

波紋を用いたインターネット上のざわめきの提示手法

岩淵 志学 久松 孝臣 高橋 伸 志築 文太郎 三末 和男 田中 二郎

筑波大学 システム情報工学研究科

知識創造の過程においてカジュアルコミュニケーションは重要である。その偶発的な発生要因として、周囲から聞こえる”ざわめき”がある。しかしながら、近年のインターネットの普及により、オフィスの物理的分散およびインターネットコミュニケーションツールの利用が進み、実世界でのざわめきをきっかけとした偶発的コミュニケーション発生が難しくなってきた。本研究では、インターネット上においても実世界と同様にざわめきをユーザに提供することで、実世界型の会話参加を支援することを目的とする。特に本論文では、ざわめきの表現手法としてデスクトップ画面上に波紋を提示する手法について述べる。また、我々は提案手法を実装したインターネットメッセージングツール RippleDesk を開発した。実験により、波紋の変化を通じて他人同士の会話の活性化がユーザに伝達可能であることが確認された。

Using Ripples to Represent Conversational Noise on Internet

Shigaku Iwabuchi Takaomi Hisamatsu Shin Takahashi
Buntarou Shizuki Kazuo Misue Jiro Tanaka

Department of Computer Science, University of Tsukuba

Conventional communication tools force the users to log into the system explicitly to knowing if other people have a conversation. In contrast, ripple sound in real world provides information of surrounding environment continuously and sometime creates an opportunity to make the listener to join to the conversation. In this paper we propose using ripple wave on the desktop screen to make people aware surrounding chat and aim to support their communication. We developed RippleDesk, an internet communication tool which uses ripple wave to notify the outbreak of other's conversation. As a result of the experiment, it is confirmed that ripples make the user aware of the start of a session of conversations.

1 はじめに

新たな知識やアイデアの創出のきっかけとして、休憩中の何気ない会話など、カジュアルなコミュニケーションは重要である。近年では、チャットやインスタントメッセージング等のインターネットを用いたコミュニケーションツールがこれらのカジュアルコミュニケーションに広く利用されている。

現在一般的に利用されているコミュニケーションツールでは、ユーザはシステムにログインする事で初めて会話に関する情報を得ることができる。そのため、自分の知らない間に興味のある話題を逃すなどの問題が発生する。また、会話が盛り上がっている場合に参加したい、と言った会話の状況に応じた機会的な会話への参加ができない。このような問題の本質的な原因は、システムが「会話をする」か「会話をしない」という二値的な選択肢を利用者に強要することにある。本研究ではこの問題の解決に向け、実世界におけるざわめきの有用性について注目している。実世界ではざわめきの受け手は常にオンライン状態であり、例えば友人の話し声を聴いて仕事の

休憩に入るなど、ざわめきから伝わる情報と自身のコンテキストによって、会話へ参加するかしないかの判断を自由に行うことが可能である。この点で、現在のインターネット上のコミュニケーション参加のプロセスは実世界と異なる。

本研究の目的は、インターネット上のコミュニケーションにおいてもざわめきの概念を導入し、実世界に近い形での会話の参加を実現することである。インターネット上のざわめきを表現するにはその表現方法が重要であるが、本論文では**波紋**によりざわめきを表現する手法を提案する。提案手法では波紋が持つ複数のパラメータにざわめきの情報をマッピングする。

さらに我々は、波紋を用いて周囲の会話のざわめきをユーザへ提供するコミュニケーションツール RippleDesk を開発した。RippleDesk はユーザ同士が簡易メッセージを交換することで会話を行うインスタントメッセージングであり、周囲のユーザの会話がデスクトップ画面の波紋によりざわめきとして提示される。

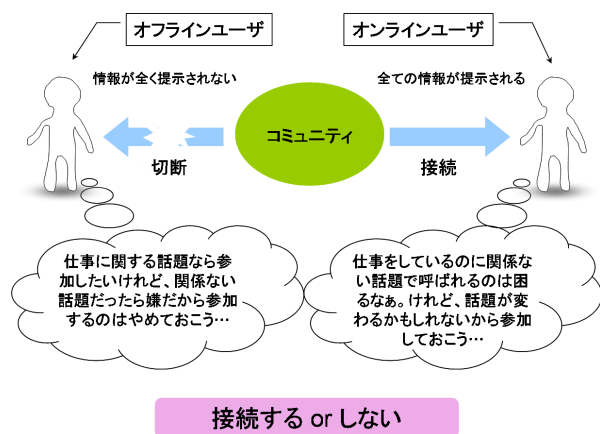


図 1: 二値的な選択の強要による問題

2 実世界との比較

表 1 に各コミュニケーションツールごとによく起きる問題をまとめた。これらの状況を分析するために、次に 2 種類のコンテキストを定義する。

会話コンテキスト チャットのチャンネルや掲示板などのインターネット上のコミュニティで起きている会話の情報を**会話コンテキスト**と呼ぶ。会話コンテキストには、会話そのものの内容の他に、会話へ参加しているユーザの名前、参加者の人数、ユーザのログインやログアウトなどの情報が含まれる。

自己コンテキスト 会話の受け手となるユーザがおかれている状況を**自己コンテキスト**と呼ぶ。自己コンテキストには、ユーザの忙しさなどの状態や、気分、興味、などが含まれる。

人は他人同士が会話している場所へ参加するかどうかを決定する際に、会話コンテキストと自己コンテキストの双方を考慮する。我々は既存のコミュニケーションツールの問題点は、会話コンテキストの情報をユーザへ完全に伝えるかもしくは完全に伝えないという二値的な提示方法しか持たないことに起因すると考えている (図 1)。オンライン状態においては、会話コンテキストをユーザに提示する際に、ユーザの自己コンテキストを考慮しない。そのため、もしユーザが他の作業で忙しい状態であっても、明示的に会話しているのと同じインタフェースで会話の情報が提示される。一方、オフライン状態においては、ユーザは会話コンテキストを得ることができない。

例を挙げる。チャットシステムでは、会話の内容、参加者のログイン・ログアウトの情報、参加者の名前、参加者の人数などが会話コンテキストとしてユーザへ提示される。ユーザはこれらの情報を知るため

には必然的に会話へ参加するためにシステムへログインしなくてはならない。また、チャットウィンドウを用いた明示的な会話は、コンピュータ上で会話をフォアグラウンドのタスクとして行っている場合には良いが、バックグラウンドで会話を行いたい場合には作業を阻害する可能性がある。しかしながら、もしシステムへログインしなければ、会話コンテキストは一切ユーザへ伝わらないため、ユーザは「盛り上がっている場合には参加したい」「自分に興味のある話題ならば参加したい」など、会話コンテキストによる選択的な参加をすることができない。

ここで、実世界におけるざわめきを考える。ざわめきは、大抵の場合には無意識のうちに無視されるが、しばしばコミュニケーションへ参加するきっかけとなる。例えば、仕事が一区切り付きそうな時にコーヒールームのほうへと歩く同僚の音が耳に入り、自分も休憩に入るような状況である。ざわめきはその受け手が必要とするか否かに関わらず常に提示されるという特徴を持つ。また、ざわめきを意識的に受け取るか、もしくは単に無視をするかは、その受け手が自由に判断することが可能である。つまり、ざわめきを完全に聴く状態か、もしくは完全に聴かないか状態か、という二値的な状態の強制がない点で、従来のインターネット上のコミュニケーション参加とは異なる。

3 提案:波紋によるざわめきの表現

実世界のざわめきによる会話への参加のプロセスをインターネット上において実現するためには、実世界のざわめきのように、聴こえて来る方向や強さなどを表現できる表現手法が重要となる。

我々はデスクトップの画面上に波紋を発生させることでインターネット上のざわめきを表現する手法を提案する。波紋を用いる主な理由は、波紋が複数のパラメータを持つことである。1つの波紋は、初期振幅と波長のパラメータを持つ。また、波紋を画面上に複数個発生させる場合には、波紋が起きる位置、波紋が起きる頻度の 2つのパラメータが加わり、合計 4つのパラメータが利用可能である。それぞれのパラメータに対してインターネット上の会話コンテキストと自己コンテキストをマッピングすることで、視覚的なざわめきとして表現することができる。

例えば、ユーザがとても興味を持つ会話が発生した時には、システムは大きな波紋によってその会話が起きていることを大々的に提示したり、逆にユーザの興味がとても小さい会話の場合には、無視できるような微かな波紋で提示を行うことが可能である。また、デスクトップ画面上の位置とざわめきの発生源となる人を関連付けることで、会話の内容を詳しく聞くことなく、波紋の発生する方向から会話をしている人を予測するようなことが可能となる。

表 1: 既存コミュニケーションツールが持つ問題の例

ツール	ストーリー
チャット	ログインしたが誰もいなかった。または、ログアウトしているうちに、楽しい話題を逃した。
インスタントメッセンジャー	AさんはBさんと仕事の会話をするためにメッセンジャーの状態を「取り込み中」から「オンライン」にした。しかし、Cさんがプライベートな話題で話しかけてきた。
大規模掲示板	自分に興味のある話題で多数の人が議論を繰り返していたが、たまたま掲示板を見なかったために機会を逃した。または、掲示板を見たが何も新しい情報が更新されていなかった。



図 2: 波紋によるざわめきの提示

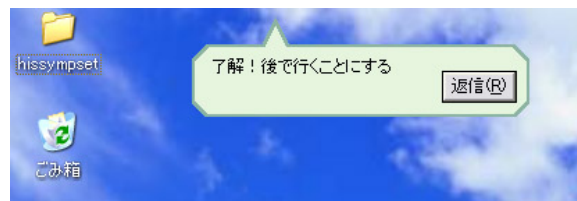


図 3: 吹き出しによる明示的な提示

4 RippleDesk の試作

我々は提案手法を用いたシステムのプロトタイプとして、コミュニケーションツール RippleDesk を開発した。

4.1 デザイン

RippleDesk は、**コミュニティ**と呼ばれる仮想的なインターネット上の場所にユーザ同士が集まり、簡易なメッセージを互いに交換するコミュニケーションツールである。ユーザは、必ず1つ以上のコミュニティに参加する。ユーザがコミュニティに参加すると、メッセージの送信など、ユーザが行うアクティビティはコミュニティに所属するメンバー全員にざわめきとして提示される。自分が所属するコミュニティは、デスクトップ画面の周囲に仮想的に配置される。これは、実世界において自分の周囲の様々な方向からざわめきが聴こえてくるメタファであり、ユーザはデスクトップの周囲のコミュニティからざわめきが聴こえるイメージを持つ。

普段、コミュニティから伝わるざわめきは、デスクトップ画面の波紋によってユーザに提示される (図 2)。もし、ユーザが明示的に会話を行いたい場合や、周囲のコミュニティで重要なアクティビティが起こった場合には、吹き出しによる明示的な提示が行われる (図 3)。

ざわめきとしてコミュニティ内のメンバーへ提供されるアクティビティの種類は、発言、ログイン・ログ

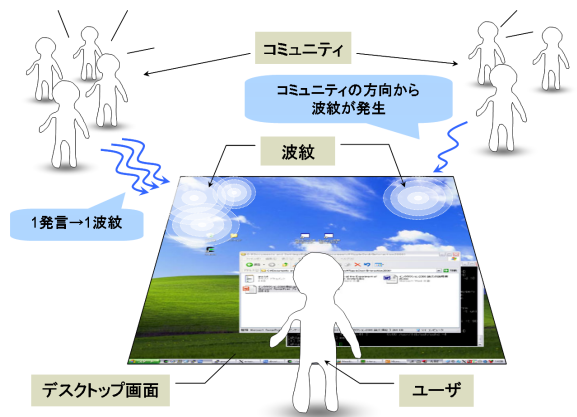


図 4: RippleDesk の概観

アウト、会話トピックの変更である。コミュニティのメンバーが1回発言すると、そのコミュニティに所属するメンバーのデスクトップ画面には波紋が1つ発生する。つまり、コミュニティ内での会話が活発になると、コミュニティの方向からは頻繁に波紋が発生し、デスクトップ画面の揺らぎが大きくなる。逆に、会話が起きていない場合には、デスクトップ画面に変化は起きない。

図 4 にコミュニティとデスクトップ上の波紋の関係を示す。また、図 5 に、RippleDesk における、会話コンテキストと波紋が持つパラメータのマッピングをまとめた。



図 5: 波紋のパラメータと会話コンテキストのマッピング

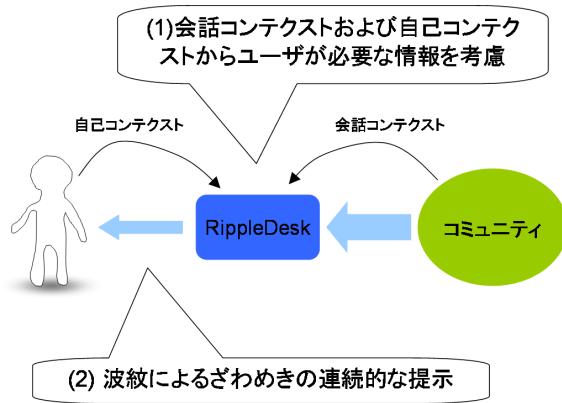


図 6: RippleDesk の位置付け

RippleDesk は常時オンライン状態であり、常にユーザーが所属するコミュニティの会話コンテキストとユーザーの自己コンテキストを取り入れる (図 6)。そして、ユーザーが所属するコミュニティでアクティビティが発生した場合には、そのアクティビティがどの程度ユーザーから必要とされるかを 2 つのコンテキストから決定する、必要の度合いに応じて提示方法を変化させる。ユーザーが忙しい時などは、デスクトップ画面に変化はほとんどなくなり、従来のシステムで言えばオフラインに近い状態となる。しかし、自分に興味がある話が発生した場合などは、吹き出しにより明示的に会話が提示され、従来システムのオンラインに近い状態となる。つまり、RippleDesk はオンライン・オフラインの二値的な選択を強制しない。

次に、メッセージの受信とコンテキストに応じた提示方法について詳しく述べる。

4.2 受信メッセージの提示

メッセージを受信した際の提示方法は、会話コンテキストと自己コンテキストの双方に依存した**インパクト係数** I の値によって変化する。

インパクト係数はすなわち、会話コンテキストと自己コンテキストを考慮した結果、受信したメッセー

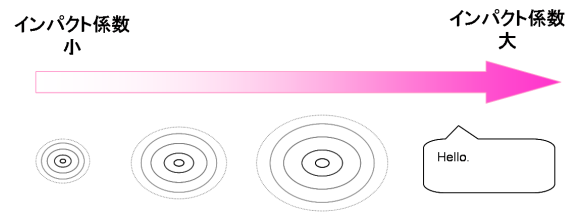


図 7: インパクト係数による提示方法の違い

ジの必要らしさを表す。 I は 0.0 から 1.0 までの正の実数である。もし I が 1.0 に近ければ RippleDesk はメッセージを吹き出しにより提示する。 I が 0.0 に近ければ波紋を用いて提示する (図 7)。

現在のバージョンの RippleDesk では I の値は次のように定義されている。

$$I = \frac{\phi_{kwd} + \phi_{com} + \phi_{app}}{3} \quad (1)$$

ここで、 ϕ_{kwd} はメッセージの中に含まれているキーワードの重みである。何もキーワードが含まれていない場合には 0.5 である。 ϕ_{com} はメッセージの発信源となるコミュニティに対してユーザーがあらかじめ設定した重みである。 ϕ_{app} はメッセージを受信した時点でフォアグラウンドで実行されているアプリケーションに対して、ユーザーがあらかじめ設定した重みである。RippleDesk ではこれらの重みをユーザーが自由に設定することができる機能を持つ。詳しくは後述する。

インパクト係数がある閾値以上では波紋による提示は行われない。 I が大きいということは、ユーザーにとって極めて重要と思われるメッセージが受信された場合や、RippleDesk がフォアグラウンドのタスクとなっているなど、ユーザーが明示的にチャットを行う状態であることを表す。その場合には、もはや波紋によってざわめきを提示するよりも明示的なコミュニケーションのためのインタフェースが適当であるから、吹き出しによる提示が行われる。現在のバー

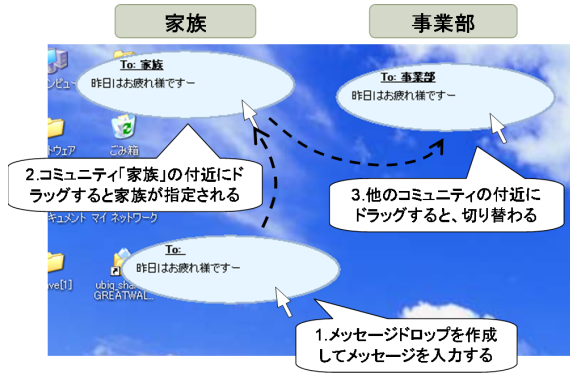


図 8: メッセージドロップの作成

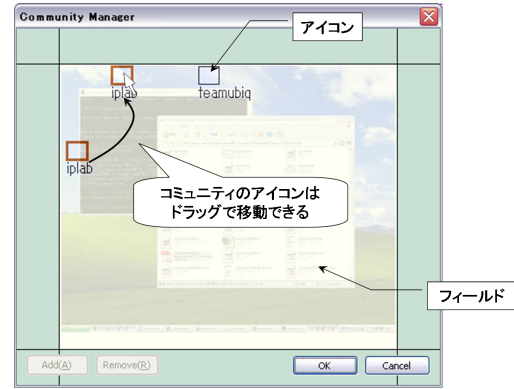


図 9: Community Manager

ジョンでは経験的な値として閾値は 0.58 に設定されている。

I の値が閾値より低く、波紋による提示を行う場合には、その初期振幅 a は次のようにメッセージの文字の長さに応じて変化する。

$$a = \sqrt{l_{msg}} \cdot I \quad (2)$$

l_{msg} は受信したメッセージの長さを表す。式が表すように、もし I が 0 に近い場合には波紋はほとんどユーザに認知されなくなる程に小さくなる。

4.3 メッセージの送信

コミュニティに向けてメッセージの送信を行うには、**メッセージドロップ**を作成する。メッセージドロップは水滴の形をしたウィンドウであり、ユーザはメッセージを書き込むことができる。図 8 にメッセージドロップの作成と送信方法を示す。メッセージドロップはシステムトレイに格納されている RippleDesk のアイコンをクリックすることで作成される。ユーザは、空のメッセージドロップにメッセージを入力した後、マウスでデスクトップ画面の周囲にドラッグする。コミュニティを設定した付近にドラッグすると、メッセージの宛先が指定される。宛先が指定された状態でマウスのボタンを離すと、宛先のコミュニティに属するメンバー全員にメッセージが送信される。

4.4 その他の機能

その他、RippleDesk は次のような機能を持つ。

4.4.1 コミュニティの設定

コミュニティをデスクトップの周囲に仮想的に配置するには Community Manager を用いる (図 9)。Community Manager は RippleDesk のアイコンを右クリックした時に表示されるメニューから起動することができる。ウィンドウ内には、デスクトップ

を模したフィールドとその周囲が図示される。追加ボタンにより、フィールドの周囲にコミュニティを示すアイコンが追加される。アイコンをマウスで自由に移動することで、ユーザはフィールドの周囲の任意の場所に仮想的にコミュニティを設置することが可能である。また、ユーザはコミュニティに対して重み ϕ_{com} を 0.0 から 1.0 で設定することができる。 ϕ_{com} は、ユーザがそのコミュニティに対して持つ関心の度合いである。

4.4.2 Cocktail Word の登録

Cocktail Word にキーワードとその重み ϕ_{kwd} を登録することで、文字数の少ないメッセージが受信された場合でも、自分にとって重要なキーワードが含まれている場合にはインパクト係数を上げることができる。実世界におけるざわめきには様々な音が混じり合っているが、人間は音の大小に関わらず聞きたい音を意識的に聞くことができる。これは Cocktail Party Effect と呼ばれる [1]。Cocktail Word はこの効果を実現するものである。Cocktail Word には、単語 1 つずつに対して 0.0 から 1.0 の範囲で重みが設定できる。メッセージ中に複数のキーワードが含まれる場合には、それぞれの関心度の最大の値が ϕ_{kwd} となる。

4.4.3 アプリケーションの重みの設定

RippleDesk は、フォアグラウンドで実行されているアプリケーションを監視し、そのアプリケーションの重み ϕ_{app} に応じてインパクト係数を調整する。 ϕ_{app} は、あるアプリケーションを実行している時にメッセージなどのアクティビティが受信された場合、ユーザがその情報にどの程度の興味を持つかの度合いである。例えばプレゼンテーションソフトなど、メッセージが届いてもまず対応する可能性が無いアプリケーションに対しては、0 に近い値を設定することで、ざわめきを抑制することができる。重みはコンピュータにインストールしてある各アプリケーションに対して、0.0 から 1.0 までの値で設定する。

4.4.4 受信履歴の参照

近い過去についてのメッセージの受信履歴を見るには、History Manager を利用する。History Manager にはスライダーコントロールがあり、これは現在から過去への時間を表す。スライダの位置を過去のある時点へ変更すると、その時から現在までのメッセージが時間軸に沿って全てデスクトップ上に再生される。表示は全てインパクト係数 1.0、すなわち吹き出しを用いて表示される。メッセージの再生速度は、デフォルト状態で実時間の 10 倍となっており、表示速度スライダーにより実時間再生から 100 倍速再生まで変更可能である。

また、システム導入時からの履歴のような遠い過去の履歴については、テキストログで確認することができる。

4.5 実装

RippleDesk は Windows プラットフォームを対象にしたアプリケーションである。実装は Microsoft Visual Studio .NET 2003 Visual C++ 7.0 を用いて行った。

RippleDesk の通信プロトコルは RFC1459 で定められている Internet Relay Chat(IRC) のプロトコルを拡張したものであり、RippleDesk はざわめきの提示機能を除いて一般の IRC クライアントと相互利用が可能である。RippleDesk は RFC1459 の PRIVMSG コマンドの送信先として、ざわめきの提示モジュールを仮想的に用意することで、会話コンテキストの情報をクライアントに提供する。ざわめきの提示モジュールは、PRIVMSG コマンドの第 2 パラメータへ RIPPLE というフィールドを指定することで利用可能である。

RippleDesk はデスクトップ画面を一定時間ごとにキャプチャし、仮想の波紋フィールドの状況に応じてキャプチャ画像を変形して描画する。仮想波紋のフィールドはデスクトップ画面と同じ大きさの 8 ビット画像であり、ピクセルの値はその場所の振幅値を表す。振幅の大きい場所ではキャプチャしたデスクトップの画像を大きく歪ませ、振幅が小さい場所ではそのまま描画することで、擬似的に水面の屈折を表現している。現在の実装では、仮想波紋フィールドの更新頻度は、計算面積とコンピュータの処理能力に依存する。開発環境である Pentium 4 2.0GHz 相当の CPU を搭載したコンピュータにおいて 1280x1024 ピクセルのデスクトップ解像度で利用する場合、アニメーションの速度は 15fps 程度である。

5 実験と考察

これまでに我々は、波紋が持つパラメータの内、特に位置のパラメータについて、波紋が発生した位置とその発信源となったコミュニティの場所の対応がどの程度正確にユーザに伝わるかを確かめた [2]。本章では、波紋の大きさや頻度を介して、会話の活発さがどの程度ユーザに伝達されるかについて、実験および考察を行う。

5.1 実験 1

実験 1 では、他人同士の会話において、発言が頻繁になる・長い発言があるなど、会話が活発になる兆候を RippleDesk の波紋によって捉えられるかどうかを検証する。

5.1.1 実験方法

まず、以下のような普段の会話のログを 6 分間記録した。

```
02:22:02(AAA) 会話例にふさわしいトークを w
02:22:11(BBB) 大丈夫、編集するよw
02:22:23(BBB) タイミングだけ正確に出ればとりあえずok
02:22:35(BBB) ナルトありがと>その
02:22:42(AAA) うい w
02:22:53(BBB) ではプログラムに戻ります
02:23:06(CCC) お互いがんばりましょう
02:23:16(BBB) ういお
02:24:08(AAA) おうー
02:25:47(BBB) 締め切り延長キター-----
02:25:53(BBB) 10・30マデ
02:25:58(AAA) なにい-----
```

このチャットのログには前半、中盤、後半と、大きく分けて 3 回の会話のセッションがある。セッションの間は、メッセージのやりとりが数秒に渡って停止する。前半のセッションの開始は緩やかに始まり、中盤のセッションは前半と比べて急に始まる。中盤のセッションの後しばらく会話が止み、突然大きな発言から後半のセッションが始まる。

実験のため、簡単なソフトウェアを作成した。実験用ソフトウェアは、ボタンをクリックすると実験開始からそのボタンがクリックされるまでの時間を記録する。

実験は、記録したチャットのログを再生し、コンピュータのデスクトップ画面上に会話に応じた波紋を 6 分間提示する。被験者には、波紋を見ながら「会話が活発になってきた」と思われる時点で実験用ソフトウェアのボタンをクリックするよう指示する。

実験の際には、インパクト係数に係わる重みの値は全て固定とし、波紋の初期振幅(大きさ)はメッセージの長さのみに依存する。被験者には、あらかじめ記録したログを再生することは伝えない。被験者は 5 名で、いずれも RippleDesk を使用した経験は無い。

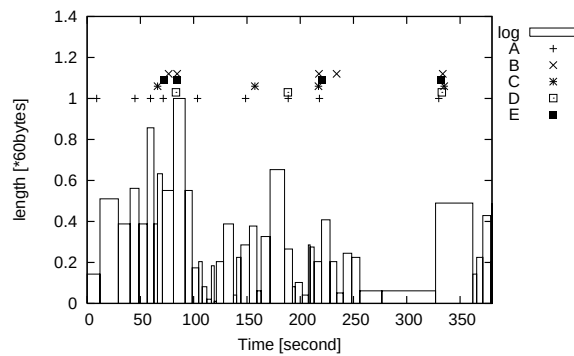


図 10: 実験結果: 発言とクリックポイントの関係

5.1.2 実験結果

ここでは、被験者がボタンをクリックした時点をクリックポイントと呼ぶことにする。図 10 に、会話量の変化および被験者 5 名のクリックポイントを示した。x 軸は時間軸であり、図中では右方向が会話の後半を表す。チャットのログは棒グラフで表されている。棒 1 本が 1 つの発言を表す。値はメッセージの長さを表し、太さは次の発言までの時間を表す。つまり、密度の濃い部分では頻繁に発言が起きていることを表す。比較のため、クリックポイントは y 軸の値を 1.0 から被験者ごとに 0.03 刻みでプロットしている。

5.2 実験 2

実験 1 において、会話の活発さがある時刻の 1 点として入力するよう被験者に求めた。実験 2 では、波紋から感じられる活発さを連続的な値として取得し、吹き出しによる明示的な会話を見た場合の活発さの変化と比較を行う。

5.2.1 実験方法

実験用に簡単なソフトウェアを作成した (図 11)。このソフトウェアはスライダーコントロールを持ち、0.1 秒ごとに値を記録する。スライダーの左端は会話が起きていないことを表し、右端に近いほど会話が活発になっていることを表す。

実験は被験者 1 人につき 2 回行った。1 回目は、波紋を用いて会話のログを再生する。被験者には、波紋の様子に応じて会話の活発さを自由に想像してもらい、スライダーを使ってリアルタイムにその度合いを入力するよう指示する。2 回目は、吹き出しによって会話のログを再生する。被験者には、吹き出しに表示されるメッセージやその文脈から会話の活発さを読み取り、リアルタイムにその度合いを入力するよう指示する。

再生するチャットのログは実験 1 と同じものを使用する。被験者は 3 名で、実験 1 の被験者とは異なる。また、実験後に 1 回目と 2 回目の違いについてインタビューを行った。

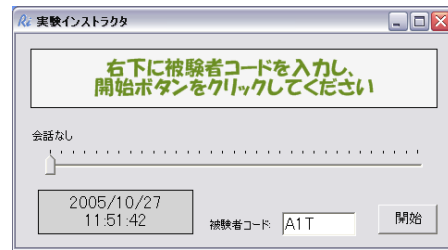


図 11: 実験インストラクタ

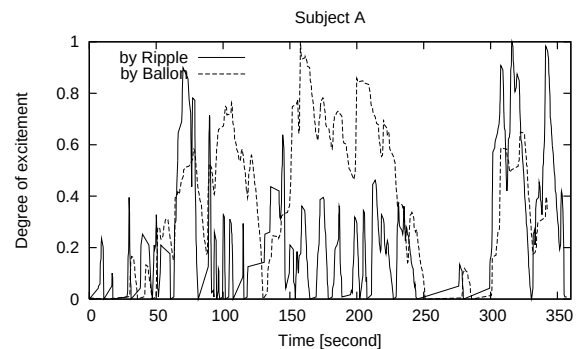


図 12: 被験者 A の実験結果

5.2.2 実験結果

実験結果の一部を図 12 に示す。図中では、1 回目の結果が実線で、2 回目の結果が破線で表されている。x 軸は時間軸、y 軸は会話の活発さの度合いである。

5.3 考察

実験 1 の結果から、会話が頻繁に起こる時や、メッセージの文字数が増加する時にクリックポイントが集中することがわかった。また、実験 2 の結果から、グラフの立ち上がりの部分に関しては、波紋を見た場合と吹き出しを見た場合の活発さの増加に、類似性が見られた。これらのことから、会話が活発になり始める状況が波紋によってユーザに伝えられたと考えられる。

実験 2 の結果から、波紋による提示の場合と吹き出しによる提示の場合を比較すると、特にグラフの立ち上がりの後に値の差が大きくなった。いずれの被験者においても、吹き出しによる提示については一度高いピークを迎えてからすぐには減衰せずにある程度のオフセットを保つ。一方、波紋による提示の場合には、ピークを迎えた直後に減衰する傾向が見られた。また、実験後の被験者へのインタビューにより、会話が続きそうなメッセージを見た時には値を下げない傾向があることがわかった。これらのことから、波紋による提示ではメッセージそのものを

読むことができないために、文脈レベルでの活発化が伝わらないという知見を得ることができた。この問題の解決は今後の課題であるが、解決策としては、システムが文脈レベルでの活発さを解釈して、それを波紋の大きさや減衰速度のパラメータへフィードバックさせることを考えている。

6 関連研究

波紋を用いてアンビエントに情報を提示するシステムとして WaterLamp[3] がある。WaterLamp は、ネットワークのトラフィックの状態に応じて水槽へ水滴を落とし、発生した波紋をランプで投影する。WaterLamp ではトラフィックの状態を水面の揺らぎの激しさへ 1 次元的にマッピングしているのに対し、RippleDesk では波紋の頻度だけでなく座標などのパラメータを利用している。

アンビエントな提示手法を用いたコミュニケーション支援システムとしては Meeting Pot[4] がある。Meeting Pot はコーヒールームでの休憩の開始を離れた場所にあるデスクの上に置かれたコーヒアロマ発生器を作動させることで知らせる。コーヒの香りが、離れた場所で休憩が起こっていることを知らせるアンビエントな提示手法となっている。また、ランプなどを用いて、Web サイトを訪問している人の活動をリアルタイムでアンビエントに提示するシステム[5] がある。これらのシステムは、離れた場所のアクティビティをアンビエントに伝える点で RippleDesk と共通している。異なる点は、RippleDesk では提示手法として波紋を利用していること、および、提示される側のコンテキストを考慮していることである。

Push&Pull[6] は、コンピュータを利用しない時には物理的に自分から遠ざけ、利用する時には近づけることで、情報提示をなめらかに変化させるインタフェースである。システムを使う・使わないという二値的な状態を強制しない点で、我々のアプローチと共通している。RippleDesk ではその点に加え、コンテキストによっては明示的にメッセージを提示する場合がある様に、システム側からの能動的な状態の移行を考慮している。

7 まとめと今後の展望

本論文では、既存のコミュニケーションツールの問題点を指摘し、インターネット上のざわめきを波紋を用いて表現する手法を提案した。また、プロトタイプとして周囲の会話のざわめきをデスクトップ上の波紋により提示するコミュニケーションツール RippleDesk を開発した。実験により、波紋によって

周囲の会話の活発さが増加している状態をユーザに伝達できる事を確認した。

今後は、波紋のパラメータを文脈に応じた会話の活発さに応じて変化させるなど、パラメータの設定をより充実させ、明示的に会話を見ることなく波紋のみからより多くの情報を得ることを実現したいと考えている。また、実世界では他人が物を食べる音や、ドアを開ける音など、様々なざわめきがコミュニケーション発生のかきかけとなるが、インターネットにもそのような偶発的コミュニケーションを導入したいと考えている。具体的には、RippleDesk がざわめきとして扱うアクティビティを、アプリケーションの起動や、メールの受信など、コンピュータの操作全般に拡張することを考えている。

参考文献

- [1] Barry Arons. A review of the cocktail party effect. *Journal of the American Voice I/O Society*, Vol. 12, pp. 35–50, July 1992.
- [2] 岩淵志学, 久松孝臣, 高橋伸, 田中二郎. 周囲の会話のざわめきを感じさせるインスタントメッセンジャー RippleDesk. ヒューマンインタフェースシンポジウム 2005 論文集, 第 2 巻, pp. 977–980, 2005.
- [3] Dahley A, Wisneski C, and Ishii H. Waterlamp and pinwheels: Ambient projection of digital information into architectural space. In *Conference Summary of CHI '98*, 1998.
- [4] 椎尾一郎, 美馬のゆり. Meeting Pot: アンビエント表示によるコミュニケーション支援. インタラクシオン 2001 論文集, 情報処理学会シンポジウムシリーズ, No. 5, pp. 163–164, March 2001.
- [5] Albrecht Schmidt and Hans-W Gellersen. Visitor awareness in the web. In *Proceedings of the tenth international conference on World Wide Web*, pp. 745–753, 2001.
- [6] 渡邊恵太, 安村通晃. Push&Pull: 「眺める」と「使う」をなめらかに移行するインタフェースの提案. インタラクシオン 2004 論文集, pp. 209–210, March 2004.