

携帯機器向けのオープンなGUI環境「式神」

大谷 浩司, 福居 宏和, 橋本 圭介, 渦原 茂, 竹岡 尚三
株式会社 アックス*

1 はじめに

我々は携帯機器向けに便利で親しみやすく、誰もがアプリケーションを開発可能であるオープンなGUI環境「式神」¹を開発している。

PDA やハンドヘルド PC などの携帯情報機器は、ハードウェア技術の進歩とコスト・ダウンによって非常に高い性能を持つようになってきた。そのため、かつてのデスクトップ・マシンのソフトウェア環境をほとんどそのまま搭載することも可能となっている。しかし、携帯情報機器では、画面の大きさ、入力デバイスなどの制約がある他、車内等での使用など利用シーンの点でデスクトップ・マシンと大きく異なっている。従って、GNOME や KDE などのデスクトップの環境をそのまま適用したのでは、不便である。そこで、我々は携帯機器向けの環境を新たに開発することにした。

「式神」は、大きく分けて以下から構成される。

1. 操作の手間を省く単純な「デスクトップ」

ミーティングでの片手間の使用、街頭や電車内などでの使用などを考えると、携帯システムでは、簡単なことが簡単にできることが重要である。また、画面のサイズが小さいために同時に多くのアプリケーションを画面上においておくことは便利とはいえない。そこで、表示するアプリケーションの数は、最小限にし、操作を単純化した。

「式神」のターゲットは、ハンドヘルド PC と PDA であるが、これらの間でも画面、入力デバイスの点で違いがある。ハンドヘルド PC は、画面は横長でキーボードを備える。PDA は、画面がさらに小さく、キーボードを備えていない。そのため、デスクトップも全く同一のものにすると、UI 上の見地から無理がある。そこで、「式神」においても、この2種類の機器向けの画面構成は異なったものとした。

* <http://www.axe-inc.co.jp/>

¹ 使いやすい道具としての、陰陽師の操る「使い魔」の名前にちなんだ名前である。 <http://www.sikigami.com/>

また、すべての API を新規のものとするのは、アプリケーション開発者の負担を負やし、動作するアプリケーションの数を減らす原因となる。そこで、デスクトップ・マシンでの標準的な API である GNOME(GTK+) とできるだけ互換性を持たせることにする。

2. 手書き文字認識システム「布目」

キーボードを持たない PDA などでは入力手段として、ペンを使った手書き文字入力は、必須の機能である。しかし、漢字が入力可能で実用的な能力を持ったオープンな手書き入力手段は、現状では存在していない。そこで、我々は新たに開発することにした。認識率として実用上問題ない水準と考えられる、ひらがな/カタカナ/漢字の平均 98% を目標とする。

認識に利用する文字パターン辞書は、オープンな形で作成したいと考えている。

3. 親しみやすい UI を作成可能な「アニメーション・ツールキット」

多くのユーザに利用してもらうには、UI が親しみやすいということは重要である。特に、常に持ち歩く携帯機器では、その傾向が強いと思われる。そこで、アプリケーションの開発者が、容易にアニメーションを UI に付加できるように、ツールキットを用意する。

4. 知的情報処理を補助する「人工知能機能」

知的情報処理を行なう場合、人工知能を利用してユーザの入力を補えば、入出力装置の機能が限られているという欠点をカバーすることができる。また、携帯機器は、ユーザと共に存在することが多く、人工知能によって秘書的な機能を果たすことができれば、非常に便利である。そこで、我々は、「式神」に人工知能機能を含めることとした。

5. 標準アプリケーション

我々は、以上のアプリケーション環境に加えて、さらに携帯情報機器に必須と思われる以下のアプリケーションを提供する。これらの標準アプリケーションは、「人工知能」機能と関係する。

- PIM(Personal Information Manager)
スケジュール、アドレス帳、メモ帳などから構成され、既存の機器においても標準的なアプリケーションである。
- リモコン
情報家電などが普及し、情報を各機器およびインターネット間でやりとりしたり、関係させたりするようになると、統一した制御インタフェースを司るものが必要になってくる。その有力な候補としては、携帯機器が考えられる。
そこで、我々は、家電や他の情報機器などを IrDA や Bluetooth などの無線装置を利用して制御するアプリケーションを標準で提供する。

以下では、標準アプリケーション以外の各部分について詳しく述べる。標準アプリケーションの詳細については、本稿では省略させていただく。

2 デスクトップ

先に述べたように、デスクトップの画面構成は、ハンドヘルド PC 向けと PDA 向けの 2 種類用意した。まず、ハンドヘルド PC 向け、次に PDA 向けを説明する。

2.1 ハンドヘルド PC 向け

図 1 に、ハンドヘルド PC 画面の例を示す。

ハンドヘルド PC 向けの大きな特徴は、画面の主な部分をメインとサブの二つの領域に分け、各領域を満たすようにアプリケーションを表示することである。ただし、電卓などは、必要な大きさのみを表示する。図 1 では、メイン領域 (1) にスケジュール、サブ領域 (2) に電卓を表示している。

2.1.1 アプリケーション・ウィンドウ配置

GNOME などのデスクトップ・マシン向けの環境では、アプリケーションは画面上の任意の場所に複数表示する。ユーザは、必要に応じて、ポインティング・デバイスなどを利用してサイズや位置を変更する。しかし、画面のサイズの小さい携帯機器では、画面上に多

数のアプリケーションのウィンドウを表示しても、不便なだけである。また、ユーザが立った状態など、機器の保持が不安定な状態でも利用することなどを考えると、ポインティング・デバイスの頻繁な利用は不便である。

一方、Windows CE などでは、画面にはひとつのアプリケーションのみを表示する。この方法では、アプリケーションは広い画面を利用できる。しかし、Web ページをみながらメールを書く、電卓で計算しつつ、メールを書くなど同時に複数のアプリケーションの表示を行ないたい場合もある。

そこで、我々は、中間的なものとして、先に述べたように二つの領域にアプリケーションを表示する方法を採用した。二つの領域で動作中のアプリケーション (以下では、タスクと呼ぶことにする) の表示は、必要に応じて瞬時に交換できるものとする。また、広い画面を利用したい場合は、簡単にひとつの画面にすることも可能とする。

2.1.2 メニュー・バー

GNOME などでは、アプリケーションは自身のメニュー・バー、ツール・バーをアプリケーションのウィンドウに表示する。しかし、表示はしているが利用していないアプリケーションのメニュー・バーを表示することは、画面リソースの無駄である。そのため、これらを表示しないこととする。その上で、現在アクティブなアプリケーションのメニュー・バー、ツール・バーを画面の上部に表示する。図 1 では、スケジュールのメニュー・バーを (3) の部分に、ツール・バーを (4) の部分に表示している。

2.1.3 仮名漢字変換領域

仮名漢字変換の状態等は、メニュー・バーと同じ画面上部の右端 (5) に表示する。

2.1.4 パネル

パネルは、GNOME の場合と同じく、以下のようなものを表示するための領域である。

- アプリケーションと独立のメニュー

図 1 では、アプリケーションの起動やシステムの状態設定等の項目を持つシステム・メニューが、右上部 (11) に表示されている。システム・メニューは必ず存在するものとする。

- タスクの一覧

現在動作しているタスクのうち、メイン、サブのどちらの領域にも表示していないものの一覧を表

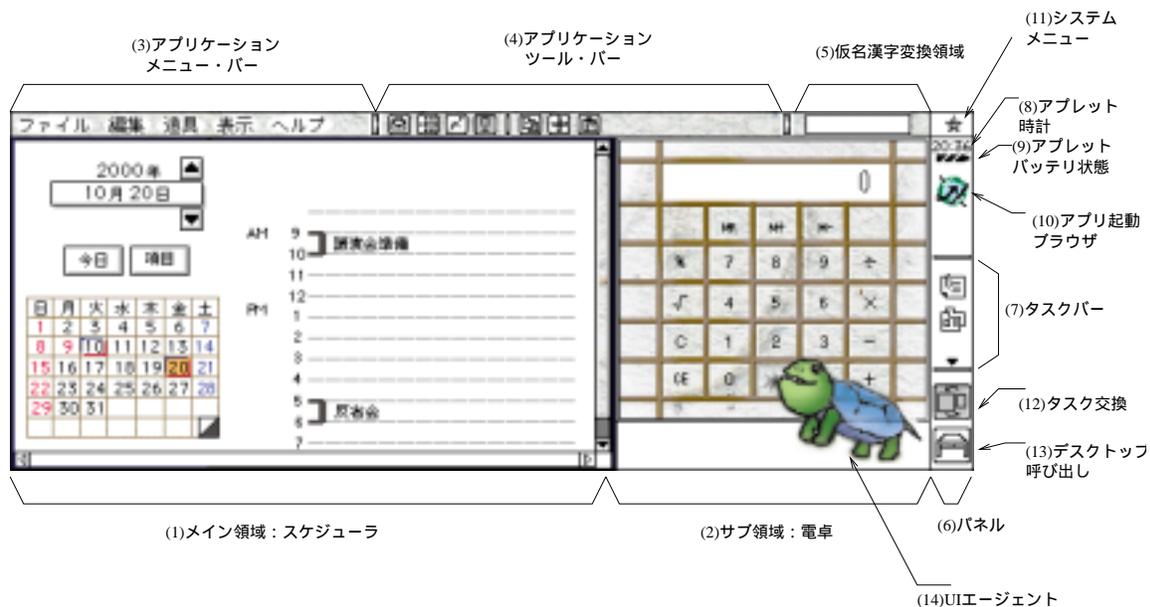


図 1: ハンドヘルド PC 画面

示す。図 1 では、(7) の部分に表示されている。

2.2 PDA 向け

● アプリケーション起動アイコン

図 1 では、ブラウザ起動用のアイコン (10) が表示されている。

● 時計などの小さな表示を行なうアプリケーション (アプレットと呼ぶ)

図 1 では、時計 (8) とバッテリー状態 (9) が表示されている。

● その他の機能アイコン

以下のものが存在する。

－ タスク交換 (12)

メインとサブの領域のタスクを交換する。

－ デスクトップ呼び出し (13)

サブ領域にデスクトップを呼び出す。デスクトップには、アプリケーション起動アイコン、ゴミ箱アイコンなどが存在する。

画面は横に長いいため、パネルは画面右端 (6) に表示する。

2.1.5 UI エージェント

後述するように「式神」の人工知能機能のユーザ・インタフェースは、UI エージェントが行なう。UI エージェントは、必要に応じて画面上に現れる。図 1 では、亀 (14) として表示されている。

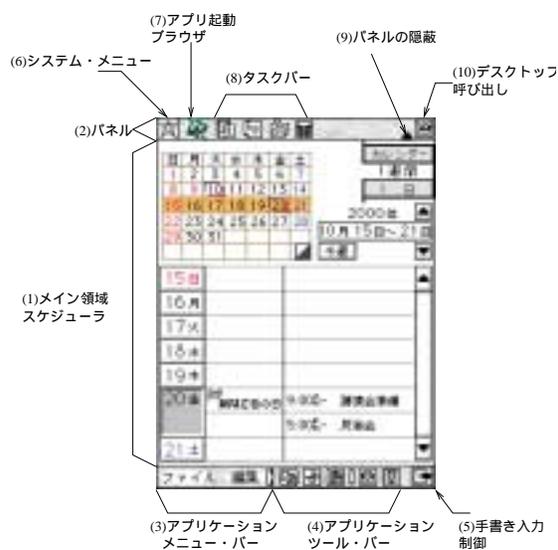


図 2: PDA 画面

図 2 に PDA 向け画面の例を示す。PDA では、画面はさらに小さいために、サブ領域についてはあきらめ、アプリケーションはひとつのみを表示する。図 2 では、メイン領域にスケジューラが表示されている。

2.2.1 パネルとメニュー・バー

メニュー・バー、ツール・バーは、ハンドヘルド PC 向けと異なり画面下部に配置する。一方パネルは、上部に表示する。これは、次の理由による。

画面は、縦に長いためこれらは、画面の上部あるい

は下部に配置するのが妥当と思われる。縦長の PDA を持ち、ペンで画面をタッチすることを考えた場合、手の平の真中に近い下部に存在する部分が安定して利用しやすい。また、上部にあった場合には、メニューが表示された場合に、ペンを持つ手で項目が隠される恐れがある。従って、アプリケーション利用中に頻繁に操作されるメニュー・バーを下部に置くことが妥当である。

2.2.2 手書き入力ウィンドウ

PDA の場合には、キーボードを備えていない。そのため、手書き入力を行なう必要がある。手書き入力用のウィンドウの表示の制御は、メニュー・バーの右端のアイコン (5) によって行なうことができる。ただし、多くの場合は、アプリケーションの文字入力領域をペンで触るだけで表示する。

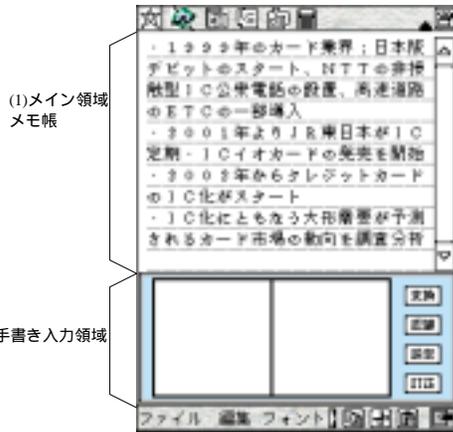


図 3: 手書き入力画面

手書き入力ウィンドウを表示した画面例を図 3 に示す。手書き入力ウィンドウは、画面下部の領域 (2) に表示される。下部に表示する理由は、メニュー・バーの場合と同様、下部の方が安定して利用しやすいと思われるためである。図 3 では、メイン領域にはメモ帳が表示されている。

3 布目

3.1 構成

図 4 に示すように、「布目」は、手書き文字認識エンジンである tegakiServer と、手書き文字入力フロントエンド tegakiInput の二つのモジュールから構成される。入力された手書き文字パターンは tegakiInput によりノイズ除去などの前処理が施された後、tegakiServer

ペンによる手書き文字入力

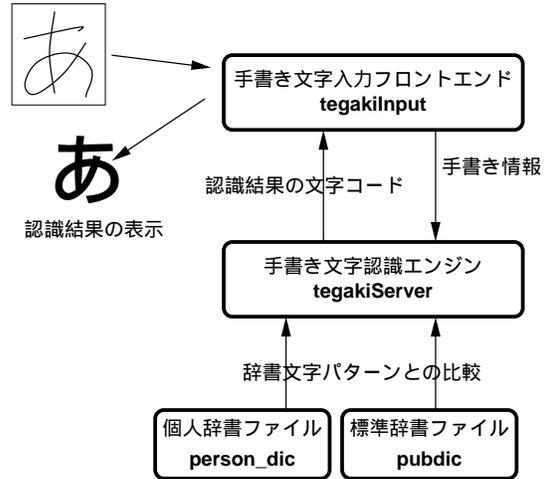


図 4: 布目の構成

へと送られる。

tegakiServer は、標準的な文字パターンが格納されている辞書ファイル中の辞書文字パターンと入力文字パターンとを比較することで、入力文字パターンがどの文字種であるかを判定する。tegakiServer は判定された文字種の文字コードを tegakiInput に送り返す。tegakiInput はその文字種をユーザに提示する。

3.2 手書き入力方法

認識結果の候補文字がポップアップウィンドウで表示される

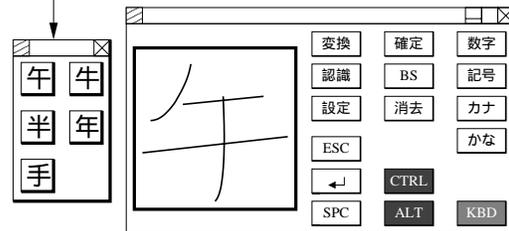


図 5: 一文字入力モード

手書き入力方法として、二種類の方法を提供する。まず、最初の方法は、図 5 に示すように、tegakiInput が手書き文字入力枠を一つ用意し、入力者は枠内に一文字ずつ手書きを行い、認識させる方法である。この方法では、認識ボタンを押すか、二秒程度何も入力しなければ、枠内に書かれた手書き文字の認識を開始する。そして、認識結果である候補文字がポップアップウィンドウに表示される。このポップアップウィンドウ中の文字を選択することで、選択された文字が入力される。すなわち、単漢字変換のように、一文字ずつ確実に入力できることを特徴としており、短い単語など

の入力に適している反面、入力速度が遅く、明示的に認識作業を行わなければならないため、文章などの入力には不向きである。

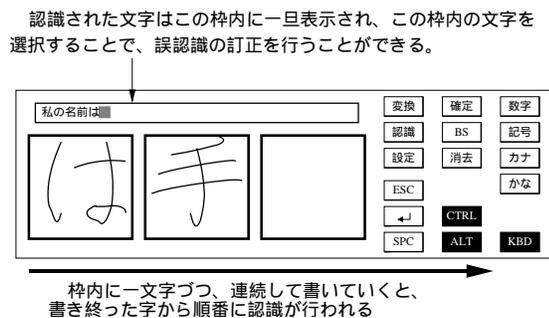


図 6: 連続入力モード

もう一つの方法は、図 6 に示すように tegakiInput が手書き文字入力枠を複数用意し、複数の入力枠に連続して入力することで、文字認識を行う方法である。この方法では、入力され文字認識された結果が入力枠上に一行分表示され、表示されている文字列の各文字を選択することで、文字認識をやり直し、認識結果の修正を行うことができる。すなわち、連続して文字入力することを特徴としており、認識作業を明示的に行う必要がなく、入力中の人間の思考を妨げないため、文章や文節単位での入力に適している。

3.3 学習機能

手書き文字には、個人ごとの書き方のくせがあるため、標準辞書を作成するためには、一つの文字種あたり数十人分の手書き文字パターンが必要とされている [1]。しかし、PDA などでは個人での使用が多く、使用者の書き方のくせに合わせた辞書を用いると認識率を向上させることができる [2]。このため、個人ごとの書き方のくせを取り入れる学習機能が必要である。

「布目」の学習機能としては二つの方法を提供する。一つは、標準辞書の辞書文字パターンに入力した文字パターンのくせを加えていく方法であり、もう一つは、個人用の辞書ファイルを用意し、個人のくせのある文字パターンを追加する方法である。標準辞書に個人のくせを加える方法は、学習効果がハッキリと現れにくい。個人用の辞書を用意する方法は、学習効果は非常に高いが、個人のくせの影響を大きく受けるため、個人辞書を作成した本人以外の文字認識率は大幅に低下する [2]。

4 アニメーション・ツールキット

ここでは、アニメーションを作成・実行するためのアニメーション・ツールキットについて述べる。このツールキットが提供するものは、アニメーションを使うアプリケーションプログラムとやり取りをしながら、独立的にアニメーションを実行する実体である。

こういったアニメーションの実体の実現例としては、アニメーション GIF や Macromedia 社の FLASH などがある。また、W3C はオープンな規格として XML ベースの SMIL を作成している [3]。しかしながら、FLASH や SMIL は基本的にドキュメントであり、ツールキットのようにプログラムの部品として使うことを目的に作られていない。Microsoft Agent では、アニメーションのキャラクタをプログラムから制御する枠組を提供しているが、汎用のアニメーション作成を目的としていない [4]。

以下では、アニメーション・ツールキットの基本的な枠組について述べる。

4.1 アニメーション・エレメント

アニメーション全体は、アニメーション・エレメントという部品から構成される。アニメーション・エレメントは、あるまとまったアニメーションの単位で、1つのエレメントが複数のサブエレメントからなる場合もある。サブエレメントは奥行方向に順序づけられ、重ね合わせた画像がメインエレメントとしての画像となる。

アニメーション・エレメントは属性として位置座標と値(透過率)をもち、拡大・縮小、回転、変形、色変換(輝度等)、クリッピングといった効果や、アニメーションの繰返し回数や制限時間を設定することができる。また、アニメーション・エレメントには、アクション(手続き)を設定することができる。アクションによって、他のエレメントの呼び出しや、エレメントのアニメーション終了などを通知することができる。

4.2 アニメーション関数

アニメーションの実行は、概念的に次の2つのことを行なう：

- (a) 時間毎に1コマを描画する
- (b) そのとき、どんな1コマを描画するかを決定する

アニメーション関数とは、(b)の決定のための手続きのことである。例えば、順々にビットマップデータを変えていくことでアニメーションを実現できるが、この場合、アニメーション関数はどのビットマップデー

タかを決定する。また、地面でバウンドするボールのアニメーションのように、ビットマップ (ボールの絵) は1つしかないが、位置を変えることでアニメーションを作ることもできる。この場合、アニメーション関数は、(放物線関数にしたがって) 各コマ (時刻) における位置座標を決定する。

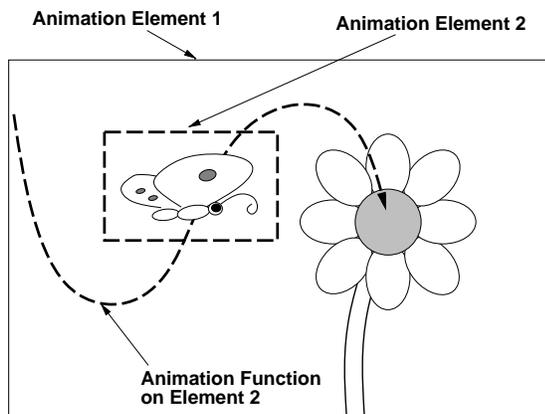


図 7: アニメーションの構成例

ツールキットは、いくつかの基本的なアニメーション関数を提供する。例えば、直線やスプライン曲線による移動経路の指定や、その際の速度変化 (コマ毎の移動距離) といったものである。(図 7 は、エレメント 2 の蝶の羽ばたくアニメーションが曲線に沿って移動する) また、ツールキットのユーザは、独自のアニメーション関数を用意することでツールキットを拡張できる。

4.3 イベント処理

イベント処理とは、マウスカーソルでアニメーション・エレメントをクリックしたときなどに起こす手続きのことである。イベント処理によって、カーソルが触れることによるアニメーションの開始や、アプリケーションプログラムへのイベント発生の通知といったことを行なう。

4.4 音との同期

アニメーション・エレメントに対して音を指定することができる。音の指定には、同期モードと非同期モードの 2 種類がある。同期モードでは、音の再生に合わせてアニメーションが実行され、もしアニメーションが間に合わなければコマを飛ばすが、非同期モードでは、このような同期はとらない。

音のデータに応じたアニメーション関数を定義することで、音に合わせて描画内容を変化させることも考えられる。例えば、音声に合わせて口の開閉をするアニメーションなどである。

4.5 アプリケーションからの制御

アプリケーションプログラムは、アニメーション・エレメントを制御できるが、アニメーション関数が参照しているパラメータを変更することでアニメーションの振舞いを細かく変更することもできる。

4.6 GUI ツールキットとの関係

アニメーションツールキットは、それによって実現されるアニメーションを既存の GUI ツールキットの枠組で使える実装もとる。これによって、例えば、ボタンのラベルなどにアニメーションを貼ることができる。

5 人工知能機能

5.1 概要

式神の特徴の一つとして、人工知能 (AI) 機能がある。式神の人工知能機能は主に学習、推論とユーザ対話を目的としている。また、式神 AI はユーザとの対話のために、ユーザ・インターフェース・エージェント (UI エージェント) の使用を前提としている。

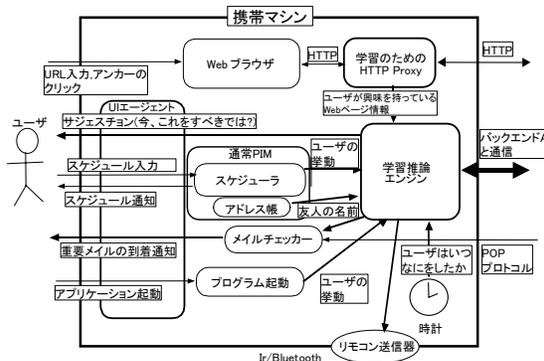


図 8: 式神 AI ブロック図

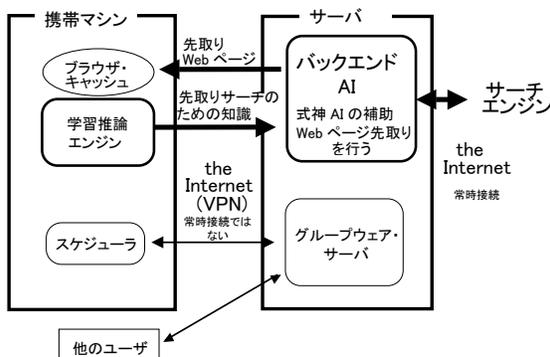


図 9: バックエンド AI との関係

式神 AI は主に学習推論エンジンと、対話メカニズ

ムから構成されている(図8)。また、式神 AI はバックエンドの AI を使用する事ができる(図9)。

学習推論エンジンは、ユーザが携帯マシンで行うことや、携帯マシン上のデータベースを常時監視し学習している。そしてユーザの行いたい事や行うべきだが忘れていたことなどを推論し、それを代行、指摘することでユーザの知的情報処理を支援する。式神 AI はユーザが常時携帯するマシンならではきめ細かい学習が可能で、それによってきめ細かい知的作業のサポートが可能である。また、個人所有のマシン内にほとんどの情報を保持するので、ユーザの個人情報の保護についても単純で判り易い。

学習推論エンジンの具体的な適用は

- Web ページの自動検索
- 重要なメールの推測とその到着の通知
- 行動予定の自動補完
- リモコンの自動制御

などを始めとし、広い分野を考えている。
以下では、式神 AI について詳しく述べる

5.2 式神 AI

式神 AI は学習エンジンと推論エンジンを一体化したソフトウェアである。また、同時に簡単な自然言語処理機構も持っており、UI エージェントの対話メカニズムのエンジンや、Web ページ解析時の単語抽出などに使用する。自然言語処理のための辞書として、仮名漢字変換システムの辞書を使用する事を検討している。学習、推論は、基本的に記号処理で行い、ニューロ・ネットワークの全面的な使用は考えていない。

式神 AI は広義のエージェント [5] の機能を一部持つが、我々は式神 AI をエージェントとは位置付けしない。式神 AI は旧来的な人工知能の実現の一つであると考える。

式神 AI からユーザへの通知は、UI エージェントを通した即時的なもの、式神 AI からユーザへのメールによるもの、単なる音によるものなど、用途、緊急度によって、多様なインタラクションを考えている。

5.3 バックエンド AI

式神 AI はバックエンドの AI を使用する事ができる。バックエンド AI とは、常時インターネット接続され、また、CPU、メモリ、外部記憶を十分に持ったサーバ機上で動作する AI エンジンである。式神 AI が携帯マシン上で学習した知識を、バックエンド AI に送

り込み処理した後、式神 AI に送り返す事で、式神 AI はより確度の高い推論が可能になる。また、バックエンド AI は携帯マシンとの接続が切れている間も、ユーザの興味ある情報を推論し、インターネット上の情報の探索、収集を実行する。情報の探索にはメタサーチエンジン技術を使用する。携帯マシンと再び接続が起こった時に、バックエンド AI は式神 AI に収集した情報を渡す。これらバックエンド AI の作業は、ユーザの知らないうちに行われる。結果として、ユーザは自分の見たい Web ページなどをいつも携帯マシン上のキャッシュにもつことになり、常時接続でない携帯マシンで快適に知的情報処理が行える。

5.4 UI エージェント

式神 AI のユーザに対するインターフェースは、UI エージェントが行う。UI エージェントは式神のアニメーション機能を使用して作成される。UI エージェントの見かけはユーザの好みで任意に作成し選んで使用できる。UI エージェントの振る舞い記述方法も公開し誰でも自由に UI エージェントを作成可能とする。

通常、UI エージェントは親しみの持てる外観を持ち、ユーザと擬人化された対話を行う。UI エージェントは、時々、本来の機能とは無関係な話題も話す。このようなノイズによって、ユーザは式神 AI に親しみをもち、また、式神 AI のサジェスションが不適當な場合も、ユーザはノイズの一種と区別をつける事無くがなく式神 AI に対する信頼度が低下する事をふせぐ効果がある。

ユーザの好みにより、画面に常時出現しない(顔を持たない)UI エージェントも選べるように、我々が開発提供しようと考えている。

5.5 式神 AI の各機能の実現について

5.5.1 Web ページの自動検索

Web ページの自動検索と先行取得は、常時接続でない携帯マシンの場合、非常に価値が高い。また、式神 AI は、忙しいユーザに代って、友人や、ユーザの興味ある Web ページを監視し、その更新を検出し、ユーザに通知する機能をもつ。

Web ページ自動検索は、大きく 2 つの機能からなる。一つは、ユーザの興味ある Web ページの推測であり、もう一つは Web ページ検索部分である。

ユーザの興味ある Web ページの推測は、ユーザが Web ブラウザでアクセスしている Web ページの URL と内容を覗きみて、その傾向を分析する事で実現する。ユーザがアクセスしている Web ページを覗き見るた

めに、携帯マシン内部で式神 AI のための HTTP プロキシを起動し、ユーザにはそれを使用させる。取ってきた Web ページの内容を解析するために、式神 AI の自然言語処理エンジンを使用する。これらの解析があまりに大変な場合は、式神 AI は処理の一部をバックエンド AI に依頼する事がある。また、PIM 内のアドレス帳データベースを監視し、ユーザの親しい友人の URL などを自動的に取得し、それらのページの更新を監視する。

Web ページ検索は、主に常時接続されているサーバ計算機内の、バックエンド AI エンジンが行う。バックエンド AI は、式神 AI に依頼されたサーチ内容にしたがって、メタサーチ・エンジン技術を使用して、Web ページの先行取得を行う。バックエンド AI は、式神 AI の推論機能の一部を補助する機能も持つ。

非-常時接続環境での AI 技術による Web ページ推測とその先行取得は、式神 AI が初めての提案だと思われる。

5.5.2 重要なメールの推測とその到着の通知

重要なメールの推測は、最近のユーザの送信相手の監視と、PIM 内のアドレス帳データベースを参照して行う。

式神 AI は、メール・サーバに接続する機会があれば、POP(やIMAP など) プロトコルでメール・ヘッダ情報を取得し、その送信相手中に、式神 AI が重要であると判断するものがあれば、UI エージェント機能を使用して、ユーザに通知する。

5.5.3 行動予定の自動補完

式神 AI は PIM 内のユーザのスケジュールや、ユーザの普段の行動を監視し、学習推論機能により、ユーザの行うべき行動を補完する。

例えば、ユーザが常に周期的にあるスケジュールに登録しているのに、たまに登録が行われていない場合、式神 AI はスケジュールの存在の可能性を指摘する。また、ある曜日のある時間にユーザがよく、特定のアプリケーションを起動していた場合、式神 AI はユーザに代って、そのアプリケーションを起動したりする。

これらの行動補完機能は学習、推論がうまくいった場合、非常に便利に動作する。

5.5.4 リモコンの自動制御

式神システムには電器製品のリモコン機能がある。式神 AI は、ユーザが行うリモコン操作を監視する。そして、ユーザの行動を学習し、推論する。

例えば、毎週特定の時間に特定の TV 番組を見てい

る事が推論できた場合、式神 AI はその時刻になると、自動的に TV のチャンネルを合わせようと試みる。

Bluetooth などの無線リモコンが一般的になると、式神 AI のこの機能は非常に効果を発揮するようになる。

6 おわりに

本稿では、携帯機器向けの GUI 環境「式神」について説明した。「式神」は、(1) デスクトップ環境、(2) 手書き文字認識システム「布目」、(3) アニメーション・ツールキット、(4) 人工知能機能、(5) 標準アプリケーションから構成されている。これらは、オープン・ソースとして開発する予定であり、本稿で述べた仕様は、その最初の提案である。今後は、さらに詳細な設計を行ない、来年の夏までにベータ版を公開する予定である。

「布目」の文字認識率向上には、多くの人の文字サンプルが必要である。そこで、標準辞書の作成は、別のオープン・プロジェクトとして、いろいろな用途に利用可能なものとして開発したいと考えている。興味のある方は、協力をお願いできればありがたい。

参考文献

- [1] 加藤 寧, 安倍正人, 根元義章: 改良型マハラノビス距離を用いた高精度な手書き文字認識, 電子情報通信学会論文誌 Vol.J79-D-II No.1 ('96/1)
- [2] 趙鵬, 佐藤幸男, 吉村ミツ: オンライン走り書き文字認識における汎用辞書の作成, 情報処理学会論文誌 Vol34 No.3 ('93/3)
- [3] Synchronized Multimedia Integration Language (SMIL) Boston Specification, W3C Working Draft 22, June 2000
- [4] Designing Characters for Microsoft Agent, msdn online Web Workshop, <http://msdn.microsoft.com/msagent/>
- [5] 長尾確編著: エージェントテクノロジー最前線, pp15-pp17 共立出版, 2000/Jul