

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

若山 史郎[†] 村瀬 正名[†] 権藤 俊一[†]
岩本 健嗣^{††} 西尾 信彦^{†††} 徳田 英幸^{†††}

近年ウェアラブルコンピューティングと呼ばれる新しい計算機環境の研究が盛んに行われている。ウェアラブルコンピューティング環境ではユーザの移動などにより、さまざまに環境が変化する。本稿では変化する環境をとらえるためのセンサーを透過的に扱うことを可能とするミドルウェアを提案する。これにより特定のセンサーに依存しない、環境に適応するアプリケーションを容易に構築可能となる。同時に環境の変化を動的に認識し、アプリケーションに対してイベントを送ることも可能となる。また、本稿では本機構の設計とプロトタイプ実装を行い、測定を行った結果、実用に十分な検索の速度が証明された。

Device-Independent MiddleWear for Wearable Computing

SHIROU WAKAYAMA,[†] MASANA MURASE,[†] SHUNICHI GONDOU,[†]
TAKESHI IWAMOTO,^{††} NOBUHIKO NISHIO^{†††}
and HIDEYUKI TOKUDA^{†††}

Recently a new study on Computing Environment, called the Wearable Computing is being actively researched. Many changes in environment occur under a Wearable Computing Environment, since the user moves along with the Wearable Computer. In this document we would like to propose, a middle-ware which allow many devices, such as sensor which detects the changes in surrounding environment, to be used without the awareness of its presence. By using this middleware, user can easily construct a software without the cure of dependency specific devices. At the same time, it can dynamically detect the surrounding environment's changes and send event to the application. Also, here we have examined, and tested with a implemented Prototype to figure out the design and the problem of this mechanism. The test has proved that the system have enough speed to performed at practical use.

1. はじめに

ウェアラブルコンピュータ(以下WC)はいくつか実際に製品化されている¹⁾が、それらはポインティングデバイスなどの入出力デバイスの携帯性に特化したハードウェア部分に重点がおかれ、その利用形態についての研究は十分に進んでいない。

本稿ではWCを用いた新しいウェアラブルコンピューティング環境を提案する。本環境において、アプリケーションは環境の変化に対して適応的にその動作を

変化させることが可能となる。

また、本システムを使用することで、従来の適応的動作を行うアプリケーションのように特定のセンサーに依存することなく、環境の変化に適応できる。また、一つのセンサーが利用できない場合、アプリケーションが意識することなく代替センサーを使用することもできる。

本稿では、はじめに本研究が提案するウェアラブルコンピューティング環境について述べ、次にユーザと共に変化する環境と適応的動作をシナリオを用いて示す。そして変化する環境を、特定のセンサーに依存することなく把握する機構の設計とプロトタイプ実装、および評価について述べる。

2. ウェアラブルコンピューティング環境

本稿では、センサーからWCが取得する周辺状況の

[†] 慶應義塾大学 環境情報学部
Faculty of Environmental Information, Keio University

^{††} 慶應義塾大学 政策・メディア研究科
Graduate School of Media and Governance, Keio University

^{†††} 慶應義塾大学 SFC研究所
SFC Research Institute, Keio University

情報をセンサー情報と定義する。また、本システムにおけるセンサーとはセンサー情報を取得可能なものを指す。例えば位置情報を取得できるGPS、周辺温度を取得できる温度計などがセンサーとしてあげられる。また、PHSなどのデバイスは基地局IDなどのセンサー情報を取得できるため、位置センサーとして扱うことができる。

本稿では想定するウェアラブルコンピューティング環境の条件を以下のように定義する。

- 利用者は常にWCを携帯している
- センサー群を保持し、WCを用いて常にセンサー情報を取得している。
- 環境に適応して動作を行うアプリケーション群がWCもしくは近隣ネットワーク上に存在する。

ウェアラブルコンピューティング環境はWCが常にユーザと同じ環境を共有しているという特徴を持つ。そして、WCが環境を認識し、その環境に応じた動作をアプリケーションが行うことによって、ユーザの利便性を増すことを目的とする。

また、ウェアラブルコンピュータ環境を構成するWCとして、以下のようなものを想定する。

- WCは持ち運びを意識しないほど小型。
- 低消費電力。よって長期に渡って使用可能。
- 無線その他のデバイスを用いたネットワークへの接続性がある。

入出力装置はネットワーク上の他のデバイスが利用可能であれば、キーボードやヘッドマウントディスプレイなどは必ずしも必要ない。

また、ネットワークへの接続には携帯電話やIEEE 802.11²⁾、Bluetooth³⁾などを想定している。

WCは、衣服のように常に着用しており、その存在を意識しないコンピュータである。この点が持ち運ぶことを意識して利用する従来のモバイルコンピューティング環境とは大きく異なる。この可搬性によりWCはユーザと日常の空間を共有し、センサーによって現在おかれている環境を把握することが可能となる。

2.1 センサー情報

センサー情報はWCが利用可能なセンサーから得られる、ユーザの状況を表す情報である。この情報はセンサーごとにデータ形式などが異なる。例えばGPSから取得できるセンサー情報は緯度経度情報であり、温度計センサーから取得できるセンサー情報は温度である。

センサーがWC自身ではなくネットワーク上に存在し、センサー情報を取得できる場合、その情報を利用することも可能である。

現在想定されるセンサーおよび取得できるセンサー情報の一例を以下に挙げる。

- 位置情報
 - GPS：緯度経度情報
 - PHS：基地局ID情報
 - Ir（赤外線）：精度の高い位置情報
 - 住所：現実世界の位置情報
- 身体情報
 - 体温
 - 心拍数
- 周辺情報
 - 温度
 - 湿度

2.2 想定するシナリオ

本稿で想定するウェアラブルコンピューティング環境を利用するシナリオを以下に示す。

あるユーザは学校で人と会わなければならない。ユーザは「学校に到着した時、ユーザに対してある人に会うということを知らせる」という要求を、保持しているWCに入力する。

ユーザが移動し学校に到着すると、WCは取得しているセンサー情報から学校に到着したことを認識する。WCはユーザに対して学校に到着し、人と会わなければならないことを伝えるように、ダイアログを開いたり、音を鳴らすなど、ユーザの注意を引く動作を行う。このようなシナリオの要素は図1に分解される。

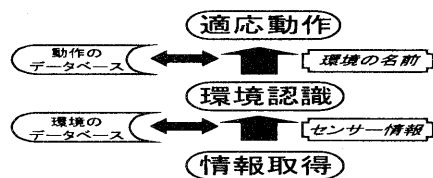


図1 シナリオの構成要素

まず、WCは常にセンサー情報を取得している。WCは取得しているセンサー情報をキーとして環境のデータベースを検索し、ユーザが学校にいるということを認識する。そして、学校という環境に応じた動作を動作のデータベースから検索し、実行する。このシナリオの場合、ユーザ自身の定義により、注意を引くというその場にあった動作を行う。

2.3 問題点

センサー情報を取得するためのデバイスはさまざまなものが利用可能である。しかし、同じ種類の情報を扱うセンサーでも、センサーごとに異なる形式の情報を提供する場合が多い。

現在の環境をWCが認識するために、センサーから取得できる情報の形式などをWCはあらかじめ知らな

ければならない。例えば地図表示ソフトはGPSのセンサー情報を有効利用している。しかし、このようなセンサー情報を直接利用するアプリケーションは特定のセンサーに依存して作成されている。そのセンサーが使用不可能な場合は、アプリケーションそのものも動作しなくなり、別のセンサーを代わりに使用することはできない。

位置情報を取得できるものとして、PHSをセンサーとして扱った場合、取得できる基地局IDなどがあるがセンサー情報の形式はGPSと異なる。そのため、PHSが利用可能であっても、そのセンサー情報を利用できず、アプリケーションは適応的動作を行えない。

ウェアラブルコンピューティング環境ではさまざまなセンサーおよびセンサー情報を使用する可能性があり、従来の特定のセンサーに依存するアプリケーションは利便性が低い。

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利便性を増すためにWCは環境を認識し、その場にあった動作を行うことが重要となる。WCはユーザと常に一緒にいるため、従来は不可能だったユーザ環境の認識および適応的動作が可能となる。しかし、前述したように従来の適応的動作を行うアプリケーションは、環境認識を行うために必要なセンサー情報を特定のセンサーに依存している。そのため、柔軟性および拡張性が少ない。

3. R2システムの設計

3.1 本システムの概要

上記の問題点を解決するために、本稿ではR2(Real-world Recognition)システムを構築する。R2システムは以下の点を目的とする。

- 同一の環境を示すセンサー情報には同一の名前を付け、透過的に扱い、センサー非依存性を実現する。
- センサー情報の変化に応じてアプリケーションに認識した環境をイベントとして送信する。

第一の機能は異なるセンサーを透過的に扱うための機能である。例えば複数種類のセンサーが利用可能であっても、適応的動作を行うアプリケーションからは「位置情報を取得可能なセンサー」として、センサーの種類や、どのセンサーが利用可能なかを意識することなく、透過的に扱うことが可能となる。また、一つのセンサーが使用不可能になっても代替センサーが使用可能であれば、アプリケーションは動作でき、その変更はアプリケーションに対して隠蔽される。また、複数のセンサーが使用可能であれば、それらを組み合わせることでさらに精度をあげることも可能である。

第二の機能は適応的動作を行うアプリケーションに

対して、ユーザが現在おかれている環境の名前をイベントとして送信する機能である。ユーザ環境の変化をアプリケーションが認識することが可能となり、その動作を変更することができる。これによりアプリケーションは環境に適応した処理を行える。

3.2 システムの動作例

送信されたイベントの利用法はアプリケーションに依存する。本節ではその具体的な利用例について述べる。

前出のシナリオでは、ユーザはあらかじめ「学校」という環境でPHSやGPSのセンサー情報に「学校」という環境の名前をつけておき、設定ファイルに書き込んでいる。

同時にユーザは設定ファイルに「学校に到着した時、beepアプリケーションに対してbeep音を鳴らすイベントを、ダイアログアプリケーションに対して人と会うことを表示するイベントを送れ」ということを記述しておく。

常にWCはセンサーからセンサー情報を取得している。ユーザが学校に到着すると同時にセンサーから取得している情報が「学校」環境という名前をつけられているセンサー情報と合致する。これにより、WCはユーザが現在「学校」という環境にいることを認識する。WCはデータベースを検索し、登録されている「beep音を鳴らす」という動作と、「人と会う用事というダイアログを表示する」という動作をするアプリケーションにイベントを送信する。

またGPSによるセンサー情報が取得不可能でも、PHSがセンサーとして使用可能であれば、WCはPHSのセンサー情報によって、「学校」環境であることを認識する。この利用するセンサーの変化はアプリケーションからは隠蔽される。

このように、本システムを使用することにより特定のセンサー情報に依存することなく環境を認識し、アプリケーションは現在のユーザ環境に適応した動作を行うことが出来る。

3.3 システムの構成

本システムの構成を図2に示す。

本システムはセンサーコントロールモジュール、データコントロールモジュール、アクションコントロールモジュールの3つのソフトウェアコンポーネントから構成される。また、センサーコントロールモジュールにはSensorML(Sensor Markup Language)、データコントロールモジュールにはDataML(Data Markup Language)、アクションコントロールモジュールにはActionML(Action Markup Language)によって記述されたそれぞれの設定ファイルを読み込むことによってさまざまな設定がなされる。

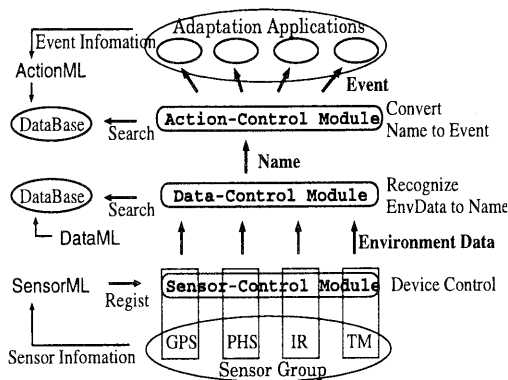


図2 センサー、アプリケーション群およびモジュール設計図

本システムでは新たな種類のセンサーやアプリケーションの追加に対応するため、これらを統一的に扱い、他のWCやネットワークから取得するなどのデータのやり取りをスムーズに行う必要がある。そのため、これらの設定ファイルはデータ構造を文書内に記述可能という特徴を利用し、XMLを用いて記述される。

3.3.1 センサーコントロールモジュール

センサーコントロールモジュールは、センサー情報を常に取得している。センサー情報を取得する方法などのセンサー固有の情報は、SensorMLによってセンサーごとに記述される。

SensorMLに含まれる情報にはセンサーの種類やデータ形式などの他にセンサー情報を取得する間隔、粒度、信頼度などが含まれる。これらの情報を用いることにより、粒度が高いセンサーを選択したり、精度を高めることができる。

例えばGPSと赤外線センサーが使用可能な場合、精度が高く、信頼性もある赤外線センサーを優先して名前を検索することなどができる。

センサーコントロールによって取得された位置情報などのセンサー情報はデータコントロールモジュールへと送られる。

3.3.2 データコントロールモジュール

データコントロールはあらかじめDataMLを読み込み、センサー情報およびセンサー情報につけた名前を保持するデータベースを構築しておく。図3にDataMLの例を示す。<ENVIRONMENT>はつけられた環境の名前を、<SENSORINFO>はどのセンサーからのセンサー情報に名前をつけたかを示す。

環境が変化していくに従い、必要となる名前情報も変化する。現在の状況により適した名前情報のみを持ち歩き、必要に応じてネットワーク上からDataMLを取得する。これにより、WC自体が大量のDataMLを保持する必要はなく、検索にかかる時間も減少する。

従ってWCに要求される計算機資源はさほど多くはない。

また、同一の名前を複数のセンサー形式のセンサー情報につけることも可能である。これにより異なる形式のセンサー情報を用いても、同じ環境の名前を検索することが可能となる。

データコントロールモジュールはセンサーコントロールモジュールから渡されたセンサー情報をキーとして、構築しておいたデータベースを検索する。

検索を行い、現在の環境に適合した名前を発見した場合、その名前をアクションコントロールモジュールに送る。これにより、個々のセンサー情報はアクションコントロールモジュールに対して隠蔽される。見つからなかった場合は意味のないセンサー情報として破棄される。

3.3.3 アクションコントロールモジュール

アクションコントロールモジュールはあらかじめActionMLを読み込み、ある環境の名前とその環境の名前に関連付けるアプリケーション、アプリケーションが送信してほしいイベントなどを保持するデータベースを構築する。

アクションコントロールモジュールはデータコントロールモジュールより渡された環境の名前をキーとして、登録されたアプリケーションのデータベースを検索する。データベースにアプリケーションが登録されていた場合、アクションモジュールはそのアプリケーションに対して、登録されたイベントを送信する。

イベントの受信によりアプリケーションはどのような環境にWCおよびユーザが変化したかを知ることができる。これによりさまざまに動作を変更し、ユーザの要求をより適切に環境に応じて実行することが可能となる。

イベントの種類は、

- Enter Event その環境になった時に起きるイベント
- Leave Event その環境を離れた時に起きるイベント

の二種類である。例えば前出のシナリオにおいては、「学校」という環境に入ったことを知らせるためにアプリケーションに対して「学校」という名前とEnter Eventを送信する。アプリケーションはEnter Eventを受け取った時点からLeave Eventを受け取るまで、ユーザが「学校」環境にいる時の動作を行う。

4. 実装・評価

実装はFreeBSD3.3RELEASE上でC++を用いて行っている。本実装ではセンサーコントロールモジュール、データコントロールモジュール、アクションコン

```

<?xml version="1.0" encoding="Shift_JIS" ?>
<!DOCTYPE dataml "dataml.dtd">
<dataml version="1.0">
<DATAML>
<ENVIRONMENT name="Keio SFC Campus o208"
  super="Fujisawa">
  <SENSORINFO type="PHS"
    devicename="MC-P100">
  <DATA>
    819A03463C80819A0343EB80
  </DATA>
</SENSORINFO>
  <SENSORINFO type="IR"
    devicename="SONY">
  <DATA>
    8931241
  </DATA>
</SENSORINFO>
</ENVIRONMENT>
</DATAML>

```

図3 DataMLの例

トロールモジュールのそれぞれに対してクラスを定義した。また、XMLパーザとしてはexpat⁵⁾を用いた。

4.1 使用センサー

本システムでは位置情報を取得するものとしてPHSとIr(赤外線)を取り上げ、プロトタイプ実装を行った。

PHSは基地局から基地局IDと電波強度を取得することができる。この基地局IDは全国で一意であり、基地局からの電波が届く範囲は200~300mと狭く、この基地局IDを取得することにより、ある程度の位置が把握できる。

また、通常基地局IDは通常複数取得でき、その組み合わせ及び電波強度によってさらに位置を絞り込むことができる。その精度は数m~数十mである。実際にこの機能を利用したサービスも始まっている⁶⁾。

同様に、赤外線で十分に大きい桁数の一意であるIDを常に送信し続ける赤外線センサーを壁面に設置し、WCで赤外線を検知することで、そのIDから位置を特定することができる。Irは通常角度30度、距離1mほど届くので、この機構により厳密に位置を検知することが可能である。

4.2 基本性能評価

DataMLは必要に応じてネットワーク上から取得される。定量的評価として、ネットワーク上のDataMLを取得し、名前を検索するためのデータベースを構築するまでにかかる時間の測定を行った。測定環境を表1に示す。

表2にネットワーク上にあるDataMLを取得し、データベースを構築するまでの時間および名前検索、イ

ベント送信にかかる時間を示す。接続するネットワークにはWaveLAN/IEEE802.11を使用した。また、一つのDataMLファイルには名前が100要素含まれており、サイズは約16KBである。各名前はPHSおよびIrセンサーのどちらか、もしくは両方のセンサー情報に対してつけられている。

表1 測定環境

マシン	VAIO PCG-505EX
CPU	Pentium 233MHz
OS	FreeBSD-3.3R
実装言語	C++
NetWork	WaveLAN/IEEE802.11

表2 名前データベース構築に要する時間

処理内容	処理時間 (msec)
ネットワーク上のDataML取得	215.5
XMLパース	613.1
データベース構築	4.0
名前検索	0.08
イベント送信	0.06

測定結果より、ネットワーク上にあるDataMLを取得し、データベースを構築するには一秒前後かかる。しかし、データベースの構築および検索、イベント送信に要する時間は秒単位より十分に小さい。そのため、DataMLの取得を一度行えば再度取得を行うまでの間ユーザの利便性を損なうことはない。

また、想定している本システムの使用形態はユーザの日常的な行動であり、認識に要する時間は秒単位であるため、ネットワークからDataMLを取得する間隔は一秒前後であっても構わない。従って、本システムは十分に実用的であると言える。

4.3 関連研究

本稿の関連研究として以下の研究について考察する。

- Active Badge⁷⁾
- POIX⁸⁾
- TEA⁹⁾
- EISS¹⁰⁾

4.3.1 ActiveBadge

ActiveBadgeシステムでは、ユーザはActive Badgeと呼ばれる単機能の小型デバイスを身につける。Active Badgeには赤外線センサーが付属しており、位置情報を取得できる。ユーザが移動するにつれて、さまざまなアプリケーションが適応的に動作する。例えば、ユーザが行くさきのドアを自動的に開閉するなどのアプリケーションが構築されている。Active Badgeは赤外線に特化してつくられているため、小型軽量であ

るが、それ以外のセンサー情報を使用することはできない。また、アプリケーションはActive Badge専用に開発しなければならない。

4.3.2 POIX

POIXはインターネット上における位置情報の交換データフォーマットを規定したものである。現在多くの位置情報を利用したサービスが存在するが、POIXを用いることによりそれらサービス間での位置情報の交換が実現可能となる。しかし、位置情報は緯度経度でのみ表され、異なる形式の位置情報を取得できるセンサーが利用可能であっても、POIXでは使用することができない。

4.3.3 TEA

TEAシステムはセンサーから取得する周囲の環境、状況、状態をコンテキストとして定義している。また、コンテキストごとに起動するアプリケーションをスクリプトに記述することで、アプリケーションからセンサー情報を隠蔽する。しかし、スクリプトで記述されているため、アプリケーションに対するインターフェースが提供されておらず、柔軟性に欠ける。

4.3.4 EISS

EISSは環境情報の変化をトリガとしてメッセージを配送する仕組みで、これを利用することで多種多様なセンサーからのトリガを扱うことができる。EISSは本システムと同等の機能を持つが、本システムではさらに構造的なアプリケーションとのインターフェースを備えている。

4.3.5 まとめ

表3に関連研究と本システムとの比較を示す。

本システムでは、現在ユーザがおかれている環境の認識も可能であるが、センサーの非依存性について考慮しており、この点において既存のシステムと異なる。ActiveBadge等関連研究では、単一もしくは複数の、特定センサーによってのみ動作している。しかし、本システムにおいてはXMLを使用してセンサーの特徴を読み込むことによって特定センサーに依存することなくさまざまなセンサーを使用することが可能となる。同時に、アプリケーションに対してイベントを送信する機構を用いることにより、環境に適応可能な特定のセンサーに依存しないアプリケーションを作成することが容易となる。

表3 関連研究との比較

名前	環境適応動作	センサー非依存
本システム	○	○
Active badge	△	x
POIX	○	x
TEA	△	○
EISS	○	○

5. まとめ及び今後の課題

本稿ではWCおよびWCをユーザを活動の中心とする新しいコンピューティング環境であるウェアラブルコンピューティング環境を提案した。また、環境に適応した動作を行うアプリケーションを構築する際のセンサー依存の問題点を指摘し、それを解決する本システムの概要について述べた。

また、本システムを用いてさまざまなセンサーを管理し、環境に応じた動作を行うアプリケーションに対してイベントを送るシステムのプロトタイプ実装を行い、実際の利用形態にあわせた機能と基本性能の評価を行った。その結果、実用に十分な速度が証明された。

今後の課題としては、本システムにおけるXMLなどのデータ構造およびアルゴリズムの改良をすすめ、さらにさまざまなセンサーを適用可能にする。また、実際に複数のセンサーによる適応動作を行い、本システムの実用性を証明する。また、粒度と精度以外のセンサー間の協調動作なども行う。同時に本システムを利用したアプリケーションの開発も行う。具体的にはユーザがおかれている環境に応じて入出力デバイスを変化させるなどのアプリケーションを想定している。

参考文献

- 1) IBM Inc, VisionPad Web page
<http://www.ibm.co.jp/News/leads/980912/>
- 2) IEEE 802.11 Working Group for Wireless Local Area Networks, "IEEE Std 802.11-1997"
- 3) Intel's Bluetooth web page
<http://www.intel.co.jp/mobile/bluetooth/>
- 4) Extensible Markup Language(XML)W3C Workin Draft
<http://www.w3.org/XML>
- 5) expat - XML Parser Toolkit
<http://www.jclark.com/xml/expat.html>
- 6) DDI Pocket 位置情報サービス
<http://www.ddipocket.co.jp/service/i.ichi.joho.html>
- 7) R. Want and Andy Hopper: "Active Badges and Personal Interactive Computing Objects," *IEEE Trans. Cons. Elect.*, Vol.38, No.1, Feb. 1992.
- 8) MOSTEC POIX
Point Of Interest eXchange language
<http://mostec.aplix.co.jp/>
- 9) Albrecht Schmidt et al.:
"Advanced Interaction in Context," pp89-101, HUC 1999.
- 10) N.Nishio and H.Tokuda:
"EISS:環境情報サーバサイトをを用いたシステムの状況適応" 情報処理学会第9回コンピュータシステムシンポジウム論文集. 1997年11月
- 11) Peter J. Brown et al.:
"context-Aware Applications: From the Laboratory to the Marketplace," *IEEE Personal Communications*, Vol.4, No.5, pp58-64, October 1997.