第 16 部

WIDE/PhoneShell

第 1 章

はじめに

PhoneShell ワーキンググループでは、昨年度に引続き WIDE/PhoneShell [208] [209] [210] や WIDE/PCS [211] [212], を活用した計算機システムの利用についてさまざまな角度から検討や実験を行った。

本報告書では、本年度の成果の中から以下の項目について述べる。

1. Sophia/PhoneShell
2. CHANT/PhoneShell
3. ページャ制御方法の改善
4. PDA の活用
5. WIDE/PCS 支援機構
6. 認証システムへの応用

このうち 1,2 は、WIDE/PhoneShell とは別の発想に基づいて設計された新しい PhoneShell についての報告である。また 3 は DTMF 制御系の、4,5 は現行の WIDE/PCS 環境を改善する方法について述べている。さらに、6 は WIDE/PCS を応用システムについて検討したものである。

なお、本報告書は、稲田龍、大野浩之、川副博、草刈千晶、新美誠、矢吹道郎 (順不同) が執筆し、大野が全体の調整を行った。

第 2 章

新しい PhoneShell

WIDE/PhoneShell は、計算機に直接アクセスできない環境にいるシステム管理者が、DTMF や音声や FAX を使って必要最小限の管理作業を実施することを目的に開発されたシステムである。言うまでもなく PhoneShell の Phone は電話 (telephone) に、Shell は UNIX のコマンドインタプリタ (shell) に由来し、本システムの特徴をあらわしている。

WIDE/PhoneShell の開発が始まったのは 1991 年で、以来 WIDE/PhoneShell は一種類だけであったが、今年度はあらたに 2 つの PhoneShell の開発がされた。両者とも、電話を介して計算機アクセスを行うという点で、従来の WIDE/PhoneShell と共通しているが、その開発動機、利用目的などは、それぞれ異なっている。このため、PhoneShell という名前は共通にして同じ発想に基づいていることを主張しつつ、接頭辞を “WIDE” 以外の文字列にして独自性を強調することにした。本章では、これら 2 つの新しい PhoneShell である Sophia/PhoneShell と CHANT/PhoneShell について述べる。

2.1 Sophia/PhoneShell

2.1.1 Sophia/PhoneShell の概要

Sophia/PhoneShell [213] は、モデムや端末などの機器を使用せずに、プッシュホンから計算機へのログインを実現するものである。入力手段としてはプッシュトーン (DTMF) を用い、応答は音声合成装置によるテキスト読み上げにより行なわれる。

当初はメイル読み上げシステムとして構築されたが、現在は汎用的な利用を可能とするため、その名の示す通りシェルとして実装され、メイル読み上げはシェル上の 1 つのアプリケーションとして実行されるようになっている。したがって、ユーザは Sophia/PhoneShell のアプリケーションを自由に作成し実行することが可能である。

過去に WIDE/PhoneShell によってプッシュトーンを用いた ping による非常時のネットワーク・リーチャビリティの確認などが実現されているが、Sophia/PhoneShell を用いるならば、より要求に即した実行が可能となる。さらに非常時には WIDE/PCS と共に利用することにより、現実的な対応が可能なネットワーク管理手段となる。

2.1.2 Sophia/PhoneShell の設計

Sophia/PhoneShell の設計方針として、汎用性とインターネットでの利用を目標とした。

- 汎用性

  汎用的なシステムとするために、ユーザが実行する機能は外部コマンドとして実現している。つまり、システムは個々に用意されたコマンド群を利用するための一種のシェルと言うことができる。通常のシェルが、キーボードとディスプレイを入出力デバイスとするのに対し、ここでのシェルは電話機の 12 のボタンと受話器からの音声を入出力とするシェルと考えることができる。

  外部コマンドはシェルスクリプトやバイナリプログラムである。しかし、従来計算機に備えられているコマンドの多くは画面に出力されることを前提としているため、音声出力には適していない。そこで、これらのコマンドの実行結果をフィルタリングする必要がある。しかし、出力のフィルタリングは各々のコマンドに依存している。多岐に渡るフィルタリングの要求をシステムが供給することは、システムを肥大化させるため望ましくない。したがって、各コマンドのフィルタリングはシェルの外部の問題としてシェルの機能から切り離し、各ユーザコマンドで個別に実現するものとする。

- インターネットの利用

  Sophia/PhoneShell は用意された機能を単に使うのでなく、個人の利用環境に合わせた個別の利用を目的としている。個人の利用環境が存在するのは通常利用している計算機（すなわちホームマシン）である。個別に利用すべき計算機がすべて電話からのアクセスシステムを持つのは現実的でない。アクセスシステムは単なるエントリと考え、インターネットを介して必要な計算機を利用できなくてはならない。この場合、不正な利用を防ぐため、セキュリティについて注意する必要がある。

2.1.3　必要となる機能

Sophia/PhoneShell の実装において必要となる機能は、ユーザインターフェイス、音声出力、ネットワーク機能などに分類することができる。

- ユーザインターフェイス

  従来の電話を用いた情報提供サービスや電話予約システムなどではユーザインターフェイスは固定的であった。しかし、汎用的なシステムを実現するためにはユーザインターフェイスは各ユーザの要求にしたがって変化しなければならない。つまり各ユーザが自由にユーザインターフェイスを定義できる必要がある。

  ユーザインターフェースとして考慮すべき項目は以下の通りである。

  – 入力モード

電子メイルや文章の読みあげ、確認作業などの単純な作業であれば機能を電話の数字キーに割り当てることによって、十分な操作環境が得られる。しかし場合により、2 ストローク入力を用いたアルファベット入力あるいは 4 ストローク入力を用いた漢字入力なども必要となるだろう。さらには、DTMF ではなく音声の入力も考慮する必要がある。音声の入力は、入力音声をコマンドとして解釈する用途や留守番電話のように入力音声を録音する用途が考えられる。これらの入力方法を切り換える入力モード機構が必要である。

– ヘルプ機能

システムの最も現実的な利用はキーによる機能の選択である。しかし機能と 12 のキーの組み合わせを結び付けるものは記憶だけとなってしまう。緊急時にのみ利用するようなユーザは必ずしも操作環境をすべて記憶しているとは限らない。このため、電話予約システムに見られるような、ユーザの利用を助けるガイダンスあるいはヘルプ機能が重要となるであろう。特に、ヘルプ機能では、目的とする機能の操作をいかに効率良く検索できるかが重要な要素となる。それぞれのキー入力に対応したヘルプメッセージが必要となる.

- コマンドの実行

機能選択を 12 のキーを通じて行なう操作環境においては、これらの設定はユーザにまかされるべきである。このためには、次の 2 点が重要となる。

– キーへのコマンドの割り当て

汎用性を持ったシステムを実現するためには、キー操作を自由に変更できるべきである。つまり、キー操作とそれに対する実行コマンドとの対応をユーザが自由に設定できるようにする必要がある。さらにヘルプ機能も必要なため、キー、実行コマンド、ヘルプメッセージの 3 つを一組として指定する。

キー操作と機能の対応付けの方法としては、階層的な構成を持つものとした。キー操作を階層構造にすることによりコマンド群がグループ化される。任意の長さのストロークのコマンドを定義することができる。また、階層的で効率的なヘルプ機能が実現できる。

– コマンドの作成

すでに用意されているものがユーザの要求を満たさないときはユーザ自身が容易に新たなコマンドを作成できなくてならない。広くユーザが利用するためには、コマンドの作成が面倒なものであってはならない。すなわち、特別の知識

を必要とせずに、かつ、既存のコマンド群を利用して、用意に作成できるものでなくてはならない。また、デバッグも容易に行なえなければならない。

2.1.4　音声制御

システムの音声出力は音声合成装置によってなされる。実際の利用においては日本語と英語の出力が必須であるが、現在利用している音声合成装置は多国語に対応していないため、システムが必要となる音声合成装置を選択し同期を取る必要がある。また、音声制御として、音声の中断、再開、繰り返し、省略等の機能が必要となる。

現状では対応していないが、音声合成装置の出力に限らず音楽なども考えられる。システムのメカニズムとして、複数の音声合成装置とともにその他の出力装置の制御もできなくてはならないだろう。

2.1.5　セキュリティ

従来の電話を用いた情報提供システムなどには不特定多数のユーザが同一のサービスを受けることを前提としたものが多いが、Sophia/PhoneShell では、通常のログインと同じく、プライベートなデータも扱うため、十分なセキュリティが必要となる。現実的な方法としては認証は電話の 12 のキーより暗証番号を入力することになるだろう。しかし、数字のみの番号では組み合わせが少ないため、通常の UNIX の login のメカニズムでは十分ではない。2 ストロークによる ascii への変換も可能であるが、変換ルールを記憶しておかなければならず、容易ではない。

長い桁数を許すなどの対策により通常端末機を用いて計算機を操作する場合と同等かそれに近いセキュリティを確保する必要がある。

2.1.6　システムの構成

現在の Sophia/PhoneShell は、以下に示すハードウェアを用いている。

- SPARCstation ELC
- PNC−3500(NCU:Network Control Unit)
- Finetalk(日本語音声合成装置)
- DecTalk(英語音声合成装置)

ソフトウェアの構成は以下の 4 つのプロセスからなっている。

- ユーザシェルプロセス
- 中央制御プロセス
- 音声制御プロセス

- NCU 制御プロセス

これらのプロセスが図 2.1のようにプロセス間通信を行って全体の機能を実現する。

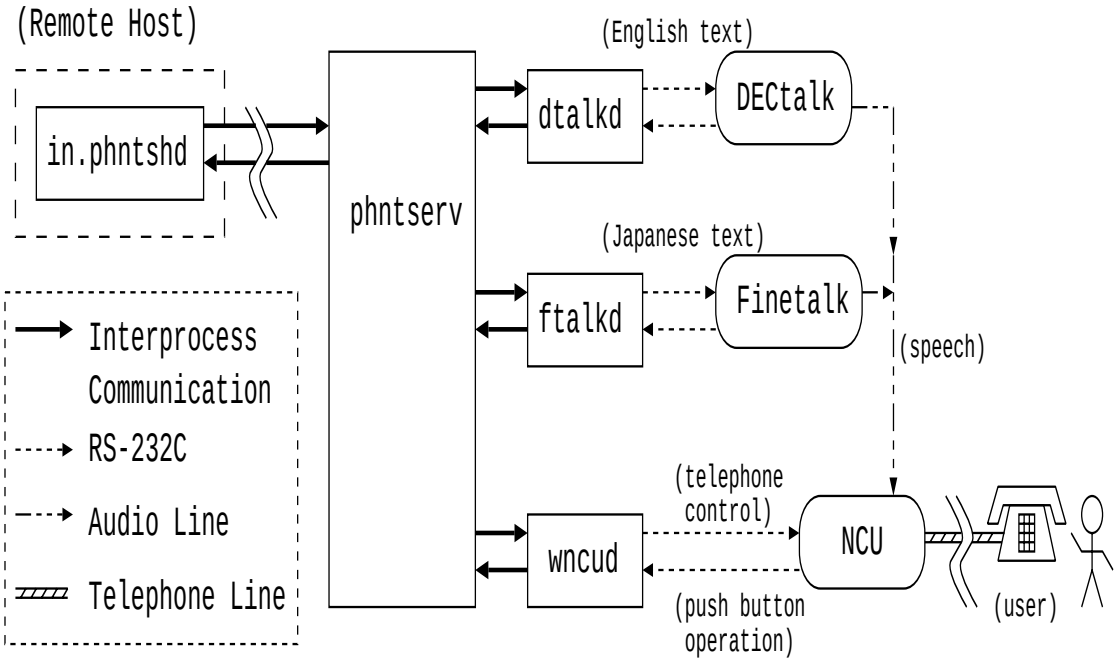

図 2.1: システムを構成するプロセスの関係図

中央制御プロセスは 4 つのプロセスの中心となりシステム全体の動作を制御するプロセスである。中央制御プロセスは入力モード切り替えや多国語を扱う際のふさわしい音声合成装置の選択も担当している。ユーザシェルプロセスはユーザが実際に利用するホスト上で起動される。ユーザシェルプロセスは起動されると最初に認証を行い、その後はユーザのキー入力にしたがって処理を行い、その結果出力を中央制御プロセスに返す。音声制御プロセスは音声合成装置またはソフトウェアを制御し、NCU 制御プロセスは NCU を制御して呼への応答、DTMF の認識、回線断等の処理を行う。

2.1.7 利用環境

ユーザの利用環境を決定するのはユーザシェルプロセスである。ユーザシェルプロセスは、最初に認証を行った後各ユーザが設定したコンフィグレーションファイルを読み込む。このファイルによりキーへのコマンドの割り当てやヘルプメッセージの定義などユーザ個人の設定が行われる。ユーザシェルプロセスにはシェル変数、モード制御（モード移行、コマンド実行)、ヘルプの機能が実装されている。

コンフィグレーションファイルの例を図 2.2に示す。この例は mh コマンドを用いて電子メールを読みあげるためのものである。この例において、mh_cmd と mh_readmail は、

```
set path /home/honma/bin/phntsh
set read_opt ( -short -header )
initcmd 'mh_cmd folder'
defmode root 'ルート' ''
  1  'goto mh_env'                    'mh 環境モード'
  4  'mh_readmail $read_opt[*]'       'メール読みあげ'
  5# 'mh_cmd scan $1'            'メッセージ番号指定'
  6  'mh_cmd prev; mh_cmd scan cur'   '前のメッセージ'
  7  'mh_cmd next; mh_cmd scan cur'   '次のメッセージ'
  8  'quit'                           'シェルの終了'
endmode
defmode mh_env 'mh 環境設定' \
                   '番号を選択してください。' return
  1  'mh_cmd folder +inbox'     'インボックスフォルダ'
  2  'mh_cmd folder +misc'      'ミスクフォルダ'
  4  'set read_opt ( -long ); speak option long' \
                                'ロングフォーマット'
  5  'set read_opt ( -short ); speak option short' \
                                'ショートフォーマット'
  6  'mh_cmd scan cur'          '現在のメール番号'
  7  'mh_cmd folder'            '現在のフォルダ名'
  8  'mh_cmd inc'               '新しいメールのインク'
```

図 2.2: メール読みあげのためのコンフィグレーションファイル例

それぞれ show、scan などのメールコマンドを実行しその出力をフィルタリングするためのプログラムである。

ユーザコマンドの例を図 2.3に示す。この例は UNIX の netstat コマンドを用いてネットワーク状態を確認する例である。

2.1.8　本節のまとめ

作成したシステムを用いて実際に電話から電子メイル読みあげや簡単なネットワーク監視機能を利用したところ十分に実用に耐えるものであることがわかった。パームトップ・コンピュータや計算機間の無線通信技術が普及すれば、それらを用いて本システムの目的をある程度果たすことができるかもしれない。しかし、それに対し本システムは特別なハードウェアを用いずに通信できることが特徴的であり、身近に電話以外の通信手段がない緊急時の計算機利用手段として非常に有効であると言える。

特に、ページャによる通知システムなどと組み合わせることにより、計算機あるいは

(例 1)
```
#!/bin/sh
# check input-errors using netstat
netstat -i | jgawk '
$1 != "Name" {
        printf("インターフェイス %s のインプットエラーは %s です。\n", $1, $6)
}'
```

(実行例)
```
インターフェイス le0 のインプットエラーは 377 です。
インターフェイス lo0 のインプットエラーは 0 です。
```

図 2.3: ユーザコマンドの例

ネットワークの管理に有効な手段となるであろう。

2.2 CHANT/PhoneShell

次に慶應義塾大学で開発中の CHANT/PhoneShell[1]について述べる。

インターネットの発展により従来の情報交換システムには見られなかった新しい文化やサービスが誕生した。例えば、電子メールという通信手段の出現により手紙、電話、ファクシミリなどといった従来のコミュニケーション手段に選択肢がひとつ増えた。その電子メールも最近は文字だけでなく、画像や音声など複数のメディアを扱えるようになり、より豊かな表現力を提供している。世界中では毎日大勢の人々が電子メールやネットワークニュース、データベースなどのインターネット上の情報やサービスを利用し、普段の生活や研究活動、業務の中で活用している。ところが現在インターネット上の情報やサービスを有効に利用するためには端末装置が必要であるため、計算機所持者と計算機非所持者の間に情報格差が生まれる可能性が高い。

この問題を解決する一つの方法として、計算機以外の装置によるインターネットの利用があげられる。PhoneShell ワーキンググループではネットワークにアクセスできない環境にいる管理者が端末装置を利用せずに管理作業を実施するための手段とそれを実現するための機構について研究している。すでに実験段階に入っている WIDE/PCS や Sophia/PhoneShell の電子メール読み上げシステムではネットワークにアクセスできない環境にいる管理者でも、端末装置を利用せずに管理作業に必要な情報の受信を行える。[2]これらのシステムでは端末装置を利用せずにインターネットの情報を入手する。

同じように、CHANT/PhoneShell は、計算機以外の装置を利用することにより端末装

[1]CHANT は，Change your Home Appliances to Network Terminals の略である。
[2]Sophia/PhoneShell の場合は管理者にとって必要な情報を要求することも可能である。

置のない環境におけるインターネットへのアクセスを可能にする。現在開発中のシステムではインターネットアクセス機構として、電話とファクシミリの共用を検討している。WIDE/PCS や Sophia/PhoneShell と異なる点は以下の通りである。

- 文字や音声だけでなく画像も扱うことができる
- 情報の受信だけでなく情報の送信も可能

CHANT/PhoneShell の実現により、広域ネットワーク管理者は端末装置の利用できない環境においても計算機ネットワークの管理に必要な情報を入手できるだけでなく、ネットワークニュースを利用したり MIME 対応の電子メールを送受信することが可能になる。つまりファクシミリと電話を共用することによってインターネットへ自由に利用できるようになる。(図 2.4)

2.2.1　実験システム

現在、PhoneShell ワーキンググループの研究成果である DTMF 制御技術と flexfax[3]の技術を利用して実験システムを構築中である。実験システムではファクシミリを文字/画像の入出力装置として利用し、電話は音声の入出力装置及びシステム側によってメニュー式に提示されるコマンド郡から実行したいコマンドを選択するための選択装置として利用する。また実験システムで提供するサービスは MIME 対応の電子メールの送受信のみとする。ファクシミリを通じて入力される文字及び画像は共に画像としてメール内に取り込まれ、送信される。メールの宛先は既存の「送信先一覧」から希望の宛先を選択して指定する。

2.2.2　本節のまとめ

これから CHANT/PhoneShell の実験的実装を行う予定である。実験システム完了後、αバージョンの設計を行い、本格的な運用に移行する予定である。

[3]Silicon Graphics Inc. の Sammuel J.Leffler 氏によって開発された、ファクシミリ関連のソフト。

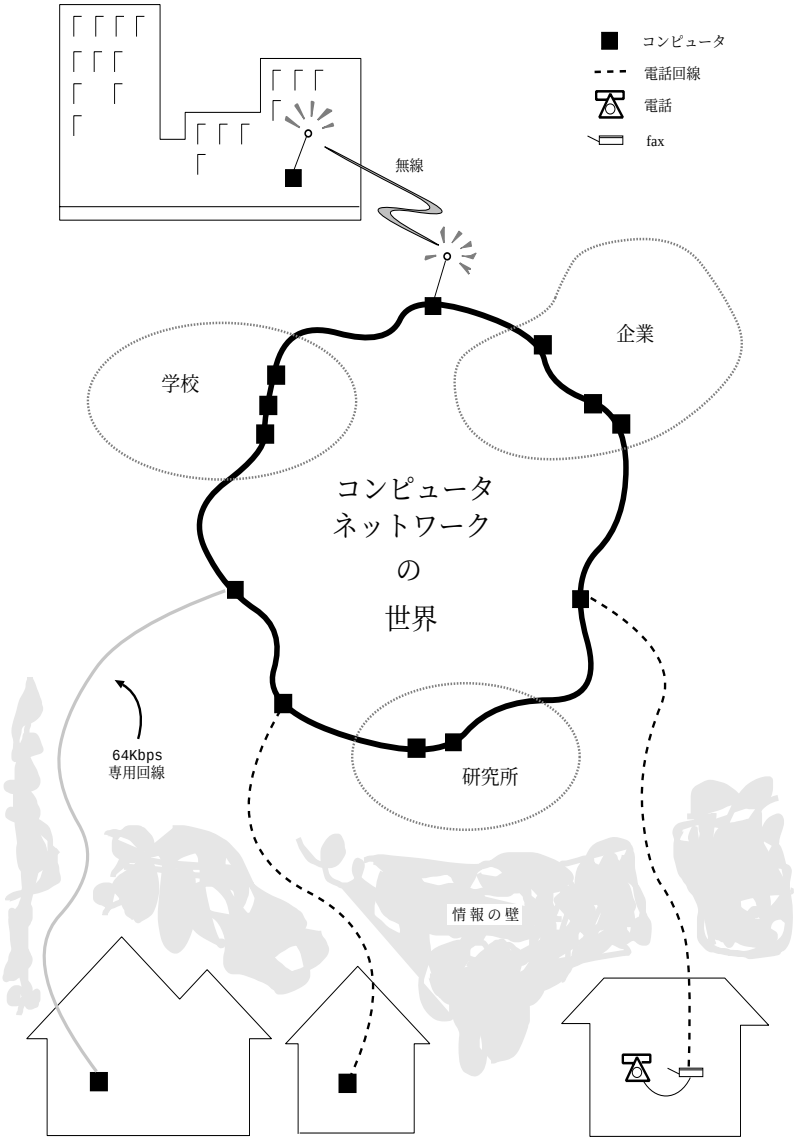

図 2.4: after CHANT/PhoneShell

第 3 章

WIDE/PhoneShell, WIDE/PCS の改良および応用

本章では, WIDE/PhoneShell および WIDE/PCS に施した改良作業および, WIDE/PhoneShell, WIDE/PCS を利用したシステムについて述べる.

3.1 WIDE/PCS のページャ呼び出し方法の改善

3.1.1 現システムの欠点

現在の WIDE/PCS がページャを呼び出す方法には、PNC-3500[1]を使う方法と AT Modem を使う方法の 2 つの方法がある。しかし、いずれの方法にもいくつかの欠点がある。

1. PNC-3500 を使う場合の欠点

   PNC-3500 は電話網制御装置と DTMF の送受信の機能があるので、現在のところページャの呼出には最適である。しかし、この装置を使った場合でも、以下のような欠点がある。

   (a) 相手に繋がった際に、電話線の極性反転がおこらない電話回線[2]では使用できない。

   これは PNC-3500 が相手に繋がったかどうかを極性反転かリングバックトーン[3]で検出する。ところがページャ呼出番号に電話した際にはリングバックトーンは一度も鳴らない。このため PNC-3500 を使用してページャを呼び出す場合はその電話回線が、極性反転する事が必要条件となる。また、PNC-3500 は相手が繋がったと認識しない限り、他のコマンドを受け付けないので、何もできない。

2. AT Modem を使う場合の欠点

   この場合の AT Modem は、ダイアラ[4]と、DTMF 発声装置として使われる。

   この場合の欠点は以下の通りである。

[1] 専用 NCU (網制御装置)
[2] 内線電話の一部、ISDN TA の一部など
[3] 電話をかけてから相手が出るまでの呼出音のこと
[4] 電話をかける装置

(a) 相手が繋がったことを判定する機能がないので、DTMF の発声はダイアル後に適当な時間をおいて行うだけである。このため、話中などで呼出が失敗している場合でも、あたかも成功したようにメッセージを返してしまう。

3. 両方式共通の欠点

上記 2 方式共通の欠点は以下の通り。

(a) DTMF を利用してメッセージを送る方法は、人間が操作する事を前提に作られている。このため、応答のメッセージは全て音声 (人の声) で行う。このため、DTMF 発信のタイミングなどを見計らう事が難しいので、文字化け、文字落ちが起きる場合がある。

(b) こちらからのデータがうまく伝わっているか確認することが出来ない。

(c) ページャの電話番号を間違えて、ページャ以外の電話 (人間など) に繋がった場合でも勝手に DTMF を送ってしまい、無事にメッセージを送ったとメッセージを返してしまう。

3.1.2 改善策 (1)『データ端末インターフェース』を利用する

NTT DoCoMo[5]は自由文型ページャを呼び出すインターフェースを記述した『ポケットベルメッセージ入力に関する技術参考資料』という資料を用意している。この資料には「PB 信号[6]入力インターフェース」と「データ端末インターフェース」についての説明がされている。この「データ端末インターフェース」にはさらに「無手順端末」、「JUST-PC 端末」、「DDX-P 端末」の 3 方式の手順が書かれている。

今回はこのうち「無手順端末」を利用して WIDE/PCS からの呼出を行うことを考える。この「無手順端末」とは 300bps(ITU-T V.21), 1200bps(ITU-T V.22), 2400bps(ITU-T V.22bis) の modem を使い、ポケットベルセンタに接続し、特定のプロトコルで通信を行い、ページャを呼び出す事である。

この方式を利用した場合に考えられる利点は以下の通りである。

1. プロトコルがあるので、文字化け、文字落ちがない。
2. 極性反転しない回線でも使用できる。
3. 使用されなくなった低速モデムでも使える。
4. ページャ以外の電話番号への間違い電話の検出が出来る。

しかし、欠点もある。

1. NTT DoCoMo 自由文型でしか使用できない[7]。

[5]NTT 移動通信網株式会社
[6]DTMF のこと
[7]テレメッセージ系のページャでも同様のインターフェースを持つ機種があるようである。

2. ポケベルセンタの電話番号は地域別、種類別[8]で分かれているので、どのページャがどのタイプかというのデータを持っている必要がある。

3.1.3　改善策 (2) 音声認識

今までの方法では、ポケベルセンタからの応答が音声 (主に人の声) であり、その内容を認識できないので、DTMF 送信のタイミングが間違ってしまい、文字化け、文字落ちなどを起こしていたことが多い。そこで、ポケベルセンタからの応答を FFT などを利用して解析し、今どのような状態かを把握すれば回避できるのではないかと考えた。認識と言っても大体以下のような物が区別できれば良いと考えられる。

- リングバックトーン (呼出音)
- ビジートーン (話中音)
- 人の声
- 第 2 発信音 (ポケベルセンタが DTMF を受け付ける状態を表す音)

幸い、最近は音声入力装置が標準装備されているワークステーションも多い。また、CPU も高速化して来ているので、充分実現可能と思われる。

3.1.4　今後の予定

現状では動作確認の為のプロトタイプや実験を行っている。今後は実際に WIDE/PCS への実装を行う.

3.2　PDA の活用

最近になって Newton/Zoomer/Zaurus など PDA(Paresonal Data Assistance) と呼ばれシステム手帳サイズの小型計算機が市場に姿を表した。
これらはいずれも

- 小型・軽量
- 文字の手書き認識機能を搭載
- (何らかの) プログラム手段を搭載
- 外部とのインターフェイス (赤外線・RS-232C) をもつ

ことを特徴としている。
本ワーキンググループでは、ユーザと計算機間のインタラクションの可能性を探っているが、この PDA はある意味で一番ユーザに近いところに存在する計算機といえる。
現在の WIDE/PCS の利用形態、すなわち

[8] 同じ NTT DoCoMo 自由文型でも広域タイプと地域タイプがある

1. コンピュータ環境監視プロセスが何らかの以上を検出
2. WIDE/PCS が管理者に Pager を使って警報を発する
3. 管理者が何らかのアクションを行なう

に、PDA をどう組み込むべきか考えてみた。

その結果、管理者がアクションを送れれば良いが、その管理者が何らかの理由で早急にアクションを起こすことができない場合、管理者の代理にたいしてアクションを転送する時に PDA を使い、管理者の代理の Pager にメッセージを送ることを検討した。

その場合、管理者は管理者代理に対して

- どこの計算機が
- どういう状態で
- どういう対処を行なう必要があるか
- 何をしてもらいたいのか

を送る必要があるであろう。

WIDE/PCS で多く使われている NTT DoCoMo の自由文型 Pager は最大 34 文字の JIS 第 1 水準の漢字を送ることができるが、送るためのユーザインターフェイスは一文字送るのに 5 回の DTMF を打たねばならない。

ある程度決まった文字列 (例えば「連絡下さい」/「会社に戻って下さい」) などは短縮系で入力可能であるが、トラブルシューティングの代行のためのメッセージの転送などにはむかない。

そこで、PDA を Pager に対する簡易な入力デバイスとして扱うことを検討した。

PDA は先にあげたように通常

- 手書き入力
- 何らかのプログラムができ
- 外部との入出力

ができる。特に Newton/Zoomer では単独で DTMF 音の発生が可能であり PDA の中の電話帳と組み合わせることにより手近に公衆電話 (携帯電話を持ち歩いている場合はこれも必要ない) があれば処理の代行をお願いすることを管理者の Pager に送り出すことができる。

現在 Phone-Shell WG では、シャープの電子手帳である Zaurus と Apple の Newton 上で動作する Pager の入力プログラムを開発中で、Zaurus 版ではシャープ製のハイパー関数カードを用い BASIC で作成中である。

機能的には Zaurus の手書き認識機能を使い送りたい文字列を入力し、オプションポート 15 を使い外部の DTMF 発信器をドライブし Pager に対して文字列を送ることになる。

3.3　WIDE/PCS 支援機構

3.3.1　WIDE/PCS エミュレータ

WIDE/PCS は、電子メールからページャにメッセージを転送する機構としてすでに３年以上利用されている [211], [212]。WIDE/PCS のユーザは、各自のメール送出用プログラム上で通常の電子メールと同じようにメッセージを作りこれを発送すれば、指定したページャにメッセージを送ることができるので、現在も広く利用されている。しかし、その際 WIDE/PCS の送出文字列の制限である 36 文字を越えないように気を付ける必要があった。36 文字を越えた部分は切り落とされてしまうためである。このためユーザは、作成したメッセージが十分短くあきらかに 36 文字以内であることが明確である場合を除き、文字数をいちいち数えなければならなかった。この不便な状況を改善するために、ページャのエミュレータを用意した。このエミュレータはユーザが作成したメッセージを読み込み、ページャ上にどのようなメッセージが表示されるかを推定して、表示するもので、 nemacs や mule 上からも利用できるので、例えば mh-e でメールを作成したユーザは、作成したメッセージをその場でエミュレータに送り、どのように表示されるかを目視確認できるようになった。このエミュレータは、自由文型だけでなく WIDE/PCS がサポートしている全てのページャをエミュレートしている。

3.3.2　WIDE/PCS 再送機構

ページャにメッセージが送られても、ページャの所有者がページャの電波の到達範囲外にいた場合にはメッセージは送られず、そのことはメッセージの発信側も受信側も知ることはできない。そこで、Sophia/PhoneShell を使い、メッセージを再送する機構を用意した。これは、ページャの所有者が Sophia/PhoneShell にアクセスし、予め決められた ID とパスワードを正しく入力することで利用可能になる。この「再送機構」の利用者は、すでに WIDE/PCS 経由で送出された自分あてのメッセージのうちのいくつかを明示的に指定し、これを再度送出するように WIDE/PCS 側に指示することができる。この機構により、ページャ所有者が移動中に地下やビル内に入り一時的にページャサービス範囲外に入ってもその間にメッセージが送られたか否かを後から確認することができるようになった。

3.4　認証システムへの応用

地理的に離れた計算機にログインできることはネットワークの利点である。ところで、通常計算機の使用資格の確認は通常 ID とパスワードとで行なわれる。現在の TCP/IP では暗号化を行なわないので、遠隔ログイン を行うと、ネットワーク上を ID とパスワードとが平文のまま流れる。したがって、LAN や通信回線をのぞき見ることができればこの ID とパスワードとを手に入れることができる。実際、今年の２月には大規模な（パスワードの盗聴）事件が報告されているが、依然としてパスワードを平文のまま送ることの危険性より、遠隔ログインの利便性をとっているところが多い。しかし最近になって、平文でパス

ワードを送る方法に代わる方法として Challenge-Response を使うところが現れはじめた。ここでは平文でパスワードを送る方法に代わる方法の WIDE/PCS,WIDE/PhoneShell 技術を用いた案について述べる。

3.4.1 問題点の確認

平文でパスワードを送る方法が危険であるのは次の理由である。

- 計算機の使用資格の確認には ID とパスワードとを使う。
- パスワードは使用者が変更するまで同一のものである。
- パスワードは平文でネットワークを流れる。
- ネットワークはのぞき見できる。
- したがって、のぞき見て得たパスワードは変更されるまで使える。

3.4.2 Challenge-Response 方式による対抗策の概要

以下に述べる方式のポイントは、使用資格の確認のための情報を暗号化することにある。使用者は携帯計算機を身につけている。 使用者が計算機にログインするために ID を送ると、計算機は ID をもとにその使用者の鍵で（時刻などを含む）乱数を DES, MD5 などで暗号化して（ネットワークを通して）使用者に提示する (Challenge)。使用者は携帯計算機を使って提示された情報を復号化し、その結果を計算機に（ネットワークを通して）送る (Response)。計算機は Response が Challenge を復号化したものであることを確認して使用を許可する。

3.4.3 WIDE/PCS を用いた対抗策について

以下に述べる方式のポイントは、計算機がパスワードをログイン毎に変更し、そのパスワードをページャを通して使用者に知らせることである。手順は次のようになる。

1. ページャを携帯している使用者が離れたところにある計算機にログインするために ID を（ネットワークを通して）送る。
2. 計算機はパスワードを作成する。
3. 計算機は WIDE/PCS に ID とパスワードとを渡す。
4. WIDE/PCS は ID をもとに電話番号を調べて、その番号のページャにパスワードを表示する。
5. 使用者はページャに表示されたパスワードを計算機に送る。

6. 計算機は入力されたパスワードと作成したパスワードとを比較し、同一であったら使用をゆるす。

この方法ではページャへ渡す情報（電波で送られる。）を盗聴し、ネットワークをのぞきみていると、ID とパスワードとを入手できるという問題点があるが、ページャへ送られる情報とネットワークを流れる情報とを同時に監視しなければ不正なログイン はできないので、現状よりは良好なセキュリティを安価に確保することが可能となる。

第 4 章

おわりに

PhoneShell ワーキンググループでは、以前から端末装置を直接アクセスできない環境下において必要最低限の計算機操作を可能にする方法を検討してきた。そして、DTMF 送出可能な電話や FAX、さらにページャを利用して計算機を操作するシステムを実装し、これがネットワーク管理に有効に働くことをしめしてきた。今年度の報告書では、こうして作られて来た環境を一層信頼性のある手法とするための試みと、ネットワーク管理以外の目的にも利用可能であることの実証例をとりあげた。

ところで、計算機に接続でき携帯可能な無線通信装置が最近普及の兆しを見せ始めている。また、端末装置も小型軽量化が進んでいる。これらの装置が普及すれば、PhoneShell ワーキンググループが目指す場所や時間に拘束されない計算機アクセスの手段として、伝統的な文字端末が復活する。今後はこれらの機材の有効利用も積極的に検討し、すでに手掛けてきた DTMF、音声、ページャ、FAX などとの組合せによる、有機的な利用形態を構築したい。