

その場プログラミング環境の実現に向けて

寺田 努[†]

宮前雅一[‡]

[†] 大阪大学サイバーメディアセンター

[‡] 株式会社国際電気通信基礎技術研究所

あらまし：近年の技術発展により、コンピュータを装着して常時利用するウェアラブルコンピューティングが実現可能になりつつある。ウェアラブルコンピュータを用いて常時サービスを受けるようになると、そのサービス定義をいつでもどこでも行いたいという要求が生じる。そこで本研究では、ウェアラブルコンピューティング環境において、状況依存のアプリケーションを容易に構築するための枠組みを提案する。提案する枠組みは Wearable Toolkit と呼び、イベント駆動型のルールエンジンおよび関連するツール群からなる。この枠組みを用いることで、いつでもどこでもシステムの動作を止めることなくサービスの定義・削除・改変等の操作を行えるようになる。実際に提案システムを用いて状況の定義を行う予備実験を行った結果、一般ユーザでも状況依存のアプリケーションが構築できる一方、改善すべき点も多く残っていることがわかった。

Toward Realizing On-site Programming Environments

Tsutomu TERADA[†] and Masakazu MIYAMAE[‡]

[†] Cybermedia Center, Osaka University

[‡] Advanced Telecommunications Research Institute International

Abstract : Wearable computing environments, where a user wear a computer anytime and anywhere, are slowly becoming a reality because of the recent technological advancements. When a user acquires various services in wearable computing environments, the user wants to define a new service by himself. Therefore, in this research, we propose a new framework for constructing context-aware applications in wearable computing environments. Our framework is called Wearable Toolkit, which consists of an event-driven rule processing engine and tools for developing applications. By using our framework, we can define, delete, and customise services anytime and anywhere. From the result of pilot study, although general users can construct context-aware applications, there are several remaining problem for easy construction of them.

1 はじめに

近年の技術発展や機器の小型化により、人々は携帯電話等の情報機器や PC を常時持ち歩くようになった。最近では、コンピュータを衣服のように装着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りつつあり、常時ユーザの行動を測定して健康管理を行うシステム^[1]や、バイクレースにおいて常時ユーザに情報を提示し続けるシステム^[9]、状況の変化に応じて旅行者のためのナビゲーションを行うシステム^{[2][8]}、看護婦の行動を認識してヒヤリハットをなくすためのシステム^[10]などウェアラブルコンピュータを活用したさまざまなシステムが研究開発されている。例えば図 1 に示すバイクレース支援システムでは、バイクチームの監督がウェアラブルシステムを装着することで、戦略支援情報やトラブル情報を他人に見られることなく閲覧できる環境を実現している。ウェアラブルコンピュータを用いることで、従来の持ち歩き可能な機器と比較



図 1: ウェアラブルシステムのバイクレースでの活用

してユーザの生活により密着したサービスが提供できるようになりつつあり、ウェアラブルコンピューティング普及を考慮した新たなサービス提供モデルやプログラミングモデルが求められている。

本研究では特に、ウェアラブルコンピューティング環境におけるエンドユーザプログラミングについて考え

る。ウェアラブルコンピュータを活用してサービスを提供する場合、たとえば「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ときにはこれを買うことを忘れないようにリマインダを提示させよう」といった要求が考えられる。このような要求は、その条件（次にこの本屋の前を通りかかる）や動作（リマインダを表示する）がその場で決定されるため、その場でそれらの要素をうまく記述し、システムに登録するメカニズムが必要となる。一方、このようなサービスの利用者・作成者は、一般ユーザを中心としてプログラミングに馴染んでいない者が多く、複雑な条件記述を行わせることは難しい。

本研究では、このような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を行える環境の実現を目指し、筆者らが開発を進めている Wearable Toolkit および関連ツール群について紹介し、その場プログラミング実現に向けての予備実験およびシステム構築について述べる。

以下、2章ではその場プログラミングを具体的な例を挙げながら説明し、3章でその場プログラミングを実現する環境を提供する Wearable Toolkit および関連ツールについて述べる。4章ではその場プログラミングにおいて特に重要な、その場コンテキスト定義機能および予備実験について述べ、最後に5章で本研究をまとめる。

2 その場プログラミング環境

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」例としては下記のようなものが挙げられる。

- 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ときにはこれを買うことを忘れないようにリマインダを提示させることにした。
- 迷いやすい場所に向かうので地図を表示したいが、常に表示されていると邪魔なので、立ち止まったときだけヘッドマウントディスプレイ上に地図を表示するようにした。
- ウェアラブルコンピュータを用いて服の点滅を制御するダンスパフォーマンスの練習中、回転時に服を派手にすれば目立つのではないかと思い、回転時の点滅パターンを変更した。
- ジョギングに出かけたが、医者に激しい運動を止め

られていることを思い出し、心拍が高かったり早く走っている期間が1分以上続いたらアラートを出すようにした。

- ある人からメールが届いてヘッドマウントディスプレイ上に表示されたが、この人からのメールは大事ではないので以降ヘッドマウントディスプレイには表示しないようにした。
- 陸上の練習を監督しているときに、手を振り下ろしてから振り上げるまでの時間を計測するようにして、ランナーのラップタイムをジェスチャで測れるようにした。
- 今月はちょっと浪費気味なので、財布を出すたびに「無駄遣いするな」と表示することにした。
- 妙にうなずく癖があると言われたので、今日は一日うなずく回数をカウントしてみることにした。
- バイクレースを観戦しているとき、あるバイクの周回ごとのラップタイムをヘッドマウントディスプレイに一覧表示して状況が把握しやすくなった。

以降、上記各アイテムのようにアプリケーションの一部を構成する機能をサービスと呼ぶ。このようなサービスをウェアラブル環境においてプログラミングするためには、システムに下記の要素が必要となる。

- (1) サービスを容易に記述できるプログラミングモデル
- (2) システム稼働中の動的なプログラム追加機能
- (3) コンテキストのその場定義機能

要求(1)は、ウェアラブル環境においてあらゆる場所でサービス定義を行うため、その場で簡単にサービス記述を行える必要があることを表す。上記のようなサービスは一般に、なんらかのイベント（本屋の前を通りかかる、立ち止まるなど）をきっかけとして、なんらかの動作（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、一般に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、複雑な構造をもつプログラミング言語はその場プログラミングには適しておらず、シンプルで簡単にサービス定義が可能な方式を採用する必要がある。要求(2)は、サービス記述を行っている際にシステム自体を止めないことが重要であることを表している。ウェアラブルシステムでは多数のサービスが並行して動作しており、重要なロギングやセンシングを行っている場合も多い。したがって、頻繁に起こるであろうサービス追加のたびにシステムの再起動を行うようではいけない。要求(3)は、サー

ビス開始のトリガとなる状況(以降コンテキスト)をその場で定義できる機能である。例で示したようなサービスにおいては、トリガとなる状況はバラエティに富んでおり、あらかじめ想定して用意しておくことは難しい。したがって、本屋の前を通りかかる、立ち止まる、といったコンテキストを現在の状況からその場で定義し、サービスのプログラミングにその場で活用できるような枠組みが必要となる。

本研究の目標は、この要求(1)～(3)を満たすウェアラブルシステムプラットフォームを構築することである。一方、ウェアラブルコンピューティング環境やユビキタスコンピューティング環境において、コンテキストウェアネスアプリケーションを構築するためのツールキットに関する取り組みはこれまでも数多く行われている。代表的なものとして、米国MITで進められているMIThrilプロジェクト^[6]は、ウェアラブルコンピューティングにおけるハードウェア・ソフトウェアプラットフォームの構築を目指したものである。コンテキストウェアシステムにおけるハードウェアの管理や、センサから抽出したデータの特徴量抽出およびコンテキスト認識を行う機能をAPIセットとして提供し、プログラマが容易にウェアラブルシステムを構築できる環境を目指している。またContext-toolkit^[4]は、センサデータからコンテキストを得る処理をセンサデータを直接取り扱いデータをカプセル化するcontext widget、複数のwidgetのデータを統合して抽象化するcontext aggregator、実際のコンテキスト計算を行うcontext interpreterの3層モデルを用いてコンテキスト認識を行うことで、使用するセンサの変化など末端のシステム変更の影響を吸収している。TEA System^[1]は、TEA Boardと呼ぶセンサが接続されたデバイスを持ち歩くことでデータを蓄積し、利用者の行動に応じてコンテキストのための閾値を自動生成する機構をもつ。LifePatterns^[3]は、100日間の自身の行動をカメラおよびジャイロセンサを搭載したウェアラブルコンピュータでモニタし、行動ログをある程度の数のコンテキストに自動的にクラスタリングする手法を提案した。R2システム^[13]では、コンテキストウェアアプリケーション構築からセンサデバイスを隠蔽することが目的であるが、センサデータからコンテキストの定義が可能となっている。LifePatternsは、実験を行った段階であり、システムやプラットフォームの提案は行われていない。また、そのほかのシステムは主な目的が、センサデータからコンテキストを抽出する部分の抽象化を行うことでアプリケーションプログラマの負荷を軽

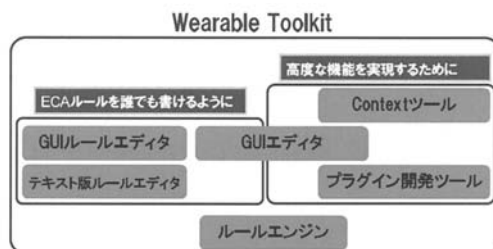


図2: Wearable Toolkitの構成



図3: ルールエンジンの構成

減することを目的としており、要求(1)～(3)は考慮されていない。

ウェアラブル/ユビキタス以外の研究分野においてもコンテキストの記述や抽象化は重要なテーマとなりつつあり、特にSemantic Web^[12]関連のプロジェクトにおいて、コンテキストの抽象化や分割などが大きなトピックとなっているが、ウェアラブルコンピューティングにおけるエンドユーザプログラミングへの適性は考慮されておらず、要求(1)～(3)も満たしていない。

3 Wearable Toolkit

筆者らは、前章で述べた3つの要求を満たすウェアラブルコンピューティングプラットフォームの構築を目指し、Wearable Toolkitプロジェクトを立ち上げた。Wearable Toolkitは図2に示す構成となっており、機能の動的追加が可能なルール処理エンジン+各種ツール群から成り立っている。

ルールエンジンは図3に示す構成をもつ。Wearable Toolkitにおけるルールエンジンは、筆者らがこれまでに開発したイベント駆動型ルール処理エンジンA-WEAR^[7]を拡張し、機能および使いやすさを高めたものである。ルールエンジンは現在WindowsOS上でミドルウェアとして稼動し、その動作は図4に示す構文に基づき、発生する事象(イベント)、実行条件(コンディ

```

DEFINE ルールID [IN 所属グループ名のコマンド区切りリスト] [FOR ルール適用範囲]
[VAR 変数名AS 変数の型]
WHEN イベントタイプ[イベント対象]
IF (左辺式演算子右辺式) (AND やOR を使用可能)
THEN
DO [実行値格納先=] アクションタイプ(アクション内容)
FOR EACH NEW.data
DO [実行値格納先=] アクションタイプ(アクション内容)
END FOR EACH

```

図 4: ECA ルール構文

表 1: 実装済みプラグインの例

プラグイン名	概要
システム情報プラグイン	ユーザの PC 使用状況取得
マルチメディアプラグイン	動画・音声の再生
GPS プラグイン	GPS による移動検出
地磁気センサプラグイン	向いている方向の変化の検出
シリアル通信プラグイン	シリアルポートでデータを送受信
カメラプラグイン	カメラによる動画・静止画の撮影
ネットワークプラグイン	ネットワークへのデータの送受信
メールプラグイン	メールの送受信
ブラウザプラグイン	ウェブブラウザの制御
デバッグプラグイン	デバッグ機能の付加
地図ビュープラグイン	地図の表示・制御
データベースプラグイン	アクティブ DB の機能を提供

ション), 実行する動作(アクション)の3つを一組とした ECA ルールにより記述する。ECA ルールを用いることで、ユーザは「こういう状態になったとき」「こうしたい」という要求をそのままイベントとアクションに記述すればよく、機能の追加・削除・変更も、それぞれ ECA ルールを追加・削除・変更することで行える。また、これらの変更操作はシステム稼働中に行うことが可能であり、前章で示した要件 (1)(2) を満たしたシステムとなっていると言える。イベントやアクションに記述できる内容は、プラグインにより自由に拡張可能である。例えば GPS のプラグインをシステムに導入すると、イベントとして「GPSMOVE」が使えるようになり、位置が移動したことをイベントとしてアプリケーションが記述できる。これまでに構築したプラグインの一部を表 1 に示す。プラグインは、C++, C#, VB.NET などさまざまな言語で記述できるようになっており、すでに存在するアプリケーションをプラグイン化することも容易である。

また、プログラマ向けのツールとして、アプリケーションの画面設計を支援する GUI エディタ (図 5)、プラグイン開発を支援するウィザードやテンプレートなどのツール群を用意している。さらに、一般ユーザがプログラミングすることも想定し、次章で述べるコンテキストツールや GUI 版ルールエディタなど、その場でプログラミングを行うことを容易にするいくつかのツールを提供している。ルール仕様の詳細やアプリケーション機能等についてはウェブサイト^[14]に掲載している。今後の予定として、Flash を用いた GUI 記述や

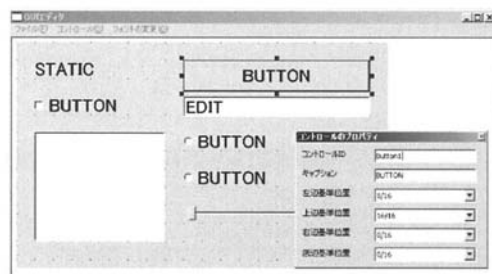


図 5: GUI エディタの画面

java への対応などさまざまな機能を追加する予定であり、プログラマからデザイナー、一般ユーザまでがウェアラブルアプリケーションを構築できるツールの提供を目指している。

4 コンテキスト定義機能

2 章で述べたように、その場プログラミングを行うためには、(1) サービスを容易に記述できるプログラミングモデル、(2) システム稼働中の動的なプログラム追加機能、(3) コンテキストのその場定義機能、の3つの要件がシステムに求められることを述べた。このうち、Wearable Toolkit のルールエンジンを利用することで、要件 (1)、(2) は実現できている。一方、2 章で挙げたような例を実現するためには、あらかじめ用意されたイベントだけでなく、その場で新たなコンテキストを表現するイベントを定義できる必要がある。

多くの従来研究においても述べられているとおり、コンテキストとはなんらかの「状況」を表すものであり^[4]、「歩行中」「自転車に乗っている」「起きている」「メール着信」「ある時点から 10 分後」などといったようにさまざまな表現が可能である。そしてこれらのコンテキスト、特にウェアラブルコンピューティングにおけるサービスで利用されるコンテキストは単一または複数のセンサの特徴量から決定される場合が多い。ここでセンサとは、GPS や加速度センサなどハードウェア的なセンサに加え、メール着信を検出するメールなど広い意味でのセンサを含む。特徴量とはセンサから出力されるそのままの値や、その値の一定時間における平均値・分散値などを表す。例えば「現在地が自宅」というコンテキストは具体的に、GPS から得られた緯度および経度の現在瞬間値が、あらかじめ登録されている自宅の緯度経度と一定値以上近いこと、と言い表せる。例として、表 2 にいくつかのコンテキストを得る際に用いられるセンサおよびその特徴量を示す。

表 2: コンテキストと用いるセンサおよび特徴量の例

コンテキスト	必要なセンサと特徴量
ある場所にいる	GPS から取得した位置の現在値
移動中	GPS から取得した位置の変化量
回転している	地磁気センサから取得した方向の変化量
立っている	加速度センサの一定期間での の平均値・分散値
歩いている	加速度センサの一定期間での の平均値・分散値
自転車に乗っている	加速度センサの一定期間での の平均値・分散値
ある人からのメール着信	メールから得られたイベントの現在値
財布をかばんから出す	RF-ID 読み取りの現在値
12 時ちょうど	時刻の現在値
ある時点から 10 分後	時刻の変化量
暑い	温度センサの値の現在値
暑くなってきた	温度センサの値の変化量



図 6: Context ツールの利用画面

4.1 コンテキストツール

前節に示したように、あるコンテキストを定義するためには、装着しているセンサ群の中からいくつかのセンサを選び、コンテキストとして利用する特徴量およびその特徴量を計算する基となるデータの範囲を指定するというプロセスが必要となる。そこで、Wearable Toolkit では、図 6 に示すようなコンテキスト登録ツールを提供している。

このツールは、システム稼働中常にすべてのセンサから得られる値をモニタリングしている。ユーザが現在の状況をコンテキストとして登録したくなったときはこのツールを呼び出して「コンテキスト登録」ボタンを押す。すると図 6 に示すような画面が表示され、ユーザは過去一定時間分のすべてのセンサデータ値を閲覧できるため、ユーザは学習に用いる部分を選択し、「平均値」「変化量」などの特徴量をコンテキストとして登録するかを選択する。複数センサや、単一センサにおける複数特徴量を組み合わせることで登録できるため、柔軟にコンテキストを表現することが可能となっている。たとえばユーザが「次にここに来たときにアラートを

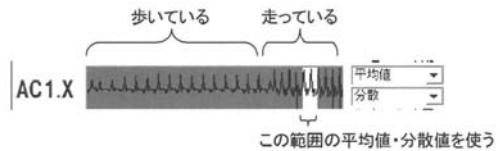


図 7: 設定例

出すために、ここという場所をコンテキストとして登録しよう」と考えた場合、このツールを呼び出して、GPS の X 値、Y 値における最新の部分の「現在値」を登録すればよいことになる。また、「走っているとき」というコンテキストを登録したい場合、しばらく走った後このツールを立ち上げて、走っている部分のデータから各加速度センサの X, Y, Z における適当な範囲を選択し、その「平均値」「分散値」両方を登録すればよいことになる。図 7 を用いて説明すると、コンテキストツールを立ち上げた際、右腕につけた加速度センサの X 値は、歩いているときと走っているときに図のように変化していることがわかる。そこで、そのうち走っている部分のここでは 2 周期分を対象とし、その平均値と分散値を選択している。このとき、その 2 周期分は特徴量の計算に使われるだけでなく、実際に認識を行う際のウィンドウサイズとしても利用される。

適切な値を設定し、コンテキスト名を登録すると、そのコンテキストがシステムに登録されて ECA ルールにおけるイベントやコンディションとして利用可能になる。例えば、2 章に示した例のひとつである「次にこの本屋の前を通りかかったときにはこれを買うことを忘れないようにリマインダを提示させることにした」という機能を実現するためには、本屋にいる状態でコンテキストツールを呼び出し、前述のようにコンテキストを登録し、次に図 8 に示すルールエディタを用いて、イベントに先ほどのコンテキスト (図では「BOOKSTORE」)、アクションにメッセージ表示を設定すればよい。

このように、Wearable Toolkit を用いることで、コンテキストの登録から実際のアプリケーション (ECA ルール) の作成までをその場で行うことができ、その場プログラミングが実現できる。

4.2 コンテキスト設定の難易度

提案するシステムおよびツールを使うことでその場プログラミングを行うことが可能になった。しかし、コンテキストツールを用いたコンテキスト定義は、一般ユーザには難易度の高い作業である。そこで、本節では予備実験として、実際に何人かのユーザにコンテキ



図 8: ルールエディタの画面

ストを設定させ、その難易度について考察する。

手続き

被験者として、本論文の筆頭著者 (被験者 A), 30 代前半の主婦 (被験者 B), 10 代前半の小学生 (被験者 C) を用意した。被験者 A はコンテキスト認識における十分な知識をもっており、ウェアラブルコンピューティングにも精通している。被験者 B は普段ウェブブラウジングやメールのやり取りを行うために PC を利用するが、ウェアラブルコンピューティングやコンテキスト認識に関する知識はまったくない。被験者 C は普段コンピュータをほとんど利用しない。

実験 1

被験者と関係のない人間に屋外を 10 分程度適当に歩かせ、GPS の値および、右手と右足にそれぞれ装着した 3 軸加速度センサの値を記録した。その後各被験者にコンテキストツール (図 6) を用いて「現在地にいる」を意味するコンテキストを定義させた。GPS および加速度センサの値が何を意味しているかは詳しく説明した。

実験 2

次に、被験者と関係のない人間に「歩行」→「走る」→「階段を下りる」→「階段をのぼる」→「自転車に乗る」→「横たわる」→「ひざ立ちをする」→「座る」→「直立する」の Context-awareness における 9 つの基本的な動作 [5] をそれぞれ 20 秒ずつ (トータル 180 秒) 行わせ、右手および右足に装着した 3 軸加速度センサの値を記録した。その後各被験者にコンテキストツールを用いてそれぞれのコンテキストを定義させた。このときは GPS は利用せず、2 つの 3 軸加速度センサの値 (合計 6 軸) のみをツールに表示してコンテキストを定義させた。特徴量としては、瞬間値、平均値、分散値を用意し、それぞれの特徴量の意味について詳しく説

表 3: コンテキストごとのレイテンシ (sec)

動作	ウインドウ幅				
	20	10	5	2	.5(秒)
歩く	20	11.1	5	4.7	0.9
走る	13.6	7.1	2.5	0.6	0
階段を下りる	15.8	6.7	2.3	1.6	2.3
階段を上がる	3.9	1.3	1.3	1.3	1.1
自転車に乗る	11.5	6.1	3	1.4	0.6
横たわる	10.1	5	2.5	1.2	0.5
ひざ立ちする	11.2	5.7	2.9	1.7	0.6
座る	14.6	7.6	3.4	1.3	0.6
立つ	9.7	2.8	1.6	1.6	3.8
平均	12.3	5.9	2.7	1.7	1.2

明した。この中から最大 2 個まで選ぶこととした。

コンテキストを定義させた後、同じ学習データを入力として、入力特徴量とコンテキストの特徴量のユークリッド距離を計算し、一番近いものを現在のコンテキストとして出力する認識器を用いて定義したコンテキストの認識精度を評価した。

実験 1 の結果

3 人の被験者ともに、「現在値にいる」というコンテキストが GPS で表現されることが理解でき、加速度センサの値は利用せずにコンテキストを定義した。被験者 A, C は GPS の現在値を、被験者 B は GPS の現在値周辺の平均値を用いたが、特に問題なくコンテキストが定義された。このような簡単なコンテキストであれば、提案するツールを用いることでその定義が正しく行えることがわかる。

実験 2 の結果

この実験では 3 人の定義傾向に差異が見られた。被験者 C は特徴量として瞬間値と平均値を選択したが、平均値は手足の向きが根本的に違う動作の区別にしか効果的でないため、認識精度は 38.1 % となった (横たわるやひざ立ち、自転車は 8 割程度の認識率であった)。被験者 A, B は特徴量として平均値と分散値を選び、そのコンテキストを表すセンサ値を正しく選択していた (「歩く」のコンテキストを決めるデータは実際に歩いているときのデータが用いられていた) が、そのセンサ区域におけるウインドウサイズがそれぞれ異なっていた。具体的には、被験者 A は 2 秒程度、被験者 B は 18 秒程度のウインドウサイズとしていた。結果、それぞれの認識精度は 79.8 %, 46.4 % となった。このように、複数のセンサと特徴量を組み合わせることが必要なコンテキストに関しては、提案するツールを用いても正しく定義を行うことが難しいことがわかる。

参考として、実験 2 と同様の条件において、ウインドウサイズを 0.5 秒、2 秒、5 秒、10 秒、20 秒と変化した場合の各コンテキストのレイテンシ (実際にそのコンテキストになってからシステムがそれを認識で

表 4: コンテキストごとの認識精度 (%)

動作	20	10	5	2	.5(秒)
歩く	0.5	31.5	46.5	64.5	55
走る	33	65.5	88.5	76	0
階段を降りる	22	67.5	76.5	43.5	33
階段を上がる	81.5	89.5	66	58.5	41.5
自転車に乗る	43.5	70.5	86	94	95.5
横たわる	50.5	76	88.5	95	98.5
ひざ立ちする	51	76.5	89	95.5	98.5
座る	28	63	84	94.5	98
立つ	52.5	87	92.5	92.5	58.5
平均	40.3	70.0	79.7	79.3	64.3

きるまでの時間)と認識精度を表3, 4に示す。表から明らかなように, ウインドウサイズを長くすればするほどレイテンシが大きくなる。例えばウインドウサイズを20秒にすると, 動作が切り替わってもそれを認識するまでに平均で12秒以上かかっており, 実際にアプリケーションで利用するのに適していない。一方, ウインドウサイズ2秒や0.5秒では問題になるほどの遅延は感じられない。また, 認識精度に関しては, ウインドウサイズ20秒で極端に悪化する。これは, 本実験では各動作を20秒ずつしか行わなかったため, レイテンシの悪さが認識精度に直接影響しているためである。このように様々な動作に切り替わり, それを認識する必要がある場合はウインドウサイズを大きくしすぎてはいけなことがわかる。逆に, 各動作が十分長く行われている場合, ウインドウサイズ5秒以上にした場合の認識精度はそれほど変わらないと予測できる。ウインドウサイズが0.5秒の場合に精度が落ちているのは, ウインドウサイズが小さすぎてノイズに対処できていないためであると考えられる。このように, 求めるコンテキストに応じて最適なパラメータ設定値は異なり, これらのパラメータを一般ユーザに直接設定させるのは難しいことがわかる。

パラメータの自動設定

前節で述べたように, コンテキスト定義のためのパラメータをユーザに手動で設定させるのは難しい場合も多くあることがわかった。その場プログラミング環境を実現するためには, このようなパラメータ設定の自動化または半自動化を行うことが必要となる。ユーザが求めるコンテキストによってパラメータの設定ポリシーは変化するため, 完全に自動化することは難しいが, 質問回答インタフェースによってユーザの目的を推測し, その結果に基づいてパラメータを自動設定する仕組みを現在実装中である。

5 まとめ

本研究では, ウェアラブルコンピューティング環境においていつでもどこでもサービスのプログラミングが行えるその場プログラミング環境の構築を目指し, イベント駆動型ルールエンジンおよびいくつかのツールを作成した。提案する枠組みは, イベント駆動型ルールで動作を記述でき, システム稼働中でもその動作を柔軟に更新できる。予備実験の結果から, 提案する枠組みはさまざまなユーザにおいてその場で容易にプログラミングを行える環境を提供している一方, 一般ユーザにとって複雑なコンテキストを定義することが難しいことがわかった。今後の予定としては, 4章最後で述べたパラメータ自動設定機能の実現および, システムの一般への配布とその普及活動が挙げられる。

謝辞

本研究の一部は, 独立行政法人情報処理推進機構2006年度下期末踏ソフトウェア創造事業「ウェアラブルコンピューティングのためのイベント駆動型ミドルウェア開発」, 文部科学省科学研究費補助金基盤研究(A)(17200006), および特定領域研究(19024046)によるものである。ここに記して謝意を表す。

参考文献

- [1] S. Albrecht, K. A. Aidoo, T. Antti, T. Urpo, L. V. Kristof, V. V. Walter, "Advanced Interaction in Context," Proc. of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC99), pp. 89-101 (1999).
- [2] K. Cheverst, N. Davies, et. al., "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences," Proc. of the 18th Conference on Human Factors in Computing Systems (CHI2000), pp. 17-24 (2000).
- [3] B. Clarkson, "Life Patterns: structure from wearable sensors," PhD thesis in Massachusetts Institute of Technology (2002).
- [4] A. K. Dey, D. Salber, and G. D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," A Special Issue on Context-aware Computing in the Human-Computer Interaction, Vol. 16, No. 2-4, pp. 97-166 (2001).
- [5] K. V. Laerhoven and H. W. Gellersen, "Spine versus Porcupine: a Study in Distributed Wearable Activity Recognition," Proc. of the 8th IEEE International Symposium on Wearable Computers (ISWC2004), pp. 142-149 (2004).

- [6] MIThril Project, <http://www.media.mit.edu/wearables/mithril/>.
- [7] M. Miyaame, T. Terada, M. Tsukamoto, and S. Nishio, "Design and Implementation of an Extensible Rue Processing System for Wearable Computing," Proc. of the 1st International Conference on Mobile and Ubiquitous Systems (MobiQuitous2004), pp. 392–400 (2004).
- [8] M. Miyamae, T. Terada, Y. Kishino, M. Tsukamoto, and S. Nishio, "An Event-driven Navigation Platform for Wearable Computing Environments," Proc. of IEEE International Symposium on Wearable Computers (ISWC2005), pp. 100–107 (2005).
- [9] M. Miyamae, T. Terada, M. Tsukamoto, K. Hiraoka, T. Fukuda, and S. Nishio, "An Event-driven Wearable System for Supporting Motorbike Races," Proc. of the 8th IEEE International Symposium on Wearable Computers (ISWC2004), pp. 70–76 (2004).
- [10] F. Naya, R. Ohmura, F. Takayanagi, H. Noma, and K. Kogure, "Workers' Routine Activity Recognition using Body Movement and Location Information," Proc. of the 10th IEEE International Symposium on Wearable Computers (ISWC2006), pp. 105–108 (2006).
- [11] K. Ouchi, T. Suzuki, and M. Doi: "LifeMinder: A wearable Healthcare Assistant," Proc. of the 2nd International Workshop on Smart Appliances and Wearable Computing (IWSAWC2002), pp. 791–792 (2002).
- [12] Semantic Web, <http://www.w3.org/2001/sw/>.
- [13] 若山史郎, 岩本健嗣, 西尾信彦, 徳田英幸, "ウェアラブルコンピュータにおけるデバイス非依存な環境情報の認識機構の設計と実装," 情報処理学会マルチメディア通信と分散処理研究会 (DICOMO), Vol. 2001(7), pp. 13–18 (2001).
- [14] WearableToolkit, <http://wearable-toolkit.com/>.