

WearMiddle: 無線センサデバイスを用いた実空間指向ゲームの開発支援 ミドルウェアの提案と実装

渡邊裕一郎† 青柳禎矩†† 高橋ひとみ†† 間博人†† 徳田英幸†,††

† 慶應義塾大学 環境情報学部

†† 慶應義塾大学大学院 政策・メディア研究科

e-mail: {nabe, sada, haru, hitomi, hxt}@ht.sfc.keio.ac.jp

概要

近年、ユーザの動作を用いた実空間指向ゲームが注目を集めている。特に小型軽量の無線センサデバイスをユーザの身体に装着することによって実空間指向ゲームを実現する WWSD (Wearable Wireless Sensor Device) 手法は、ユーザの動作制限が少ないという特徴を有する。しかし、WWSD 手法を用いた実空間指向ゲームの開発はセンサからの情報取得やコンテキスト情報の解析が困難であるため、現状では無線センサデバイスに関する多くの知識を必要とする。

本研究では、WWSD 手法を用いた実空間指向ゲームの開発を支援するミドルウェアシステム WearMiddle (Wearable Middleware) を提案する。WearMiddle はユーザに装着された無線センサデバイスでコンテキストを解析し、実空間指向ゲームに特化したコンテキスト定義リストを保持する。そのため、ゲーム開発者が自らコンテキスト解析の手法を考案する必要なく、ユーザコンテキスト情報をゲームシステムに反映できる。ゲーム開発者は WearMiddle を使用することで、無線センサデバイスのハードウェアやネットワークに関する知識を必要とせずに、実空間指向ゲームの開発が可能となる。

WearMiddle: A Construction and Implementation of Middleware which Supports to Develop Real-World Game using Wireless Sensor Devices

Yuichiro Watanabe† Sadanori Aoyagi†† Hitomi Takahashi†† Hiroto Aida††
Hideyuki Tokuda†,††

† Keio University, Faculty of Environmental Information

†† Keio University, Graduate School of Media and Governance

e-mail: {nabe, sada, haru, hitomi, hxt}@ht.sfc.keio.ac.jp

Abstract

Recently, the real space aiming game with user's actions attracts attention. Especially, the WWSD (Wearable Wireless Sensor Device) technique achieving the real space aiming game by installing small wireless sensor devices in user's body has the feature with a little limitation of user's actions. However, under the present situation, the development of the real space aiming game using the WWSD technique require a lot of knowledge concerning the wireless sensor devices because the acquiring from the sensor and the analysis of context information are difficult.

In this research, We propose middleware system called WearMiddle(Wearable Middleware) that supports the development of the real space aiming game using the WWSD technique. WearMiddle analyzes the context with the wireless sensor devices installed by the user, and maintains the context definition lists that specialize in the real space aiming game. Therefore, user context information can be reflected in the game system unnecessary the game developers think of the technique of the context analysis. The game developers don't needs neither the knowledge about hardware of the wireless sensor devices nor the knowledge about networks using WearMiddle, and the development of the real space aiming game becomes possible.

1 実空間指向ゲーム

1.1 概要

近年、任天堂 [8] から発売されたゲーム機の Wii [9] に代表されるような実空間指向ゲームが提案されている。実空間指向ゲームとはユーザの動作や行動がそのままゲームの進行に反映されるゲームのことであり、ユーザはより直感的にゲームを遊びとして体験できる。実空間指向のゲームでは、プレイヤーの行動や周辺環境をコンテキストとして得るために、様々なセンシング技術が用いられている。実空間

指向ゲームに用いられる使用デバイスの例として、GPS、カメラ、ウェアラブルコンピュータ、無線センサデバイスが挙げられる。

GPS (Global Positioning System) は、地球上の現在位置を調べるための衛星測位システムであり、実空間指向ゲームにおいてユーザの位置を取得するために利用される。GPS を用いた実空間指向ゲーム例として、Space Race [4] が挙げられる。Space Race は、プレイヤーが実空間上を GPS が搭載された携帯電話を所持し、移動する。ゲームはプレイヤーとオンライン上のナビゲータ役に分かれ、プレイヤー

はナビゲータの指示に従いつつ実空間を移動し、実空間指向ゲーム上の仮想空間内に隠された目的物を探す。

次にカメラによる画像認識技術を用いた実空間指向ゲーム例として、EyeToy カメラ [10] を挙げる。カメラは、カメラ画像の情報からプレイヤーのコンテキストを認識できるため、実空間指向ゲームにしばしば使用される。EyeToy カメラは、SCEI から発売されているゲーム機である PS2 (Play Station 2) 専用のカメラデバイスである。実空間指向ゲームを操作する際に PS2 専用のコントローラで操作するのではなく、EyeToy カメラを介し、テレビに写るプレイヤー自身をコントローラとして操作する。

またウェアラブルコンピュータを用いた実空間指向ゲームの例として、Human Pacman[2] が挙げられる。ウェアラブルコンピュータとは服やカバン、腕時計のように身につけて利用するコンピュータの事である。Human Pacman においてプレイヤーは HMD (Head Mounted Display) やカメラ、GPS を含めた様々なコンピュータを装着し、Pacman と Ghost に分かれ実空間を自由に移動する。両チームのプレイヤーは実空間上ではお互いを判別不可能であるが、両方のプレイヤーを判別できるオンライン上の Helper が、Pacman と Ghost が装着しているウェアラブルコンピュータから位置や状態の情報を把握し、伝達する。

また、ユビキタスコンピューティングにおいて実空間の人や環境の情報を収集する用途で用いられている無線センサデバイスというデバイスも、実空間指向ゲームに利用されている。ユビキタスモンスター [13] は、実空間と仮想空間の相互作用で動的に変化し続ける世界を舞台としたモンスター収集ゲームである。プレイヤーは実空間中のあるエリアへ接近し、そのエリアに対応するゲーム中のフィールドに生息するモンスターを実空間を動き回りながら収集してゆく。図 1 のような無線センサデバイスを取り付けた網 (モンスター捕獲ネット) を用い、無線センサデバイスの加速度センサのデータから、「網を振る=モンスターの捕獲」というコンテキストを用いている。

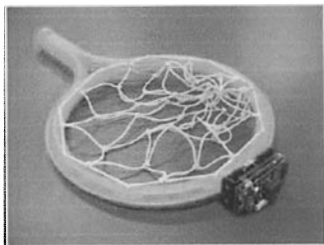


図 1: モンスター捕獲ネット (文献 [13] より引用)

1.2 実空間指向ゲーム機能要件

実空間指向ゲームを構成する機能要件として、インタラクティブ性、動作性、遍在性の 3 つを挙げる。プレイヤーとゲームが対話を行い、相互に影響し合うというインタラクティブ性はあらゆるゲームにおい

て求められるが、特にプレイヤー自身の動作や行動が直接仮想空間に反映できるという点で実空間指向ゲームはインタラクティブ性が高いといえる。また実空間をプレイヤーが動き回ることができるという動作性は、家庭内で専用コントローラを用いて遊ぶ形式のゲームにはない、実空間指向ゲーム特有の強みであるといえる。そして、実空間に使用デバイスがいかにか自然な形で配置されているかという遍在性もまた、実空間指向ゲームの一般化という面において必要な要件であるといえる。よって、これらの要件をより多く満たすことが可能な実空間指向ゲーム開発が求められる。

そこで実空間指向ゲームにどのデバイスを用いることが現時点で最良かということ調べるため、以上に挙げた実空間指向ゲーム要件をデバイス別に比較して表 1 にまとめた。

表 1: 実空間指向ゲーム要件のデバイス別比較

	インタラクティブ	動作	遍在
GPS	×	○	○
カメラ	○	×	△
ウェアラブル	○	○	×
無線センサデバイス	○	○	○

GPS はプレイヤーの位置情報のみを取得しているもので、プレイヤーの詳細なコンテキストは分からずインタラクティブ性に欠けている。カメラはプレイヤーの位置がカメラの認識可能範囲からずれてしまうと正しくプレイヤーのコンテキストが取得できないため、動作性に欠ける。また、ウェアラブルコンピュータはプレイヤーの体に多くの機材を装着することが前提であるため、プレイヤーの動作に支障をきたす。

一方、無線センサデバイスは環境に遍在させるために小型軽量化されているものが多く、遍在性に優れている。また無線センサデバイスは、種類にもよるがセンサデータを無線で 10m から 30m の距離まで送信することが可能なため、携帯した際の動作性にも優れているといえる。また無線センサデバイスは、実空間内に居るユーザのコンテキストを解析するためのシステムによく用いられていることから、インタラクティブ性に優れているといえる。よって、無線センサデバイスが最も実空間指向ゲームの要件を満たしている。

2 WWSD 手法の提案とプロトタイプの実装

本研究では実空間指向ゲームの開発手法として、無線センサデバイスを使用した WWSD 手法 (Wearable Wireless Sensor Device 手法) を提案する。

前節で述べたように、無線センサデバイスは実空間指向ゲームの開発に適しているデバイスである。そこで、更に無線センサデバイスをプレイヤーの体に取り付けるというアプローチを取ることで、プレイヤーにより密着したコンテキストが取得可能となり、インタラクション性がより増す。本研究ではまず、WWSD 手法の有用性を確かめるためのプロトタ

イブとして「UBI-SHOOT!!-ver1.0」[12]を実装した。UBI-SHOOT!!-ver1.0は、自機と敵機が攻撃し合う簡単なシューティングゲームである。UBI-SHOOT!!-ver1.0ではMICAz MOTE[5]とuPart[7]という無線センサデバイスを用い、トラの着ぐるみの頭上と手のひらにそれぞれ取り付けた。プレイヤーは頭上のMICAz MOTEと手のひらのuPartを操作し、移動、溜め、攻撃という三種類のコンテキストを元にゲームを行う。

UBI-SHOOT!!-ver1.0は実装において無線センサデバイスから送られるセンサデータ（0～256の数値）を元に、移動や攻撃というプレイヤーのコンテキストを判別した。頭部に取り付けられたMICA MOTEの加速度センサから送られるセンサデータが閾値を超えると右または左移動、プレイヤーが手のひらを握ることでuPart 照度センサからのデータが閾値を超えると溜めというコンテキストを、また同様に溜めの状態からプレイヤーが手を突き出し、uPartのボールスイッチセンサが腕の動きを感知したら攻撃というコンテキスト定義した。図2は溜め状態から攻撃状態にコンテキストが移行する際のプレイヤー動作を、図3はゲームスクリーンショットを示している。



図 2: UBI-SHOOT!!-ver1.0: 「溜め」から「攻撃」へ変化時のプレイヤー動作 (左: 「溜め」 右: 「攻撃」)

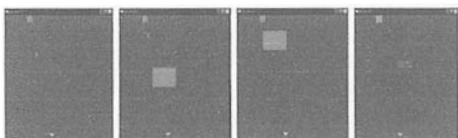


図 3: UBI-SHOOT!!-ver1.0: 「溜め」から「攻撃」へ変化時のスクリーンショット (左: 「溜め」 右: 「攻撃」が敵機に命中した瞬間)

3 問題意識と提案

3.1 問題意識

UBI-SHOOT!!-ver1.0の実装によって、WWSD手法を用いた実空間指向ゲームを提案した。しかしその開発において、UBI-SHOOT!!-ver1.0の開発の敷居が高いという結論を得た。WWSD手法を用いた実空間指向ゲーム開発が困難となる要因は以下の3つである。

開発ステップ数の増加

UBI-SHOOT!!-ver1.0の実装において、ゲームのイベント処理に用いるコンテキスト定義は開発者の

試行錯誤によってされた。従来のアプリケーションにおいて、センサデータから解析をするコンテキスト定義は、多くの場合そのアプリケーションやゲームにおいて必要な要件を満たすためだけのものとして設計されやすく、広く汎用的に使用できるシステムを構築することは困難である。よって、実空間指向ゲームをWWSD手法を用いて開発する場合、ゲーム開発者は図4の手順を踏まなければならない。

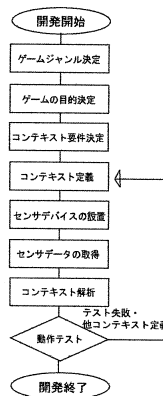


図 4: WWSD 方式における開発者の開発ステップの流れ

まず、ゲーム開発者は開発したい実空間指向ゲームのジャンルを決め、ゲームの目的を決める。その後、ゲームに使用するためのコンテキストの定義を行う。そして、その定義方法にそって無線センサデバイスの設置、センサデータの取得、コンテキストデータへの抽象化、動作テストを繰り返し、期待どおりの結果が得られたときにコンテキスト定義が終了となる。特に後半の5ステップは、コンテキスト定義をやり直す場合や開発に用いるコンテキストの数だけ繰り返し行う必要がある。よって、開発ステップやコード数が多くなる点が問題となる。

無線センサデバイス知識の必要性

無線センサデバイスから送られるセンサデータでコンテキストを解析するには、複数のセンサデータを抽象化する必要がある。また受信したセンサデータから、コンテキストの解析に必要なデータのみを選択して使用する必要もある。これらの作業には無線センサデバイスに関する知識が必要となる。

ネットワーク知識の必要性

無線センサデバイスからのセンサデータの受信、コンテキストデータを実空間指向ゲームのシステムに送信するために、ネットワークに関する知識が必要となる。

3.2 WearMiddle の提案

本論文では、WWSD手法を用いた実空間指向ゲーム開発の問題点を解決するミドルウェアシステムとしてWearMiddleを提案する。ゲーム開発者は本機

構を用いることにより、WWSO 手法を用いた実空間指向ゲームの開発が可能である。

4 WearMiddle の設計

4.1 機能要件

WearMiddle には大きくわけて、コンテキスト定義という 1 つのデータテーブルと、コンテキストエディタ機能、センサデータ統一機能、コンテキスト解析機能の 3 つの機能で構成されている。

コンテキスト管理表

コンテキスト管理表は、WearMiddle がコンテキスト解析を行う際に必要となるコンテキスト定義の情報を格納している。コンテキスト管理表のコンテキスト定義は、ゲームに多く用いられるものを設定する。また、コンテキスト定義はゲーム開発者も自由に書き加えられる方針を理想としているが、今回の設計においては決め打ちのコンテキスト定義を使用する。

コンテキストエディタ機能

コンテキストエディタ機能は、ゲーム開発者が必要なコンテキストを取得するための情報を提供する役割を持ち、WWSO 手法の開発ステップの増加問題を解決する。実空間指向ゲームの開発者は、ゲームに使用するコンテキストの詳細を設定できる。

センサデータ統一機能

センサデータ統一機能は、複数の規格が異なるセンサデータをコンテキスト解析機能が解析できるデータ形式に統一する役割をもつ。WearMiddle は複数種類の無線センサデバイスへの対応を想定しているため、それらの規格の違うセンサデータを同等に扱う。

コンテキスト解析機能

コンテキスト解析機能は、センサデータからプレイヤーのコンテキストを解析し、ゲームシステムにそのコンテキストデータを提供する。統一センサ機能が統一したセンサデータを元に、コンテキスト解析機能がコンテキストを解析し、ゲームシステムにコンテキストデータを送信する。本機能は、WWSO 手法における無線センサデバイスとネットワーク知識の必要性問題を解決している。

4.2 全体の設計

3 つの機能から成り立つ WearMiddle の構成図は図 5 の通りである。

ゲーム開発者は、実空間指向ゲーム開発のステップであるコンテキスト要求を、コンテキストエディタ機能を用いて行い、センサデバイスの設置場所等のコンテキスト解析情報を得る。またプレイヤーが身につけた無線センサデバイスは、センサデータをセンサデータ統一機能に送り、統一センサデータにする。そしてコンテキスト解析機能がコンテキストデータをゲームシステムに送ることによって、実空間

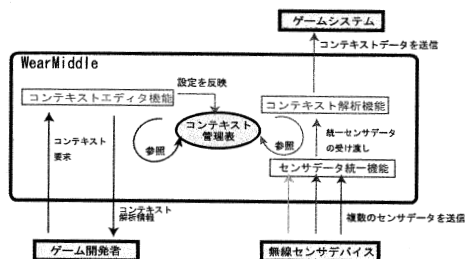


図 5: WearMiddle の全体構成図

間指向ゲームが成り立つ。また、コンテキストエディタ機能とコンテキスト解析機能は、コンテキスト管理表を参照する。

4.3 コンテキスト管理表の設計

コンテキスト管理表は、ゲームに使用するコンテキスト解析情報を管理し、表 2 に示す形でデータを格納する。コンテキスト管理表は、コンテキスト解析機能のセンサデータ解析モジュールによって参照される。また、コンテキスト管理表のデバイスの種類、デバイスの ID、デバイスの種類は、コンテキストごとに選択肢がいくつか用意されており、コンテキストエディタ機能のコンテキスト選択モジュールによって変更、決定ができる。

表 2: コンテキスト管理表格納データ構成

項目	説明
コンテキスト	使用できるコンテキスト
デバイスの種類	使用するデバイスの種類
デバイスの ID	使用するデバイスの ID
デバイスの設置位置	使用するデバイスの設置位置
使用センサ	解析に使用するセンサの種類
閾値	コンテキスト解析に用いる閾値
プレイヤーの動作	プレイヤーが行う動作
送信文字列	ゲームプログラムが WearMiddle から受け取る文字列

4.4 コンテキストエディタ機能の設計

WearMiddle のコンテキストエディタ機能を構成する 2 つのモジュールについて説明する。

コンテキスト選択モジュール

コンテキスト選択モジュールは、使用可能なコンテキストをコンテキスト管理表から参照し表示する役割をもつ。ゲーム開発者は、このリストからゲームに使用したいコンテキスト定義を選択する。

コンテキスト詳細設定モジュール

コンテキスト詳細設定モジュールは、コンテキスト選択モジュールで選択したコンテキスト定義の詳細を決定する役割をもつ。表 2 のように格納されているデータには、いくつかの選択肢が用意されているものもある。ゲーム開発者は、コンテキスト選択モジュールにおいて選択肢の中からコンテキストを定義できる。コンテキスト詳細設定モジュールで設定したコンテキストの情報は、データテーブルに記憶されセンサデータ解析機能において用いられる。

4.5 センサデータ統一機能の設計

WearMiddle のセンサデータ統一機能を構成する 2 つのモジュールの設計について詳細に説明する。

受信モジュール

受信モジュールは、プレイヤーに装着された無線センサデバイスからのセンサデータを受け取る役割をもつ。無線センサデバイスから送信されるデータは、無線センサデバイスの種類によってデータの構成が異なる。センサデータ受信モジュールは、実装に使用する無線センサデバイスの種類数によって複数個設計する必要がある。

センサデータ統一モジュール

センサデータ統一モジュールは、受信したセンサデータの中から、コンテキスト解析機能に必要な数値データを抽出し、無線センサデバイスの種類とその ID 情報を付加した統一センサデータをコンテキスト解析機能に受け渡す。無線センサデバイスの種類によってコンテキスト解析に必要な数値データの抽出方法が異なるため、センサデータ統一化モジュールも、実装に使用する無線センサデバイスの種類数によって複数設計する必要がある。

4.6 コンテキスト解析機能の設計

WearMiddle のコンテキスト解析機能を構成する 2 つのモジュール設計について説明する。

センサデータ解析モジュール

センサデータ解析モジュールは、センサデータ統一モジュールから統一センサデータを受け取り、その統一センサデータを解析してコンテキストデータにする。本モジュールは、まず統一センサデータから、データ元の無線センサデバイスの種類と ID の情報を取得する。その結果と加速度や照度、温度センサ等からの数値データ値を元にデータテーブルの中の情報を参照する。そしてプレイヤーのコンテキストを解析し、コンテキストデータを送信モジュールに受け渡す。

送信モジュール

送信モジュールは、センサデータ解析モジュールから受け渡されたコンテキストデータを、ゲームシステムへと送信する役割をもつ。

5 WearMiddle の実装

5.1 実装環境

WearMiddle は、Java 言語を用いて実装した。また、MICAz MOTE の事前実装に NesC 言語を用いた。本実装は無線センサデバイスとして MICAz MOTE と uPart を対象とした。

5.2 事前実装

事前実装として、使用する無線センサデバイスの送信間隔を狭める設定を行った。無線センサデバイスの送信間隔が広い程、プレイヤーが行動してからセンサデータが WearMiddle に送られるまでの遅延が多く生じてしまい、ゲーム進行の妨げとなる。よって、無線センサデバイスのデータ送信間隔を可能な限り狭く設定した。

MICAz MOTE は、TinyOS[3] 上のデータ送信のプログラムを書き換えることによって、センサデータの送信間隔を変更した。無線センサデバイスは、極端に短い頻度でセンサデータを送信するように想定されておらず、初期値は 1000~5000msec 程であった。本実装では、MICAz MOTE のプログラムを、送信頻度が 50msec 毎になるように変更した。

uPart のセンサデータ送信間隔は TecO[11] の Web サイトのコンフィギュレーションページで変更を行った。uPart のデータ送信間隔は最短で 288ms までしか変えられないため、288ms に設定した。

5.3 コンテキスト管理表の実装

コンテキスト管理表は、以下のような記述方法をとって実装した。

表 3: コンテキスト管理表

左移動,mote,1 2 3, 頭上 肩, 加速度, ...
右移動,mote,1 2 3, 頭上 肩, 加速度, ...
ジャンプ,mote,2 1 3, 足の甲, 加速度, ...
しゃがむ,mote,4 2 3, 腿の裏, 加速度, ...

左から順に、コンテキスト、デバイスの種類、デバイスの ID、デバイスの設置位置、使用センサ、閾値、プレイヤーの動作、送信文字列といった項目の情報を、カンマで区切って保持している。また同じ項目の中に選択肢が複数ある場合は、選択肢を半角スペースで区切り保持している。

5.4 コンテキストエディタ機能の実装

コンテキストエディタ起動時のスクリーンショットを図 6 に示す。図 6 の決定ボタンから上半分がコンテキストエディタのコンテキスト選択モジュール、下半分がコンテキスト詳細設定モジュールである。

コンテキスト選択モジュール

コンテキストエディタが起動すると同時に、コンテキスト選択モジュールがコンテキスト管理表を参照し、コンテキストの一覧を表示する。



図 6: Context Editor 起動時クリーンショット

コンテキスト選択モジュールがコンテキスト管理表を参照する際、まずコンテキスト管理表に格納されている 8 種類の項目をコンテキストをキーとしてハッシュテーブルに格納する。ハッシュテーブルを用いた理由としては、検索が高速であることが挙げられる。このコンテキスト管理表は、WearMiddle が起動している間は保持される。

コンテキスト詳細設定モジュール

コンテキスト一覧からコンテキストを選択し、決定ボタンを押すと、コンテキスト詳細設定モジュールが起動する。その後、コンテキスト管理表にコンテキストをキーとして保存されていたデータが呼び出され、表示される。コンテキスト管理表が保持する使用センサと閾値の値は、開発者が実空間指向ゲームの開発に不要な情報であるため、コンテキストエディタには表示しない。

図 7 は、明るさ変化のコンテキストを選択した際の表示結果である。この場合、無線センサデバイスの種類・無線センサデバイスの ID・無線センサデバイスの取り付け場所を選択肢があり、コンテキストエディタのプルダウンに表示される。ゲーム開発者はこの選択肢の中から、コンテキスト要求に合った設定を選択する。また、Send Buff 欄に表示されている文字列は、コンテキスト解析機能からゲームシステムに送られる文字列である。この文字列は開発者がゲームシステムを実装する際に必要となる。



図 7: Context Editor コンテキスト選択時スクリーンショット

コンテキストの詳細設定が終わった後、コンテキストエディタの設定ボタンが押されることによって、開発者が選択したコンテキストをキーとした 8 項目のデータがコンテキスト管理表に保存される。

5.5 センサデータ統一機能の実装

受信モジュール

センサデータ統一機構は、使用する無線センサデバイスの種類数に応じて実装する必要がある。MICAz MOTE と uPart のセンサデータを受信するため、それぞれ UDP ポートの 10000,10001 番を使用した。受信モジュールは、無線センサデバイスから受信したセンサデータと無線センサデバイスの種類情報をセンサデータ統一モジュールに受け渡す。

センサデータ統一モジュール

uPart と MICAz MOTE から送信される各センサの数値データの抽出は、Micaz Mote や uPart の送信データの構造に基づいて行った。

まず、受信モジュールから受け取ったセンサデータから、センサデバイスの ID と数値データを取得する。そして、無線センサデバイスの種類と ID、数値データの 3 種類を統一した統一センサデータを、コンテキスト解析機能に送っている。uPart に関しても同様で、無線センサデバイスの種類と uPart の ID、数値データを加えた統一センサデータをコンテキスト解析機能に送っている。

5.6 コンテキスト解析機能の実装

センサデータ解析モジュール

センサデータ解析モジュールは、統一センサデータに含まれる無線センサデバイスの種類と ID、数値データからコンテキストデータを解析する。コンテキスト管理表に格納されているコンテキストの中で、無線センサデバイスの種類と ID が一致するものを探索する。そして、その中でさらに数値データが閾値以内である場合、そのコンテキストに対応している送信文字列が送信モジュールに渡される。

送信モジュール

コンテキスト解析機能の送信モジュールは、センサデータ解析モジュールから渡された文字列情報を、ゲームシステムに送信する。ゲームシステムに文字列情報を送信する際は、初期値で 9000 番の UDP ポートを使用した。

また本実装環境では WearMiddle とゲームシステムが同一 PC 上にあつたため、同一 PC の IP アドレスの 9000 番ポートから送られてくるデータを着信するように設定した。WearMiddle とゲームシステムの IP アドレスが異なる場合は、WearMiddle 側にゲームシステム側の IP アドレスを設定する必要がある。

6 評価

6.1 ECN-Slider の実装

WearMiddle の評価には、ECN-Slider という WearMiddle を使用して開発した実空間指向ゲームを用いた。ECN-Slider は Macromedia Flash Professional 8[1] を用いて実装しており、WearMiddle で提供している左右移動、ジャンプ、しゃがみと明るさ変化の 4

つのコンテキストを使用している。プレイヤーは移動、ジャンプ、しゃがみの動作を実空間で行いながら、ゲーム内のキャラクタを操り、障害物をよけて進む。スクリーンショットを図 8 に示す。

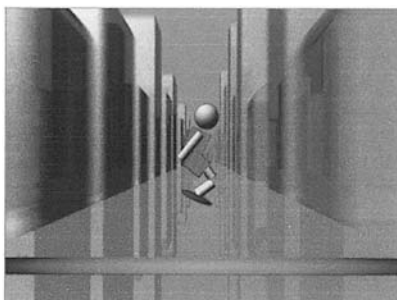


図 8: ECN-Slider スクリーンショット

6.2 定量的評価

WearMiddle の定量的評価として、ECN-Slider を動かす際の WearMiddle による処理遅延を計測した。

評価環境

WearMiddle と ECN-Slider を同時にユーザ P C (CPU: Pentium M 1.4GHz, Memory 512MByte, OS: Windows XP) 上で起動した。また、無線センサデバイスとして Mote を 4 個、uPart を 4 個の計 8 個を使用し、各無線センサデバイスのセンサデータを受信するシンクノードは、ルータ (ASUS Wireless Router WL-500gPremium) を経由して、有線 LAN ケーブルでユーザ P C と接続されている。

評価手法

評価手法は、WearMiddle 単体の処理遅延とゲーム全体の処理遅延を Mote と uPart の個数を 1 個から 4 個まで増やして計測した。WearMiddle 単体の処理遅延は、WearMiddle が無線センサデバイスからのセンサデータを受け取ってから ECN-Slider に受け渡すまでの処理時間を計測した。また、ゲーム全体の処理遅延は、プレイヤーの動作が ECN-Slider 上で反映されるまでの処理時間を計測した。

評価結果

WearMiddle 単体の処理遅延時間を図 9 に、ゲーム全体の処理遅延時間を図 10 に示す。横軸は無線センサデバイスの個数、縦軸が処理遅延時間 (ms) である。Mote、uPart 共に無線センサデバイスの個数が増加するに連れて、コンテキストデータを ECN-Slider に反映させるまでの時間が増加している。これは、センサデータ統一機能とコンテキスト解析機能において、多くのセンサデータを処理したことによる遅延だと考えられる。

また図 10 では、Mote を 4 個使用した時のゲーム全体の処理遅延時間は 200ms を超えていた。ゲームの処理遅延時間の許容範囲はゲームの種類やプレイヤーによっても異なるが、ECN-Slider のように遅延

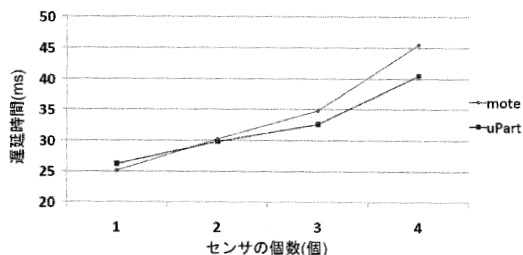


図 9: WearMiddle 単体の処理遅延時間 (センサの個数別)

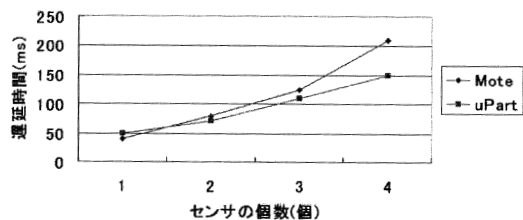


図 10: ゲーム全体の処理遅延時間 (センサの個数別)

による影響が大きいリアルタイム性の強いゲームにおいては 100ms の遅延でプレイヤーは不快感を覚え始め、250ms では操作に著しく影響を与える [6]。したがって 200ms を超える遅延は、一般的なプレイヤーの許容範囲を超えているといえる。WearMiddle の実装において Mote を使用する際は、最大で 3 個までは遅延が最低限の許容範囲である実空間指向ゲームを開発できる。

uPart の場合も Mote と同様なことがいえるが、uPart のデータ送信間隔が最短で 288ms であるため図 10 に示した遅延よりも体感遅延は長くなる。よって今回の WearMiddle の実装において uPart を使用する場合は、左右移動やジャンプのように遅延がゲーム進行の妨げとなりやすいコンテキストではなく、昼夜変更のような遅延の影響が少ないコンテキストに使用する等の工夫が必要であるといえる。

6.3 定性的評価

他 WWSD 手法の実空間ゲームとの比較

本節では WearMiddle を使用していない WWSD 手法を用いた実空間ゲームとの比較評価を行う。本評価で比較基準として用いる要素は、問題意識でも述べた開発者の作業ステップ数、無線センサデバイス知識の必要性、ネットワーク知識の必要性である。作業ステップ数は、コンテキストの定義等で試行錯誤のステップを行う回数である。無線センサデバイスやネットワークの知識の必要性については、必須な場合は×を、必要としないのは○を多少の知識を必要とするものを△とした。

比較結果を、表 4 に記す。

表 4: 他 WWSD 手法実空間指向ゲームとの比較評価

	ステップ数	デバイス	ネットワーク
ユビキタスモンスター	×	×	×
UBI-SHOOT!!	×	×	×
ECN-Slider	○	○	△

WearMiddle を用いて開発した ECN-Slider では、コンテキストの解析や定義は全て WearMiddle で行えるため、開発者の作業ステップを削減できる。よって、開発者が開発のために要するプログラムのコード数も、UBI-SHOOT!!-ver1.0-の際は 1000 行程度だったものが、ECN-Slider では 200 行程度に収まった。また無線センサデバイスやセンサネットワークに関しても、開発者はほぼ意識をしないうですむ。唯一、ゲーム開発者がコンテキストデータをゲームシステムに組み込む際に、ソケットプログラムを書く必要があるためネットワーク知識の必要性に関しては△となっているものの、他の実空間指向ゲームに比べて格段と開発が容易になったことがわかる。

7 まとめと今後の課題

WearMiddle において、コンテキスト管理表、コンテキストエディタ機能、センサデータ統一機能、コンテキスト解析機能を実装した。WearMiddle は WWSD 手法を用いた実空間指向ゲーム開発を支援する。WearMiddle を使用することによって、実空間指向ゲームの開発におけるステップ数は WearMiddle を使用しないものと比べて大きく減少し、また開発者への無線センサデバイスやネットワークに関する知識の要求もほぼ必要となくなった。

今後の課題としては、遅延の減少、汎用性の向上、開発者の自由度向上があげられる。今回の WearMiddle の実装では、無線センサデバイスの個数を増やす程遅延が大きくなった。この原因としては無線センサデバイスからの高頻度のセンサデータのやりとりによるデータ送信の遅延等が挙げられる。よってより多くのセンサデータを並行して高速に処理できるアルゴリズム等を WearMiddle に導入する必要がある。またコンテキスト管理表があらかじめ実装されたものに限定されており、用意したコンテキスト定義も 7 個と少なかった。よって、実空間指向ゲーム開発を支援するミドルウェアとして汎用性に欠ける点があったため、コンテキスト管理表の内容をより充実させる必要がある。またゲーム開発者による独自のコンテキスト作成の自由度が低かった。よって開発者側が自由にコンテキストの定義をコンテキスト管理表にて行える機能や、センサデータからコンテキストの定義が行えるキャリブレーション機能も必要である。

謝辞

本研究は総務省「ユビキタスネットワーク制御・管理技術の研究開発 (ubila プロジェクト)」の一部として実施した。

参考文献

- [1] Adobe. *Macromedia Flash Professional 8*: <http://www.adobe.com/products/flash/flashpro/>, 2005.
- [2] Adrian David Cheok and Siew Wan Fong and Kok Hwee Goh and Xubo Yang and Wei Liu and Farzam Farzbiz and Yu Li. Human Pac-man: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal And Ubiquitous Computing*, Vol. 8, No. 2, 2004.
- [3] D.Gay and P.Levis and R.Brewer and D.Culler. The nesC Language :A Holistic Approach to Network Embeded System. In *Proceedings of the International Conference on Management of Data*, 2003.
- [4] Drab S. A. and Binder Gerald. Spacerace: A Location Based game for mobile phones using Assisted GPS. In *Workshop on Pervasive Gaming Applications at Pervasive 2005(Pergames 2005)*, 2005.
- [5] J.Hill and R.Szewczyk and A.Woo and S.Hollar and D.Culler and K.S.J.Pister. System Architecture Directions for Networked Sensors. In *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2000.
- [6] L. Pantel and L. C.Wolf. On the impact of delay on realtime multiplayer games. In *In Proc. of NOSSDAV*, May 2002.
- [7] Micheal Beigl, Christian Decker, Albert Krohn, Till Riebel, Tobias Zimmer. upart: Low Cost Sensor Networks at Scale. In *UBI-COMP 2005*, 2005.
- [8] Nintendo. <http://www.nintendo.co.jp/>.
- [9] NINTENDO. Wii:<http://www.nintendo.co.jp/wii/>, 2006.
- [10] Sony Computer Entertainment Inc. *EyeToy*: <http://www.eyetoy.jp/>, 2004.
- [11] TecO. <http://particle.teco.edu/>.
- [12] Yuichiro Watanabe and Sadanori Aoyagi and Hitomi Takahashi and Hiroto Aida and Hideyuki Tokuda. UBI-SHOOT!! -ver1.0- : A Context-Aware Game Using Wearable Sensor-Networks. In *INSS2006 :Third International Conference on Networked Sensing Systems*, 2006.
- [13] 川西直, 川原圭博, 森川博之, 青山友紀. 実空間指向ネットワークゲームへのユーザアクションの適用機構. エンターテインメントコンピューティング 2004, Aug 2004.