

ウェアラブルコンピューティングのための その場プログラミング環境

秋山翔太郎^{†1} 寺田 努^{†1,†2} 塚本昌彦^{†1}

ウェアラブルコンピューティング環境ではユーザ状況が絶えず変化するため、ユーザがその場の状況に応じてサービスを定義できるシステムが必要となる。しかし、デスクトップ環境でのアプリケーション開発手法をウェアラブルコンピューティング環境に適用するのは難しい。そこで本研究では、ユーザがその場の状況に応じて必要とするサービスを、容易にかつ短時間でプログラミングできるシステム On-site Development Tool を提案する。提案システムでは、スクリプトを意味単位に分割したものをアイコンとして提示し、それらの中から必要なものを順番に選択していくことによってプログラミングを行う。提案システムはユーザが過去に定義したサービスを利用して、新しいルールを生成する遺伝的アルゴリズム機能やユーザと関係のある人が定義したルールを取り込むソーシャルネットワーク機能をもつ。

An On-site Programming Environment for Wearable Computing

SHOTARO AKIYAMA,^{†1} TSUTOMU TERADA^{†1,†2}
and MASAHICO TSUKAMOTO ^{†1}

In wearable computing environments, it is difficult for users to prepare applications that are used beforehand since there are various situations and places. Therefore, they want to define new services by themselves. However, there are many problems to use service developing methods for current desktop environments in wearable computing. In this study, we propose On-site Development Tool that we can define context-aware services easily and quickly. It employs a simple model where a user can define a new service by a simple operation such as selecting an event-driven rule indicated by an icon. The proposed system has two functions that to generate new services using genetic algorithm from existing services and to get services from his/her friends on social networks.

1. はじめに

携帯電話などの小型端末を持ち歩くことが一般的になり、ユーザが計算機を常に身に付けて利用するウェアラブルコンピューティングに対する注目が高まっている。

ウェアラブルコンピューティング環境では、ユーザのおかれている状況が絶えず変化するため、前もって定義しておいたサービスだけではコンピュータを様々な状況に応じた形で利用する事が難しい。そのため、その場で新しくサービスを定義しコンピュータで実行したいという欲求が生じる。ここでいうサービスとは、ウェアラブルコンピューティング環境においてユーザの操作や状況に応じて、定義された処理を実行するものである。しかし既存システムでは、ウェアラブルコンピューティング環境でのアプリケーションの構築が容易ではなく、その場で逐次新しいサービスを定義したりサービスを改変することは現実的ではなかった。

そこで本研究では、ウェアラブルコンピューティングにおけるユーザ状況を考慮したプログラミング環境 On-site Development Tool を提案する。提案システムでは、ウェアラブルコンピューティングにおいてサービスの定義を可能にするため、サービスを構成するスクリプトの文字列を特定の意味単位で区切り、必要な引数などと共にアイコンとしてまとめたものを提示する。それらの中からユーザが必要なものを選んで配置していくことによって、簡単な操作で自由なサービス定義を実現する。また、ウェアラブルコンピューティング環境におけるユーザは、いままで使用したことのない全く新しいサービスより、以前定義したサービスに似たサービスを必要とすることが多いと考えられるため、ユーザが過去に定義したサービスを利用して新しいルールを生成する遺伝的アルゴリズム機能と、ユーザと関係のある人が定義したルールを取り込むソーシャルネットワーク機能をもつことで、ユーザ自身が複雑なプログラムを定義しなくても利便性の高いサービスを利用できるようにする。

本研究では遺伝的アルゴリズム機能について評価を行い、過去に定義したサービスを元に遺伝的アルゴリズムで生成したサービスがある程度有用なものであることを示した。

本論文は以下のように構成されている。2章では関連する研究について述べ、3章で提案システムの設計について述べる。4章でシステムの実装について説明し、5章では提案シス

^{†1} 神戸大学

Kobe University

^{†2} 科学技術振興機構さきがけ

JST PRESTO

テムの評価を行う。最後に6章で本論文のまとめを行う。

2. 関連研究

2.1 ビジュアルプログラミング

大量の文字入力を必要としないプログラミング手法として、ビジュアルプログラミングと呼ばれるプログラミング手法がすでに数多く提案されている¹⁾。ビジュアルプログラミングではテキストでプログラム要素を記述するのではなく、テキストやグラフィックシンボルなどで表現されたプログラム要素を、空間に配置することでプログラミングを行う。

ESPranto SDK²⁾は、実世界にあるタンジブルインターフェースをプログラム内でオブジェクトとして取り扱うようなプログラミングを行える。必要なプログラミング要素をパズルのように組み合わせることで、ぬいぐるみを感じると鳴き声が再生されるようなプログラムを簡単に作成できる。また、LEGO MINDSTORMS³⁾は、レゴで作ったロボットの動作をビジュアルプログラミングで定義できる。

プログラム要素の配置ではないビジュアルプログラミングとしては、原田によるViscuit⁴⁾がある。Viscuitでは絵を動かすためのプログラムを、GUIを用いて絵とそれに関する動きのルールを定義することで作成できる。

しかしこれらの研究は、プログラムの可視化によって得られる利便性や、GUIによる操作の簡易化によりエンドユーザのプログラミング活動を支援することが目的であり、ウェアラブルコンピューティングでの使用を想定している本研究とは目的が異なる。また、GUIを用いること自体はウェアラブルコンピューティングにおけるプログラミングを単純化することに直接は寄与しない。

2.2 モバイルコンピューティングにおけるプログラミング

コンピュータを持ち運ぶという点において、ウェアラブルコンピューティングとよく似ているモバイルコンピューティングにおいても、その特性を考慮したプログラミング手法が提案されている。BLOCCO⁵⁾は携帯電話でよく使うサービスをIF-THENルールで定義し、実行できる。BLOCCOでは、IF-THENルールの定義は上級者に向けたエキスパートモードの一機能として提供されており、オススメのルールを実行できる初心者モードだけでもある程度の機能を利用できるように設計され、プログラミングの知識をもたないエンドユーザの利用を強く意識した設計となっている。提案システムも、上記問題点の解決とエンドユーザにも簡単に利用できるプログラミングシステムを目指す。

2.3 対話型遺伝的アルゴリズム

遺伝的アルゴリズムは様々な分野で活用されている。遺伝子を交配し進化させることで最適解に収束させるのが一般的な利用方法だが、最終的な解が無数に存在すると考えられる場合はアルゴリズムにより生成された次の世代に対して人間による操作を加えることで、より目的に近いものを生成していく対話型の利用方法が多数提案されている。

LiveCraft LOGO3⁶⁾は提示された複数のロゴからユーザが気に入ったものを選ぶことで、次の提示では選択されたロゴの遺伝子をもつものが優先され、ユーザの感性にあうロゴが提示されていくシステムである。他にもインタフェースのボタン配置⁷⁾やヘッドホンの音質調整⁸⁾など、普遍的な最適解が存在せず各個人の個性や感性によって最適な解が変わる場合に多く用いられている。提案システムでも、プログラミングの簡易化と各個人に合わせたルール提示を両立するための手法として対話的遺伝的アルゴリズムを用いる。

2.4 ソーシャルネットワークサービス

近年、個人の間関係がインターネット上に表れるソーシャルネットワークサービスに注目が集まっている。代表的なソーシャルネットワークサービスにはmixi⁹⁾やfacebook¹⁰⁾などがある。

一般にソーシャルネットワーク上の友人の間には、実世界での友人関係だけでなく、興味や趣味などの関連性があることが知られている¹¹⁾。そして、ソーシャルネットワークによる情報の伝搬には有効なフィルタリング効果があることがわかっており¹²⁾、一定に達していない品質の情報は伝搬しづらい。多数の人々がウェアラブルコンピュータを身につけそれぞれに必要なサービスをプログラミングするようになった場合にも、自分とつながりのある人物の作ったプログラムは自分にとって有用性が高いものである可能性が高いと考え、提案システムではソーシャルネットワークを利用したサービス提示を行う。

3. システム設計

本研究では、ウェアラブルコンピューティングのユーザが必要とするサービスをいつでもどこでもプログラミングすることを「その場プログラミング」と呼ぶ。ウェアラブルコンピューティング環境において「その場でプログラミングする」例としては以下のようなものが考えられる。

- 会合に来てみると初めて会う人がたくさんいそうなので、誰かと握手をした時に装着型カメラで自動的に相手の写真を撮って保存することで誰と会ったか覚えておけるようにした。

- 本屋に来て立ち読みをしたところ、この本を購入したいと思ったが手持ちのお金がない。次にこの本屋の前を通りかかった時にはこれを買うことを思い出すようリマインダを表示させることにした。
- 迷いやすい場所に向かうので頭部装着型ディスプレイ上に地図を表示していたが、常に表示されていて邪魔になったので、立ち止まった時だけ表示するようにした。
- 座りっぱなしは体に悪いので、2時間以上座っていたら体を動かすようにリマインダを表示させることにした。

このようなプログラミングでは、一般になんらかのイベント(握手をする、本屋の近くを通りかかるなど)をきっかけとして、なんらかの動作(写真を撮る、リマインダを提示するなど)を行うことが想定されているため、イベント駆動型のプログラミングモデルが適している。

また、ウェアラブルコンピューティングでは、ユーザが常に画面を注視して操作に集中できるとは限らない。ユーザ自身の状況として、立ち止まっている、歩いている、走っている、寝転んでいる、電車に乗っている、車を運転している、など様々なものが考えられ、それぞれの状況によってユーザの画面に対する注意力やユーザに可能な操作の量、複雑さなどが異なると考えられる。そのため、マウスやキーボードのようなデスクトップコンピューティング環境で一般的な入力装置は、ある程度のスペースを確保する必要があるため使用できないと考えるべきである。

さらに、ウェアラブルコンピューティングはユーザの日常生活との結びつきが強いいため、必要となるサービスも高度な情報処理を行うものより、ユーザの周囲状況を取得して何らかの情報を出力するようなシンプルなものが多く、少しだけ条件の異なる類似サービスを多数の人が利用することも考えられる。加えて、それらを必要とするユーザは、プログラムや情報処理の専門的な知識をもっていないことも十分考えられるため、プログラミングに関する知識に乏しいエンドユーザによるプログラミングも想定する必要がある。一般にエンドユーザはそのシステムで何ができるかわからないため、「どんなサービスが可能で何をプログラミングしたらよいのか」という初期段階で躓くことが多い。したがって、いきなりプログラミングをはじめめるのではなく既存サービスを参考にしながらいかにうまく求めるサービスに近づけるかを考える必要がある。

3.1 要求事項

これまでに述べたその場プログラミングの特徴を考慮すると、ウェアラブルコンピューティングにおけるプログラミング環境は、以下のような条件を満たすことが要求される。

- (1) イベント駆動型のプログラミングモデル
ユーザの「こういう状態になった時、こうしたい」という要求をそのまま記述できる必要がある。
- (2) 状況変化に柔軟に対応できるサービス定義
状況に応じてサービスの必要性が変化するため、サービスの稼働状態をそれに合わせて柔軟に変更できる必要がある。また、ウェアラブルシステムでは多数のサービスが並行動作しているため、頻繁に起こるサービス書き換えの度にシステムが停止してはいけない。
- (3) 簡易で少ない操作量
思いついた時にすぐサービスを必要とする上に、マウスなど高性能な入力装置は使えないため、ユーザによる操作は少数のボタン程度で済み、なおかつ操作量は少なくすることが望ましい。
- (4) 既存サービスを活用したサービス定義
必要とされる頻度が高いサービスは簡易によく似たパターンで構成されると考えられるため、新規にサービスを作成するより、既存サービスの改変や他人が定義したサービスの取り込み機能を充実させ、サービス定義を容易にする必要がある。

これらの要求を満たすため、提案システムは定義されたルールに基づきサービスを実行するエンジンである Wearable Toolkit^{(13),(14)} と、ユーザがサービスを開発するツールである On-site Development Tool の 2 つから構成される。

3.2 システム構成

提案システムは図 1 に示すように、定義されたサービスを実行するエンジン部分の Wearable Toolkit と、ユーザが実際に操作してサービスを定義するインタフェース部分の On-site Development Tool とで構成される。以下、それぞれについて概説する。

3.2.1 Wearable Toolkit

3.1 節で述べたようにウェアラブルコンピューティングにおけるプログラミング環境では、ユーザ状況の取得やその変化にあわせてサービスの実行が求められる。そこで提案システムでは、ウェアラブルコンピューティングにおけるプログラミングを念頭においたアプリケーションプラットフォームである Wearable Toolkit を利用する。

Wearable Toolkit の動作は、ある事象 (Event) が生じた時に呼び出され、変数の値などが条件 (Condition) に適合するか調べ、適合した場合は操作 (Action) が実行される、というように定義される ECA ルールで記述する。

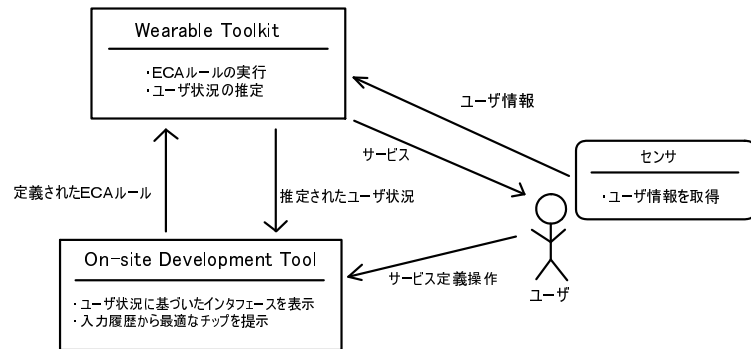


図 1 システム構成

Event, Condition, Action に記述できる内容は外部のプラグインによって定義されており、外部プラグインに関する仕様が公開されているため、将来ウェアラブルコンピューティングに有用な新しいデバイスが開発されても、デバイスの動作に必要な処理を外部プラグインとして実装すれば、システムの全体に変更を加えることなくデバイスを使用できる。

また、プラグインやルールは外部ファイルからの読み込みや外部プロセスとの通信によって、起動中でも動的に変更できるため、状況変化に対するルール変更をシステムを停止させることなく行え、3.1 節で述べた要件 1, 2 を満たす。

3.2.2 On-site Development Tool

On-site Development Tool は、ウェアラブルコンピューティングにおいて困難なテキスト入力を用いなくてもプログラミングが行えるように、ECA ルールにおける、Event, Condition, Action それぞれを単位とするアイコンとして表したものを、カーソルキーで選択することによりプログラミングを行う。

On-site Development Tool はアイコンの配置を行うルール定義モード、ユーザが過去に定義したルールから新しいルールを生成する遺伝的アルゴリズムモード、ユーザと友人関係にある人が定義したルールを取り込んで利用するソーシャルネットワークモードの 3 つの機能をもつ。それぞれの画面間は上下カーソルキーの長押しによって移動できる。以下それぞれを詳しく説明する。

3.2.3 ルール定義モード

On-site Development Tool のルール定義インタフェースを図 2 に示す。

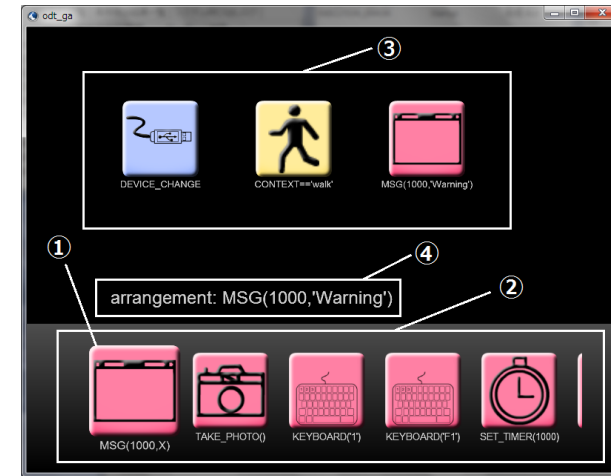


図 2 ルール定義モード

現在選択されているアイコンは ① のように、他のアイコンよりやや大きく表示されている。② のアイコン提示エリアには Event, Condition, Action いずれかの種類のアイコンが横に並べて提示されており、上下キーで種類の選択し、左右キーでアイコンを選んでエンターキーで ③ に配置する。タイマの秒数などアイコンが引数をもつ場合、それぞれのアイコンにはあらかじめデフォルト値が定義されているが、整数値の引数に限り配置時に上下左右のカーソルキーで値の変更が可能である。定義したい ECA ルールに必要なアイコンを全て配置し終わったら、エンターキーの長押しによって対応する ECA ルールが出力される。同時に、定義したルールがルールアイコンとして Event 行に並べられる。これによって、以前定義したルールを構成するアイコン全てを一度の選択操作で配置でき、ルールの改変を効率的に行える。④ にはアイコンの詳細などシステムメッセージが表示される。

また、提案ツールは、宮前らが実装したジェスチャ登録・認識機能をもつ ContextTool(図 3) と連携する機能をもっており、ContextTool で新しいジェスチャを登録すると、そのジェスチャを表す Event アイコンが Event 行に自動的に追加されるため、ルールのトリガとしたい動作をその場で自由に登録して利用できる。アイコン配置やルール出力操作は左カーソルキーの長押しで取り消し、右カーソルキーの長押しでやり直しができる。

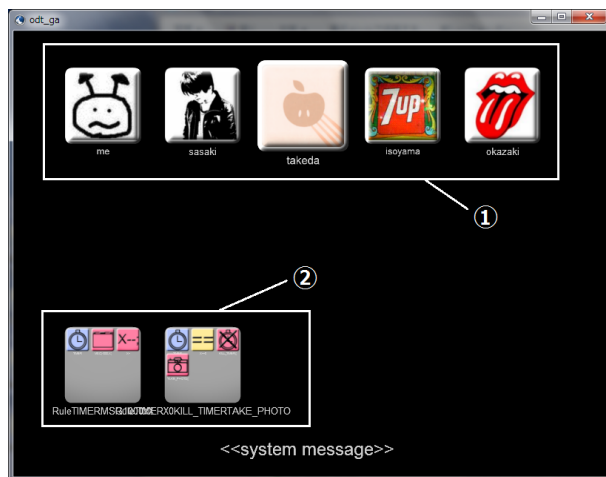


図 5 ソーシャルネットワークモード

2つのルール間における類似度を、2つのルールのどちらにも含まれるアイコンの枚数の2乗と定め、各友人の定義したルール全てについて、自分の定義したルールとの類似度を計算し、その合計が高い友人から順に提示する。

ソーシャルネットワークモードにより取得したルールは自分で定義したルールと同様に扱われ、3.2.4節で述べた遺伝的アルゴリズムモードの親ルールとしても用いることができる。

4. 実 装

3章で述べたウェアラブルコンピューティングにおけるプログラミング環境のプロトタイプを実装した。

プロトタイプでは Wearable Toolkit の起動時に呼ばれるルールによって、On-site Development Tool の起動が行われる。これらのソフトウェア間の接続はそれぞれがシリアルポートに対して接続を行い、それを橋渡しすることで行った。この時、On-site Development Tool 側からシリアル接続するために必要となるソケット・シリアルプロキシとして、Flash Net Comport Connector を利用した。また、コンピュータ上の2つのシリアルポートを接続するために、仮想シリアルポートソフトとして com0com を利用した。

これらの接続関係によって、ユーザが On-site Development Tool 上でアイコン配置を

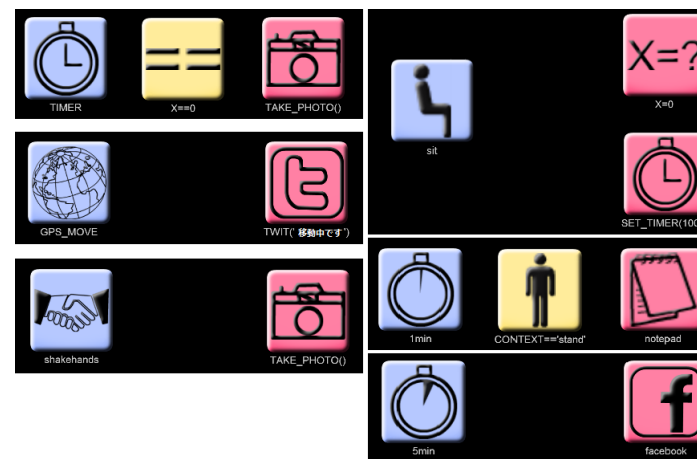


図 6 ルール定義例

を行い、出力操作を行うことによってECAルールが書き出されると、Flash Net Comport Connector にソケットを用いて通知を行い、Flash Net Comport Connector がシリアルで通知を行う。通知は com0com によって橋渡しされ、Wearable Toolkit が受け取る。通知を受け取った Wearable Toolkit が書き出されたスクリプトを新しいルールとして読み込み、実行することでウェアラブルコンピューティングにおけるプログラミング環境を実現した。システムの操作には4方向のカーソルと決定の入力があればよく、小型のテンキーやジョイスティック、ゲームパッドなどを想定している。

図6に提案システムによるルール定義の例を示す、上から「タイマーが0になったら写真を撮る」「GPSの値が変わった時に移動中だとTwitterに投稿する」「握手をしたときに写真を撮る」「座ったらカウントを0にしてタイマーをスタートし、座っている時間を計測する」「1分の空き時間があったら、メモ帳を起動してアイデアをメモする」「5分の空き時間があったら、facebookをチェックする」というサービスである。特に最後2つの例は、ウェアラブルコンピューティングを用いて時間を有効活用する例で、ユーザが現在空き時間であることをシステムに伝えると、やるべきことをシステムが教えてくれる。

5. 評 価

実際のユーザの利用を想定したシナリオに従って遺伝的アルゴリズムを適用することで、

生成されるルールの評価を行った。

想定シナリオ 1

ウェアラブルコンピュータを身につけたユーザが散歩に出かけることを想定する。ユーザは散歩中に見つけた物の写真を撮るために、「ボタンを押した時に写真を撮る」ルールを定義している。また立ち止まっている時間を計測するために、「立ち止まったらカウントを0にしてタイマーをスタートする」、「タイマーが動いている間カウントを増やす」ルールを定義し、他にも、「走っている時はシステムを一時停止しておく」、「現在地が移動した時に現在の天気情報を見る」ルールの合計5つのルールを定義しているとする。

この時、提案システムの遺伝的アルゴリズム機能を用いて、ユーザによる優先度の操作を行わずに新しいルールを生成すると、「走っている時にカウントを増やす」、「ボタンを押した時にカウントを0にしてタイマーをスタートする」、「ボタンを押した時に写真を撮ってタイマーをスタートする」、「立ち止まった時に写真を撮ってタイマーをスタートする」という新しいルールが生成された。これらのルールの内、2つ目を利用すると、動作しているタイマーの数値が何かの拍子に異常値を示した場合に手動でリセットできるようになる。4つ目を利用すると、散歩中に何かを見つけて立ち止まった際に自動的に写真を撮るシステムができる。

このようにユーザが構成したルールに類似したルールを提示することで、最初にユーザが必要だと考えたシステムの骨格を維持しながら、ウェアラブルコンピュータをより便利に利用するためのルールを生成できた(図7)。

想定シナリオ 2

ウェアラブルコンピュータを身につけたユーザが、多くの人が集まる会合に出向くことを想定する。多くの人に会い、話をする機会になるため、誰とどのような会話をしたのかをある程度記録しておきたい。そこで、「握手をした時に30秒間の録音を始める」、「録音を始める時にメッセージを表示する」、「録音が終わった時にもメッセージを表示する」ルールを定義し、会話を録音しておく。また、「手を挙げた時にメモ帳を起動する」ルールを定義することで、会話を終えた時に印象に残ったことをメモできるようにしておく。その際に気になったことをその場で調べたくなることも考えられるため、「その場で足踏みをしたら検索用のテキストボックスを表示する」ルールも定義しておく。そして、記念写真を撮るための簡易タイマー付きカメラの役目を果たすため、「ボタンを押したらカウントを10にしてタイマーをスタートする」、「タイマーが動いている間カウントを表示しながら1づつ減らす」、「カウントが0になった時にタイマーを止めて写真を撮る」というルールを定義していると想定

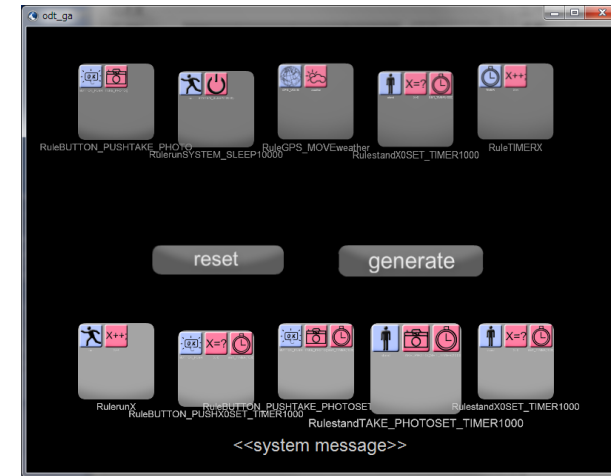


図7 想定シナリオ 1: ルール生成の様子

する。

ユーザはもう少し便利に写真を撮るサービスが必要だと考えて、サービスの具体的なイメージをもたずに「写真を撮る」アイコンが含まれている最後のルールの優先度を1だけ上げて、遺伝的アルゴリズムによりルールを生成する(図8)。その後「写真を撮る」アイコンが含まれているルールの優先度を上げながら生成を繰り返すと、出てくるルール全てに写真を撮るアイコンが含まれるようになった。

「録音を始めるときに写真を撮る」や「ボタンを押したときに写真を撮る」のような新しいルールは生成されたが、特に利便性を感じなかったためもう少し生成を続けると、遺伝的アルゴリズムの突然変異によって「写真を撮る」アイコンが置き換えられた「ボタンを押した時にツイッターにつぶやく」ルールが生成された。このルールをそのまま利用してもあまり便利だとは言えないため、ユーザはこのルールのルールアイコンを改変して「ボタンを押した時に写真を撮りツイッターにつぶやく」というルールを定義した。

このように提案システムの遺伝的アルゴリズム機能を利用し何世代かに渡って優先度を設定することで、「写真を撮る」のような必要としている部分を保ちながら新しいルールを提示することができる。また、突然変異によってユーザが思いつかなかったルールを生成でき、そこからヒントを得ることによってより便利なルールを構成できた。

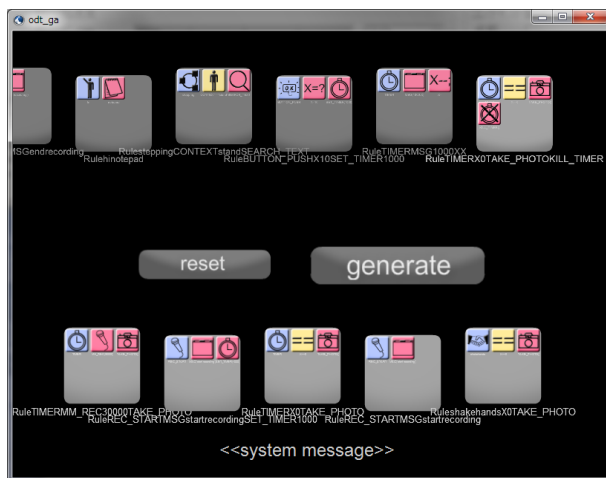


図 8 想定シナリオ 2: 優先度を上げてルール生成の様子

6. ま と め

本論文では、ウェアラブルコンピューティングにおけるユーザ状況を考慮したプログラミング環境を提案した。提案システムを利用することで、ウェアラブルコンピューティング環境において、少数のキーのみを使用した少ない操作でプログラミングを行えた。また、ユーザがプログラミングを意識することなく必要なサービスを定義できる、遺伝的アルゴリズム機能とソーシャルネットワーク機能を提案した。

評価により、過去に定義したルールから遺伝的アルゴリズムを用いてルールを生成する手法により、ある程度有用なサービスが得られることを示した。

今後の課題として、現状のインターフェースだけでは多数のルールを取り扱うのが難しく、現在登録、実行されているルールの一覧や自由な変更、削除ができるインターフェースの考案が必要だと考えられる。

謝 辞

本研究の一部は、科学技術振興機構戦略的創造研究推進事業(さきがけ)および文部科学省科学研究費補助金基盤研究(A)(20240009, 23240010)によるものである。ここに記して

謝意を表す。

参 考 文 献

- 1) 増井俊之: GUI ベースのプログラミング, 平川, 安村(編), ビジュアルインタフェースポスト GUI を目指して, 第 2.2 章, pp. 45-64 (1996).
- 2) R. van Herk, J. Verhaegh, and W. F. J. Fontijn: ESPranto SDK: an Adaptive Programming Environment for Tangible Applications, *Proc. of the 27th International Conference on Human Factors in Computing Systems (CHI '09)*, pp. 849-858 (2009).
- 3) レゴマインドストーム: <http://www.legoeducation.jp/mindstorms/>.
- 4) 原田康徳: アニメーションで簡単にプログラムを学ぼう, 電子情報通信学会誌, Vol. 87, No. 8, pp. 674-677 (2004).
- 5) BLOCCO: <http://www.blocco.jp/>.
- 6) Bisca/LiveCraft LOGO3: <http://www.livecraft.co.jp/Logo3.htm>.
- 7) 米田有佑, 田中美里, 廣安知之, 佐々木康成, 三木光範, 横内久猛: 対話型遺伝的アルゴリズムのインターフェースにおけるボタン配置が評価に与える影響の検討, 情報処理学会研究報告, Vol. 136, No. 15, pp. 1-5 (2010).
- 8) 中村広美, 梶川嘉延, 野村康雄: IGA を用いたヘッドホン受聴のための音質フィッティングシステムの構築とその有効性の検証, 電子情報通信学会技術研究報告, Vol. 107, No. 470, pp. 7-12 (2008).
- 9) ソーシャル・ネットワーキング サービス [mixi(ミクシィ)]: <http://mixi.jp/>.
- 10) Facebook - フェイスブック: <http://www.facebook.com/>.
- 11) 松尾 豊, 安田 雪: SNS における関係形成原理 mixi のデータ分析, 人工知能学会論文誌, Vol. 22, No. 5, pp. 531-541 (2007).
- 12) 竹内 亨, 寺西裕一, 春本 要, 下條真司: ソーシャルネットワークに基づいた情報伝播型コミュニケーションの実証実験による有効性評価, 情報処理学会論文誌, Vol. 47, No. 2, pp. 555-565 (2006).
- 13) Wearable Toolkit Home Page: <http://wearable-toolkit.com/>.
- 14) 寺田 努, 宮前雅一, 山下雅一: Wearable Toolkit: その場プログラミング環境実現のためのイベント駆動型ルール処理エンジンおよび関連ツール, 情報処理学会論文誌, Vol. 50, No. 6, pp. 1587-1597 (2009).