

卒業論文 2006 年度 (平成 18 年度)

利用者の状況と好みに基づいた適切な  
コミュニケーション手段の選択手法に関する研究

慶應義塾大学 環境情報学部

奥村 祐介

## 利用者の状況と好みに基づいた適切なコミュニケーション手段の 選択手法に関する研究

### 論文要旨

本研究は、インターネット上のコミュニケーションについて、受信者の“状況”と“コミュニケーションサービスの好み”を考慮した、最適な受信方法の選択する手法を提案する。

インターネット上のコミュニケーションは、技術の進化と利用者のニーズの変化によって多様化している。また、多くの利用者はコンピュータ、携帯電話、ゲーム機などの複数の通信機器を所持し、電子メール、インスタントメッセージングなど複数のサービスを状況によって使い分けている。送信者と受信者が複数のコミュニケーションサービスを利用している場合、送信者は、受信者が利用している複数のサービスから一つを選択する。一方、受信者は、同時に複数のサービスを利用していても、どのサービスで受信するかを選ぶことはできない、このことから、送信者が受信者の状況を把握できなければ、受信者に最適なサービスを利用したコミュニケーションにならない可能性がある。

そこで本研究は、受信者の状況に“最適”なコミュニケーションサービスを決定するために、キーボードやマウスの入力イベントとその入力先アプリケーション情報から状況を類推し、利用者の好みに基づいて最適なコミュニケーションサービスを選択する手法を提案した。また、本手法を評価するために、入力イベントを取得し状況を推測するソフトウェアを実装し、被験者が提示されたシナリオに沿ってシステムを利用する検証実験を行った。結果として、コンピュータを継続的に利用するシナリオでは 97.4% という高い中率を確認できた。しかし、一定時間以上コンピュータを利用しない状況が含まれるシナリオでは、42.0% という中率であった。

この結果から、本研究は、利用者のニーズを考慮したメッセージの送受信を実現するための、入力イベントを用いた推測手法の有用性と問題点を明らかにすることができた。

### キーワード

1. ユニファイド・メッセージング, 2. コンテキスト・アウェアネス

<p style="text-align: center;">A Research on a Selection Technique of Appropriate Communication Path Based on User Context and Preferences</p>
--

Summary:

This research proposes a method to choose the best scheme for receiving path that considers its “situation” and “preference of communication service” for the Internet communication .

The communication on the Internet has been diversified by the evolution of technologies and the transformation of user's demands . A lot of users have a couple of communication devices like computers , mobile phones and portable game devices . And these devices provides e-mail , instant messaging and other multiple communication services .

In case that both of a sender and a receiver are able to use same communication services , the sender can choose one from those . However, the receiver can not choose the most appropriate even multiple services are available .

To solve this issue , we propose a new method to choose the most appropriate communication service depends on the inferred situation of the receiver .

Our method infer the situation of the receiver from the events from keyboards and mouses , and the target application software .

To evaluate our method , we implemented the prototype system which grabs input events and infers the situation . We tried to evaluate this prototype with the evaluation experiment which subject users use the system in keeping with the proposed scenario . We confirmed the system could achieve the high hit rate 97 . 4% with the scenario which the subject used the computer continuously , while the hit rate is 42 . 0% for the scenario which includes periods without computer usages .

From these results , we proved the effectiveness and problems of our proposed method , which provides the appropriate communication service selection depends on the input events of the users .

Keywords:

1 . Unified Messaging , 2 . Context awareness

# 目次

第 1 章	序論	1
1.1	はじめに	1
1.2	本研究の目的	1
1.3	本論文の構成	2
第 2 章	背景	3
2.1	メッセージとは	3
2.2	メッセージングサービスの現状	3
2.2.1	通信機器の多様性	3
2.2.2	機器の普及率	6
2.2.3	通信サービスの多様性	7
2.3	メッセージの即時性消失の問題	8
第 3 章	関連研究	10
3.1	Tangible Chat	10
3.2	Sentient Computing Project	11
3.3	MIRAI	11
3.4	EAPEC	12
第 4 章	既存のユーザ状態取得手法の問題点	13
4.1	ユーザ状態情報の不足	13
4.2	ユーザごとに異なるコミュニケーション手段の好みへの対応不足	14
4.3	導入障壁が高い	14
第 5 章	提案するアプローチ	15
5.1	イベント情報とユーザの状態の関連性	15
5.2	入力イベントを取得するための導入障壁	15
5.3	推測手法検証のための予備実験	16

5.3.1	入力イベントの記録蓄積	16
5.3.2	入力イベントログの解析	16
5.4	コミュニケーション手段決定手法の提案	17
5.4.1	ユーザ状態情報の取得	18
5.4.2	ユーザによるサービス優先度の指定	18
第 6 章	設計	20
6.1	設計概要	20
6.2	ユーザコンテキストの生成	21
6.2.1	入力イベントの取得と記録	21
6.2.2	各ユーザ状態の実行状態判定	22
6.3	ユーザプリファレンスの指定	22
6.3.1	ユーザ状態機能毎のサービスプリファレンス	23
6.3.2	ユーザ状態機能の優先度プリファレンス	23
6.4	最適なコミュニケーションサービスの決定	23
第 7 章	実装	26
7.1	実装の動作環境	26
7.2	実装概要	26
7.3	入力イベント検知モジュールの実装	27
7.3.1	グローバルフックの利用	27
7.3.2	DLL の利用	28
7.3.3	入力イベントの検知	28
7.4	アプリケーション調査モジュール	28
7.5	情報蓄積モジュール	29
7.6	ユーザ行動監視モジュール	30
7.7	最適サービス決定モジュール	31
第 8 章	評価	33
8.1	検証実験	33
8.2	実験結果	34
8.2.1	被験者 A	35
8.2.2	被験者 B	36
8.2.3	被験者 C	37
8.3	検証実験の考察	38
8.3.1	入力イベント数が多いほどの中率が高い	38

8.3.2	設定ファイルの記述が困難 . . . . .	38
第9章	結論 . . . . .	40
9.1	まとめ . . . . .	40
9.2	今後の課題 . . . . .	41
9.2.1	入力イベント以外の情報の導入 . . . . .	41
9.2.2	システム設定の簡易化 . . . . .	41
9.2.3	メッセージの内容を考慮できない . . . . .	41
	参考文献 . . . . .	43

# 目次

2.1	携帯電話普及率の推移 . . . . .	6
2.2	インターネット利用者数の推移 . . . . .	6
2.3	インターネット利用者内の利用端末の割合 . . . . .	7
5.1	記録される入力イベントログの形式 . . . . .	16
5.2	アプリケーション毎の入力イベント数 . . . . .	17
6.1	設計概要図 . . . . .	20
6.2	入力イベントの記録 . . . . .	22
6.3	データの概要 . . . . .	24
6.4	サービスの決定手順 . . . . .	25
7.1	実装概要図 . . . . .	27
7.2	グローバルフックの設定に利用する関数 . . . . .	28
7.3	入力イベントが発生したアプリケーションの特定 . . . . .	29
7.4	ユーザ行動のデータ構造 . . . . .	30
7.5	利用する構造体のデータ構造 . . . . .	31
7.6	ユーザ行動監視モジュールの動作 . . . . .	32
8.1	設定ファイルに記述する要素 . . . . .	34
8.2	被験者 A の設定内容 . . . . .	35
8.3	被験者 B の設定内容 . . . . .	36
8.4	被験者 C の設定内容 . . . . .	37

# 表目次

5.1	アプリケーションの分類 . . . . .	18
6.1	生成されるユーザコンテキストの例 . . . . .	23
8.1	実験結果 . . . . .	34



# 第 1 章

## 序論

### 1.1 はじめに

インターネットを用いたコミュニケーション手段として、E-mail やインスタントメッセージング、IP(Internet Protocol) 電話などが存在する。これらのコミュニケーション手段は、インターネット利用者にとって欠かせないものになりつつある。例えば、E-mail で仕事の報告をしたり、海外の友人と連絡を取り合う、といった利用方法は一般的なものとなった。インターネットでのパケット伝送の遅延が解消するにつれ、インターネットを用いた音声通話が普及し、オフィスで IP 電話を利用する例も多く見られる。

コミュニケーションサービスの多様化も進んでいる。文字を用いたサービスでは、複数人が一度にメッセージを送受信するチャットや、短いメッセージを送受信するインスタントメッセージングサービスが普及した。また、音声を用いた IP 電話サービスや、映像と音声を利用した遠隔会議システムも実用化されている。

近年では、コミュニケーションを可能にする通信機器を複数利用する利用者も増えてきた。据え置き通信端末ではなく、持ち運ぶことのできる通信端末の登場によってその傾向は強まった。常に携帯電話を持ち歩く利用者は多く、さらにポケットサイズのコンピュータやラップトップコンピュータを持ち歩くユーザも増えた。

これらの背景により、利用者にとってコミュニケーションサービスと通信端末の選択肢が多様化してきている。

### 1.2 本研究の目的

現在の多様化したコミュニケーション環境下では、メッセージの発信者は相手の状況を推測し、多種多様な通信機器やサービスを自分で使い分けなければならない。本研究の目的は、受信者の状況を考慮した最適なコミュニケーション手段の推測を実現することで、

ユーザのコミュニケーションを支援することである。本研究では、ユーザの状況の取得にコンピュータに対する入力操作情報を利用し、ユーザにコミュニケーションサービスについての好みを指定させることにより、最適なサービスを決定する手法を提案する。そしてこれらの手法を実現するシステムを設計、実装し、サービスの決定に有用な手法であることを検証する。

### 1.3 本論文の構成

本論文は9章から構成される。第2章では、メッセージングサービスの現状を述べ、インターネット利用者を取り巻くコミュニケーション環境について述べる。第3章では、コミュニケーション環境の向上にむけた関連研究について述べる。第4章では、第3章で述べた関連研究における問題点について述べる。第5章において、その問題点を解決することのできるアプローチを提案する。第6章において、提案するアプローチを元に考案した、本システム設計に関して述べる。第7章で設計に基づいた実装に関して述べる。第8章で本システムの手法と実装に対する評価を行い、既存の問題を解決したか否かを述べる。最後に、第9章で本研究のまとめと今後の展望を述べる。

## 第 2 章

# 背景

### 2.1 メッセージとは

コミュニケーションとは，人と人が意思や感情，情報を伝え合うことを指す．メッセージとは，コミュニケーションにおいて相手に伝えたい内容を，送受信する単位でまとめたものである．メッセージは，直接対話できる状況において，メディアを介さないコミュニケーションができる場合には会話や身振り手振りで伝えられる．しかし会話ができない状況では表現形式が変化し，メッセージを文字や音声，画像，映像のといったメディアを介して相手に伝送する．このための伝送手段を，コミュニケーションサービスとする．メッセージには，電話での会話の一つ一つの言葉，手紙における文面，E-mail における件名や本文，ポケベルメッセージの本文，インスタントメッセンジャーにおける一文などがメッセージにあたる．これらのコミュニケーションサービスを用いてメッセージを送受信することを，メッセージングと呼ぶ．

### 2.2 メッセージングサービスの現状

本節では，本研究の重要な背景であるコミュニケーションの多用性を明らかにするために，通信機器及びメッセージングサービスの普及と多様化の現状について述べる．

#### 2.2.1 通信機器の多様性

計算機の著しい発展に伴い，様々な形態の情報通信端末が登場し，普及した．その多様な通信機器の特徴を以下に示す．

### デスクトップ型コンピュータ

デスクトップ型コンピュータとは、主に机の上に備え置いて使用する用途で作られたパーソナルコンピュータのことである。以下、デスクトップ PC と呼ぶ。デスクトップ PC は、移動して利用するには適していない。なぜなら、一般的に本体とディスプレイが別になっており、また体積が比較的大きいために持ち運ぶには適していないからである。

一方でデスクトップ PC は、後述するラップトップ、携帯電話などと比較して機能の拡張性が高く、ディスプレイやキーボードなどの、PC を操作する機器を自由に選択できる。さらに、その他の機器も必要に応じて自由に追加が容易であるため、ユーザの趣向に合わせた PC を作成できる。また、自由にアプリケーションを追加することができる。アプリケーションとは、コミュニケーションサービスを利用するために必要なソフトウェアである。利用したいコミュニケーションサービスに応じてアプリケーションを追加することにより、新しいサービスを利用できるようになる。これらの機器とアプリケーションのカスタマイズができる特徴を持つため、デスクトップ PC ではユーザにとって使いやすい環境を整えやすい。

インターネットへの接続は、機器が移動しないことを前提としているため、有線を利用することがほとんどである。最近では、インターネットへの常時接続サービスが普及しており、インターネットに接続されている時間は長くなってきている。

一人のユーザが複数台のデスクトップ PC を所持している事も多い。例えば、自宅とオフィスに一台ずつという環境を持つユーザも少なくない。

### ラップトップ型コンピュータ

ラップトップ型コンピュータとは、小型で持ち運びしやすい、携帯型パーソナルコンピュータの事である。ラップトップという名前は、膝の上で操作することから付いたものである。以下、ラップトップ PC と呼ぶ。ラップトップ PC は、持ち運びが容易であるため、出掛けた先や移動中など、様々な場所で PC を使用できる。

ラップトップ PC の機器拡張性は、USB 規格 [6][7] の普及により向上している。以前ラップトップ PC は、機器を接続するために採用している規格が複数あり統一されていなかった。そのため、ラップトップ PC と接続機器の規格を合わせるのが面倒であり、機器拡張性は低かった。しかし、機器を接続しデータを伝送するための汎用的な規格である USB が普及し、ラップトップ PC と接続機器の両方が USB に対応したため、様々な機器をラップトップ PC に接続することが容易になった。またラップトップ PC は、デスクトップ PC と同じように、アプリケーションの追加も容易である。

最近では、ラップトップ PC がインターネットに接続するために、様々な方法が提供されている。PHS の回線を利用した無線アクセス通信網を提供するサービス [10][11] や、街

中で無線のアクセスポイントを提供するサービス [5][12] などがある。これらのインターネット接続サービスの登場と普及により、ラップトップ PC が場所と時間を問わずインターネットに接続できる環境が整ってきている。

一人のユーザが、デスクトップ PC とラップトップ PC を併用することも多い。例えば、移動中にラップトップを利用し、オフィスではデスクトップを利用するという状況も一般的な光景となった。

### 携帯電話

携帯電話は、無線通信を利用した持ち歩ける電話機である。携帯電話はデスクトップ PC やラップトップ PC と比べ体積と質量が小さく、大変持ち運びやすい。そのため、ユーザがコンピュータからメモや移動中閲覧したい内容を携帯電話に転送し、閲覧する使い方もある。

携帯電話の機器拡張性は低い。携帯電話を持つことなく通話をするために、マイクとイヤホンを接続することがあるが、それ以外に機器を接続することはほとんど無い。アプリケーションの追加の自由度も低い。一部の機種は汎用 OS(オペレーションシステム)の上システムが構築されており、アプリケーションの追加が可能なものもあるが、多くの機種は購入時にサポートされているコミュニケーションサービス以外を利用することはできない。

携帯電話の普及率は高く、日本人口の 89.6% の人が所持している [14]。普及率の推移を図 2.1 に示す。コミュニケーションサービスの中で広く普及している電話と E-mail の機能を利用し、携帯電話だけでコミュニケーションを行う人も少なくない。

### PDA (Personal Digital Assistant)

PDA(Personal Digital Assistant) とは、手の平に収まるサイズの持ち運びしやすい小型コンピュータである。小型の液晶を持っており、キーボードは機器に内蔵されているタイプと、液晶のタッチパネルを操作することで入力をするタイプがある。PDA には携帯電話機能を備えたタイプも登場し、それらは Smartphone と呼ばれる。

機器の拡張性は低い。新たな機器を接続することはあまり考えられておらず、外部記憶メディアや外付けキーボードなどに限られている。機種にもよるが、多くの PDA は汎用 OS の上にシステムが構築されているため、アプリケーションの追加は容易である。携帯端末専用の OS が利用されるため、アプリケーションによっては導入が不可能な場合もある。

インターネットへの接続は、無線 LAN を利用して行うことが多いため、移動中の利用が可能である。また、Smartphone の場合電話網を利用してインターネットに接続できるため、電話網の電波が受信できる場所であればインターネットを利用できる。

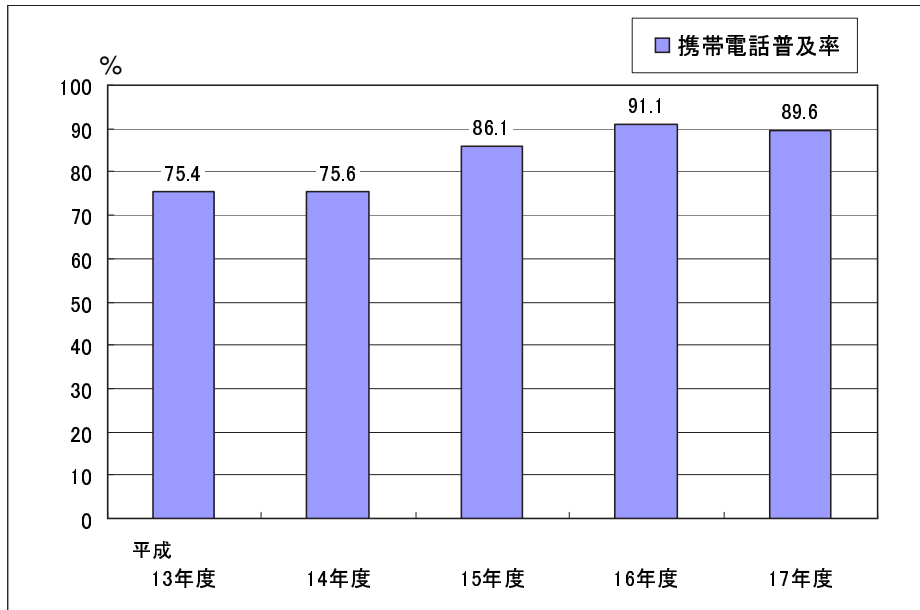


図 2.1 携帯電話普及率の推移

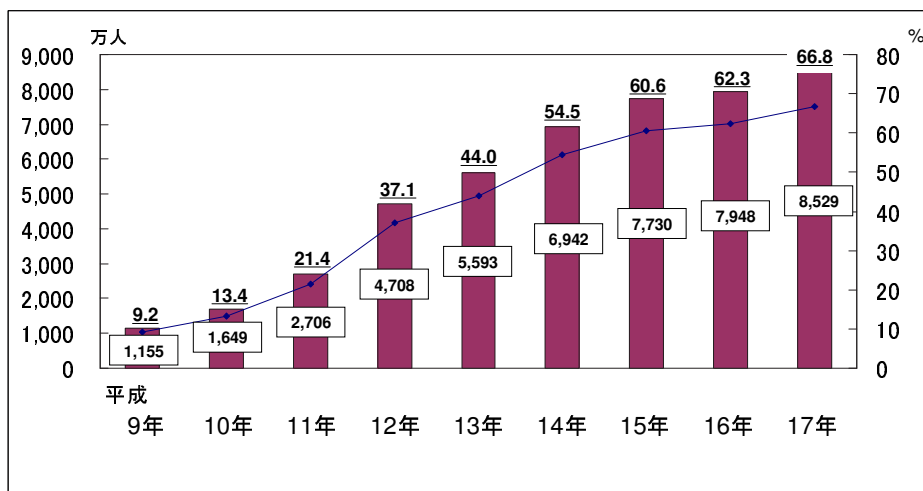


図 2.2 インターネット利用者数の推移

これまで PDA は、携帯電話より持ち運びづらく、ラップトップより性能が低いという点で中途半端なイメージを拭えず、ユーザは少なかった。しかし、高性能で小型の PDA が登場することにより日本国内でも普及の兆しが見え始めている。

## 2.2.2 機器の普及率

総務省の平成 17 年度通信利用動向調査 [14] の結果によると、インターネット利用者推計 8529 万人に達した。インターネット利用者数の推移を図 2.2 に示す。これは日本全人

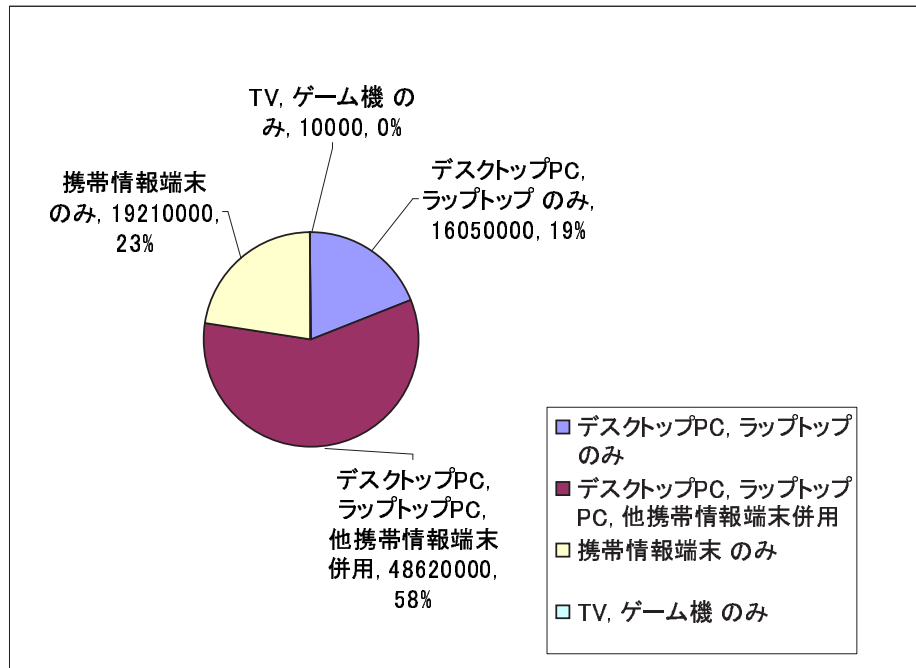


図 2.3 インターネット利用者内の利用端末の割合

口の約 67% であり、多くの人インターネットを利用していることが分かる。また、図 2.2 から分かるように、インターネット利用者は右肩上がりに増加しており、今後ますますインターネットは普及すると考えられる。

また、図 2.3 にインターネット利用者の所持する通信機器の割合を示した。図 2.3 によると、半数を超える利用者が、PC や携帯電話、PHS などの端末を併せて所持していることが分かる。

### 2.2.3 通信サービスの多様性

ここでは、現在広く利用されているインターネットを用いたコミュニケーションサービスについて述べる。

#### E-mail

E-mail は、電子メールとも呼ばれる。PC における E-mail は、インターネットを通じて文字メッセージを交換する。メールの送信者は、送信者の利用するサーバを介してメールを送信する。送信されたメールは SMTP を用いて受信者のメールサーバまで運ばれ、受信者は任意のタイミングでサーバにアクセスし、メールをダウンロードする。受信者による能動的なダウンロードによって配送が実現されるプル型のコミュニケーションサービスである。

携帯電話での E-mail も，基本的には PC の E-mail と同じである．しかし，メール配送時のみ異なっており，メールサーバにメールが到着すると，携帯電話の場合自動的にメールの配送が行われる．つまり，携帯電話においては，E-mail はプッシュ型のサービスになっている．

### インスタント・メッセージング (IM)

インターネット上で同一のサービスを利用している知人がオンラインかどうかを調べ，オンラインであればチャットやファイル転送などを行えるサービスを，インスタントメッセージング (IM) と呼ぶ．オンラインとは，インターネットに接続し，サービスを利用できる状態にあることを指す．IM では，会話をしているかのようにリアルタイムに文字のメッセージを交換できる．欠点として，直接の対話とは違いメッセージの作成状況が相手に正確に伝わらないため，お互いが同時にメッセージが送信してしまうと意見が交差してしまう点が挙げられる．また，短いメッセージをお互いに送信し合う使われ方が一般的であり，雑談をするには有効であるが，まとめた内容をやりとりするには適さない．普及している IM サービスとして，MSN Messenger[3] や Yahoo!Messenger[8] などがある．

### インターネット電話

インターネットを利用した音声通話サービスである．最近では，知人のリストを作成し，オンラインかどうかを調べた上で通話を始めることが可能なサービスが普及している．文字でのメッセージングではなく，音声を用いてコミュニケーションできる．メッセージとして音声を利用するため，マイク等の，機器に音声を入力するためのデバイスが必要である．デバイスとは，機能を追加するために必要なコンピュータの周辺機器のことである．また，文字を利用したメッセージングとは違い，実際に声を出さなければならない．そのため，電車の中や会議中など声を出ることができない場所では利用できず，メールや IM と比べて利用場所が制限される不便さがある．

## 2.3 メッセージの即時性消失の問題

送信者は多くの場合，メッセージを送信する際にメッセージが受信者にできるだけ速く確実に配送されるよう，送信先の機器やコミュニケーションサービスを決定する．しかし，現在の多様化したコミュニケーション環境において，受信者はメッセージを受信するための通信機器とサービスの組み合わせを複数持つため，メッセージを受信者が実際に受け取るまでに時間がかかってしまう場合がある．

例えば，受信者が携帯していない通信機器にメッセージを送信してしまう場合が考えられる．他にも E-mail では，電池の無くなった携帯電話のアドレス宛に E-mail を送ってし



まったり、会社でしか受け取れないメールアドレスにメッセージを送信してしまったり、などの可能性がある。また、IM にログインしたまま出掛けてしまった PC に、メッセージが送られてしまう可能性もある。

これらの問題は、受信者が通信機器やサービスを複数利用している現状では避けられない問題である。

このように、現在の多様化したコミュニケーション環境において、インターネット利用者は通信機器やサービスの多くの選択肢の中から、一つを選んでコミュニケーションを行っている。メッセージの送信者は、受信者の状況を考慮しながらコミュニケーション手段を選択している。この選択肢の決定は、送信者にとって大きな負担になるものである。この問題を解決するためには、受信者のユーザの状態とコミュニケーション手段の好みを考慮し、ユーザにとって最適なコミュニケーション手段にメッセージを配送する必要がある。

## 第3章

# 関連研究

第2章, 第4章において, 現在のコミュニケーション環境化でメッセージを迅速に受信者に届けるには, 受信者のユーザの状態を考慮し, ユーザにとって最適なコミュニケーション手段にメッセージを配送する必要がある点を指摘した. そこで本節では, 以下の点に着目し既存手法の調査を行った.

- ユーザの状況を推測する
- 状況に応じて, 利用するコミュニケーションサービスを変更する

以下では, 本研究との関連性を明確にするために, 各手法のアプローチのあと, どのようにしてコミュニケーション環境を支援するのかを述べる. また, 考える利点と欠点を定性的に評価した.

### 3.1 Tangible Chat

Tangible Chat[15] はチャットにおいて, 文字だけの送受信ではなく打鍵情報を共有することで, 対話をよりスムーズにすることを目的とした研究である. この研究では, 送信者が発話した心境や状況が, 文字の送受信だけでは分からないことを問題点に挙げている. TangibleChat では, 打鍵振動を座面の振動として通信相手に伝えることで, 様子を伝え合う. 例えば, チャットにおいて「はい」という一言であっても文字情報だけでは, 本当に了解したのか, しつこく了解したのかの判別は付きにくい.

利点は, 打鍵の様子を伝えることで, 途中の様子が明確化されることである. ゆっくり入力しているか速く入力しているか, 何度も書き直しをしているかなどの状況が伝わることで, 受信者は送信者の様子を推測できる.

欠点として, 送信者の様子の推測は, 受信者の主観に影響される点が挙げられる. そのため, 何度も書き直しているという情報を受信しても, 単に書き間違えているのか, 入力

をためらっているかを明確に判断できず、誤認する可能性がある。他にも、必要な機器として加速度センサと振動子を内蔵したクッションが必要である。加速度センサはキーボードに内蔵され、打鍵の振動を検知する。また、クッションは、通信相手の打鍵の様子を振動でユーザに伝える。

## 3.2 Sentient Computing Project

Sentient Computing Project[1] は、ユーザの物理的な位置を検出し、コンテキストとして活用する研究である。Sentient Computing Project では、位置検出に超音波を利用する。各ユーザは Bat と呼ばれる超音波送信器を持ち歩き、室内の天井に備え付けられた複数の超音波受信機が音波を受信する。Bat との距離は各受信機によって異なるため、Bat から発信された音波は各受信機に到達するまでの伝播時間に違いが出る。その遅延時間差を利用し、屋内でのユーザの正確な位置や向きを求めることができる。

屋内での位置を正確に検出できることにより、ユーザが外出しているのか、机で作業しているのか、電話を利用しているのかなどの状態を推測することができる。そうして取得できるユーザコンテキストは、ユーザの場所に依じた電話転送などの、コミュニケーション環境の向上にむけて応用できると考えられている。

欠点として、導入コストの問題が挙げられる。この問題については、第四章で後述する。

## 3.3 MIRAI

異なる無線通信技術をユーザが意識することなく柔軟に利用できるシームレス通信環境の実現のために、異種無線統合ネットワーク MIRAI アーキテクチャ [13] が提案されている。現在インターネットに接続できる無線環境は 802.11a/b/g や PHS、携帯電話の 3G など多岐に渡る。それらの無線通信技術の性能は異なっており、無線技術によっては帯域の問題から電話はできるがテレビ電話は利用不可能である、などの状況が発生する。MIRAI では、通信機器の状態を高い精度で取得し、その状態情報を元に接続するネットワークや利用できるサービスを決定する。具体的には、通信機器の位置、搭載インターフェイスとその状態、利用ネットワークの優先順位、通信状態（通信中/非通信中）、搭載アプリケーション、利用可能帯域、利用可能アプリケーションなどの情報を通信端末ごとに取得し、MIRAI が接続ネットワークと利用アプリケーションを制御する。

MIRAI を利用することにより、ユーザはネットワーク状態を意識する必要性が減少する。また、MIRAI によるアプリケーション選択の必要性が低下し、通信に集中できる環境を提供できる。

欠点は、状況の決定をデバイスの状態のみでしているため、ユーザの状態が軽視されて

しまう点である。例えば、ユーザが会議で発表をしている時と、レポートを作成している時という違いがあっても、上に挙げたデバイス状態が同じであればこれらは同一であると検知されてしまう。

### 3.4 EAPEC

コミュニケーションを行うユーザは、時間や場所に関わらず多種多様な通信端末やサービスを利用することになる。以上の背景を元に、メッセージ到達性の向上と、ユーザの複雑な操作を軽減させるためのシステム“EAPEC: Environment-Adaptive Personal Communicaions”[9]を提案している。

アプローチとして、時々刻々と変化するユーザの通信環境をシステムがリアルタイムに把握する。その情報を元に、受信者がメッセージを受信するために最適な通信端末とサービスを自動的に選択し、それに合わせてメッセージを変換し配送する手法を用いている。ユーザの通信環境として、“ネットワーク情報から判断できる端末の位置”，インターネット接続性を持つ端末，端末が利用できるサービス，利用可能な通信メディア（画像，音声など），サービスとメディア変換に関する優先度の情報をシステムが収集する。そして，その情報を元に受信者にとって最適な手段でメッセージを配送する。

欠点として，デバイスの状態情報のみを用いてコミュニケーションサービスを決定している点が挙げられる。デバイスの状態情報のみを用いた場合，ユーザの状況が把握できず状況に応じたサービスの決定ができない。他にも，コミュニケーションサービスに対するユーザの好みは考慮されていない点が問題である。コミュニケーションサービスの好みは人によって異なるため，好みに対応しなければ最適なサービスの決定はできない。

## 第 4 章

# 既存のユーザ状態取得手法の問題点

第 3 章で、ユーザの状況を把握することによりコミュニケーションを支援する研究について述べた。本章では、これらの先行研究が抱える問題点について述べる。

### 4.1 ユーザ状態情報の不足

ユーザ状態情報の不足とは、システムがユーザの状況を十分に把握できない状態である。先行研究において、ユーザ情報の不足に関する問題は二種類ある。

一つ目は、コミュニケーション手段を決定する際、システムがユーザの状態を取得せずに、デバイスの状態情報のみを用いる手法が多い点である。第 3 章で取り上げた研究の中では、MIRAI アーキテクチャがそれに当たる。ここでは、デバイスの状態情報として通信機器の位置、搭載インターフェイスとその状態、利用ネットワークの優先順位、通信状態（通信中/非通信中）、搭載アプリケーション、利用可能帯域などを利用している。このようにデバイスの状態情報だけを利用しては、PC の前で変化するユーザの状況に対応することはできない。

二つ目は、取得しているユーザ状態情報が、ユーザにとって最適なコミュニケーション手段の決定には不十分である点である。第 3 章で取り上げた研究の中では、Sentient Computing Project と TangibleChat を紹介したが、これらの研究がそれにあたる。例えば Sentient Computing Project では、ユーザの屋内でのユーザの位置を正確に知る事ができるが、位置の情報だけではユーザの状況を特定することができず、コミュニケーション手段の決定はできない。また、TangibleChat では、ユーザの状態情報として打鍵情報を利用しているが、これはチャットにおけるユーザの状況を表すものであり、他のコミュニケーションサービスを考慮したユーザ状態情報は取得していない。

## 4.2 ユーザごとに異なるコミュニケーション手段の好みへの対応不足

ユーザが、状況に応じてコミュニケーションサービスを選択できるべきという主張は、第2章で述べた。その際、ユーザは自分の好みに応じてサービスを選択できなければならない。なぜなら、状況が同じであっても、その時に全ての利用者が同じコミュニケーションサービスを利用したいとは限らないからである。しかし第3章で述べた先行研究では、そのために必要なユーザの好みをコミュニケーション手段の決定に反映できない点が問題である。

例えば MIRAI では、無線ネットワークの切替え、及びネットワークに応じたコミュニケーションサービスの切替えをシステムが自動的に行う。この切替えは、デバイス状態を元に MIRAI が独自に行うものであり、ユーザの希望は反映されない。また EAPEC でも、ユーザの好みの必要性は十分に考慮されていない。

## 4.3 導入障壁が高い

関連研究で挙げた先行研究は、コミュニケーション環境の改善のために行われている研究である。これらの研究は、インターネットコミュニケーションの利用者全体を視野に入れて進められている。しかし、導入障壁が高いことにより、インターネットの一般利用者がシステムを利用する事が困難になってしまっている研究が多く存在する。

主に導入障壁となるのは、システムが必要とする機器の導入である。先行研究ではシステムを利用するために、通信機器に新しい機器を取り付けたり、屋内に専用の機器を設置することを前提として進められている研究が多い。例えば、SentientCoumputingProject ではユーザの状態情報を収集する際に、各ユーザが Bat と呼ばれる超音波送信器を持ち歩き、屋内の天井に超音波受信機が敷き詰められていなければならない。システムの導入には、必要機器として各ユーザ分の Bat と多くの超音波受信機が必要である。さらに、機器の設置工事も必要になる可能性がある。また、TangibleChat では、システムがユーザの状況を把握するために、キーボードに加速度センサを組み込み打鍵の振動を検知する。さらに、取得した情報を受信するユーザは、状況を表す打鍵情報を振動として体感するために、振動子を内蔵したクッションを利用しなければならない。

このように、専用の機器を用意し、システムに必要なインフラを整えることはインターネット利用者にとって大きな負担となってしまう。多くの人が利用するためには、導入に必要なコストは抑えられるべきである。

## 第 5 章

# 提案するアプローチ

第 4 章において、ユーザ状態情報の取得が十分に行われていないこと、ユーザの好みが反映されていないこと、導入障壁が高いことを問題点として述べた。本章ではこれらの問題を解決するために、キーボードやマウスの入力イベントを利用しユーザの状況を推測する。さらに、推測した状況に対してコミュニケーションサービスの好みを指定することにより、状況と好みに応じたサービスの決定手法を提案する。

### 5.1 イベント情報とユーザの状態の関連性

キーボードでの打鍵行為とマウスボタンの押下情報を、キータイプ情報と呼ぶ。キータイプ情報と、その入力先アプリケーションの情報のセットをイベント情報と呼ぶ。

本研究では入力イベントとして、キーボードとマウスの入力イベントと、入力先アプリケーションを取得することにより、ユーザの様々な状態の推測を試みる。例えば、入力イベントをカウントすることにより、各アプリケーションに対するキータイプ数を知ることができる。入力イベントを用いてアプリケーションの利用状況を取得することによりユーザの状態を把握できると考える。ユーザの状態をリアルタイムに取得する

### 5.2 入力イベントを取得するための導入障壁

第 4 章において、導入障壁の問題を指摘した。この問題点は、ユーザの状態情報を取得するためにはセンサ類を用いることが多く、その場合必要な機器が増えてしまう、というものだった。今回のキーボードとマウスのイベントを取得し利用するという手法は、通信機器にキーボードとマウスが付属していれば他に機器は必要ない。

第 2 章で紹介した通信機器の中でキーボードとマウスを持ち、本手法を利用できる通信機器はデスクトップ PC、ラップトップ PC、PDA の 3 つである。これらの機器の普及率

**形式**

月日, 時刻, 入力先アプリケーション名, イベントの種類

**例**

```
2006/11/15,15:25:57:887,  
C:\Program Files\Mozilla Firefox\firefox.exe,  
MOUSE_L_CLK
```

図 5.1 記録される入力イベントログの形式

は第2章で述べた通り, 高い。

以上のことより, 本手法を利用するために必要な機器はキーボードとマウスだけであること, またその二つを持つ通信機器は普及していることが分かる。よって, 本手法の導入障壁は低いと言える。

### 5.3 推測手法検証のための予備実験

本節では, 日常的な入力イベントのデータを収集することにより, 推測手法検証のためのデータを求めた。

#### 5.3.1 入力イベントの記録蓄積

入力イベント発生時に月日, 時刻, 入力先アプリケーション名, イベントの種類を取得し記録するアプリケーションを作成した。記録する形式を図 5.1 に示す。

筆者のコンピュータで, 普段の生活における筆者の入力イベントとその入力先アプリケーションをログに記録した。期間は, 2006/09/19 - 2006/11/30 である。総入力イベント数は 391619 だった。記録フォーマットは以下のようにした。

#### 5.3.2 入力イベントログの解析

上述の予備実験において収集したログを解析した。



順位	アプリケーション名	入力イベント数
1	TeraTermPro	130688
2	Firefox	126640
3	PowerPoint	42137
4	MsnMessenger	35581
5	不明	23755
6	devenv . exe	22452
7	LimeChat	10213
8	Explorer . exe	5755
9	メモ帳	5636
10	Word	1982
11	InternetExplorer	1965
12	Excel	1727
13	bluewind	1363
14	xyzyy	840

図 5.2 アプリケーション毎の入力イベント数

#### 日常的に利用するアプリケーションの判別

利用者の状況は、利用しているアプリケーションを利用することで特定できると推測した。そこで、入力イベント数の多い順にアプリケーションを並べ、順位を付けた。その結果、14位と15位には大差があり、上位14位のアプリケーションでの入力イベント数の割合が、全入力イベント数の87.9%であった。この様子を図5.2に示す。アプリケーション名が不明のアプリケーションは、何らかの理由でファイル名が正常に取得できていなかった。

よって、筆者が日常的に利用しているアプリケーションは15個程度であり、その数は全て把握できる範囲内であることが分かった。

## 5.4 コミュニケーション手段決定手法の提案

本節では、提案する手法の概要を説明する。

ユーザ行動	アプリケーション例
Web	Firefox , InternetExplorer , Opera ...
E-mail	Thunderbird , Becky , OutlookExpress ...
Chat	MSN Messenger , LimeChat ...
Music	Winamp , foobar2000 ...
Movie	WindowsMediaPlayer , VLC , WinDVD ...
Game	Starcraft , Warcraft3 , solitaire ...
Edit	Notepad , xyzy , Word , PowerPoint , Illustrator ...
Terminal	Teraterm , Putty ...

表 5.1 アプリケーションの分類

#### 5.4.1 ユーザ状態情報の取得

予備実験の結果，入力イベントの 8 割は上位 10 種類程度のアプリケーションにおいて発生していることが分かった．それら上位のアプリケーションを，ユーザに提供する機能を元に分類した．本手法では，PC における各機能の実行状態を判定し，その情報を利用することでユーザ状態の取得を行う．具体的に，表 5.1 のように機能に分類した．これらの機能を，ユーザの状態を表すための行動としてユーザ行動と呼ぶ．アプリケーション例は筆者の利用しているアプリケーションを用いた．

次に，各ユーザ行動の実行状態を判定する方法を説明する．利用状況はその時によって変化するため，リアルタイムに判定する必要がある．よってユーザ行動毎に，過去数分間の合計イベント入力数を算出し，その値が一定以上だったら実行中とする．

こうして，PC 上でユーザが実行しているユーザ行動を判定し，ユーザの状態情報とする．

#### 5.4.2 ユーザによるサービス優先度の指定

本手法ではユーザの好みを反映するために，ユーザ行動に対応したコミュニケーションサービスの優先度とユーザ行動の優先度の 2 種類の優先度を指定する．

##### ユーザ行動に対応したコミュニケーションサービスの優先順位指定

1 つ目は，受け取りたいコミュニケーションサービスの優先順位指定である．あるユーザ行動が実行されている時には，そのユーザ行動に対応した，利用したいサービスがある

と考える。例えば、レポートを作成中などの状況で Edit 機能を実行中の時と、ゆっくりとゲームをしている Game 機能を実行中の時では、利用したいコミュニケーションサービスには違いが出る可能性がある。そのため、ユーザ行動ごとに最適なコミュニケーションサービスを指定する必要がある。そして、そのユーザ行動が実行されている時には、指定された優先順位に基づいてサービスを決定する。

#### ユーザ行動の優先度指定

2つ目は、実行中のユーザ行動の優先度である。表 5.1のように分類したユーザ行動は、同時に複数利用される状況が考えられる。その場合、それらに対応するサービスの優先順位指定も複数になってしまう。その際ユーザ行動の優先順位を決定するために、重視する行動の優先度を指定する。例えば、Edit が Music より優先順位が高ければ、Edit の優先順位が適応される。

ユーザ行動に対応したコミュニケーションサービスの優先順を指定することにより、ユーザがコンピュータ上で実行中の行動を元に、コミュニケーションサービスを特定できる。さらに、ユーザ行動を複数実行時の優先度を指定することにより、複数実行されている行動の中でも、行動に対するユーザの優先度を反映したコミュニケーションサービスの特定ができる。これら二つの優先度を用いることにより、ユーザの好みを反映したコミュニケーションサービスの特定を行う。

## 第6章

# 設計

本章では、ユーザの状態と好みを反映し、コミュニケーションパスを決定するためのシステムの設計について述べる。

### 6.1 設計概要

設計の概要を、図 6.1 に示す。はじめに、システムは入力イベントを取得する。そして、その情報を元にユーザ行動の実行状態を判定する。次に、ユーザ行動の実行状態とユーザの優先度を表すユーザプリファレンスの情報を用いて、最適なサービスを特定する。ユーザプリファレンスは、予めユーザが入力する。各行程の詳細を、以下に示す。

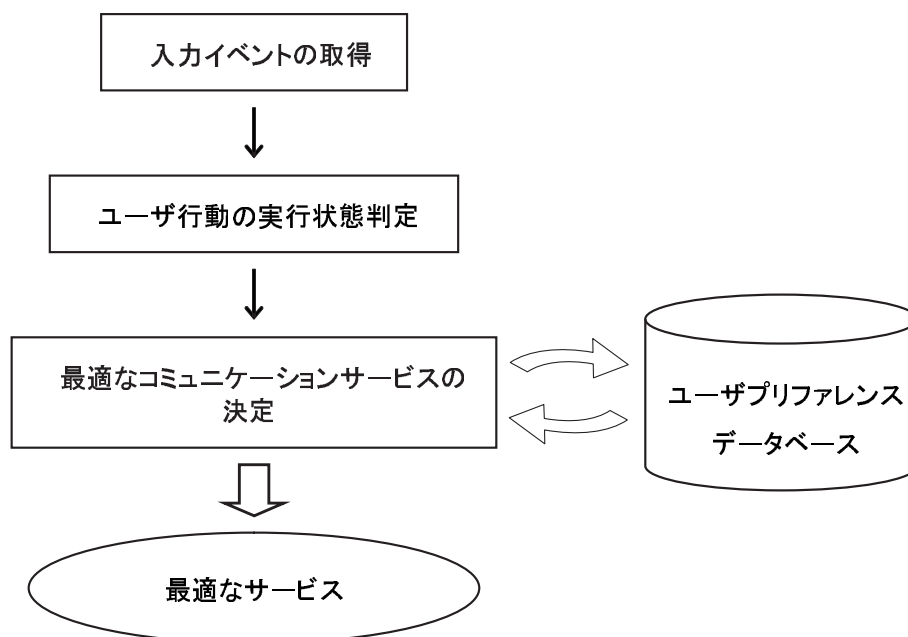


図 6.1 設計概要図

## 6.2 ユーザコンテキストの生成

ユーザの状況のことを，ユーザコンテキストと呼ぶ．本節では，ユーザコンテキストの生成手法について述べる．

### 6.2.1 入力イベントの取得と記録

コンピュータに対する操作情報を利用し利用者の状況を把握するため，ユーザコンテキストの推測には入力イベントを利用する．そのために，コンピュータに対する入力イベント情報を取得する必要がある．入力イベント取得モジュールは，入力されたキータイプ情報を取得し，さらにそのキータイプの入力先アプリケーションを取得するモジュールである．入力イベントとして取得する情報を以下に示す．

#### キーボード関連イベント

アルファベット a-z, A-Z のアルファベットキー

数字 0-9 の数字キー

ファンクション F1-F12 までのファンクションキー

カーソル 矢印キー

特殊キー Ctrl, Alt, Esc, Tab, 半角/全角

その他 記号, NumLock キー, Pause キーなど

#### マウス関連イベント

左クリック 右ボタン

右クリック 左ボタン

真ん中クリック ホイールボタン

入力イベントが発生した場合，その入力先アプリケーションを調査する．次に，アプリケーション名を元に関連するユーザ行動を特定し，入力イベントの発生を記録する．こうして記録をするために，ユーザ行動はアプリケーションリストと入力イベント履歴を持つ．アプリケーションリストは，そのユーザ行動に関連するアプリケーションを指定する．入力イベント履歴には，入力イベントの発生時にイベントの種類と時刻が記録される．その様子を図 6.2 に示す．

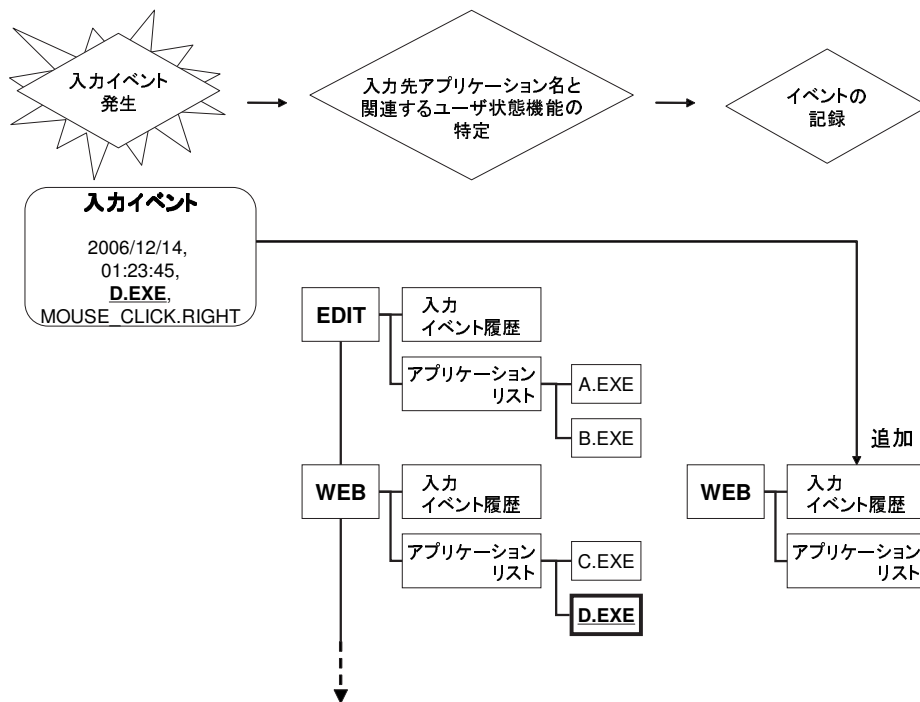


図 6.2 入力イベントの記録

### 6.2.2 各ユーザ状態の実行状態判定

ユーザのコンテキストを生成するために、ユーザ行動の実行状態を判定する。

ユーザ行動の実行状態判定は、過去一定時間の合計入力イベント数と一定の値を比較して行う。そのために、各ユーザ行動は入力イベント履歴、参照時間、入力イベント数の閾値の3つの情報を含む。入力イベント履歴は、過去の入力イベント情報の履歴である。参照時間は、入力イベント数を合計する時に遡る時間を表す。入力イベント数の閾値は、参照時間内の合計入力イベント数と比較し、実行状態を判定するためのものである。

判定手法を説明する。まず、入力イベント履歴の情報から、参照時間内の合計入力イベント数を求める。そして、その合計イベント数を閾値と比較し、閾値を越えていた場合はそのユーザ行動は実行中であるとする。

この手法によって生成されるユーザコンテキストの例を、表 6.1に示す。

### 6.3 ユーザプリファレンスの指定

プリファレンスとは、ユーザの希望を反映させるための優先度情報である。本節では、ユーザの好みを反映させるためのプリファレンス指定方法について述べる。

ユーザ行動	利用状況
Web	
E-mail	×
Chat	
Music	
Movie	×
Game	×
Edit	×
Terminal	×

表 6.1 生成されるユーザコンテキストの例

### 6.3.1 ユーザ状態機能毎のサービスプリファレンス

ユーザ行動毎のサービスプリファレンスとは、ユーザ行動を実行中に、受信したいサービスを表すプリファレンスである。利用できるサービス名を優先順に指定することにより、受信可能なサービスとその優先順位を表現できる。

### 6.3.2 ユーザ状態機能の優先度プリファレンス

ユーザ行動のプリファレンスとは、優先するユーザ行動を指定するプリファレンスである。ユーザ行動は、入力イベントの発生に応じてそれぞれに実行状態が判定される。そのため、複数のユーザ行動が実行中となる場合がある。その際、ユーザ行動に対応するサービスプリファレンスも複数になってしまうため、用いるサービスプリファレンスをその中から選択する必要がある。そのために、優先するユーザ行動を指定する。

## 6.4 最適なコミュニケーションサービスの決定

ユーザコンテキストの取得方法と、プリファレンスの指定方法を前述した。本節では、それらの情報を元に最適なサービスの決定方法を示す。これまで説明した手法を実現し、状況と好みを考慮したコミュニケーション手段を決定するために必要となるデータ構造を、図 6.3にまとめる。

本システムは図 6.3のようにデータを持つ。ユーザ行動をつなぐ順番で、ユーザ状態機能の優先度プリファレンスを表現する。

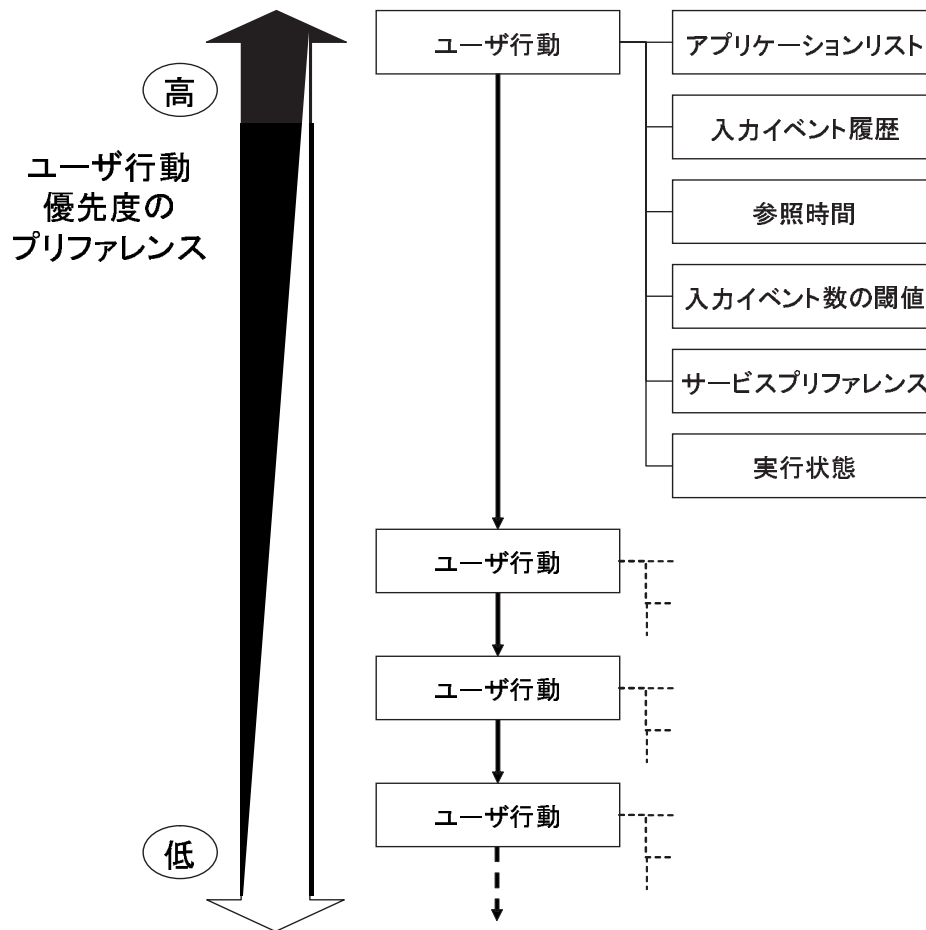


図 6.3 データの概要

これらのデータを用い、前述した手法を用いることにより、ユーザ状態機能ごとに実行状態を判定し、サービスプリファレンスを取得する。これらのデータの中で、サービス決定に利用するデータは利用状況とサービスプリファレンス、ユーザ行動の優先度プリファレンスである。

具体的なサービス決定手順を、図 6.4を用いて説明する。

1. 最初のユーザ行動である Edit に移動
2. Edit の実行状態を確認
3. 実行中ではないため次のユーザ行動へ移動
4. Chat の実行状態を確認
5. 実行中のためサービスプリファレンスを確認
6. IM, Mail, インターネット電話が利用可能。優先順位は IM > Mail > インターネット電話となる。



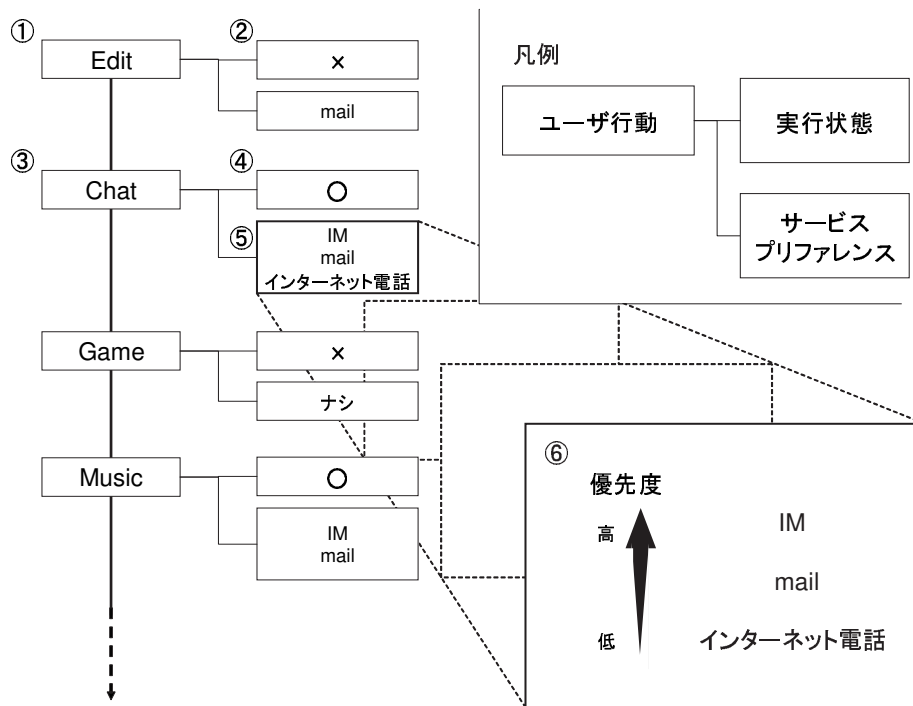


図 6.4 サービスの決定手順

## 第 7 章

# 実装

ユーザの状態情報を取得するために、キーとマウスのイベントを取得するためのソフトウェアを実装した。本章では、その詳細について述べる。

### 7.1 実装の動作環境

本実装は、コンピュータを利用し生活しているユーザの状態情報を取得する。そのため、インターネット利用者の多くが利用している、Windows プラットフォーム [2] 上で動作するソフトウェアを実装した。本実装は、WindowsXP 上で VisualStudio .Net[4] を用いて、C++ 言語で開発した。

### 7.2 実装概要

実装の概要を、図 7.1 に示す。

キーボードとマウスから入力があると、入力イベント検知モジュールが入力イベントを検知する。入力イベント検知モジュールは、その入力イベントが何のアプリケーションに対して行われたのかを調べるためにアプリケーション調査モジュールを利用する。アプリケーション調査モジュールは、呼び出されると現在ユーザが利用しているアプリケーション名を返す。入力イベント調査モジュールは、利用アプリケーションが判明すると、入力イベントの発生時刻、種類、利用アプリケーション名を履歴管理・状態保存モジュールに保存する。これらの動作は、入力イベントが発生する度に行われる。

履歴管理・状態保存モジュールは、各ユーザ行動の、入力イベントの履歴と実行状態を保存するためのモジュールである。

ユーザ行動監視モジュールは、履歴管理・状態保存モジュール内の履歴情報を参照し、実行状態を判定する。そして、履歴管理・状態保存モジュール内の状態情報を更新する。

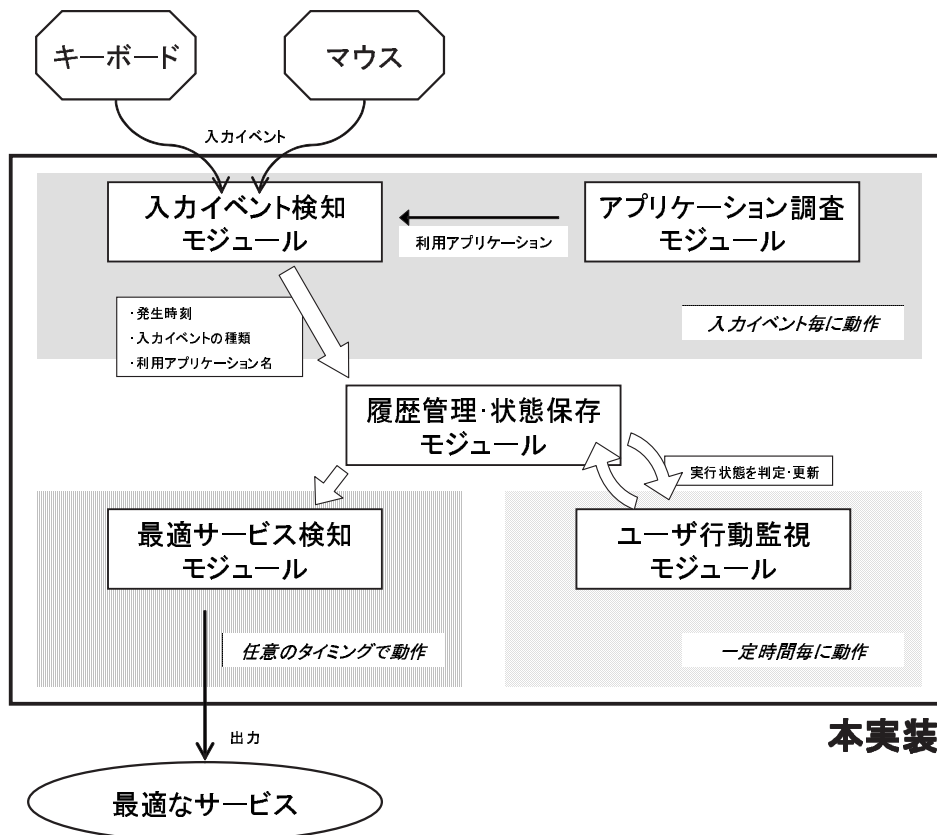


図 7.1 実装概要図

このユーザ行動監視モジュールが一定時間毎に呼び出されることにより、実行状態の情報は常に最新の情報に更新される。

最適サービス決定モジュールは、履歴管理・状態保存モジュール内の情報から、最適なサービスを決定するためのモジュールである。このモジュールは任意のタイミングで呼び出せるため、自由なタイミングで最適なサービスを出力することができる。

各モジュールの詳細を、以下の節に示す。

### 7.3 入力イベント検知モジュールの実装

入力イベント調査モジュールは、キーボードとマウスの入力を検知するためのモジュールである。本節では、その実装について述べる。

#### 7.3.1 グローバルフックの利用

本システムでは、入力イベントの検知をするためにグローバルフックという仕組みを用いた。Windows では、キー入力が発生すると内部的にメッセージが発行される。アプリ

```
SetWindowsHookEx (WH_KEYBOARD , ...  
SetWindowsHookEx (WH_MOUSE , ...
```

図 7.2 グローバルフックの設定に利用する関数

ケーションの開発者は、特定のメッセージを指定しそれを検知するためのフックを仕掛けることで、指定したメッセージが発行された場合にそのメッセージを受け取ることができる。

Windows におけるフックの設定方法には、ローカルフックとグローバルフックの二種類がある。ローカルフックではフックを仕掛けたアプリケーション内でのメッセージのみを受け取る。グローバルフックでは、アプリケーションに関係なく、Windows 内で発行された全てのメッセージを受け取ることができる。今回は、どのアプリケーションに対する入力であってもその全てを検知しなければならないため、グローバルフックを利用した。

### 7.3.2 DLL の利用

DLL(Dynamic Link Library) とは、複数のアプリケーションソフトが共通して利用するために汎用性の高いプログラムを部品化したライブラリである。今回は、入力イベント検知モジュールを DLL として実装した。Windows の制約として、グローバルフックを行うためにはフックを行う部分を DLL 内に実装する必要があったため、DLL を用いた。こうしてグローバルフックの機能を用いた DLL を作成し、その DLL を利用することで、システムはコンピュータ上で発生する全ての入力イベントを検知することができる。

### 7.3.3 入力イベントの検知

フックの組み込みには、SetWindowsHookEx() 関数を利用した。この関数は、引数に監視するメッセージを指定することでグローバルフックを利用することができる。図 7.2 に利用した関数を示す。

## 7.4 アプリケーション調査モジュール

アプリケーション調査モジュールは、入力イベントが入力されたアプリケーションを調査するためのモジュールである。本説では、その実装について述べる。

入力先アプリケーションの調査は、ユーザが操作しているウィンドウを特定し、そのウィンドウを制御しているアプリケーションを調査する方法を採用した。本実装ではアプリ

```
hWnd = GetForegroundWindow();
GetWindowThreadProcessId( hWnd , &ActiveProcessId);
hProcess = OpenProcess(PROCESS_QUERY_INFORMATION |
                      PROCESS_VM_READ , FALSE , ActiveProcessId);
EnumProcessModules(hProcess , ModuleHandles , 1024 , &RetSize);
for()
    GetModuleFileNameEx(hProcess , ModuleHandles[0] ,
                      FileName , BUFSIZE);
```

図 7.3 入力イベントが発生したアプリケーションの特定

ケーション名を特定するため、ウインドウを制御しているプロセスにアクセスし、プロセスがどの実行ファイルを用いているかを調べた。

まず、ユーザが操作しているウインドウの特定に `GetForegroundWindow()` 関数を利用した。この関数は、戻り値にユーザが操作しているウインドウの識別子を返す。次に、`GetWindowThreadProcessId()` 関数を利用した。この関数は、ウインドウの識別子を第一引数に与えることで、ウインドウを制御しているプロセスの識別子を第二引数に格納する。次に、`OpenProcess()` 関数を利用した。この関数は、プロセスにアクセスするためのハンドルを戻り値として返す。次に、`EnumProcessModules()` 関数を利用した。この関数は、プロセスのハンドルを引数に指定することで、プロセスにアクセスし依存しているファイルのリストを第二引数に格納する。最後に、`GetModuleFileNameEx()` 関数を利用した。この関数は、引数に依存ファイルのリストを渡すことで、実際のファイル名を第二引数に格納する。

詳細を、図 7.3 に示す。

## 7.5 情報蓄積モジュール

入力イベント蓄積モジュールとは、入力イベントや各ユーザ行動の実行状態などの情報を蓄積し、後述するユーザ行動監視モジュール、最適サービス決定モジュールに提供するためのモジュールである。

蓄積するためのデータ構造を、図 7.4 に示す。この `FunctionInfo` 構造体には、ユーザ行動の情報を格納される。入力イベントが発生する度に、入力が行われたアプリケーション名を元にユーザ行動を特定し、この中の `EventHistoryList` のカウントが加算される。

```
struct FunctionInfo{
    char *FuncName; //ユーザ行動の名前
    int Doing; //実行状態
    struct App *AppList; //関連するアプリケーションのリスト
    int TermSecond; //実行状態の判定に利用する時間幅
    int Threshold; //実行状態の判定に利用する入力イベント数
    struct EvntHistory *EventHistoryList; //入力イベントの履歴
    int EventHistoryNum; //入力イベントの保存秒数
    struct Service *ServiceList; //このユーザ行動が実行中に利用したいコ
    ミュニケーションサービスのリスト
    struct FunctionInfo *next;
    struct FunctionInfo *back;
}
```

図 7.4 ユーザ行動のデータ構造

図 7.4の中に含まれている構造体の詳細を、図 7.5に示す。

EventHistory 構造体は、発生した入力イベントの数を一秒毎に分けて保存するための構造体である。App 構造体は、ユーザ行動に関連するアプリケーション名を保存する構造体である。Service 構造体は、ユーザが指定する、そのユーザ行動が実行中に利用したいサービス名を保存する構造体である。

## 7.6 ユーザ行動監視モジュール

ユーザ行動監視モジュールは、ユーザ行動の実行状態を一定時間ごとに判定するモジュールである。本実装ではタイマーをセットし、一秒ごとに実行状態の判定を行った。

実行中の判定は、一定時間内に、設定した値以上の入力イベント数があるかどうかを算出することで行う。FunctionInfo 構造体内の EventHistoryList が入力イベントの履歴を表す。TermSecond は一定時間にあたる、時間の幅を表す。Threshold は、一定時間内に実行中と判断する入力イベント数の閾値を表す。

判定部分の実装を、図 7.6に示す。

```
struct EventHistory{
    time_t tm_t;
    int EventNum;
    struct EventHistory *next;
    struct EventHistory *back;
};

struct App {
    char *Appname;
    struct App *next;
    struct App *back;
};

struct Service {
    char *ServiceName;
    struct Service *next;
    struct Service *back;
};
```

図 7.5 利用する構造体のデータ構造

## 7.7 最適サービス決定モジュール

最適サービス決定モジュールは、入力イベント蓄積モジュールに保存されている情報から、ユーザに最適なサービスを決定するためのモジュールである。FunctionInfo 構造体のリストは、ユーザが優先的に扱いたいユーザ行動が順番に繋がっている。そのため、FunctionInfo 構造体をリスト順に調べ、その中で実行中のユーザ行動を調査する。そして、一番最初に見付かった実行中のユーザ行動に設定されている、ユーザの利用したいコミュニケーションサービスを、最適なサービスとする。

```
time_t tm_t;
int Totalcount;

struct FunctionInfo *fi_p;
struct EventHistory *eh_p;

tm_t = time(NULL);

for ( fi_p = fi_root; fi_p!=NULL; fi_p=fi_p->next){
    if (fi_p->EventHistoryList == NULL)
        continue;
    Totalcount=0;
    for (eh_p = fi_p->EventHistoryList; eh_p->next!=NULL;
        eh_p=eh_p->next){
        }
    while (fi_p->TermSecond > (int)difftime(tm_t , eh_p->tm_t)){
        Totalcount = Totalcount + eh_p->EventNum;
        eh_p = eh_p->back;
        if (eh_p==NULL)
            break;
    }
    if (Totalcount >= fi_p->Threshold){
        fi_p->Doing = DOING;
    }else{
        fi_p->Doing = NOT_DOING;
    }
}
```

図 7.6 ユーザ行動監視モジュールの動作



## 第 8 章

# 評価

本研究で実装したシステムを評価し，提案手法の検証を行う．

### 8.1 検証実験

システムが状況と好みに応じたコミュニケーション手段を推測できているかを確認するために，被験者を募り実際にシステムを利用してもらった．各被験者にシナリオと時間を提示し，日常的な動作に従って実行してもらった．

被験者は三人で，それぞれ個別に実験した．被験者にはまず，本研究の概要とシステムの説明をした．実験におけるキータイプ情報のログはすべて保存するため，パスワードやカード番号など，個人情報への入力には留意するよう被験者に伝えた．そして，システムの設定を行ってもらった．システムに設定できる項目とは以下の通りである．

- ユーザ行動に関連するアプリケーション
- ユーザ行動ごとのサービスプリファレンス
- ユーザ行動のプリファレンス
- ユーザ行動が実行状態を推測するための参照時間と入力イベント数の閾値

システムの設定にかかる時間と設定内容は，制限しなかった．参照時間と入力イベント数の閾値は，筆者の考える最適な値を初期値として与えた．設定ファイルに記述する要素を図 8.1 に示す．

次に，被験者に異なるシナリオを提示した．そして，そのシナリオを実行する際に利用可能なコミュニケーションサービスとその優先順位を，シナリオを実行する前にあらかじめ記録してもらった．指定されたコミュニケーションサービスの中でもっとも優先順位の高いものを，シナリオを実行中のユーザにとって最適なサービスとする．

次に，制限時間を提示し，シナリオを実行してもらった．その際，被験者が日常的に行っ

## - 形式

ユーザ行動:参照時間:入力イベントの閾値:関連アプリケーション:利用可能サービス

上記の形式をユーザ行動の優先順位順に上から列記する。

## 例

Edit:120:70:TeraPad . exe,WINWORD . EXE,EXCEL . EXE,POWERPNT . EXE:Mail

Terminal:120:30:ttermpro . exe,putty . exe:Mail

Chat:300:50:MsnMsgr . Exe,IPMSG . exe:IM

...

図 8.1 設定ファイルに記述する要素

シナリオ番号	被験者	的中率	的中に要した時間	最適サービス
1: レポート作成	A	42.0%	512 秒	Mail
2: 新たに購入する靴の検討	A	66.2%	135 秒	IM
3: メールの読み書き	A	94.0%	54 秒	IM
4: 一番かわいいと思う帽子の検討	B	92.5%	55 秒	IM
5: レポート作成	B	56.8%	48 秒	Mail
6: ゲームをながら IM	B	0.0%	-	IM
7: メールの読み書き	C	97.4%	23 秒	Mail

表 8.1 実験結果

ているように実行してもらった。例えば音楽を聞く、動画を再生する、などの行動に制限はかけなかった。

制限時間終了後、システムを終了し、実験を終了した。

## 8.2 実験結果

あらかじめ記録してもらった最適なサービスと、システムが推測した最適なサービスを比較し、システムの的中率を算出した。被験者の三人を、それぞれ A, B, C とする。各被験者の実験結果を、表 8.1 に示す。項目の的中率とは、推測が的中していた時間の割合を、全体を 100 的中に要した時間とは、システムが最適なコミュニケーション手段を的中する

```
Edit:120:70:TeraPad . exe,WINWORD . EXE,EXCEL . EXE,POWERPNT . EXE:Mail
Terminal:120:30:ttermpro . exe,putty . exe:Mail
Chat:300:50:MsnMsgr . Exe,IPMSG . exe:IM
Mail:600:20:B2 . exe:IM,Mail
Web:60:10:FIREFOX . EXE,IEXPLORE . EXE:IM,Mail
Movie:180:10:wmplayer . exe:IM,Mail
Game:120:20:spider . exe:IM,Mail
Music:180:10:winamp . exe:IM,Mail
```

図 8.2 被験者 A の設定内容

までに要した時間である。最適サービスとは、被験者が、シナリオを実行中に利用したいと指定したサービスである。シナリオ 6 では、システムがユーザの希望するサービスを的中できなかったため、的中に要した時間は“-”と示した。

被験者とシナリオに分けて、以下に実験結果の詳細を示す。

### 8.2.1 被験者 A

被験者 A のシステム設定を、図 8.2 に示す。

#### シナリオ 1: レポート作成

シナリオ 1 実行時にユーザが指定した最適なサービス: Mail

レポート作成のシナリオを提示した。被験者 A は、バスケットボールについて考察するレポートを作成した。レポートの作成は、テキストエディタを用いて行っていた。被験者 A は、本を資料としてレポートを作成した。

的中率: 42.0%

的中までに要した時間: 512 秒

#### シナリオ 2: 新たに購入する靴の検討

シナリオ 2 実行時にユーザが指定した最適なサービス: IM

新たに購入する靴を探し、検討するシナリオを提示した。被験者 A は、終始 Web を利用して靴を探した。

的中率: 66.2%

```
Edit:120:70:xyzzzy . exe,notepad . exe,WINWORD . EXE,EXCEL .  
EXE,POWERPNT . EXE  
:Mail  
Mail:600:20:thunderbird . exe:Mail,IM  
Chat:300:50:MsnMsgr . Exe,LimeChat . exe:IM,Mail  
Terminal:120:30:ttermpro . exe,putty . exe:Mail  
Web:60:10:firefox . exe,IEXPLORE . EXE,iexplore . exe:IM,Mail,VoIP  
Game:120:20:sol . exe,winmine . exe:Mail  
Movie:180:10:vlc . exe,wmpplayer . exe:Mail  
Music:180:10:winamp . exe:Mail,IM,VoIP
```

図 8.3 被験者 B の設定内容

的中までに要した時間: 135 秒

### シナリオ 3: メールの読み書き

シナリオ 3 実行時にユーザが指定した最適なサービス: IM

E-mail の送受信を行うシナリオを提示した。被験者 A の元には海外の友人からメールが届いており、その返事を作成していた。被験者 A が返信するメールでは日本のニュースを紹介しており、Web でその内容を確認していた。また、被験者 A は英単語を調べるために和英辞典の Web ページを活用していた。

的中率: 94.0%

的中までに要した時間: 54 秒

### 8.2.2 被験者 B

被験者 B のシステム設定を、図 8.3 に示す。

### シナリオ 4: 一番かわいいと思う帽子の検討

シナリオ 4 実行時にユーザが指定した最適なサービス: IM

被験者 B が一番かわいいと思う帽子を検討し、見付け出すシナリオを提示した。被験者 B は、Web を使って理想の帽子を探した。

的中率: 92.5%

```
Game:120:20:system4 . 0 . exe,winmine . exe:Mail
Edit:120:70:xyzyy . exe,WINWORD . EXCEL . EXE,POWERPNT . EXE,notepad .
exe
:Mail,IM,VoIP
Terminal:120:30:ttermpro . exe,putty . exe:Mail
Web:60:10:IEXPLORE . EXE:IM,Mail,VoIP
Movie:180:10:mplayerc . exe:Mail
Chat:300:50:MsnMsgr . Exe:IM,Mai,VoIP
Mail:600:20:B2 . exe:Mail,IM,VoIP
Music:180:10:mplayerc . exe:Mail,IM
```

図 8.4 被験者 C の設定内容

的中までに要した時間: 55 秒

シナリオ 5: “通信事業法について説明せよ” というテーマのレポート作成

シナリオ 5 実行時にユーザが指定した最適なサービス: Mail

被験者 B は、Web を利用して情報収集しながらレポートを作成した。レポートの作成は、テキストエディタで行っていた。

的中率: 56.8%

的中までに要した時間: 48 秒

シナリオ 6: ゲームをしながら友人とインスタントメッセージング

シナリオ 6 実行時にユーザが指定した最適なサービス: IM

ゲームをしながら、IM を用いて友人と会話を行うシナリオを提示した。被験者 B は、ゲームと IM を交互に切替えていた。この実験では、推測結果は全く的中しなかった。

的中率: 0.0%

的中までに要した時間: -

### 8.2.3 被験者 C

被験者 C のシステム設定を、図 8.4 に示す。

### シナリオ 7: メールの読み書き

シナリオ 7 実行時にユーザが指定した最適なサービス: Mail

メールを受信し、届いていたメールに返信するシナリオを提示した。被験者 C には、友人からメールが届いており、web を利用し様々な話題について調べながら返信していた。

的中率: 97.4%

的中までに要した時間: 23 秒

## 8.3 検証実験の考察

### 8.3.1 入力イベント数が多いほどの的中率が高い

本システムは、継続的にコンピュータを利用し入力イベント数が多いシナリオほど、推測結果の的中率が高い。シナリオ 3 とシナリオ 7 は、メールを読み書きするシナリオだった。これらのシナリオでは、被験者はまずメールソフトを使いメールを受信し、メールを読んだ。次に、英単語を調べたり、返信するメールの内容を調べるために web を利用しながらメールを作成し、送信した。この間、被験者はメールと web の機能を交互に利用し続けており、継続的に入力イベントが発生していた。このシナリオでの的中率はシナリオ 3 が 94.0%、シナリオ 7 が 97.4% であり、高いものとなった。またシナリオ 2 とシナリオ 4 では、被験者が好みの靴と帽子を調査した。これらのシナリオでは、被験者は終始 web を用いて自分の好みのものを探していた。このシナリオでの的中率はシナリオ 2 が 66.2%、シナリオ 4 が 92.5% であり、これも比較的高いものとなった。

逆に、一定時間コンピュータを利用しない状況が含まれるシナリオでは、的中率が低くなった。シナリオ 1 では、被験者は体育のレポートを作成した。このシナリオで被験者は、資料として本を参照しながらレポートを作成した。このように、コンピュータが操作されず、入力イベントが発生しない状況が断続的に訪れることにより、本システムはユーザの状況が正確に把握できなかったのだと思われる。このシナリオでの的中率は、42% であった。

### 8.3.2 設定ファイルの記述が困難

シナリオ 6 の実験結果により、設定ファイルの記述を直感的に行うことが難しい場合があることが分かった。シナリオ 6 で、被験者はゲームをしながら IM を使って友達と会話をした。このシナリオでの本システムの的中率は 0% であった。

的中率が 0 被験者がシナリオに対して指定した最適なサービスが食い違っていたためであった。被験者は、このシナリオ中に一番利用したいサービスとして IM を指定した。

被験者は Chat のユーザ行動に対応するものとして IM を選択しており、さらに Chat は Game よりも優先度が高かった。また、Game のユーザ行動に対応する最適なコミュニケーションサービスとして Mail を指定していた。各ユーザ行動の実行状況が正確に判定出来ていれば、Chat が実行中であり、最適なサービスは IM であると推測できるはずだった。しかしこの被験者の場合、Chat の実行状況を判断するための適切な閾値の設定がされておらず、システムでは Chat が実行中と判定されていなかった。この時、Game の実行状態は正確に判定できており、Game は実行中であると判定されていた。その結果、Game のユーザ行動に対応したコミュニケーションサービスが優先され、被験者の希望とは違うコミュニケーションサービスが推測されてしまった。

このように、本システムの設定に関して、閾値の設定が難しいという問題点が明らかになった。また、この閾値の設定によって、推測結果に大きな影響が出ることが分かった。

## 第9章

# 結論

本章では、本論文をまとめ、今後の課題について述べる。

### 9.1 まとめ

本研究では、コミュニケーション環境が多様化している現状と、その問題点について考察した。そして、その問題点を解決するための研究を、関連研究として述べた。関連研究の問題点として、ユーザ状態情報の取得が困難である点を挙げた。関連研究では、機器の状態情報を取得し、コミュニケーションサービスを決定する手法が多く用いられていた。ユーザの状態情報を取得する手法もあったが、それらは赤外線センサや超音波センサなどが必要であり、導入障壁が高かった。

これらの問題点を踏まえた上で、本研究はそれらを解決するために入力イベントとその入力先アプリケーションの情報をを用いてユーザ状態情報を取得した。利用者によって異なるニーズに対応するために、ユーザ状態における受信したいサービスを指定するための各ユーザが設定するプリファレンスを提案した。これらのユーザ状態情報と、プリファレンスを用いて利用者にとって最適なサービスを決定するシステムを設計、実装した。

システムの評価として検証実験を行った。検証実験では、被験者に提示したシナリオに沿ってシステムを利用してもらい、本システムの的中率を測定した。結果として、コンピュータを継続的に利用するシナリオでは97.4%という高い的中率を確認できた。しかし、ある一定時間以上コンピュータを利用しない状況が含まれるシナリオでは、42.0%という的中率であった。

この結果から、本研究は、利用者のニーズを考慮したメッセージの送受信を実現するための、入力イベントを用いた推測手法の有用性と問題点を明らかにすることができた。



## 9.2 今後の課題

### 9.2.1 入力イベント以外の情報の導入

検証実験を通して、一定時間以上コンピュータを利用しない状況が含まれるシナリオでは的中率が低くなることが分かった。本手法では、生成されるユーザ状態情報の精度は入力イベント数に依存している。そのため、入力イベントが一定時間発生しない場合は的中率が低くなってしまう。例えば、検証実験のシナリオ1では、被験者が本を資料としてレポートを作成したため、本を読んでいる間は入力イベントが発生せず、ユーザ状態情報の精度が落ちた。そのため、ユーザ状態情報の生成には、入力イベント情報以外の有用な情報の取得が望まれる。

### 9.2.2 システム設定の簡易化

検証実験を通して、システムに対して行う設定の難しさも明らかになった。シナリオ6では、ゲームをしながらIMを利用し友人と会話をした。被験者は事前の設定のミスにより、実行中であり優先されるはずのChatのユーザ行動が実行していないと判定され、Gameのユーザ行動に対応するコミュニケーションが優先的に選択されてしまった。その際に被験者の望むサービスはIMであったが、システムの推測結果はMailであり、間違った推測結果を出してしまった。このように、閾値の設定の難しさによって設定にミスが生じ、的中率は0.0%になってしまった。この問題の解決方法はまだ見付けていないが、設定の方法を見直す必要性を感じた。

### 9.2.3 メッセージの内容を考慮できない

本システムでは、メッセージの内容を考慮したコミュニケーションサービスの決定ができない。ユーザにとってメッセージの内容は、コミュニケーションサービスの決定に重要な役割を果たすはずである。例えば、メッセージが緊急の場合とそうでない場合では、ユーザが利用したいコミュニケーションサービスは違って来るはずである。他にも、メッセージの差出人が上司か妻かの違いによって、利用するサービスを変更したいユーザがいるかもしれない。よって、メッセージの内容を考慮した上での、コミュニケーションサービスの決定が必要になる。

# 謝辞

本研究を進めるにあたり，ご指導をいただきました慶應義塾大学 環境情報学部教授 村井純博士，並びに同学部教授 徳田英幸博士，同学部教授 中村修博士，同学部助教授 楠本博之博士に感謝致します。

絶えず御指導と御助言を頂きました，慶應義塾大学大学院政策・メディア研究科特別研究講師 南政樹氏，同研究科博士課程 小原泰弘氏，白畑真氏，同研究科修士課程 小椋康平氏，水谷正慶氏に深く感謝致します。

本論文の作成にあたり協力して下さった，慶應義塾大学村井研究室 SING グループの佐藤龍氏，尾崎隆亮氏，波多野敏明氏に感謝致します。

3年間の長きに渡って苦楽を共にした，金井瑛氏，空閑洋平氏を始めとする同期諸君に心より感謝の意を表します。

最後に，僕の大学生活を最大限に応援してくれた，父 睦夫，母 和子，兄 保之 和明に精一杯の感謝の意を表します。

2007年1月24日

奥村 祐介

## 参考文献

- [1] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, A. Hopper, J. Weatherall, A. Jones, G. Girling, et al. Implementing a Sentient Computing System. *IEEE Computer Magazine*, Vol. 34, No. 8, pp. 50–56, August 2001.
- [2] Microsoft Corporation. Microsoft Windows ファミリ ホーム.
- [3] Microsoft Corporation. MSN Messenger.
- [4] Microsoft Corporation. Visual Studio ホームページ.
- [5] NTT コミュニケーションズ株式会社. 無線 LAN サービス ホットスポット.
- [6] USB Implementers Forum, Inc. USB Spec., Rev. 1.1.
- [7] USB Implementers Forum, Inc. USB Spec., Rev. 2.0.
- [8] Yahoo! Japan. Yahoo!メッセンジャー.
- [9] 上岡英史, 山田茂樹. ユービキタスコンピューティング・ネットワーク実現に向けた環境適応型パーソナル通信, May 2002.
- [10] 株式会社ウィルコム. WILLCOM.
- [11] 株式会社エヌ・ティ・ティ・ドコモ. PHS — サービス・機能 — NTT ドコモ.
- [12] 株式会社エヌ・ティ・ティ・ドコモ. 無線 LAN サービス ホットスポット.
- [13] 井上真杉, マハムドカレド, 村上誉, 長谷川幹雄, 森川博之. シームレス通信環境のためのコンテキスト情報を利用したネットワーク・アプリケーション制御機構. 情報処理学会論文誌, Vol. 46, No. 9, pp. 2236–2249, September 2005.
- [14] 総務省. 平成 17 年「通信利用動向調査」の結果, May 2006.
- [15] 山田裕子, 平野貴幸, 西本一志. TangibleChat, May 2003.