasayaki”と入力することで、以後発言内容に関わらず、フキダシを図 3.6 のBまたは図 3.6 のCのような形にすることができる。

他にも、“f\_happy”または“f\_sad”と入力することで、フキダシの色をピンク色もしくは水色に変更することができる。

なお、これらのコマンド操作を取り消すときには、“f\_reset”とテキストフィールドに入力すればよい。

フキダシの形に影響する文字列とそれに対応するフキダシの形の一覧を、表 4.1 に示す。また、実際にさまざまなフキダシで会話をしている様子を、図 4.5 に示す。

表 4.1 : フキダシの形に対応するコマンドと文字列

フキダシの形や色	コマンド	文字列
図 3.9 の B の形	f_bikkuri	! !
図 3.9 の C の形	f_sasayaki	... . . . . .
ピンク色	f_happy	:-) :) ^^ ^_^ ^_^
水色	f_sad	:-( :( :-(- :-(-



図 4.5 : 様々なフキダシで会話をする様子

## 4.5 ログ閲覧の方法

図 4.6 は、図 4.3 の時刻からさらに時間が経過したときの画面である。このように、フキダシが出現してからある程度時間が経つと、古い発言は画面の左外へ流れて表示されなくなってしまう。



図 4.6 : フキダシが画面外に流れてしまった様子

しかし、画面上部のスライダを動かすことによって、以前に行われた発言を見ることが可能である。

スライダの位置は、一定時間ごとに保存された一連の画像と対応しており、スライダの位置に応じた時刻の部屋の画像が画面上に表示される。

また、ひとつひとつの発言には、どの画像のときに発言されたのかものであるかを示す情報が関連付けられており、スライダで指定した画像のときに発言されたフキダシを表示する。ある特定の画像のときに複数の発言があった場合は、左から発言順に表示される。図 4.7 は、スライダが指定した時刻において 2 つの発言が行われていたことを示す。



図 4.7 : スライダの操作によって過去のチャットの様子が表示された様子

## 第5章 フキダシチャットシステムの実装

### 5.1 ハードウェアの構成

フキダシチャットシステムは、部屋の画像を撮影するライブカメラ1台とサーバーPC1台、そして複数台のクライアントPCから構成される。

#### 5.1.1 ライブカメラ

ライブカメラには、AXIS 205 ネットワークカメラ[2]を使用した。ライブカメラはウェブサーバ内蔵型であり、http サーバとしてネットワーク上に存在している。カメラのスナップショットは、“http://[ライブカメラの IP アドレス]/jpg/image.jpg” という URL にアクセスすることによって取得することができる。

#### 5.1.2 サーバーPC

サーバーPCが起動すると、次の ~ で示すタスクを一定時間ごとに行う。図 5.1 は、処理の大まかな流れを示したものである。

ライブカメラから画像データ取得する。

共有フォルダ内に、画像データを保存する。

共有フォルダ内のログファイルに、現在の最新画像番号を書き込む。

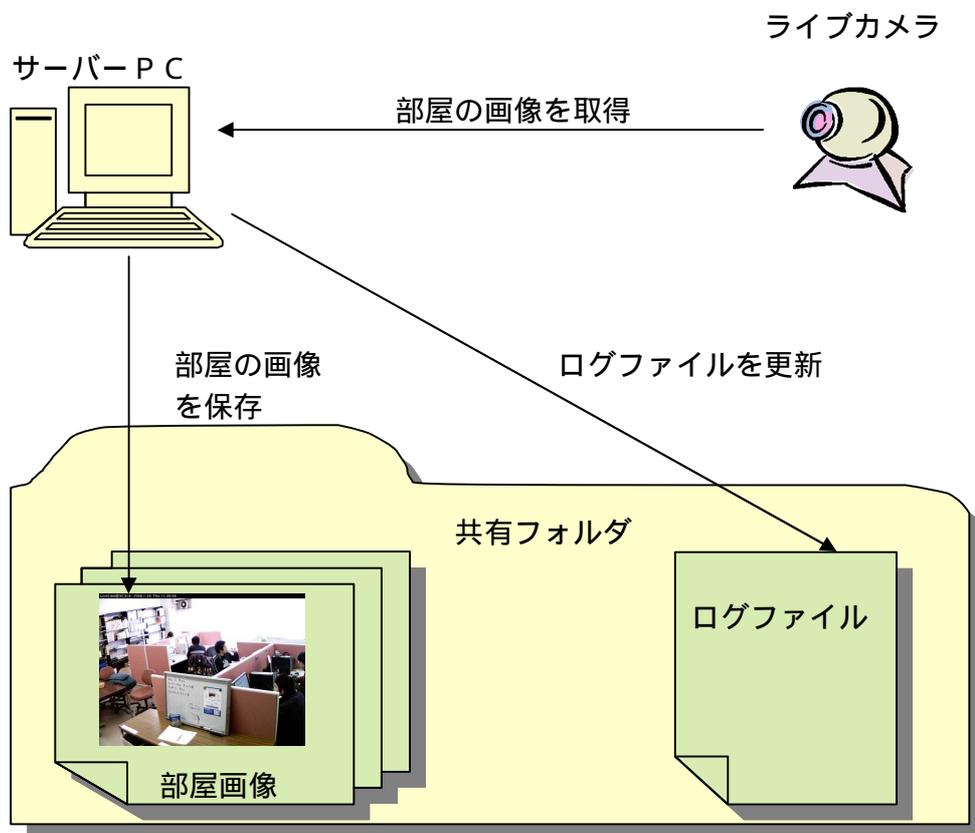


図 5.1 : サーバーPCの大まかな処理の流れ

### 5.1.3 クライアントPC

クライアントPCには、チャットモードとログ閲覧モードの動作がある。

#### チャットモード時の動作

クライアントPCは、次の ~ で示すタスクを、一定時間ごとに繰り返す。図 5.2 は、処理の大まかな流れを示したものである。

共有フォルダのログファイルから、未取得の発言を取得する。

ログファイルの画像番号が増加していた場合、共有フォルダから最新の画像を取得し、ディスプレイに表示する。

最新の発言を表示し、表示されているフキダシを移動させる。

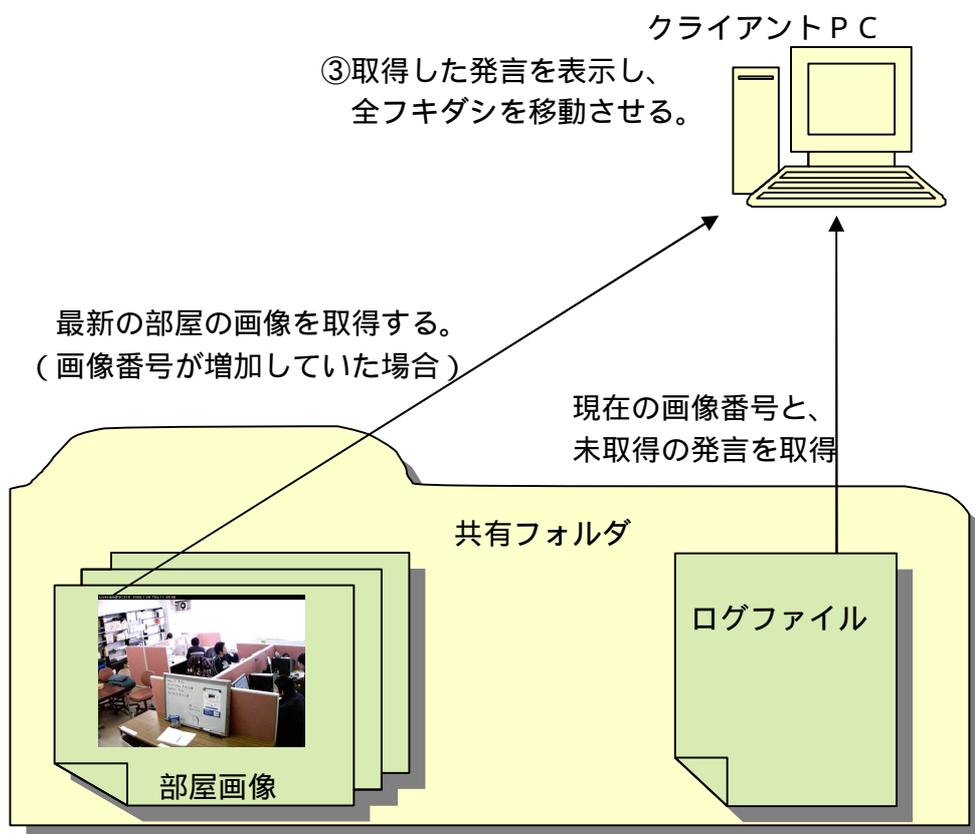


図 5.2 : クライアントPCの大まかな処理の流れ

ユーザが発言を行ったときは、上で示したタスクに割り込む形で、以下のA～Cの動作を行う。図 5.3 に、このときの処理の流れを示す。

A) 最新の画像番号を取得する。

B) ログファイルに発言内容を書き込む。

C) 発言を画面に表示する。

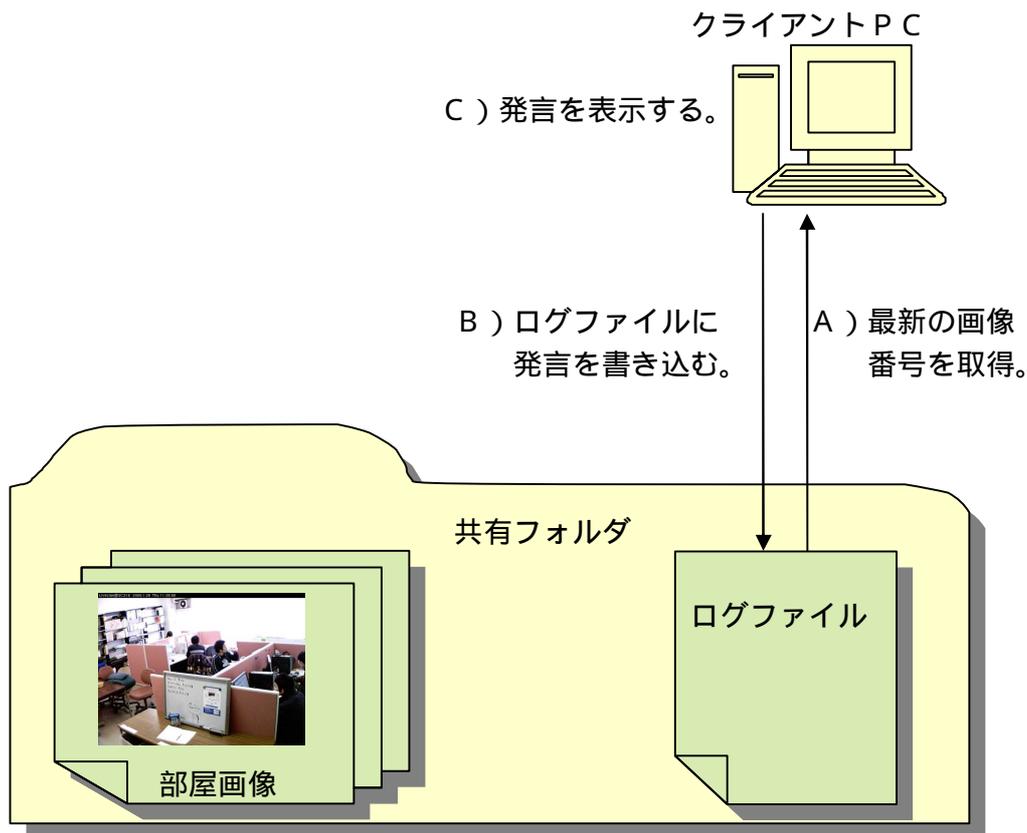


図 5.3 : クライアント P C の発言時の処理

#### ログ閲覧モード時の動作

ユーザがスライダを動かした場合、クライアント P C はログ閲覧モードに入る。このときの処理の流れを ~ で示す。図 5.4 に、このときの処理の流れを示す。

スライダの位置に応じた画像ファイルを取得する。

取得した画像番号のときのログを取得する。

ログの内容をフキダシとして、画像とともに表示する。

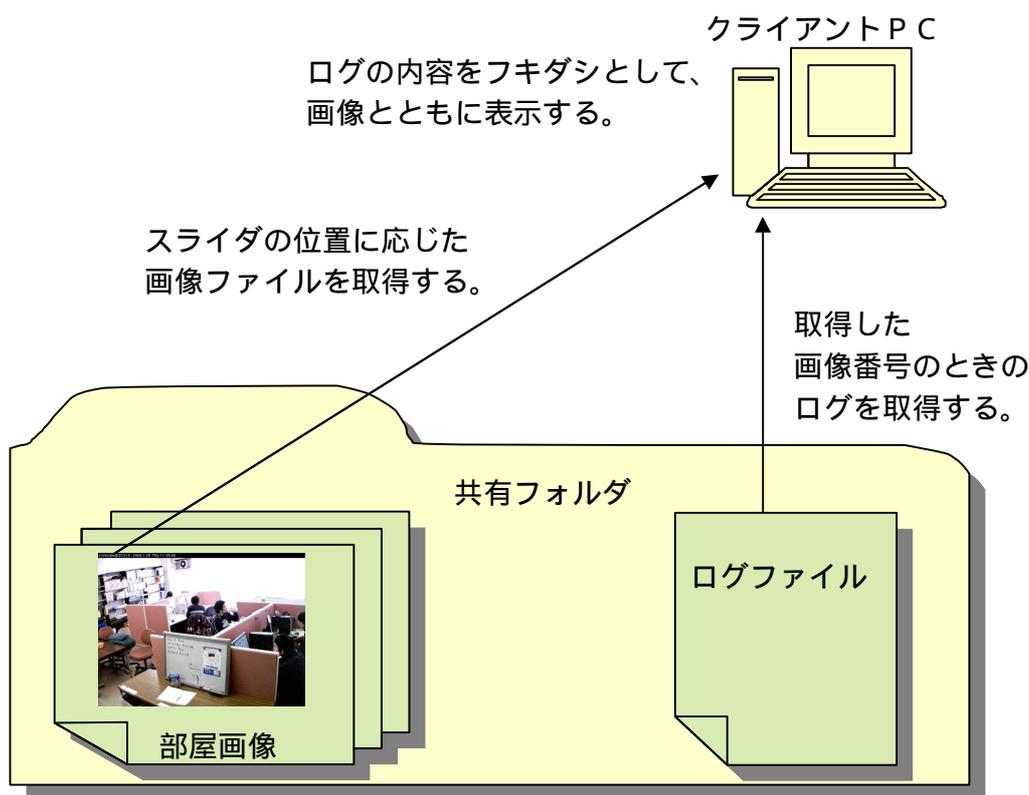


図 5.4 : クライアント P C のログ閲覧時の処理

## 5.2 共有フォルダの利用

本システムでのデータ処理は、チャットサーバは使わずに、簡単のために共有フォルダ内のファイル操作で擬似的にチャットを再現している。

1 台のサーバー P C と、複数台のクライアント P C はそれぞれ直接通信をする手法を持たない。そのためそれぞれのクライアント P C は、現在何番目のライブカメラ画像が最新であるかを知ることができない。それらを同期させるために、ログファイルの画像番号を用いる。

### 5.2.1 ログファイル

ログファイルはテキストファイルであり、表 5.1 の形式で表されるものである。実際は発言番号から発言内容までの 8 つのフィールドからなり、それぞれのフィールドは半角カンマ ( , ) で区切られている。

発言番号は、一連のチャットセッションにおいて何番目の発言であることを示す。画像番号は、一連のチャットセッションにおいて何番目のライブカメラ画像のときに発言されたかを示す。x 座標と y 座標は、フキダシのリンクの先端の座標を示す。フキダシの発生位置は、フキダシが上部と下部のどちらに生成されたのかを整数で保持する。フキダシの形、フキダシの色は、それぞれについての情報を整数で保持する。発言内容は、テキストフィールドから入力された文字列である。

表 5.1 : 画像更新時のログファイルの内容

フィールド	発言番号	画像番号	x座標	y座標	フキダシ発生位置	フキダシの形	フキダシの色	発言内容
例 1	12	9	300	200	0	0	0	学食行かない？
例 2	null	10	null	null	null	null	null	null
例 3	13	10	50	300	0	0	0	もう昼休みか。

表 5.1 には、例 1 で示す発言番号 12 と、例 3 で示す発言番号 13 のフキダシの情報が含まれている。これは、クライアント PC が発言をする際に書き込まれるログである。クライアント PC が発言する際には、発言番号の値を 1 増加させ、画像番号は前と同じ値を書き込む。

一方で、サーバー PC が画像を更新する際に書き込むログは、表 5.1 中の例 2 のような形式であり、画像番号以外のフィールドが全て null で満たされている。サーバー PC が画像を更新する際に書き込む画像番号は、以前の値から 1 増加させたものである。

表 5.1 の例 1 と例 3 の画像番号が異なるのは、例 2 によって最新の画像番号がインクリメントされた結果である。

#### 5.2.2 同期の方法

サーバー PC と複数台のクライアント PC は処理や処理速度の違いから、画像更新や画像取得のタイミングがずれてしまうことが考えられる。しかし、クライアント PC が画像を取得するタイミングを、ログファイルの画像番号が変更されていたときに定めることで、サーバー PC とクライアント PC を擬似的に同期させることができる。

### 5.3 サーバー PC の仕組み

サーバー PC は主に、ネットワーク上のライブカメラから一定時間ごとに画像を取得して共有フォルダに保存する ImageManager クラスと、共有フォルダ上のログファイルに現在の最新画像番号を書き込む LogManager の 2 つのクラスからなる。

ImageManager オブジェクトは、新しい画像を取得するたびに連番のファイル名の画像ファイルとしてサーバーに保存している。画像番号はこのときにインクリメントされ、LogManager の更新に反映される。

なお、実装は Java で行った。

### 5.4 クライアント PC のソフトウェア

#### 5.4.1 主なクラス

クライアント PC の主要なクラスは以下の通りである。サーバー PC と同様に、実装は Java で行った。

##### **FukidasiChat** クラス

チャットモードとログ閲覧モードの切り替えを管理している。また、マウスアクション、

テキストフィールド、スライダなどのリスナーを保持している。

### **MyCanvas クラス**

共有フォルダに保存されたライブカメラ画像を、一定時間ごとに取得し画面に表示する。同時に画像を保存している。またフキダシの描画もしている。

### **FukidasiManager クラス**

画面上に表示されているフキダシの移動等を管理する。新しいフキダシの作成や、画面外に流れたフキダシの消去も担当している。

### **Fukidasi クラス**

それぞれのフキダシの色や形、フキダシそのものの位置や発言者の位置を保持している。

### **LogManager クラス**

ログファイルへの書き込みと、過去のログを表示する機能を持つ。

## 5.4.2 処理の流れ

### **画像の取得**

システムが起動されると、FukidasiChat オブジェクトは MyCanvas オブジェクトに対して画像取得を命令する。タスクを受け取った MyCanvas オブジェクトは、共有フォルダ内にある最新画像の取得と表示を行う。

### **ログイン**

本システムでは、ユーザがクリックした位置をフキダシのリンクの先端の位置とする。FukidasiManager によって取得されたクリックの座標情報は FukidasiManager オブジェクトの `point_x`, `point_y` 変数に格納される。将来各 Fukidasi オブジェクトのリンク部分は、座標(`point_x`, `point_y`)を先端として描画される。

### **フキダシの生成**

ユーザが、画面下のテキストフィールドに発言を入力し、エンターキーを押下するか、右の `send` ボタンをクリックすることで、FukidasiManager オブジェクトの `makeFukidasi()` メソッドが実行される。

`makeFukidasi()` の引数として FukidasiManager に渡された文字列は、Fukidasi クラスのコンストラクタに渡されて Fukidasi オブジェクトとなる。

### **フキダシの色や形に関する処理**

FukidasiManager に渡された発言内容を示す文字列に“！”や“...”などが含まれている場合、Fukidasi オブジェクトを生成する際に、形を表すメンバーが変更される。また、“:-)”や“:-("などが含まれる場合は、色をあらわすメンバーが変更される。

Fukidasi オブジェクトの形や色を表すメンバーの値によって、MyCanvas の `draw` メソッドにそれぞれの色や形を描画させるタスクを渡す。

FukidasiManager に渡された文字列が、“`f_bikkuri`”などフキダシの色や形を変化させるコマンドに一致した場合、FukidasiManager が保持する形や色を変更する。以後の Fukidasi

オブジェクトは、変更された形や色の情報を保持して生成されることになる。具体的には、通常の形は0、図 3.9 の B の形は1、図 3.9 の C の形は2であり、通常の色は0、ピンク色は1、水色は2である。

FukidasiManager に渡された文字列が “ f\_reset ” であった場合、FukidasiManager が保持する色や形を表す変数が通常の変数値に変更される。

### 複数の話題に関する処理

FukidasiManager に渡された文字列が “ toggle\_y ” に一致した場合、FukidasiManager のフキダシ発生位置を管理するメンバーが変更される。表 5.1 の「フキダシ発生位置」は通常0の値を持つが、この操作を行うたびに0と1の値が交互に代入される。これによって以後に生成されるFukidasiオブジェクトの発生位置を表すメンバーが変更され、フキダシは0のときは画面上部、1のときは画面下部に描画される。

### ログファイルの更新

新たなFukidasiオブジェクトが生成される度に、LogManager が呼び出され、ログファイルに発言の情報が書き込まれる。

### ログファイルの確認と新規画像取得

FukidasiChat オブジェクトは、一定時間ごとにログファイルを読み込み、新しい発言があるかどうかを確認する。新しい発言があった場合は、ログファイルに記述された情報をもとにフキダシを生成する。

このとき、ログファイルの画像番号が増加していた場合、MyCanvas オブジェクトに画像の取得を命令する。

その後FukidasiChat オブジェクトは、FukidasiManager の moveFukidasi()メソッドを実行する。これにより、画面上に表示されているそれぞれのFukidasi オブジェクトの move()メソッドが実行される。この操作により、各Fukidasi オブジェクトの位置座標が変更される。

その後FukidasiChat オブジェクトは、MyCanvas オブジェクトの再描画関数 repaint() を実行し、各Fukidasi オブジェクト移動が完了する。

### ログ観閲の処理

スライダが操作されると、共有フォルダ上の最新の画像取得やフキダシの移動などの処理はいったん停止し、ログ観閲モードに切り替わる。

ログ観閲モードでは、FukidasiChat オブジェクトはスライダの位置に対応した画像番号の画像を共有フォルダから取得し、MyCanvas 上に表示する。また、その画像番号に対応する発言をログファイルから取得し、FukidasiManager の makeFukidasi() を実行する。

最後にFukidasiChat オブジェクトがMyCanvas を再描画することによって、画面上に過去の画像とフキダシが表示される。

## 第6章 関連研究

高橋伸らによる[3]は、ライブカメラの画像をプレゼンス情報として利用するシステムであり、本研究と同様にライブカメラの画像をプレゼンス情報として用いている。また、画像だけではわかりにくい活動状況を、マンガの効果線やフキダシによって表現している。これはプレゼンス情報の伝達が主な目的であり、会話を主な目的としている本研究とは異なるものである。

D. Kurlander らによる[4]は、発言と同時にキャラクタとフキダシを画面上に表示し、マンガのコマを新たに作成する感覚でチャットを行うものである。フキダシを用いることによって、ユーザの理解を補助する点は本研究と同様である。しかしこれは仮想空間上においてアバターを用いることでコミュニケーションを行うもので、実世界の画像をプレゼンス情報として使用している本研究とは異なるものである。

Hua Wang らによる[5]は、テキストの色や形、大きさをアニメーションで変化させることによって、チャットにおける感情情報を強調するものである。文字以外の情報を用いることでユーザのコミュニケーションの理解を支援する点と、ユーザの顔の画像を使用する点は、本研究と類似している。しかし、ユーザの周囲の情報を表示する機能がない点と、画像とチャットの表示部分が分離している点が本研究とは異なる。

## 第7章 結論

本論文では、まずPCを用いたコミュニケーションツールの特徴を示し、同期式のコミュニケーションツールであるチャットとインスタントメッセージングサービスを拡張できる可能性について述べた。そしてその拡張手法として、ライブカメラ画像にフキダシを重ねることで、相手の様子を見ながらチャットを行うシステムの提案と実装を行った。その際、フキダシの形や表示位置を考慮することで、ユーザに直観的なコミュニケーションを提供することができた。また、過去のチャットを表示する場合にも、当時の画像にフキダシを重ねることで、ユーザに臨場感を与えることが可能になった。

今後は、画像の表示方法やフキダシのあり方について考察し、複数の部屋においても使用できるように拡張することを考えている。また、インターフェースを改良し、ユーザにとってさらに自然なチャットシステムとなるように開発することを考えている。また、システムのユーザ評価を行って、更なる開発の方向を定めたいと考えている。

# 謝辞

本論文を執筆するに当たり、指導教員である田中二郎先生をはじめ、三末和男先生、志築文太郎先生、ならびに高橋伸先生には、幾度となく丁寧なご指導と適切な助言をいただきました。心より感謝申し上げます。

また、田中研究室の皆様にもゼミなどを通じて大変貴重なご意見をいただきました。特にNAIS チームの皆様には、チームミーティング以外にも日常的に多くのご意見やご指摘を頂きました。本当にありがとうございました。

また、家族、すべての友人、ならびにアルバイト先のご家族にも、いろいろな面で支えて頂きました。この場をお借りして御礼申し上げます。

## 参考文献

[1]神田 陽治著：『わかる！インスタントメッセージング』，オーム社，2002.

[2] AXIS 205 ネットワークカメラ  
<http://www.axiscom.co.jp/prod/205/>

[3] 高橋 伸, 岩淵志学, ジャッキーノ ヤン, 山田徹, 久松孝臣, 中村卓, 土持幸久, 金春明, 田中二郎. "ライブカメラ画像を用いたプレゼンス情報の表示手法", 第13回インタラクティブシステムとソフトウェアに関するワークショップ, 日本ソフトウェア科学会 WISS 2005, 2005年12月, pp. 15-18.

[4] D. Kurlander, T. Skelly, and D. Salesin. "Comic Chat", ACM Computer Graphics, Annual Conf. Series, pp. 225-236, 1996.

[5] Hua Wang, Takeo Igarashi, "Animated Chat", Demonstration Paper, 16th Annual Symposium on User Interface Software and Technology, ACM UIST'03, Demonstration paper, Vancouver, Canada, November 2-5, 2003.