

筑波大学大学院博士課程

システム情報工学研究科修士論文

ユビキタス環境における
情報提示手法に関する研究

岩淵 志学

(コンピュータサイエンス専攻)

指導教員 田中 二郎

2006年3月

概要

近年のコンピュータの遍在とインターネットの普及は、M. Weiser の唱えるユビキタスコンピューティングおよびD. A. Norman の唱えるインビジブルコンピューティングの流れと一致する。本論文では、来るユビキタス時代における2つの問題提起し、新たに情報提示手法を提案することでそれらの解決を試みる。1つ目に、ユビキタス環境におけるストレージに注目し、自分自身の体を利用する柔らかいストレージの概念を提唱する。柔らかいストレージを実現するために、利用者に対して体内にストレージが存在する錯覚を与えるような情報の提示手法を考案する。さらにプロトタイプを開発し、提案手法について議論を行う。2つ目に、インターネットを用いたコミュニケーションの広まりに注目し、実世界のざわめきの効果をインターネット上で導入することにより、実世界型の偶発的なコミュニケーションの参加支援を試みる。デスクトップ画面上の波紋にインターネット上のざわめきの情報を含めて提示する手法について述べ、試作システムについて述べる。実験の結果から、提案手法によってインターネット上におけるざわめきの情報が伝達可能であることが確認された。

目次

第1章	はじめに	1
1.1	ユビキタスコンピューティング	1
1.2	情報提示手法の変化	2
第2章	ユビキタス環境におけるストレージ	3
2.1	従来型ストレージの問題	3
2.2	柔軟いストレージの提案	4
2.3	ユビキタス環境における利用シナリオ	5
2.3.1	カメラ型アプライアンス	5
2.3.2	音楽再生アプライアンス	5
2.3.3	商品販売端末	6
2.4	実装方針	7
2.4.1	個人とストレージを対応付ける	7
2.4.2	体内にストレージがあるように錯覚させる	8
第3章	NSシリーズの開発	10
3.1	Natural Storage Mirror	10
3.1.1	概観	10
3.1.2	写真データを閲覧する	10
3.1.3	他人同士で写真を交換する	12
3.1.4	実装	12
3.2	Natural Storage Camera	13
3.2.1	概観	13
3.2.2	写真の撮影と保存	14
3.2.3	実装	15
3.3	利用例	15
第4章	ユビキタス環境におけるざわめきの提示	17
4.1	インターネットを用いたコミュニケーションの問題	17
4.2	実世界におけるざわめき	18
4.3	インターネット上におけるざわめき	18
4.4	波紋を用いたざわめきの表現手法の提案	19

第 5 章	RippleDesk の開発	20
5.1	システム概要	20
5.2	メッセージの受信と波紋の提示	21
5.3	メッセージの送信	23
5.4	その他の機能	24
5.4.1	コミュニティの設定	24
5.4.2	Cocktail Word の登録	25
5.4.3	アプリケーションの重みの設定	25
5.4.4	受信履歴の参照	25
5.5	実装	26
5.5.1	通信プロトコル	26
5.5.2	波紋の描画アルゴリズム	26
第 6 章	柔らかいストレージの議論	27
第 7 章	波紋による情報提示の実験	28
7.1	実験 1: コミュニティの解像度	28
7.1.1	実験方法	28
7.1.2	実験結果	28
7.2	実験 2: 会話の活発化の伝達	29
7.2.1	実験方法	29
7.2.2	実験結果	30
7.3	実験 3: 連続的な会話状況の伝達	30
7.3.1	実験方法	31
7.3.2	実験結果	31
7.4	実験結果についての考察	31
第 8 章	関連研究	33
8.1	柔らかいストレージに関して	33
8.2	波紋によるざわめきの提示に関して	33
第 9 章	まとめと今後の展望	35
	謝辞	36
	参考文献	37

目次

2.1	共有ディレクトリにアクセスするためのインタフェース例	3
2.2	利用者がイメージする柔らかいストレージ	4
2.3	情報アプライアンスを用いた写真の撮影と印刷	5
2.4	情報アプライアンスを用いた音楽データの記録と再生	6
2.5	柔らかいストレージを利用したデジタルストア	6
2.6	柔らかいストレージを実現する情報アプライアンスの一般構成	7
2.7	利用者によってイメージされる柔らかいストレージ	8
3.1	Natural Storage Mirror のイメージ	11
3.2	Natural Storage Mirror の利用	11
3.3	Natural Storage Mirror 処理の流れ	12
3.4	Natural Storage Camera のイメージ	13
3.5	Natural Storage Camera の利用	14
3.6	プレビュー画面	14
3.7	転送中の画面	14
3.8	Natural Storage Camera 処理の流れ	15
3.9	NS シリーズの具体的な利用例	16
5.1	RippleDesk システム概観	20
5.2	波紋によるざわめきの提示	21
5.3	吹き出しによる明示的な提示	21
5.4	波紋のパラメータと意味のマッピング	22
5.5	インパクト係数による提示方法の違い	22
5.6	メッセージドロップの作成	23
5.7	Community Manager	24
5.8	Cocktail Word と重みの関連付け	25
7.1	波紋の間隔と正答率の関係	29
7.2	実験結果: 発言とクリックポイントの関係	30
7.3	実験インストラクタ	31
7.4	被験者 A の実験結果	32

表目次

2.1 見かけの保存先の例	9
4.1 実世界とインターネットにおけるざわめき情報の違い	18

第1章 はじめに

1.1 ユビキタスコンピューティング

ユビキタスコンピューティング [1] は、今から 15 年以上前に Mark Weiser によって提唱されたコンピュータの利用についての概念である。今日では、インターネットが普及し、コンピュータが小型化・大型化してきた。また、コンピュータの偏在化が進み、多くの人々は携帯電話を持つことで常にネットワークで相互接続されるようになった。これらはユビキタスコンピューティングが実現されることを示している。

また、Weiser は、ユビキタスコンピューティングにおける自然なインタフェースの重要性を唱えている。自然なインタフェースとは、従来のマウスやキーボードに変わって、手書きなどのより人の情報処理能力に近いインタフェースのことである。近年では、手書きや身振り手振り、指差し等を用いた多くの技術が提案されており、マウスやキーボードは限定的な場合のみで利用されるようになっていくと思われる。

さらに、ユビキタスコンピューティングに関連した技術に D. A. Norman によって唱えられたインビジブルコンピューティングがある [2]。これは人々がコンピュータという技術を意識せずにコンピュータを利用できるインタフェースの重要性を提唱したものである。インビジブルコンピューティングによれば、情報アプライアンスと呼ばれる、コンピュータが埋め込まれた“道具”がコンピュータの究極の形である。例として、次に電気モータに関するストーリーを挙げる。

電気モータは当初、高価なものであったのでモータそのものが販売されていた。モータだけでは何もできないので、利用者はアタッチメント装置を別途に購入し、モータに取り付けることで様々な用途に使用した。しかし、モータを扱うためには専門的な知識が必要とされたこともあり、なかなか一般には広まらなかった。その後、モータは大量に生産されるようになってコストが下り、色々なものの一部として埋め込まれるようになった。現在では、人は道具の中にモータが入っているかどうかを意識することなく利用するようになった。

この例は、汎用的で複雑なコンピュータが、単機能に特化され、情報アプライアンスへと進化することを示している。身近な例としては、デジタルカメラや携帯音楽プレーヤ等は、1 つのタスクを処理するように特化されたコンピュータである。近年の研究では、家具にコンピュータを埋め込む試み [3] 等がなされており、将来は生活用品の多くがコンピュータの搭載された情報アプライアンスになると予想される。

なお、本論文中では便宜上、情報アライアンスを単にアライアンスとも呼ぶ。また、本論文中で用いるユビキタス環境とは、情報アライアンスが遍在し、それらがネットワークによって相互接続され、人々が定常的にコンピュータによる生活の支援を受ける環境を指す。ユビキタス時代とは、そのような環境が一般化する時代を指す。

1.2 情報提示手法の変化

ユビキタス時代においては、情報を提示するためのインタフェースについて考慮することが重要である。システムの入力インタフェースはもちろん、出力(提示)インタフェースについても前述の自然なインタフェースが実現されなければならないからである。ここで、ユビキタス時代における情報提示に関する2つの変化をまとめた。

- (1) 情報を提示するための表現手法が増える 従来の典型的な表現手法は、ディスプレイ画面におけるメッセージの表示や警告音であった。しかし、携帯電話、街頭の大画面など多くの端末が遍在するユビキタス環境においては、情報の表現手法が多彩になる。
- (2) 提示される情報の種類そのものが増える 従来は、個別のコンピュータが、それぞれ単体で持つ情報を処理し、提示するだけであった。それに対して、ユビキタス環境においてはあらゆるコンピュータが相互接続され、膨大な量の情報を共有する。例えば、あらゆる場所にセンサーを配置し、それらをネットワークに接続することで、コンピュータが外界の状況を把握できるようになる。そのため、新たなタイプの情報を扱うことができる。

本研究ではユビキタス環境における2つの問題点を提起し、(1)および(2)の視点から新たな情報提示技術を提案した。

まず(1)について、ユビキタス時代におけるストレージのインタフェースについて問題提起を行い、自分自身の体を利用する柔らかいストレージの概念を提唱する。柔らかいストレージを実現するためには、その利用者に対して、体内にストレージが存在する錯覚を与えるような情報の提示手法が重要になる。これについては、第2章で提案の詳細について述べ、第3章では実際に開発した試作システムについて述べる。

次に(2)について、ざわめきによる実世界型のコミュニケーション参加をインターネット上で実現する必要性を述べ、ざわめきの提示手法として波紋を利用することを提案する。第4章で提案手法について述べ、第5章で試作システムについて述べる。

第2章 ユビキタス環境におけるストレージ

2.1 従来型ストレージの問題

ユビキタス時代においても、データを記憶するためのストレージの利用は変わらず欠かせないものである。ストレージはデジタルデータの記憶媒体である。情報アプライアンスの中には、データを書き出したり、読み込むものが多い。現在のアプライアンスについても、例えばデジタルカメラでは写真データをストレージに書き出すし、携帯音楽プレーヤではストレージから音楽データを読み込んで再生する。

ユビキタス時代においては、多くのアプライアンスはネットワークで相互接続される。そのため、ネットワークストレージが広く利用されるものと思われる。ネットワークストレージの利用者は普段、ネットワーク上のストレージのアクセスに必要なIDやパスワードなどのキーだけを持ち歩くことで、いつでもどこでもデータにアクセスできる。ネットワークストレージの代表的な例としては、OSから提供される共有ディレクトリの機能 (WindowsにおけるSMB、LinuxにおけるNFSなど) や、インターネット上におけるストレージ提供サービス (Yahoo!におけるブリーフケース) がある。

しかしながら、これらの現在のネットワークストレージのインタフェースは、典型的なグラフィカルユーザインタフェース (GUI) による操作を前提としている (図 2.1)。IDとパスワードの入力などにはキーボードを用いなければならない。また、データの閲覧や移動にはマウスを用いる。

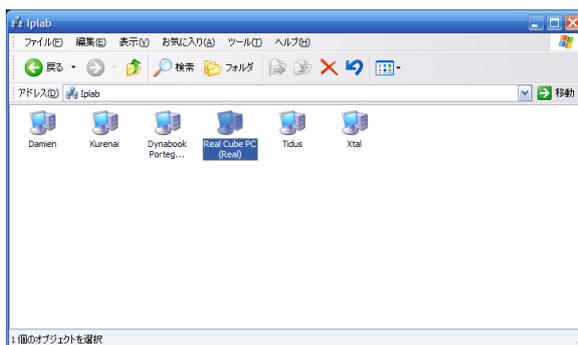


図 2.1: 共有ディレクトリにアクセスするためのインタフェース例

このような従来型の入力インタフェースは、前章で述べたようにユビキタスコンピューティングにおいては非現実的である。また、Normanの唱えるインビジブルコンピューティングで

は人々が情報アプライアンスを利用するに際してそのテクノロジーは意識されないと言われて
いるが、ネットワークの利用にはしばしばホスト名やプロトコルの指定を求められる。

このように、情報アプライアンスがコンピュータのテクノロジーを隠すように、ストレージ
もそのテクノロジーを隠匿する必要がある。

2.2 柔らかいストレージの提案

本研究では、情報アプライアンスの利用者の“体”そのものをネットワークストレージのキー
として利用し、さらにアクセスのためのインタフェースにも体を用いる手法を提案してきた
[4][5]。この手法によって利用者は、自分の体内に自分専用のストレージがあり、データを自
分の体に記録する錯覚を持つ。体が持つストレージは柔らかいストレージ (Natural Storage) と
呼ばれる。

柔らかいストレージの基本的なコンセプトは、体をデジタルデータの保管場所として利用
することである。柔らかいストレージは、日常生活の中で情報アプライアンスと共に利用
される。利用者は自分の体の中にデータを記録し、携帯するイメージを持つ (図 2.2)。

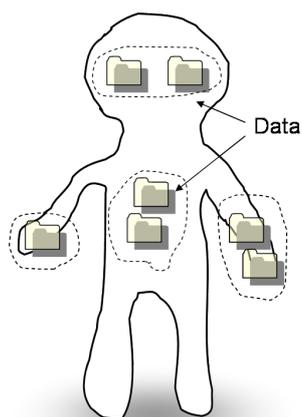


図 2.2: 利用者がイメージする柔らかいストレージ

情報アプライアンスの利用者は、データのサイズ制限やデータパスなどのコンピュータ側
の知識なしで、柔らかいストレージを使用することができる。また、体をストレージとして
利用することで、物理的なメディアを用いる DVD やメモリカードなどのストレージと比較し
て次のような利点がある。

- データの持ち運びの負担の解消。柔らかいストレージは、重さや大きさが無い。
- 紛失の恐れへの解消。体にデータを記録するため、人が存在するところには、いつでもどこでもデータがある。

- 物理的なデータの破損がない。物理的なメディアを伴うストレージは、メディアを壊せばデータが消えてしまう。

これらは携帯電話などの小型デバイスやウェアラブルデバイスにデータを記録する場合と比較しても、同様の利点が言える。人の体とは別の”物”を扱うことを利用者に強制すると、自分の体以外の物を扱う負担は避けられない。また、それらを十分に小型化しても、紛失するリスクが増える。体をキーとして用いる手法は、この問題を解決する唯一の方法であると考えている。

2.3 ユビキタス環境における利用シナリオ

柔らかいストレージの利用シーンをアプライアンスの例と共に述べる。

2.3.1 カメラ型アプライアンス

デジタルカメラで撮影した写真を見ることは日常によくあるシーンである。現在は、デジタルカメラにメモリカードを挿入し、撮影したデータを記録した後、パソコンで再生するか、プリントアウトして見る。これに柔らかいストレージを適用した例を図 2.3 に示す。まず、デジタルカメラが生成する写真データは体内に格納される。そして、プリンタなど出力に関するアプライアンスを利用する時には、体内から直接データを出力する。

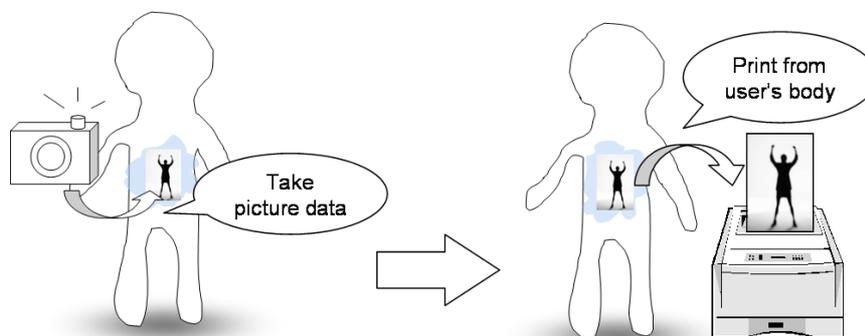


図 2.3: 情報アプライアンスを用いた写真の撮影と印刷

2.3.2 音楽再生アプライアンス

音のデータを体内に記録するマイク型のアプライアンスを考える。

ユーザがマイクを手を持って話すと、その音は逐次マイクを持った人の体内に記録される。そしてヘッドフォンを装着すると、装着した人の体内に記録された音のデータが再生される(図 2.4)。

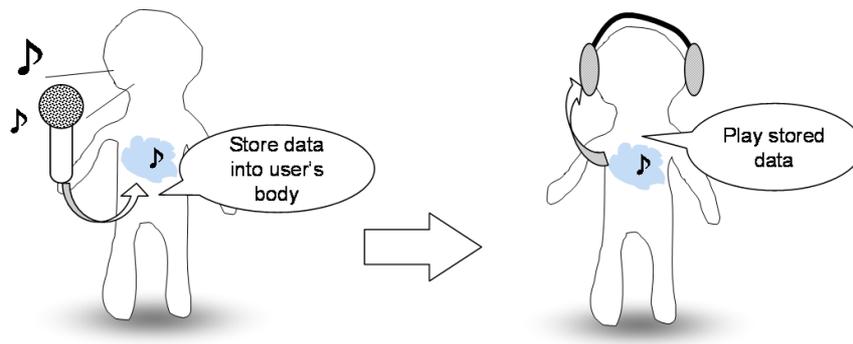


図 2.4: 情報アプライアンスを用いた音楽データの記録と再生

2.3.3 商品販売端末

デジタルデータの販売に柔らかいストレージを適用することも考えられる(図 2.5)。最近では、特に音楽や映画のコンテンツについて、物理的なメディアを用いないインターネットを通じてのダウンロード販売が広がっている。現在では、ダウンロードしたデータをそのままハードディスクから再生するか、もしくはCDに書き込む、メモリに転送するなどして再生する。このようなシーンに対して柔らかいストレージを適用する場合、店先でコンテンツを買うと同時に体内へデータをダウンロードし、記録されたデータを先に述べたようなアプライアンスを用いて再生するような事が考えられる。

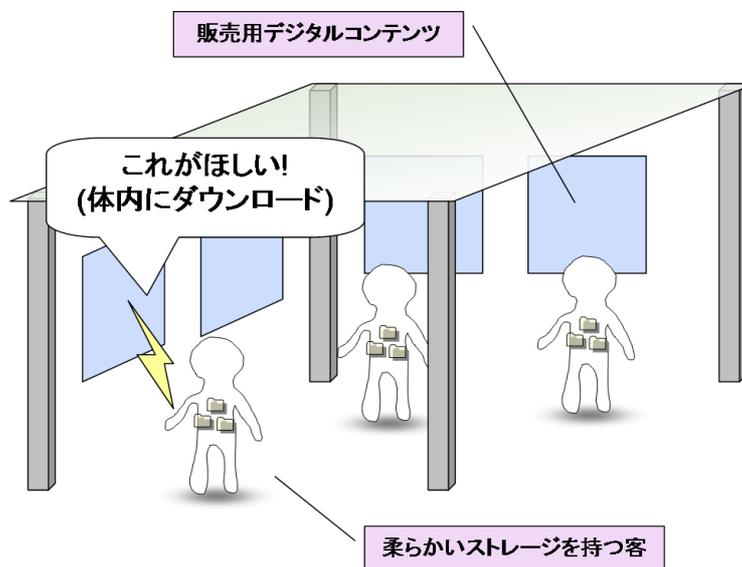


図 2.5: 柔らかいストレージを利用したデジタルストア

2.4 実装方針

柔らかいストレージは、体にデータが存在するイメージを利用者に与える機能を備えた情報アプライアンスによって、仮想的に実現される。具体的には次の2つの機能が情報アプライアンスに実装されることが求められる:

1. 個人がその人専用のストレージを持つこと
2. ストレージが体内に存在するように錯覚させること

次の節でそれぞれについて詳しく説明する。

2.4.1 個人とストレージを対応付ける

まず、アプライアンスの構成をシステムの側面から説明する。柔らかいストレージを実現するために、アプライアンスは個人の持つ生体情報から固有のIDを得て、そのIDに関連付けられたネットワークストレージにアクセスする手法を用いる必要がある。ネットワークストレージを使うことで、IDがあればどこからでも個人に関連付けられたストレージにアクセスすることができる。この場合、IDは体そのものであるから、人体を仮想的なりムーバブルストレージの様に見ることができる。また、仮想的にストレージを実現する方法には、ネットワークストレージ同様に、アクセスの権限付けや容量の制限などをソフトウェア的に解決できるという利点を持つ。図2.6に、柔らかいストレージを利用するアプライアンスの設計をまとめる。

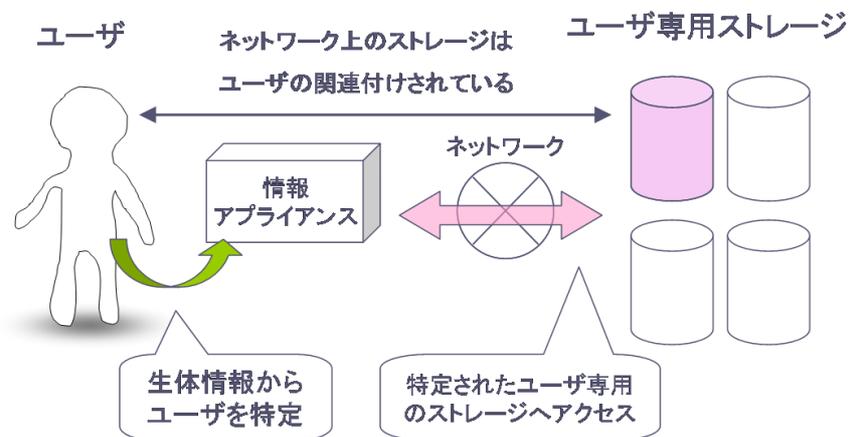


図 2.6: 柔らかいストレージを実現する情報アプライアンスの一般構成

2.4.2 体内にストレージがあるように錯覚させる

次に、アプライアンスの構成についてインタラクションの面から説明する。ストレージが個人の体内にあるように錯覚させるためには、情報アプライアンスと人との間に起こる2種類のインタラクションを実装する必要がある。

まず、2つのインタラクションについての一般的な説明を次に示す:

アドレッシング アプライアンスに対して操作対象のストレージを指定するインタラクションである。これは従来のシステムにおけるフロッピーディスクの挿入や、URLの指定などに相当するインタラクションである。

フィードバック アプライアンスがストレージへアクセスしたことを人に対して伝えるインタラクションである。これは従来のシステムにおけるパイロットランプの点滅、デバイスからのシーク音、振動などに相当する。

これらのインタラクションの必要性について説明する。図2.7に、柔らかいストレージを実現する情報アプライアンスが持つ一般的なインタラクションの構成を示す。

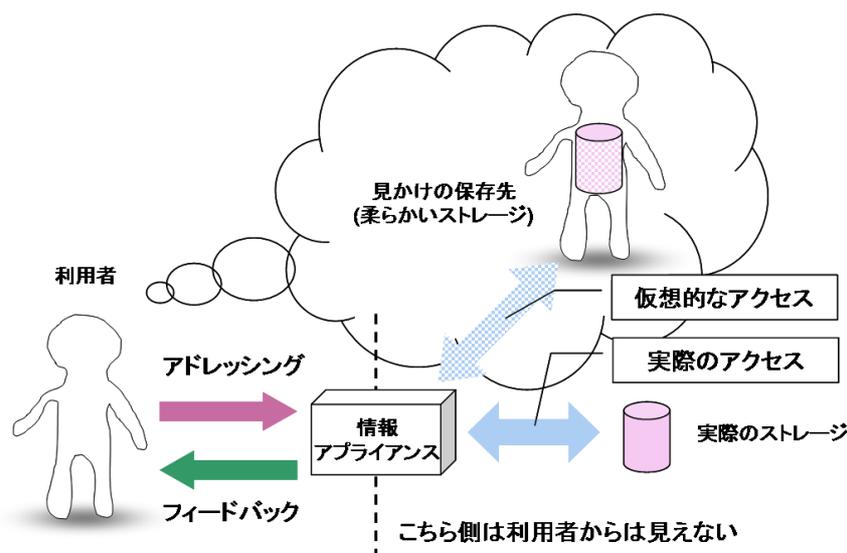


図 2.7: 利用者によってイメージされる柔らかいストレージ

利用者は、これら2種類のインタラクションから得られる情報のみによって、ストレージのイメージを想像する。本研究では、アプライアンスとのインタラクションから利用者がイメージするストレージを、見かけの保存先と呼ぶ。見かけの保存先は、現在のストレージの利用においても一般的によく見られる。その例を表2.1に挙げる。

柔らかいストレージは、見かけの保存先が利用者自身の体となるようなインタラクションによって実現される。この時、ユーザに見かけの保存先を明確に意識させるためには、アド

表 2.1: 見かけの保存先の例

見かけの保存先	実際の保存先
メールはメールボックスに届く	メールサーバのディスクに届く
Web 掲示板の書き込みは掲示板に保存される	Web サーバのディスクに保存される
電子マネーはカードの中にある	カード会社のデータベースに記録される

レッシングが想像させる見かけの保存先と、フィードバックが想像されるそれを一致させなければならない。なぜならば、もし各インタラクションからイメージされる見かけの保存先が不鮮明であれば、人はデータがどこに保存されているか直感的に把握することができないからである。コンピュータの知識が少ない人が、ハードディスクのファイルシステムよりフロッピーディスクのほうが使いやすいとしばしば述べることがある。これは、ハードディスクと人とのインタラクションよりも、フロッピーディスクと人とのそのほうが、見かけの保存先を具体的にイメージしやすいからと言える。

次に、柔らかいストレージを実現するアプライアンスに必要となるアドレッシングとフィードバックについて述べる。

アドレッシングには指紋・虹彩・顔認識などのバイオメトリクス技術による生体情報の検出を用いる。どの認証方法を用いるかは、アプライアンスによって異なる。たとえば、マイクやデジタルカメラなど、手に持って使うアプライアンスには指紋が有効である。人と離れた位置で動作するアプライアンスならば、カメラによる個人認識が利用できる。

フィードバックについては、ストレージにアクセスしていることを体と密接に関わる表現手法を用いてユーザに示す。例えば、アクセスランプに相当する機能を体に取り付けるなどが考えられる。また、ストレージにアクセスしている際に仮想的な音や振動などをフィードバックさせることができれば、より利用者が認知する見かけの保存先はたしかなものになる。

第3章 NSシリーズの開発

本研究ではこれまでに、柔らかいストレージを利用するアプライアンスである NS シリーズを試作した [6]。本章では、体内に記録されている画像を閲覧するための鏡である Natural Storage Mirror(以下、NS ミラー)と、体内に写真データを保存するデジタルカメラ Natural Storage Camera(以下、NS カメラ)について述べる。NS カメラは従来のデジタルカメラがメモリカードに写真データを保存するのに対して、撮影者の柔らかいストレージに保存する。そして、NS カメラで写真を撮影した人が NS ミラーの前に立つことで、体内に記録してある写真を閲覧することができる。

3.1 Natural Storage Mirror

3.1.1 概観

NS ミラーは、体内に記録されている写真データを映し出す鏡である。普通の鏡と異なる点は、自分の体のまわりに自分の柔らかいストレージに記録されている写真が映し出されることである。また、鏡の中に映し出された自分の手を使って写真データを他人に転送することが可能である。NS ミラーのイメージを図 3.1 に示す。このように、鏡の前に利用者が立つと体が映り込み、映り込んだ体から写真が飛び出す。それらの画像は、映し出された体の周囲を回転しながら表示される。

次に、NS ミラーのプロトタイプを図 3.2 に示す。試作した NS ミラーはディスプレイとカメラから構成される。この図では、ディスプレイの前に立ったユーザの写真が表示されている。現在のバージョンでは、鏡に映り込んだ人の位置と ID を検出するために、利用者には固有の四角形のマーカをあらかじめ取り付けられている。

3.1.2 写真データを閲覧する

写真データを閲覧するには、利用者は NS ミラーの前へ立つ。すると、自分の体に保存してある写真が体の中央から飛び出し、自分の体の周囲をゆっくりと回転する。写真を詳しく見たい場合には、鏡に近づくことで写真が拡大される。逆に、鏡から遠ざかると写真は小さく表示される。もし複数人が同時に映りこんだ場合には、それぞれの体の周囲に写真を表示することで、互いの写真を閲覧し合う事が可能である。

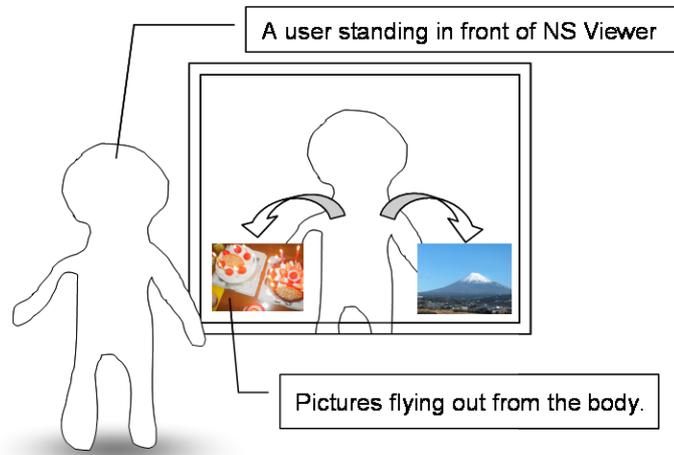


図 3.1: Natural Storage Mirror のイメージ

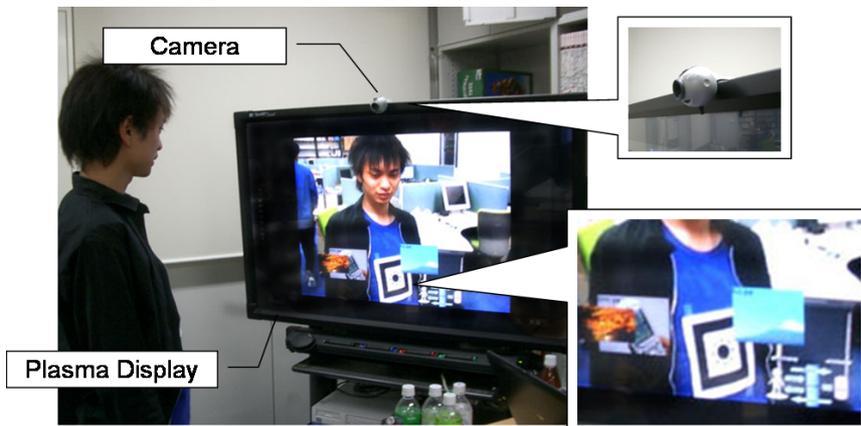


図 3.2: Natural Storage Mirror の利用

3.1.3 他人同士で写真を交換する

NS ミラーは映り込んだ手によってハンドカーソルを移動することができる。ハンドカーソルは、鏡の中に仮想的に映る四角形のカーソルである。このカーソルは、最も近くの写真を吸い付ける効果がある。

写真を他人の体へ移動するシーケンスについて説明する。例として、A さんが自分の持つ写真を B さんに移動させる場合を考える。まず、A さんが手の平を鏡に向ける。手の平が一定時間検出されると、鏡の中のハンドカーソルが有効となる。次に、ハンドカーソルを自分の周囲を回る写真に近づけると、その写真がハンドカーソルに吸い付く。吸い付けた写真は手でそのまま移動することができるので、B さんの体の近くまで手を移動させる。ここで、A さんが手を裏返して、ハンドカーソルを無効にする。すると、ハンドカーソルに吸い付いていた写真は B さんの体へ吸い寄せられ、B さんの体へ移動する。この時点で移動は完了である。新しく B さんに追加された写真は、B さんの体の周囲を回転する。

3.1.4 実装

試作では、大画面ディスプレイとして幅約 2 メートル、高さ約 1 メートルのプラズマディスプレイを用いた。また、解像度が 11 万画素の USB 接続の PC カメラをキャプチャー用として使用した。カメラはプラズマディスプレイ上部に取り付けられており、ディスプレイの前面の風景を撮影するように設置してある。

システムの処理の流れを図 3.3 に示す。

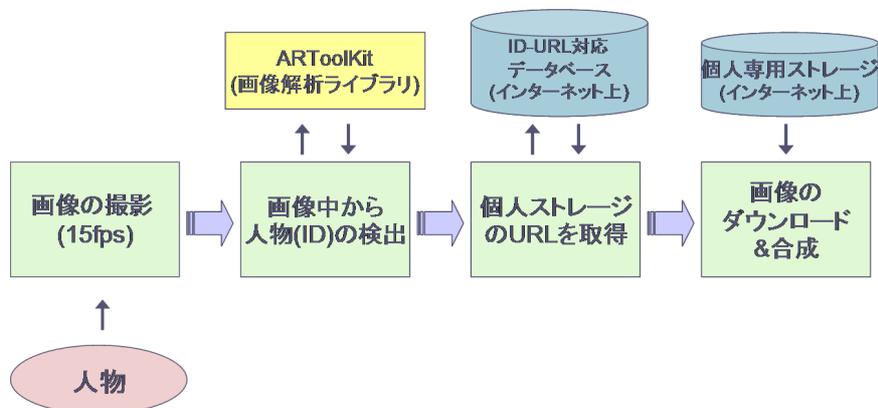


図 3.3: Natural Storage Mirror 処理の流れ

まず NS ミラーは、カメラで撮影した映像から 1 フレームを取得し、ARToolKit[7] を用いて静止画からマーカを検出する。もし、マーカから個人を識別する ID が得られると、次にその ID に対応付けられたストレージの場所をデータベースから取得する。現在は、ストレージの場所は URL を用いて表現される。次に、URL が指し示すディレクトリから、画像ファイルを全てダウンロードする。そして、マーカのまわりに画像を回転するように合成し、プラ

ズマディスプレイに出力している。出力される映像は鏡のように見えるよう、水平方向にフリップさせている。また、フレーム中にハンドカーソル用のマークが検出された場合には、手の位置にカーソルを表示する。もし、カーソルの中心から一定距離の中に写真があれば、その写真をカーソルに関連付け、元の所有者との関連を解消する。その後カーソルが消えるまで写真を保持し、カーソルが消えた場合には、最も近くのマークを新しい所有者と見なして、新たな関連付けを行う。もし、消えた位置の近傍にマークが存在しない場合には、元の所有者と再び関連付ける。

ネットワーク上の個人専用のストレージは、Windows の標準的なファイル共有プロトコルである SMB を用いて、共有フォルダによって実現されている。

プログラムは Microsoft Visual C++ .NET を用いて実装されている。

3.2 Natural Storage Camera

3.2.1 概観

NS カメラは、柔らかいストレージにアクセスするデジタルカメラである。従来のカメラがストレージに写真を保存するのに対して、NS カメラでは撮影者の体に写真を保存する。NS カメラの概要を図 3.4 に示す。

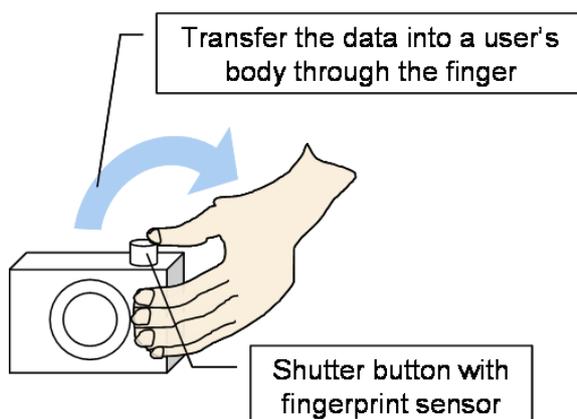


図 3.4: Natural Storage Camera のイメージ

次に、試作した NS カメラを図 3.5 に示す。プロトタイプを試作するには、NTT Docomo の携帯電話 F900iT を用いた。F900iT はモバイルカメラ、指紋センサ、Bluetooth およびインターネットにアクセスできるネットワーク機能を備えており、これらはプログラムから制御可能である。また、デバイスの大きさや可搬性の点からも実際のデジタルカメラに近いという理由で利用することにした。

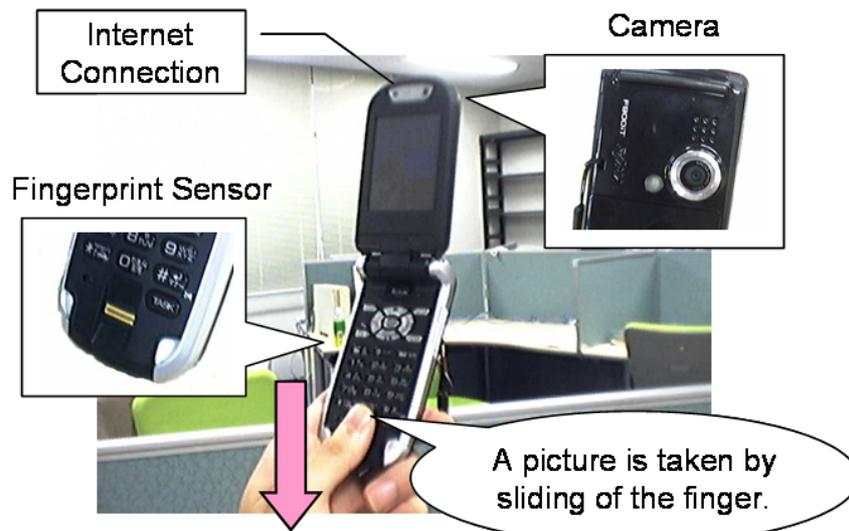


図 3.5: Natural Storage Camera の利用

3.2.2 写真の撮影と保存

撮影および保存のシーケンスについて説明する。まず、利用者はカメラ機能を使って写真を撮影する。写真が撮影されると、画面にはプレビューが表示される(図 3.6)。次に、利用者が指紋センサに指をスライドさせると、写真が体内に保存される。この時の画面には体内に転送している事を示す画面が表示される(図 3.7)。



図 3.6: プレビュー画面



図 3.7: 転送中の画面

3.2.3 実装

カメラ側のプログラムは、NTT Docomo から提供されているアプリケーション開発環境である Doja を用いて実装されている。Doja には i アプリと呼ばれる携帯電話用の java アプリケーションを開発するためのライブラリが含まれている。ライブラリには F900iT のモバイルカメラや指紋センサを扱うためのクラスがあり、通常の java 環境を拡張する形で i アプリのプログラミングが可能となっている。

NS カメラを構成するシステム全体の処理の流れを図 3.8 に示す。

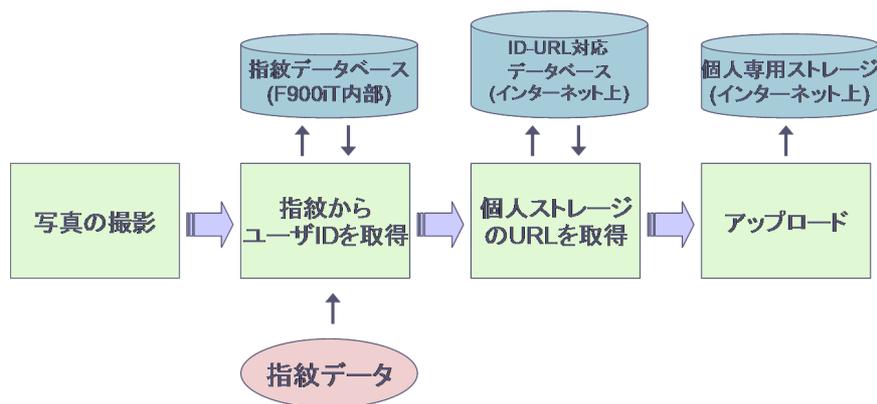


図 3.8: Natural Storage Camera 処理の流れ

まず、F900iT のカメラによって写真が撮影され、写真データが準備され、プレビューされる。ユーザが保存を選択すると、システムは指紋の入力を利用者へ求める。指紋が入力されると、システムは F900iT 内部の指紋データベースから関連付けられたユーザ ID を得る。指紋認証に失敗した場合にはシーケンスを中止し、プレビューに戻る。次に、その ID に対応する個人専用のストレージのネットワークパス (URL) をデータベースから取得する。そして、写真データを HTTP プロトコルを介して URL へアップロードする。

サーバ側のプログラムは、Apache 上で動作する Perl によって CGI として実装されている。CGI プログラムは、カメラからアップロードが要求されると、指定された個人専用のストレージにファイルを保存する機能を持つ。

3.3 利用例

図 3.9 に、NS ミラーと NS カメラの具体的な利用イメージを示す。NS カメラで富士山の写真を 1 枚撮影すると、撮影した人の体内にデータが記録される。この時に NS ミラーで体を映すと、富士山の写真が表示される。次に、ケーキの写真をもう 1 枚撮影する。この時には、その人の体内には 2 枚の写真が格納されている。NS ミラーの前に立つと、2 枚の写真が回転しながら体の周囲に表示される。もし、NS ミラーの前に立って NS カメラで写真を撮影すると、その場で写真が体から飛び出してくるように映る。

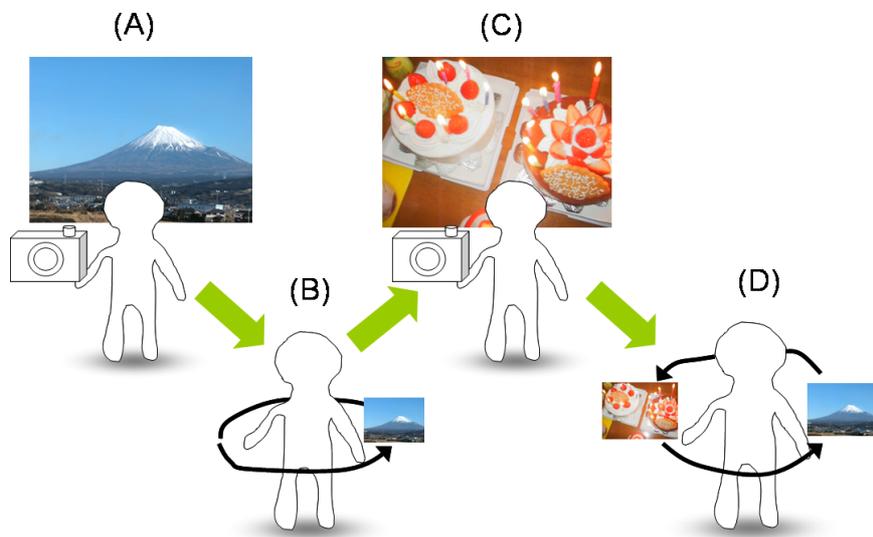


図 3.9: NS シリーズの具体的な利用例

第4章 コビキタス環境におけるざわめきの提示

4.1 インターネットを用いたコミュニケーションの問題

近年ではインターネットの普及によってオフィスの物理的分散が進み、同じ部署やチームであっても、離れた場所で作業する機会が増えてきた。将来のコビキタス環境においては、現在より一層に分散が進み、固定されたオフィスを持たずにあらゆる場所でビジネスを行うことが可能になると考えられる。

このような環境において、離れた人同士の会話にはインターネットを利用したコミュニケーションツールが用いられる。インターネットコミュニケーションツールは、具体的にはチャット、インスタントメッセージ、Web 掲示板などがある。これらのツールは本来プライベートでの会話に利用されてきたが、近年ではインターネット会議などに利用され、ビジネスシーンにおいて重要な役割を果たしている。

しかしながら、現在のインターネットコミュニケーションツールの問題として、偶発的なコミュニケーションの発生が難しい点があげられる。偶発的コミュニケーションとは、偶然居合わせた人同士で行われるインフォーマルなコミュニケーションである。例えば、休憩中のコーヒールームでの何気ない会話である。そのようなカジュアルコミュニケーションは、協調作業や知識創造のためには重要である。コーヒーを飲みながら雑談する中で新しいアイデアが創出されたり、各個人のコミュニケーションが深まることで作業効率が上昇するなどの効果がある。

実世界におけるカジュアルコミュニケーションは、しばしば周囲のざわめきによって偶発的に起こる。例えば、他人がコーヒールームに行く足音を聞いて、自分も休憩に入るといった場合である。このように、実世界のざわめきは偶発的な会話への参加を支援する点で有用である。一方、インターネット上におけるざわめきを考えると、ざわめいている状態は存在するものの、それを伝えるシステムが存在しない。このためユーザは受動的にざわめきを得ることができない。

本研究の目的は、ざわめきの提供によるコミュニケーション支援である。ユーザがインターネット上のざわめきを受動的かつ定常的に得ることで実世界型の会話への参加が実現できるものと考えられる。

4.2 実世界におけるざわめき

ざわめきは、人々の会話によって発生し、音として周囲の人へ伝達される。実世界のざわめきの持つ特徴を以下の3点にまとめた。

- (a) 定常性 ざわめきは定常的かつ能動的に人に働きかける
- (b) 選択性 ざわめきは聞き手のコンテキストによって取捨選択される
- (c) 多次元情報 大きさ、方向、音色によって話し手を推測できる

これらの特徴は偶発的コミュニケーションのきっかけとなる。つまり、(a) 常に聴こえてくるざわめきは、(b) 受け手が必要であると判断した場合にのみ意識的に聴き取ることが可能であり、それらの判断材料になるのは (c) 聴こえてくる方向や音量の変化などである。

4.3 インターネット上におけるざわめき

近年、インターネット上においても人々はコミュニケーションツールを用いて会話を行う。インターネット上においても、常にまわりで会話が起きているのであるから、実世界と同様に“ざわめいている”状況は存在する。しかしながら、掲示板やチャットにおけるコミュニケーションの発生を知るにはユーザが能動的に調べなくてはならない、という点で実世界とインターネット上では事情が異なる。例えば、ユーザは自分でブラウザを開いたり、チャットへのログインが必要である。表 4.1 に、実世界での会話とインターネット上における会話の間の違いについて、前節で述べたざわめきの3つの特徴の観点からまとめた。インターネット上の会話についてはいくつかのコミュニケーションを例に取った。

表 4.1: 実世界とインターネットにおけるざわめき情報の違い

	実世界	チャット	メッセージ	掲示板
定常性				×
選択性		×	×	×
多次元情報		×	×	×

チャットやメッセージにおいては、オンライン(ログイン)状態に限り、メッセージは定常的にユーザに伝達される。もしオフラインの場合には、他人が会話していてもその情報は一切ユーザには伝わらない。掲示板については、こちらから能動的に掲示板の内容の更新を能動的に要求しない限り、情報は定常的に伝わらない。また、いずれのコミュニケーションツールについても、ユーザが忙しいかどうかは考慮されない。MSN Messenger を例にとると、ユーザが何か作業をしている場合でも、メッセージが受信される度にウィンドウがフラッシュし、通知音が鳴る。さらに、これらのツールは文字ベースであって、その内容を明示的に見ない限り、発言者が誰であるかは不明である。

4.4 波紋を用いたざわめきの表現手法の提案

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、定常的に情報が与えられ、それらを状況によって選択的に受け取ることが可能であり、さらに多次元情報を表現できる手法が必要である。本研究では、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。

これらのパラメータをユーザの状況に応じて変化させることで、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ことが可能となる。

第5章 RippleDeskの開発

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。

5.1 システム概要

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。RippleDeskシステムの概観を図5.1に示す。

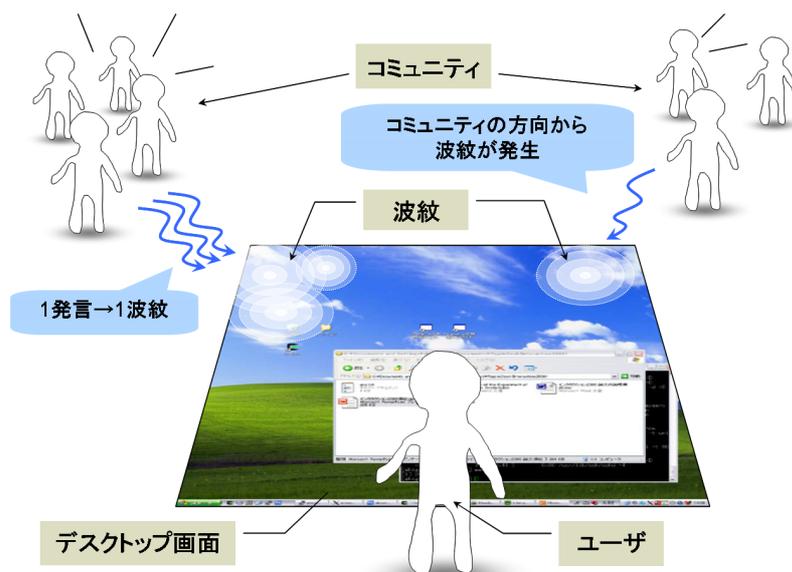


図 5.1: RippleDesk システム概観

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 5.2)。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 5.3)。



図 5.2: 波紋によるざわめきの提示

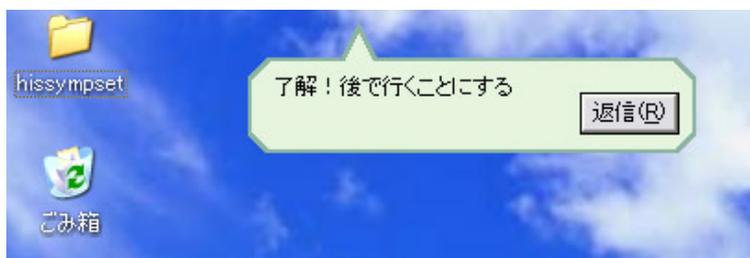


図 5.3: 吹き出しによる明示的な提示

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 5.4 に、波紋が持つパラメータとその意味のマッピングについてまとめた。このように波紋に複数の情報をマッピングすることで、4.2 節で述べたざわめきの性質の内、多次元情報が実現されている。

5.2 メッセージの受信と波紋の提示

RippleDesk は、システムを利用しているユーザのコンテキストに応じて、メッセージの提示手法を動的に変化させる。コンテキストとは、ある時点でのユーザの状態である。実世界において、忙しいので話をしたくない、暇である、というようにざわめきの受け手の状態は常に変化し、同様にインターネットを介した会話においてもユーザの状態は変化する。



図 5.4: 波紋のパラメータと意味のマッピング

メッセージを受信した際の提示方法は、受信した時点のコンテキスト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 I はすなわち、コンテキストに応じて算出される、受信したメッセージの“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 5.5)。

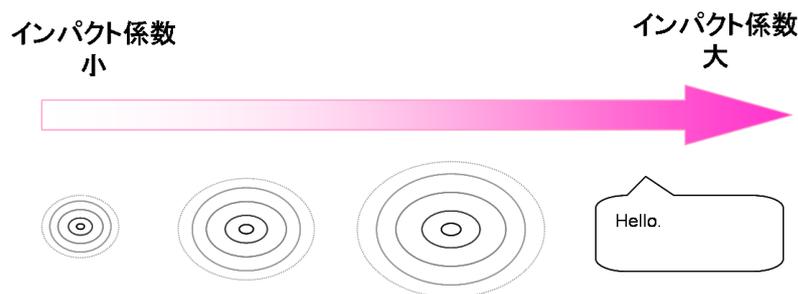


図 5.5: インパクト係数による提示方法の違い

インパクト係数によるユーザのコンテキストを考慮したざわめき情報の提示は、4.2 節で述べた実世界のざわめきの性質のうち、定常性および選択性を実現している。すなわち、実世界においてユーザは定常的にざわめきを聴きながらもコンテキストによって無意識に取捨選択するのと同様に、RippleDesk は定常的にコミュニティにおける会話の様子を受け取り、コンテキストに応じて波紋の規模を変化させる。

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。

$$I = \frac{\phi_{kwd} + \phi_{com} + \phi_{app}}{3} \quad (5.1)$$

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。

$$a = \sqrt{l_{msg}} \cdot I \quad (5.2)$$

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。

5.3 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 5.6 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

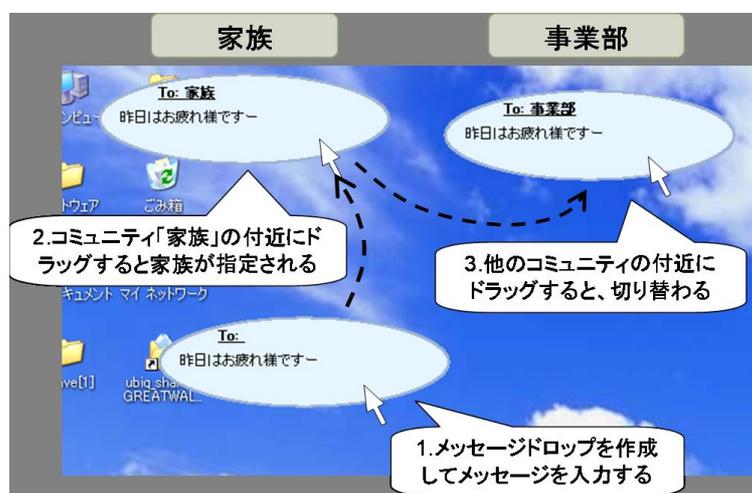


図 5.6: メッセージドロップの作成

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。

5.4 その他の機能

その他、RippleDesk は次のような機能を持つ。

5.4.1 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 5.7)。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

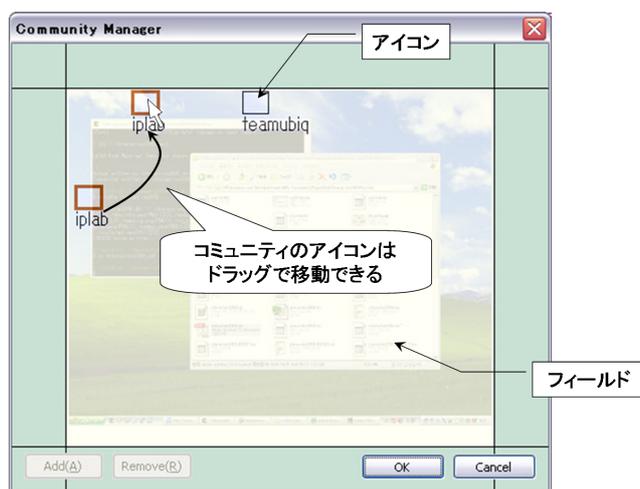


図 5.7: Community Manager

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。コミュニティはデスクトップの周囲に複数配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。

5.4.2 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 [8]。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。

RippleDesk システムでは、専用のウィンドウにより Cocktail Word を登録することができる (図 5.8)。それぞれの単語に対して、スライダーコントロールを用いて重みを関連付けることができる。



図 5.8: Cocktail Word と重みの関連付け

5.4.3 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。

5.4.4 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。

5.5 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。

5.5.1 通信プロトコル

RippleDesk の通信プロトコルは RFC[9] により定められている Internet Relay Chat プロトコル (RFC1459) 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。

5.5.2 波紋の描画アルゴリズム

システムは波紋フィールドとよばれる仮想の水面を持つ。波紋フィールドは、デスクトップ画面と同じ大きさの 8 ビットのメモリ領域であり、それぞれの要素はその場所の振幅値を表す。

システムは、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する。その後、変形されたキャプチャ画像をデスクトップに描画する。振幅の大きい場所では、キャプチャされたデスクトップの画像は大きく歪み、また、振幅が小さい場所ではそのまま描画される。結果的に、波紋フィールドに応じた擬似的な水面の屈折が表現される。

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。

第6章 柔らかいストレージの議論

柔らかいストレージを実現する方法のひとつに、人体に直接無線アクセス可能なメモリを埋め込む方法も考えられる。人体に埋め込む RFID はすでに実用化されているが [11]、身体的および精神的な不安が伴う。そのため、人体から ID を検出し、その ID とストレージを対応付けることで仮想的にストレージを実現する方法を用いている。アプライアンス間でデータを移動するために、リムーバブルストレージを介さずに、ネットワーク共有ストレージを利用する方法がある。ネットワーク共有ストレージの利点は、柔らかいストレージと同様に、物理的負担を伴わないことである。利用者の視点からは、ネットワーク共有ストレージは形の無い論理的な概念であり、見かけの保存先があいまいなものである。

柔らかいストレージは仮想的なネットワークストレージである。これに対して物理的な媒体を用いるシステムがいくつか提案されている。The tangible file[12] は、RFID タグを取り付けた紙をファイルとして扱うシステムである。また The Personal Server[13] は、個人専用のデータを持ち歩くための小型サーバである。これらのシステムは、いずれも物理的な媒体を利用してストレージを実現しているため、持ち運ばなくてはならない、盗難、破損などの物理的負担や制約がある。これに対して柔らかいストレージは人体をキーとして利用しているため、そのような問題がない。

日常生活で扱うコンテンツの量を考えると、柔らかいストレージには多くのデータが記録されることが予想される。記録された多くデータをどう扱うかの問題は、本研究の大きな課題である。現在のファイルシステムに習えば、柔らかいストレージにディレクトリ構造を持たせることが考えられる。しかしながらディレクトリ構造を持たせることで、見かけの保存先が体そのものではなく、体の中のディレクトリとして認知される恐れがある。これにより、体内のディレクトリ内を検索する必要性が出てくるため、検討が必要である。

柔らかいストレージにアクセスするアプライアンスは、無線ネットワークなどの設備が整った環境を必要とする。応用される場所の候補としては、まず ActiveSpace[14] のようなコンピュータが遍在する空間が挙げられる。このような環境が将来、社会全体に整備されるかどうかは興味深いが、大規模な空間でなくとも、例えば教室やオフィスのような限定された空間は柔らかいストレージを利用するのに適していると考えられる。

また、柔らかいストレージを、大きな規模の空間で利用するにはセキュリティの問題がある。これは、(1)ID の盗難や不正な利用、(2) ネットワークを通じたデータの保管、(3) アクセス権の付与、に分類される。(1) については、今後バイオメトリクスが普及する上での問題と共通している。(2) については、暗号化された通信が必須となる。(3) について、柔らかいストレージは個人利用のストレージとして考えている。

第7章 波紋による情報提示の実験

本章では RippleDesk システムの実験について述べる。実験 1 では、波紋の発生する位置とコミュニティの関連付けがどの程度の解像度で認識できるかについて、実験 2 および 3 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調べた。

7.1 実験 1: コミュニティの解像度

RippleDesk は波紋の中心点によってコミュニティの場所を表現する。しかし、コミュニティの場所を多く設定すればするほど、波紋がどのコミュニティから発生したものが認識しにくくなる。そこで、波紋によってどの程度の数のコミュニティが認識できるかを実験した。

7.1.1 実験方法

まず事前準備として、デスクトップ画面の上部に、波紋を発生させるポイントを一定の間隔で設定する。また、それぞれのポイントに、左端のポイントを 1 番目としてコミュニティ番号を書き込んだ付箋紙を物理的に取り付ける。被験者には実験の前に 1 分間、番号と波紋の発生するポイントの対応をあらかじめ覚えてもらう。次に、付箋紙を取り払い、ランダムなポイントに波紋を発生させる。被験者には、それが何番のポイントから発生したか、番号で答えてもらう。これを 20 回行い、1 セットとする。被験者は 3 名で、いずれも学生である。1 人につき、ポイントの間隔を 200 ピクセル、100 ピクセル、50 ピクセルに変え、計 3 セット行った。波紋の大きさは、中心から 100 ピクセルの距離で初期振幅の 10% 以下に減衰する値に設定した。インパクト係数の値としては約 0.5 に相当する。実験に用いたディスプレイは 17 インチの液晶モニターで、デスクトップ画面の解像度は 1024 × 768 ピクセルである。デスクトップの壁紙には一般的な森林の画像を使用した。

7.1.2 実験結果

結果を図 7.1 に示す。

被験者 A,B,C とともに、波紋の発生ポイントの間隔が狭くなるほど正答率が低下した。200 ピクセルの間隔では平均して 9 割以上の確率で、被験者は正しい番号を答えた。100 ピクセル間隔では、7 割程度までに低下した。50 ピクセル間隔で波紋を発生させると、半分以上、正しい発生ポイントの番号を答えることができなかった。

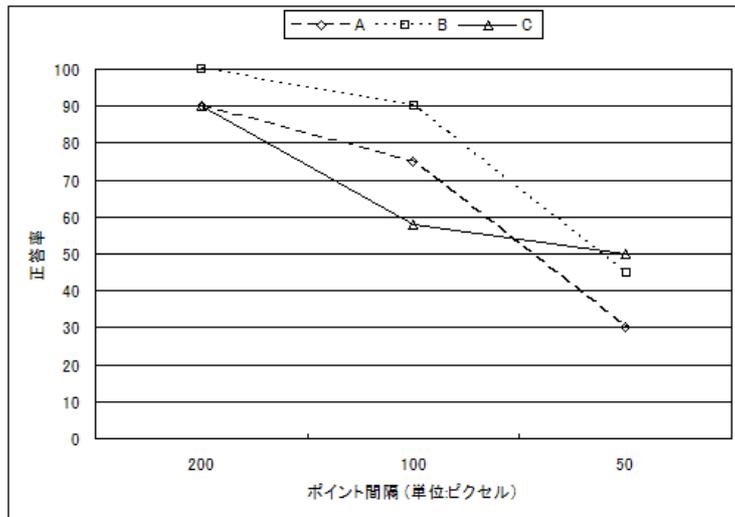


図 7.1: 波紋の間隔と正答率の関係

7.2 実験 2: 会話の活発化の伝達

実験 2 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。

7.2.1 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お k
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には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。

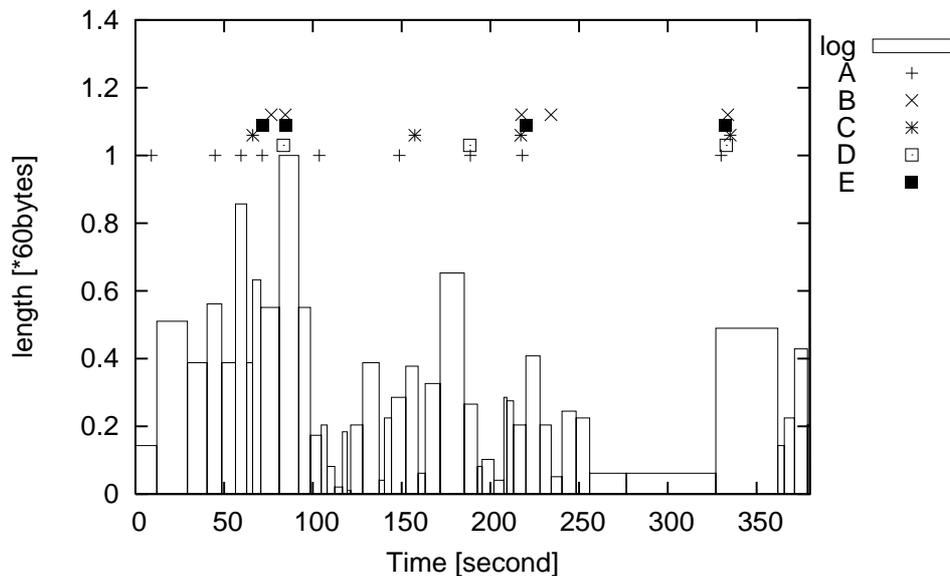


図 7.2: 実験結果: 発言とクリックポイントの関係

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。

7.2.2 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 7.2 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。

7.3 実験3: 連続的な会話状況の伝達

実験2において、会話の活発さをある時刻の1点として入力するよう被験者に求めた。実験3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。

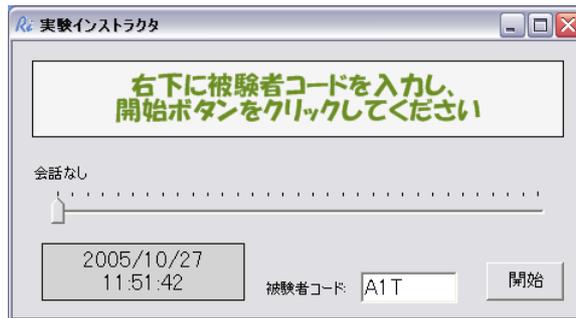


図 7.3: 実験インストラクタ

7.3.1 実験方法

実験用に簡単なソフトウェアを作成した(図 7.3)。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。

7.3.2 実験結果

実験結果の一部を図 7.4 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。

7.4 実験結果についての考察

実験 1 の結果からは、200 ピクセル以上の間隔を空けてコミュニティの場所を設定することが実用的であると考えられる。それ以下の間隔では、波紋の発生位置から予想したコミュニティと実際のコミュニティが異なる場合に、誤ったコミュニティにメッセージを送信してしまうなど、トラブルが発生する可能性がある。一般的なデスクトップ画面の解像度を 1280x1024 ピクセルとすれば、1 辺あたり 5~6 個程度であるから、デスクトップ上部以外に左右の辺を利用すればその 3 倍程度の数のコミュニティに所属することが可能と思われる。

また、実験後の被験者の話から、3 名ともに、実験に用いた森林の壁紙に存在する特徴的な部分と、波紋の発生するポイントの番号を関連付けてコミュニティの場所を覚えることがわ

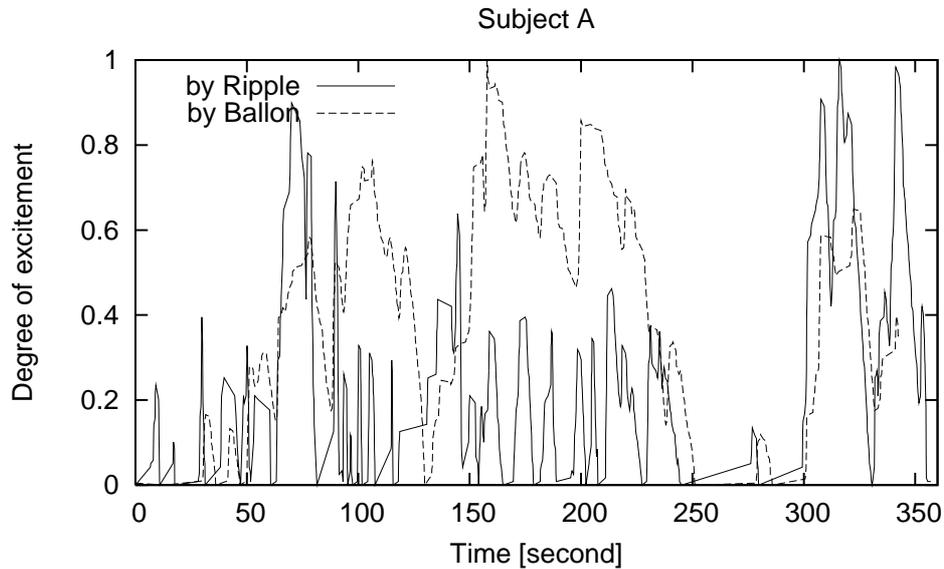


図 7.4: 被験者 A の実験結果

かった。(写真の中の白っぽい木から発生した波紋は3番のコミュニティである、等)つまり、単純にデスクトップ画面のx座標だけでなく、壁紙の特徴的な部分をコミュニティの場所を手がかりとしていた。このことから、壁紙の種類によって認識できるコミュニティの数に変化する可能性がある。

実験2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3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3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。

第8章 関連研究

8.1 柔らかいストレージに関して

物にIDを付与してオンライン上のデータと結び付けるシステムは多くある。mediaBlocks[15]は、RFID タグを取り付けた木のブロックを、オンラインデータとリンクさせて、デジタルメディアデータのコンテナとして扱っている。このシステムでは見かけの保存先が木のブロックである。Pick and Drop[16]は、コンピュータ間におけるデータの移動を行うインタフェースについての研究である。ペン型のデバイスを用いて、紙からコンピュータへの見かけ上のデータの移動を可能としている。IconSticker[17]は、デスクトップのアイコンを実世界に取り出すメタファを用いている。紙アイコンと呼ばれるバーコードを印刷した紙によって、データの移動を行うことができる。これらのシステムは、全て実世界の日用品もしくは専用のメディアを利用するので、リムーバブルメディアと同様に、物理的な扱いに起因する問題がある。それに対し、柔らかいストレージは利用者の体以外の物を扱う必要がない。

8.2 波紋によるざわめきの提示に関して

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[18]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[19]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ーアロマ発生器を作動させることで知らせる。コーヒーの香りが、離れた場所で休憩が起き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[20]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[21]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。

RippleDesk では、実行しているアプリケーションから忙しさを推定してメッセージの提示方法を変化させる。ユーザの状況に応じて適切なインタラクションを提供することを目的とした研究としては、実世界のデスクワークの状況から忙しさを自動判定する研究 [22] がある。

第9章 まとめと今後の展望

ユビキタス環境において情報アプライアンスと共に利用されるストレージとして、柔らかいストレージを提案した。柔らかいストレージは、見かけ上、人の体に埋め込まれたストレージであり、情報アプライアンスが適切な情報の提示手法を実装することで実現される。今後、各プロトタイプを用いた利用実験を行い、実際に体にストレージがあるかの様に人を錯覚させられるかどうかについて、評価を行う。さらに、柔らかいストレージを用いたアプライアンスの利用について、従来型のストレージと比較を行い、定性的な評価をしたいと考えている。

次に、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、文脈に応じた会話の活発さに応じて波紋のパラメータを変化させることで、ユーザが明示的に会話を見ることなく波紋のみからより正確な情報を得られるようにしたいと考えている。

謝辞

本研究の機会を与えて頂きました筑波大学システム情報工学研究科 田中二郎教授には、終始懇切なご指導とご助言を頂きました。ここに深く感謝致します。また、筑波大学システム情報工学研究科 三末和男助教授ならびに志築文太郎講師には大変有益な議論の機会を与えて頂きました。心から感謝申し上げます。さらに、筑波大学システム情報工学研究科 高橋伸講師には、研究内容に留まらず進捗等についてもご指導頂きました。心から感謝致します。筑波大学システム情報工学研究科 インタラクティブプログラミング研究室のメンバーの方々にも大変お世話になりました。この場を借りて厚く御礼申し上げます。

参考文献

- [1] Mark Weiser. The computer for the twenty-first century. In *Scientific American*, pp. 94–104. Scientific American, September 1991.
- [2] Donald A. Norman. *The Invisible Computer*. MIT Press, 1998.
- [3] 椎尾一郎, Jim Rowan, 美馬のゆり, Elizabeth Mynatt. Digital Decor: 日用品コンピューティング. In *The 10th Workshop on Interactive Systems and Software*, 2002.
- [4] 岩淵志学, 志築文太郎, 田中二郎. 柔らかいストレージ: 体と融合したストレージの提案と実装. 日本ソフトウェア科学会第 21 回大会 論文集, 2004.
- [5] 岩淵志学, 志築文太郎, 三末和男, 田中二郎. 柔らかいストレージ: 見かけ上体に存在するストレージの利用. *インタラクション 2005 論文集*, 2005.
- [6] Shigaku IWABUCHI, Buntarou SHIZUKI, Kazuo MISUE, and Jiro TANAKA. Natural Storage in Human Body. In *(KES2005) 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, Vol. 4, pp. 430–436, 2005.
- [7] ARToolKit. <http://www.hitl.washington.edu/artoolkit/>.
- [8] Barry Arons. A review of the cocktail party effect. *Journal of the American Voice I/O Society*, Vol. 12, pp. 35–50, July 1992.
- [9] Request for Comments. <http://www.ietf.org/rfc.html>.
- [10] 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.
- [11] VeriChip. <http://www.4verichip.com/>.
- [12] Alexandros Karypidis and Spyros Lalis. The tangible file system. In *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)*, pp. 268–273, May 2003.

- [13] Roy Want, Trevor Pering, Gunner Danneels, Muthu Kumar, Murali Sundar, and John Light. The Personal Server: Changing the Way We Think about Ubiquitous Computing. In *Proceedings of Ubicomp 2002: 4th International Conference on Ubiquitous Computing*, pp. 194–209, Sep 2002.
- [14] Manuel Roman, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, and Roy H. Gaia: A middleware infrastructure to enable active spaces. In *IEEE Pervasive Computing*, pp. 74–83, Oct-Dec 2002.
- [15] Brygg Ullmer, Hiroshi Ishii, and Dylan Glas. mediaBlocks: Physical Containers, Transports, and Controls for Online Media. In *SIGGRAPH'98, Computer Graphics Proceedings*. ACM, July 1998.
- [16] Jun Rekimoto. Pick-and-drop: A direct manipulation technique for multiple computer environments. In *Proceedings of UIST'97*, pp. 31–39, 1997.
- [17] Itiro Siio and Yoshiaki Mima. IconStickers: Converting Computer Icons into Real Paper Icons. In *Human-Computer Interaction, Ergonomics and User Interfaces*, Vol. 1 (HCI International '99), pp. 271–275, August 1999.
- [18] 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.
- [19] 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.
- [20] Albrecht Schmidt and Hans-W Gellersen. Visitor awareness in the web. In *In Proceedings of the tenth international conference on World Wide Web*, pp. 745–753, 2001.
- [21] 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.
- [22] 水口充, 竹内友則, 倉本到, 渋谷雄, 辻野嘉宏. デスクワークにおける忙しさの自動推定. ヒューマンインタフェース学会論文誌, Vol. 6, No. 1, pp. 69–74, 2004.