

卒業論文

2006年度(平成18年度)

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構築

指導教員

慶應義塾大学 環境情報学部

徳田 英幸

村井 純

楠本 博之

中村 修

高汐 一紀

湧川 隆次

慶應義塾大学 環境情報学部

渡邊 裕一郎

卒業論文要旨 2006年度(平成18年度)

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構築

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、無線センサデバイスを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、実空間の様々な無線センサデバイスを利用し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。

慶應義塾大学 環境情報学部
渡邊 裕一郎

Abstract of Bachelor's Thesis

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

Recently, the research and the development of the context analysis system with wireless sensor devices have attracted attention. There, it is a general method to use the technique that constructs the network with arranging a small, cheap wireless sensor devices in various places of a real space and from the sensor, collects data of the state of the environment, the person and the thing in a real space automatically then analyzes these data as context information.

However, there is another method using advantage of smallness of the wireless sensor devices. With arranging these wireless sensor devices directly in the human body, we can acquire the data that sticks to the person and analyze the context. It can be said that this technique is useful for the application development that needs the user interactions, mainly in the field of the game development. Especially, in the real space game operated by the user action, technologies such as image recognitions and infrared sensors with the camera are used for the Sensing of user's free movements. The wireless sensor devices can be used to develop the real space game as well as these, too.

However, there is a problem when the wireless sensor devices are used for the game development. The problem is that it is difficult to develop the game considering of acquisition of the information from the sensor and the analysis of context information if you are not a specialist on the sensor devices and the network. This is because an analytical technique and a definition of the context when the environment of a real space is analyzed from the sensor data are more complex than the case to use other technologies. So, the development of the game with wireless sensor devices were beyond our capacities besides the specialist of the wireless sensor device under the present situation.

In this thesis, I propose a Middleware system called as Wearable Middleware that supports development of game using a variety of SenSynge information on real space. WearMiddle can maintain the context definition list that specializes in the game by using an analytical technique of the context to which the user is required to install the wireless sensor device in the body. And User context information can be reflected in the game program and for the game developer it is unnecessary to design the technique of the context analysis.

With using WearMiddle, it becomes possible for game developers to develop the real space game without knowledge on the hardware side and the network side of the wireless sensor devices.

Yuichiro Watanabe

**Faculty of Environmental Information
Keio University**

目次

第1章	序論	1
1.1	研究動機	1
1.2	本研究の目的	2
1.3	本論文の構成	2
第2章	研究背景	4
2.1	実空間指向ゲームの概要	4
2.2	使用デバイスによる実空間指向ゲームの分類	4
2.2.1	GPS	4
2.2.2	カメラ	5
2.2.3	ウェアラブルコンピュータ	6
2.2.4	無線センサデバイス	8
2.3	無線センサデバイスの概要	8
2.3.1	概要	8
2.3.2	必須要件	9
2.3.3	構成要素	10
2.3.4	無線センサデバイス例	11
2.4	本章のまとめ	13
第3章	問題提起	14
3.1	実空間指向ゲームの機能要件	14
3.2	既存実空間指向ゲームの問題点の比較	15
3.2.1	GPS	15
3.2.2	カメラ	15
3.2.3	ウェアラブルコンピュータ	16
3.2.4	無線センサデバイス	16
3.3	WWSD手法と問題点	16
3.3.1	WWSD手法の提案	16
3.3.2	WWSD手法を用いたプロトタイプの実装	17
3.3.3	WWSD方式の問題点	18
3.4	WearMiddleの提案	21
3.4.1	コンテキスト定義における問題解決	21
3.4.2	コンテキスト解析における問題解決	21
3.5	センサネットワークにおけるミドルウェアの要件	21

3.6	関連研究	23
3.7	本章のまとめ	24
第4章	WearMiddle の設計	26
4.1	WearMiddle の概要	26
4.2	想定環境	26
4.3	機能要件	27
4.3.1	全体構成	28
4.4	モジュールの設計	29
4.4.1	コンテキスト管理表	29
4.4.2	コンテキストエディタ機能	29
4.4.3	センサデータ統一機能	31
4.4.4	コンテキスト解析機能	33
4.4.5	動作手順	34
4.5	本章のまとめ	34
第5章	WearMiddle の実装	35
5.1	実装の概要	35
5.1.1	実装環境	35
5.1.2	事前実装	35
5.2	モジュールの実装	36
5.2.1	コンテキスト管理表	36
5.2.2	コンテキストエディタ機能	37
5.2.3	センサデータ統一機能	38
5.2.4	コンテキスト解析機能	40
5.3	本章のまとめ	41
第6章	WearMiddle の評価	42
6.1	ECN-Slider の実装	42
6.2	評価概要	43
6.3	定量的評価	43
6.4	定性的評価	44
6.4.1	WWSD 手法の実空間ゲームとの比較	44
6.5	本章のまとめ	45
第7章	結論	47
7.1	今後の課題	47
7.2	まとめ	48

目次

1.1	次世代ゲーム機の初動1週間の売上台数の比較(データ元:週刊ファミ通 No.941 [8])	1
2.1	Space Race における、GPS を用いたシステム構成(文献 [7] より引用)	5
2.2	EyeToy カメラセットアップ画像 ([22] より引用)	6
2.3	EyeToy カメラ対応実空間指向ゲームスクリーンショット ([22] より引用)	6
2.4	Human Pacman に用いられたウェアブルコンピューター(文献 [4] より引用)	7
2.5	Human Pacman のゲーム画面(文献 [4] より引用)	7
2.6	モンスター捕獲ネット(文献 [29] より引用)	8
2.7	MICA MOTE (出典:Crossbow 社ウェブページ [5])	12
2.8	uPart とセンサ	12
3.1	UBI-SHOOT!!-ver1.0- システム図	18
3.2	UBI-SHOOT!!-ver1.0-: 「パワーチャージ」から「攻撃」へ変化時のアクション(左: 「パワーチャージ」 右: 「攻撃」)	19
3.3	UBI-SHOOT!!-ver1.0-: 「パワーチャージ」から「攻撃」へ変化時のスクリーンショット(左: 「パワーチャージ」 右: 「攻撃」が敵機に命中した瞬間)	19
3.4	WWSD 方式における開発者の開発ステップの流れ	20
4.1	WearMiddle の全体構成図	29
4.2	コンテキストエディタ機能のモジュール図	30
4.3	センサデータ統一機能におけるモジュール図	31
4.4	uPart の送信データの構造	32
4.5	uPart の送信データの concom-u の中身	32
4.6	MICA MOTE の送信データの構造	32
4.7	コンテキスト解析機能におけるモジュール図	33
5.1	uPart Sensor Configuration ページ ([24] より引用)	36
5.2	Context Editor 起動時クリーンショット	37
5.3	Context Editor コンテキスト選択時スクリーンショット	39
6.1	ECN-Slider スクリーンショット	42
6.2	無線センサデバイスの使用個数別処理遅延時間(1種類)	44

表 目 次

2.1	無線センサデバイスの代表的な通信規格 (出典:Sensor Network Technology [21])	10
2.2	uPart の構成	13
3.1	既存実空間指向ゲームの問題点の比較	15
3.2	UBI-SHOOT!!-ver1.0-におけるコンテキスト定義表	18
4.1	コンテキスト管理表格納データ構成	30
5.1	MICAz MOTE センサデータ送信頻度変更擬似コード	35
5.2	コンテキスト管理表	36
5.3	センサデータ統一モジュール擬似コード (Mica Mote 用)	39
5.4	センサデータ解析モジュール擬似コードの一部	40
6.1	他 WWSD 手法実空間指向ゲームとの比較評価	45

第1章 序論

1.1 研究動機

近年、任天堂 [16] から発売されたゲーム機「Wii」[18] に代表されるような、実空間指向の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。昨今のゲームコンテンツは、Computer Graphics (CG) や映像をコンテンツの特有性として全面に出したものが多く、毎日何時間もゲームをするようなユーザ向けに高難易度化・シリーズ化する事が一般的となり、初心者やゲームを趣味程度に遊ぶようなユーザには敷居が高かった。しかし最近では Wii に代表される実空間指向ゲームが台頭し、単純で直感的なゲーム性が好評を博している。このことは、図 1.1 が示すように、Wii の初動一週間の売上台数を PS3[23]、XBOX360[13] といった次世代ゲーム機と比較してみても明らかである。

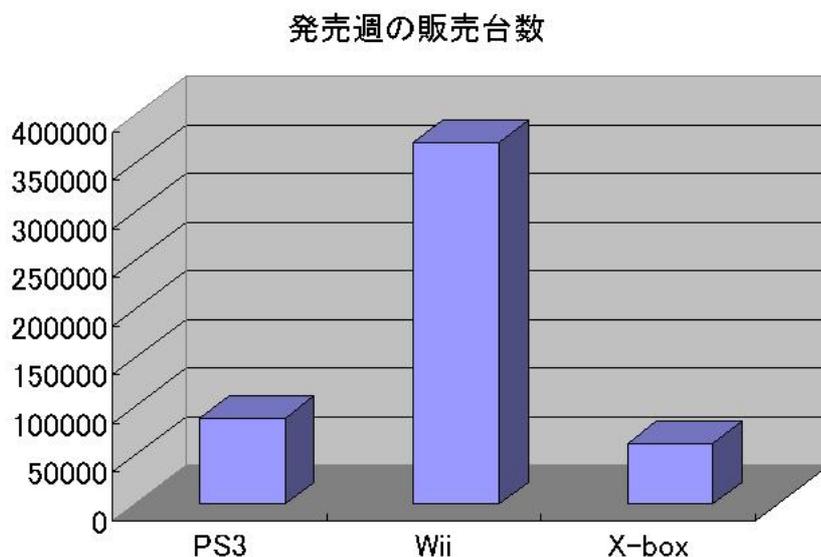


図 1.1: 次世代ゲーム機の初動 1 週間の売上台数の比較 (データ元:週刊ファミ通 No.941 [8])

実空間指向ゲームには、実空間の情報を取得するために様々なセンシング技術が利用されている。例として、カメラによる画像認識や GPS (Global Positioning System) センサ等が挙げられる。また、ユビキタスモンスター [30][28][29] に代表されるように、

無線センサデバイスを網などの物に取り付け、実空間指向ゲームに利用するという試みもある。

本研究の最初の動機は、無線センサデバイスを人に直接装着し、センサデータからユーザの行動や周辺環境を「コンテキスト」として抽象化し、実空間指向ゲームに使用するというものだった。無線センサデバイスを用いた実空間指向ゲームは、他のセンサデバイスを用いた実空間指向ゲームと比べて、ユーザの動作範囲や実空間指向ゲーム中の仮想空間と実空間との相互作用性に優れており、今後の実空間指向ゲームの開発において重要な技術となりうる。そしてそのプロトタイプとして、実空間指向ゲームである「UBI-SHOOT!! -ver1.0-」[26]を提案した。ユーザの移動、パワーチャージ、攻撃という三種類のコンテキストの解析を行い、シューティングゲームに反映した。

しかし、UBI-SHOOT!!-ver1.0の開発において、無線センサデバイスの装着部位や、コンテキストの定義、解析手法は常に開発者の試行錯誤が必要であり、ユビキタスモンスターを始めとする無線センサデバイスを用いた他ゲームの開発例でも同様であった。センサデータのみで様々なコンテキストを解析するためには、用途に応じて最適なセンサを用いる必要があるため、何種類かの無線センサデバイスを使用することになる。無線センサデバイスの種類によって機能や性能は異なるため、様々なセンサを1つの実空間指向ゲームが利用するためには、そのような無線センサデバイスの種類や機能の違いを吸収するための、ゲームに特化したミドルウェアの存在が必要となる。

1.2 本研究の目的

本研究の目的は、無線センサデバイスを用いた実空間指向ゲームの開発の敷居を低くするためのミドルウェアの構築である。そこで、本研究では、WearMiddle (Wearable Middleware) を提案する。

従来の実空間指向ゲームの開発は、1.1で述べたような開発者の試行錯誤によって行われるものであった。WearMiddleは、ゲームに最低限必要なコンテキストと、コンテキストの解析方法を開発者に提供する。そして、ゲーム開発者はWearMiddleから送られてくるコンテキストデータのみをゲームプログラムに組み込み、実空間指向ゲームの開発できる。

本機構により、ゲーム開発者は実空間指向ゲームの開発において無線センサデバイスに関する試行錯誤を行う必要がなくなり、実空間指向ゲームの開発にかかる期間の減少につながる。

1.3 本論文の構成

本論文は全8章から構成される。第1章では、本研究の動機及び目的・意義について述べた。第2章では、本研究の背景である無線センサデバイスの概要、実空間指向ゲームの概要についてを説明し、それぞれの関連研究を挙げる。次の第3章では、まず無線センサデバイスを用いた実空間指向ゲームの開発の優位性について論じた後、

従来の手法での問題点を指摘する。そしてその問題点の解決策として、第4章で無線センサデバイスを用いた実空間指向ゲームの開発を支援する機構である WearMiddle の提案を行い、第5章でその設計方法を説明する。次の6章では前章の設計に基づき行った本機構の実装の詳細について説明する。7章にて WearMiddle を用いて開発したコンテンツの定性的評価と、WearMoiddle の定量的評価を行い、本機構を用いた実空間指向ゲームの開発の有用性を実証する。最後に8章において本論文をまとめ、今後の課題について展望する。

第2章 研究背景

本章では、本研究の背景を述べる。まず本研究の対象である実空間指向ゲームについて概説し、その既存研究・実装例を示す。また、本研究で用いる基盤技術である無線センサデバイスについて概説する。

2.1 実空間指向ゲームの概要

本節では、本研究の対象アプリケーションである実空間指向ゲームの概要について述べ、この分野においての様々なセンシング技術の有用性を述べる。なお、本節よりゲームで遊ぶ人をプレイヤーと定義する。

実空間指向ゲームは、プレイヤーが実空間で起こした様々なイベント（歩く、ジャンプする等）が、仮想空間において反映されるゲームの総称である。専用のコントローラは用いず、プレイヤー自身の感覚的な操作を行えることが特徴である。身近な例として、ゲーム機のWiiに対応しているゲームソフトが挙げられる。Wii対応のゲームソフトである「ゼルダの伝説 トワイライトプリンセス」[17]では、プレイヤーがリモコン型のコントローラを振り下ろすことで、「攻撃」のアクションをゲームキャラクタが行う。また右手のコントローラを空中に振り上げ、左手のコントローラをリールを巻くように動かすことで、ゲームキャラクタが「釣り」を行うことも可能である。

このように、実空間上のプレイヤーがあたかも仮想空間上に自らがいるような没入感を感じられるという点が、実空間指向ゲームの特徴である。

2.2 使用デバイスによる実空間指向ゲームの分類

実空間指向のゲームでは、実空間でのプレイヤーのコンテキストを得るために、様々なセンシング技術が用いられている。本節では、実空間指向ゲームに用いられる使用デバイスの観点から、既存研究および実装例を4つに分類する。特に本研究で用いる無線センサデバイスについては、概要や必須要件、構成要素やデバイス例についても詳しく述べる。

2.2.1 GPS

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の研究例として、Space Race[7] が挙げられる。Space Race において、プレイヤーは実空間上を、GPS を搭載した携帯電話を所持して移動する。ゲーム内はプレイヤーとオンライン上のナビゲータに分かれ、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、実空間指向ゲーム内に繰り広げられるの仮想空間上に隠された目的物を探す。

図 2.1 に Space Race のシステム構成を示す。

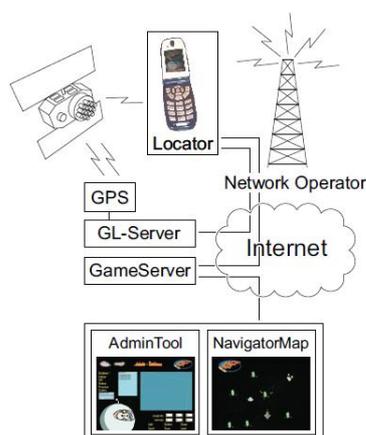


図 2.1: Space Race における、GPS を用いたシステム構成 (文献 [7] より引用)

プレイヤーの実空間内での位置情報は、GPS を搭載した携帯電話からゲームサーバへと送信され、ナビゲータが閲覧できるナビゲータマップに表示される。また、プレイヤーは目的物の発見を、プレイヤーが閲覧できるアドミンツールによって把握できる。

また Newt Games 社のモギィ アイテムハント [15] も、GPS を使った実空間指向ゲームの一例である。モギィ アイテムハントは、プレイヤーが実際に東京のあらゆる場所を巡り、携帯電話に搭載された GPS を使って仮想的なアイテムを探す実空間指向ゲームである。花や果物、生物などのアイテムが東京中に隠されており、プレイヤーの携帯電話に表示される街の地図にその位置が示される。プレイヤー達はチームを組んで街をくまなく探し、様々なアイテムを組み合わせたコレクションの作業を目的としている。またアイテムを交換するために、自分達とは違うコレクションを集めている他のチームのプレイヤーの探索も行う。

2.2.2 カメラ

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使用される。カメラによる画像認識技術を用いた実空間指向ゲームの実装例として、まず AceSpeeder2[19] が挙げられる。

AceSpeeder2 は、反重力空間を高速で走るレーシングゲームである。このゲームは、プレイヤーの頭部背面に設置したカメラの画像から、プレイヤーの体の傾き具合を

GPUVision[2] を用いて判別する。図の左側が元のカメラ映像であり、プレイヤーは、ゲーム上の乗り物を体の動きで操作できる。

また、SCEI (Sony Computer Entertainment Inc) から発売されている EyeToy カメラ [22] というデバイスも、カメラによる画像認識技術を用いたものである。EyeToy カメラは、同じく 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。



図 2.2: EyeToy カメラセットアップ画像 ([22] より引用)



図 2.3: EyeToy カメラ対応実空間指向ゲームスクリーンショット ([22] より引用)

EyeToy は、カメラに映る自分の姿と、図 2.2 内の白い枠線との大きさが同じになるようにカメラのピントやプレイヤーの位置を微調整する。そして、図 2.3 のように画面に映し出されたプレイヤー自身のカメラ画像を見ながら、プレイヤー自身が動いて遊ぶことができる。

2.2.3 ウェアラブルコンピュータ

最も理想的なコンピュータの携帯手法として、コンピュータを着るというアプローチであるウェアラブルコンピューティングの研究は、携帯電話、PDA の進歩に見られるコンピュータの小型化に伴い近年盛んになっている。

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[4] が挙げられる。Human Pacman においてプレイヤーは HMD (Head Mounted Display) 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 (図 2.4 参照)。Pacman チームは HMD を通して見える仮想空間上における目標物の収集と、Ghost による Pacman への接触がゲームの目的である (図 2.5 参照)。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ため、両方のプレイヤーを判別できるオンライン上の Helper が、その位置を伝達する。実空間内でのプレイヤー同士の直接的なコミュニケーション効果を得るため、Human Pacman では相手にタッチするという要素を取り入れている。



図 2.4: Human Pacman に用いられたウェアブルコンピューター (文献 [4] より引用)



図 2.5: Human Pacman のゲーム画面 (文献 [4] より引用)

プレイヤーは、Human Pacman において図 2.4 のような GPS センサの他にもカメラや無線モジュール、バッテリー等を装着する。これにより、複数のセンシング技術を同時に使用しながらプレイヤーが実空間を広範囲で移動できる。

2.2.4 無線センサデバイス

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がある。無線センサデバイスは、様々なセンサデータを無線技術で送信可能なデバイスであり、実空間指向ゲームに利用されている研究例もある。東京大学の川西氏らが開発したユビキタスモンスター [30][28][29] がその例である。

ユビキタスモンスターは、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 [29]。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。その実装の中で、図 2.6 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の定義をしている。



図 2.6: モンスター捕獲ネット (文献 [29] より引用)

2.3 無線センサデバイスの概要

本節では 2.2.4 で触れた無線センサデバイスの概要について述べる。まずユビキタスコンピューティングにおけるネットワークセンシングの概要を論じた後、そのプラットフォームである無線センサデバイスの概要について説明する。

2.3.1 概要

近年、我々を取り巻くネットワーク環境は、あらゆるものがネットワーク接続される「ユビキタスコンピューティング (Ubiquitous Computing) [11]」環境へと変化を遂げ

ようとしている。ユビキタスコンピューティング環境では、我々の生活する実空間とコンピュータネットワークの作り出す仮想空間が密に相互接続されることにより、実空間中で起こるあらゆる出来事を仮想空間で把握でき、また仮想空間での処理結果を実空間に反映できる。

このような実空間とのインタラクションを特徴とするユビキタスコンピューティングにおいて、実空間の情報を取得できるセンシング技術は必要不可欠である [21]。そして特に、センシングシステムとネットワークの結合はコンピュータネットワークが作り出す仮想空間と実空間を密に連携させることを可能とする。SSLab(Smart Space Lab.)[25] では、環境に埋め込まれた様々なセンサによるネットワークで構成された知的空間 (Smart Space) とユーザのインタラクションに基づく、次世代コラボレーションスペースの実現を果たしている。

ユビキタスコンピューティングを構築するセンサネットワークにおいて必要不可欠なものが、センシングデータを無線で送信、ないしは受信が可能な「無線センサデバイス」である。無線センサデバイスは、「スマートダスト (Smart Dust)[10]」のプロジェクトを発端として研究者の注目を集め、より環境に組み込み易いものへとなっていた。スマートダストは「無数の無線センサデバイス (スマートダスト) が空中を漂いながら、互いに通信し、実空間の情報を集める」というセンサネットワークの究極の姿を提案している。

2.3.2 必須要件

スマートダストのコンセプトを実現させるために、無線センサデバイスには主に以下の4つの要件が必須である。

- 小型・軽量化
実空間の中のありとあらゆる物や場所に配置をするため、またより多くの数を空間に散りばめるためには、デバイスはできる限り小さく軽い方が望ましい。
- 安価化
実空間にデバイスを散りばめるには多くの数を必要とするため、単価は安い方が望ましい。
- 省電力化
実空間で長時間稼働が必要になるため、電力消費ができるだけ少ない方が好ましい。またデバイスの小型・軽量化の為に小さい電池でも長時間稼働できると望ましい。
- 耐故障性の強化
長時間に渡りデバイスを放置する必要があるため、天候や事故が原因で故障してしまうこともしばしある。そのためなるべく対故障性に優れている方が望ましい。

2.3.3 構成要素

一般的に無線センサデバイスは、通信モジュール・マイクロプロセッサ・センサ基盤・電源の4つの要素で構成される。

- 通信規格

無線センサデバイスに用いられている代表的な通信規格は、表 2.1 の通りである。

表 2.1: 無線センサデバイスの代表的な通信規格 (出典:Sensor Network Technology [21])

デバイス	RFM TR	ZigBee	Bluetooth
周波数帯	300 ~ 900MHz 帯	2.4GHz 帯	2.4GHz 帯
規格	微弱無線	IEEE802.15.4	IEEE802.15.1
最大チャンネル数	単チャンネル	16 チャンネル	32 チャンネル
最高通信速度	111.5kbps	250kbps	1Mbps
最大通信距離	約 10m	約 30m	約 10m
変調方式	ASK/OOK	DSSS	FHSS

現在、無線センサデバイスに用いられているモジュールは、通信速度が数十 kbps ~ 百数十 kbps 程度の狭帯域なものが多いが、近年は、ZigBee(IEEE802.15.4)[3] をセンサネットワーク用の無線通信方式として利用する動きが見られる。ZigBee は Bluetooth や無線 LAN と同じ 2.4GHz 帯の電波を使っており、省電力性に優れているため、無線センサデバイスに用いる通信規格に適している。

- マイクロプロセッサ

無線センサデバイスに用いられるマイクロプロセッサは、8 ビットあるいは 16 ビット程度の処理能力で、数 MHz 程度の動作周波数の組み込み用マイコンがよく利用される。その理由としては無線センサデバイスのプロセッサの主な役割は、センサからデータ取得とデータの目的地までの配送のみであり、性能をそれほど必要としないためである。

マイクロプロセッサとして最もよく用いられるのは、Microchip 社の PIC マイコンと Atmel 社の AVR マイコンが挙げられる。

- センサ基盤

無線センサデバイスで用いられるセンサ基盤には、組み込み型と取り付け型がある。前者は、センサ基盤がマイクロプロセッサと同じ基盤上に組み込まれているため、センサの種類はある程度限られている。代表的なものに、MillennialNet 社の i-Bean[14] がある。後者は、センサ基盤がマイクロプロセッサ基盤上にあるコ

ネクタピンへ取り付けが可能であり、多くのセンサが用意されている。代表的なものは Crossbow 社 [5] の MTS310 というセンサ基盤で、温度・照度・音・2 軸加速度・2 軸磁気センサがついている。

- 電源

無線センサデバイスは空間のあらゆる場所に偏在させる用途が多く、場合によっては電気の通らないような場所にも配置させることがある。したがって、電源は外部電源ではなくバッテリー電源を用いる。バッテリー電源には、乾電池を用いる場合とボタン電池を用いる場合がある。

2.3.4 無線センサデバイス例

本節では無線センサデバイスの中で特に、本研究で用いる MICA MOTE と uPart という端末について概説する。

MICA MOTE

MICA MOTE はスマートダストの流れを汲み込み、早い段階からハードウェアとソフトウェアの開発が行われ、最も早く市販化がされたデバイスである。最初に発売された MICA MOTE は、マイクロプロセッサに 8 ビットの AVR マイコンを搭載、無線モジュールには RFM 社の TR1000 (表 2.1 参照) が用いられた。そして単 3 乾電池 2 本によって構成されるボードを基本とし、その上にコネクタを介して各種センサが搭載されたボードがとりつけられるようになっている。センサボードには、温度センサ、加速度センサ、光センサ、磁気センサが搭載されており、マイクロフォンとスピーカもボード上に実装されている (図 2.7 左下参照)。また、無線モジュールに Zigbee が用いられている MICAz MOTE (図 2.7 右上参照) は、本研究の実装にも使用しているデバイスである。現在ではさらに、MOTE MICAz のサイズが切手型になった MICAz OME Module (図 2.7 右下参照) という端末も市販されている。

また MICA MOTE には、TinyOS[9] というオペレーティングシステム (OS) が組み込まれている。TinyOS とは、カリフォルニア大学バークレー校で開発された MICA MOTE のハードウェアを制御するための OS である。無線センサデバイス用の OS は、汎用的な OS に比べて要求される役割が違う。ハードウェア資源、特に CPU・メモリ・電力量が極めて限られており、また OS の制御対象であるシステムの役割は外部状態をセンシングし、そのデータを別の場所や装置に送信するのみである。よって TinyOS は、省電力と物理世界とのインタラクションの 2 点に焦点を絞って設計されている。また TinyOS には、NesC[6] と呼ばれる C 言語を拡張した言語が用いられており、取得したセンサデータをどのように処理するかという命令を記述することで、開発者が自由にプログラミングによってカスタマイズできる仕様になっている。

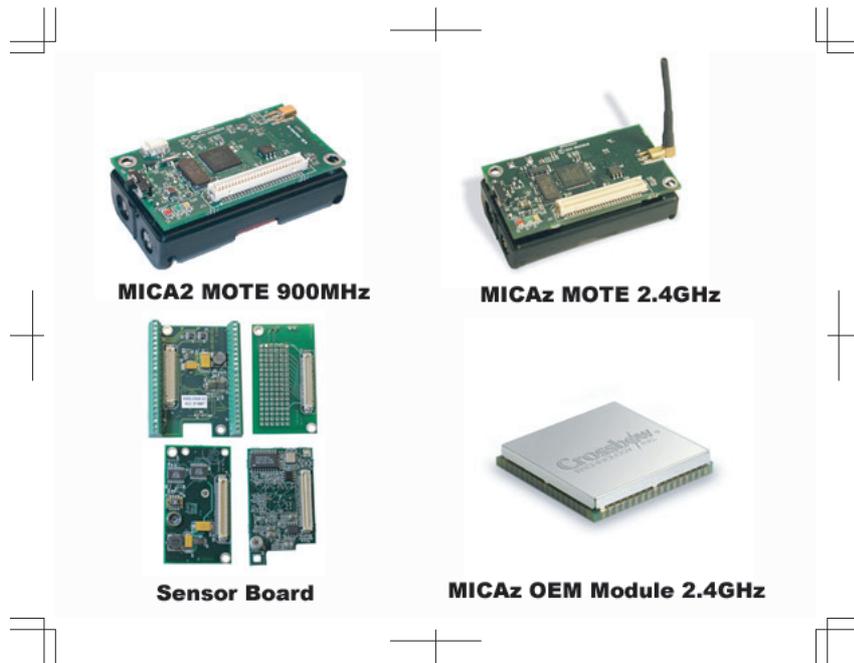


図 2.7: MICA MOTE (出典:Crossbow 社ウェブページ [5])

uParts

uParts[12] は、カールスルーエ大学 TecO[24] で開発された無線センサデバイスである。uPart の最大の特徴は、全長が横 20mm 縦 17mm と超小型であることがあげられる(図 2.8 参照)。また低コストで駆動時間が長いことや、MICA MOTE 等の他の無線センサデバイスに比べ、非常に安価(約 2000 円程度)であることが利点である。

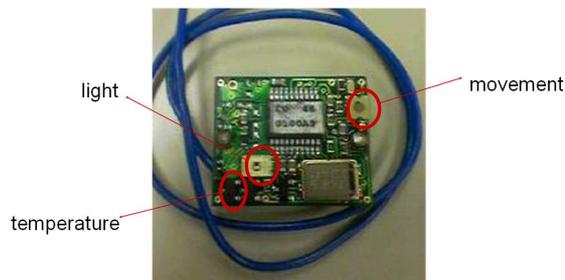


図 2.8: uPart とセンサ

以下に uPart の構成をまとめる。

センサ基盤は組み込み型で、ボールスイッチと温度・照度センサのみとなっている。また CPU の処理能力も MICA MOTE より低く、パケットの送信はできるが受信はできない。また、センサデータプログラミングによる変更もできない。

表 2.2: uPart の構成

CPU	Microchip 12F675 at 4 MHz 1.4KBytes Programming Memory
Wireless	315MHz, 19.2KMbps,
Sensors	Ball-switch Temperature sensor(Microchip TC1047) Light Sensor (Taos TSL13)
Battery	Lithium Coin cell types

デバイス自体の性能は MICA MOTE と比べて劣るものの、その小型軽量さ・安価さ・省電力さは、無線センサデバイスとして十分な要件をそろえているといえる。

2.4 本章のまとめ

本章では研究の利用対象である実空間指向ゲームの概要にふれ、様々なセンシング技術を用いた実空間指向ゲームの研究例について述べた。そして、その中でも本研究の実装で用いる無線センサデバイスについての概要を述べた。

第3章 問題提起

本章ではまず実空間指向ゲームの機能要件について述べ、既存の実空間指向ゲームの問題点をあげる。そしてその問題点を解決する手法として、WWSD手法 (Wearable Wireless Sensor Device 手法) をまず提案する。また本手法の有用性を確かめるために実装したプロトタイプについての説明をする。次に、本手法に関する関連研究やプロトタイプの実装の現状の設計では、WWSD手法を用いた実空間指向ゲームの開発が困難であることを示し、その解決機構として WearMiddle を提案する。

3.1 実空間指向ゲームの機能要件

実空間指向ゲームには、他のゲームには無いいくつかの特徴がある。本節では、実空間指向ゲームを構成するための機能要件について述べる。実空間指向ゲームの開発において重要な要件は、以下の3つが挙げられる。

- 相互作用性

実空間上のプレイヤーがあたかも仮想空間上に自らがいるような没入感を感じ、実空間上のプレイヤーの行動と仮想空間であるゲームの間に相互作用 (インタラクション) があるかどうか。

- 動作性

プレイヤーが実空間内を動き回り、様々なポーズや行動といったアクションを実際に動作できる、少なくとも 10m 四方程度の動作範囲に対応しているかどうか。

- 偏在性

実空間情報を取得するためのデバイスを、実空間へ偏在させることが可能かどうか。

あらゆるゲームの開発において常に相互作用性は求められるが、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されるという点で、実空間指向ゲームの相互作用性は高い。また動作性があるという点は、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。またその分空間にデバイスがいかにか偏在しているかという偏在性をいかにか持たせるかが重要な要件であるといえ、この要件をいかにか実現させるかで、動作性や相互作用性にも影響を及ぼすといえる。よって、これら3つの要件をより多く満たすことのできる実空間指向ゲームの開発が求められる。

3.2 既存実空間指向ゲームの問題点の比較

前章で実空間指向ゲームの既存研究において、様々なデバイスが用いられていることを述べた。本節ではこれらのデバイスを用いた既存研究を前節で述べた機能要件と照らし合わせそれぞれの問題点を指摘し、その中で一番優れているデバイスが無線センサデバイスであることを述べる。

まず、表 3.1 に各デバイスにおいての問題点をまとめた。

表 3.1: 既存実空間指向ゲームの問題点の比較

	相互作用	動作範囲	偏在性
GPS	×		
カメラ		×	
ウェアラブル			×
無線センサデバイス			

相互作用はプレイヤーの具体的な行動や状況をセンシングしている場合は、していない場合を×とした。動作範囲は少なくとも10m四方以上、動作可能なものを、それ以外を×とした。偏在性は、プレイヤーがデバイスの存在に不自然さを感じないものを、不自然さを感じるものは×とした。

3.2.1 GPS

まずGPSを利用しているSpace Raceでは、プレイヤーの動作範囲が広く、すでに携帯端末等にGPSが搭載されていることから装着性は極めて高い。しかし問題点として、GPSはあくまで人の位置のみをセンシングしており、プレイヤーの行動をセンシングしているわけではないため、プレイヤーとゲームとの相互作用が少なく相互作用が十分であるとは言えない。

3.2.2 カメラ

カメラでの画像解析技術を用い、プレイヤーのアクションをそのままゲームのインターフェースとしているAceSpeeder2に関しては、相互作用の問題はクリアしている。また、ゲームセンタやアミューズメントパークにおけるゲームを想定した場合に限り、偏在性もある。しかし、カメラが固定であることが前提であるため、必然的にプレイヤーの行動範囲はカメラが認識できる範囲に限られてしまい、動作範囲が十分だとはいえないという問題点が挙げられる。

3.2.3 ウェアラブルコンピュータ

ウェアラブルコンピュータを用いた HumanPacman は、プレイヤーが広範囲の実空間を動き回ることができ、HMD によってプレイヤーが相互作用による没入感を得ることに成功している。しかし重量デバイスを体に装着しなければならず、プレイヤーの円滑な行動に支障をきたす。

3.2.4 無線センサデバイス

無線センサデバイスを用いた Cubix Monster は、プレイヤーの網を振る行動がそのままモンスターの捕獲という結果に繋がっており、また無線でデータを通信していることからプレイヤーは自由に空間を動き回ることができるため相互作用性、動作性を共に持っているといえる。偏在性においては、網に取り付けている無線センサデバイスが若干大きいサイズのものであるものの（図 2.6 参照）、網を振る際の行動の邪魔にはほぼならないため、偏在性も持っているといえる。

よって無線センサデバイスによる実空間指向ゲームの開発は、実空間指向ゲームの機能要件をすべて満たしているといえる。

3.3 WWSD 手法と問題点

前節で取り上げたように、既存の実空間指向ゲームに用いられているデバイスにはそれぞれの問題点があったが、その中で唯一無線センサデバイスは実空間指向ゲームの要件を満たしていた。Cubix Monster は道具に無線センサデバイスを取り付けるアプローチを取っていた。ただ、無線センサデバイスの取り付け対象を道具にしてしまうと、どうしてもその道具の使用が前提となる実空間指向ゲームになってしまうため、汎用性が少ない。

よって本研究のアプローチは、無線センサデバイスをプレイヤーの体に取り付け、実空間指向ゲームの開発に使用する手法を用いることにする。我々はこの手法を「WWSD 手法」(Wearable Wireless Sensor Device 手法)と名付けた。本節は、その WWSD 手法の概要と本手法の優位性を示すために、WWSD 手法を用いて実装したプロトタイプである「UBI-SHOOT!!-ver1.0-」の概説を行う。また、UBI-SHOOT!!-ver1.0-の実装の際の問題点について述べる。

3.3.1 WWSD 手法の提案

前節で実空間指向ゲームを構成する上での必要な要件として相互作用性・動作性・偏在性の三つを挙げ、既存研究に足りない部分を指摘した。その中で、我々は既存研究の中の無線センサデバイスとウェアラブルコンピューティングのアプローチに着目した。

プレイヤーがセンサデバイスを体に取り付けるというアプローチは、プレイヤーのアクションとゲームとの相互作用が密になるため、ゲームへの没入感を得ることができる。また、プレイヤー自身が無線センサデバイスと一体化して動き回ることが可能なので、動作性という面もかなり自由度が高い。唯一のネックである偏在性の欠如に関しては、従来のウェアラブルコンピューティングに用いられるデバイスが大型なデバイスが多く用いられていることが原因であった。しかし、小型で軽量である無線センサデバイスをプレイヤーの体に装着させることで、プレイヤーがデバイスを装着する際の不自然さを解消できる。よって我々は偏在性を高く保ち、相互作用と動作範囲のメリットを損なわない手法である WWSD 手法を提案する。

2.1 節において、無線センサデバイスの発達により小型軽量・安価・省電力・耐故障性に優れたものが開発されていることは述べた。無線センサデバイスは、Human Pacman に用いられているどのデバイスよりも小さく軽量であり、またバッテリー電源も内蔵しているため長時間稼働し、ウェアラブルに適しているといえる。また無線センサデバイスは、センサデータを受信するためのシンクノードとの通信可能距離が、無線センサデバイスの種類にもよるが 10m ~ 30m あるためプレイヤーの動作範囲について問題点はない。そして前章でも述べたように、無線センサデバイスには加速度・照度・温度などのあらゆるセンサが取り付け可能なため、プレイヤーがどのように動いたかというモーションだけでなく、プレイヤーの置かれている環境情報もセンシングできる。よって、実空間と仮想空間との相互作用の面でも様々な効果が期待できる。

3.3.2 WWSD 手法を用いたプロトタイプの実装

WWSD 手法の有用性を確かめるためのプロトタイプとして「UBI-SHOT!!-ver1.0」を実装した。UBI-SHOOT!!-ver1.0-は、自機と敵機が打ち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 と uPart という無線センサデバイスを用い、トラの着ぐるみにデバイスを取り付けた。プレイヤーは頭上に MICAz MOTE を、手のひらの uPart を取り付け、「移動」、「パワーチャージ」、「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、移動や攻撃というゲームイベントをセンサデータからプレイヤーのコンテキストを解析し処理していた (図 3.1 参照)。UBI-SHOOT!!-ver1.0-に用いたコンテキストの定義表は表 3.2 に示す。

頭部に取り付けた 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たら「右または左移動」、手のひらにつけた uPart 照度センサからのデータがある閾値より低くなったら「光が遮られた状態 = 手を握った状態 = パワーチャージ」というコンテキストの定義をし、同様に、パワーチャージの状態から uPart のボールスイッチセンサが腕の動きを感知したら「手を突き出した状態 = 攻撃」という定義を独

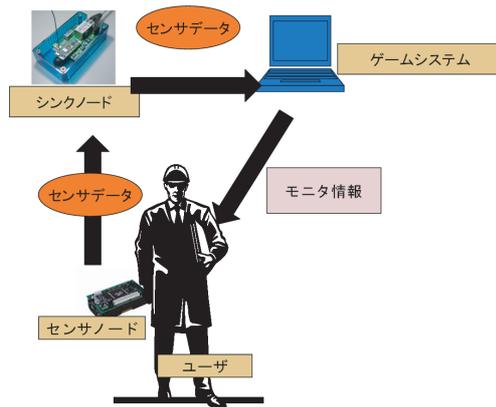


図 3.1: UBI-SHOOT!!-ver1.0- システム図

表 3.2: UBI-SHOOT!!-ver1.0-におけるコンテキスト定義表

コンテキスト	ノードタイプ	位置	行動
左右移動	MICAz MOTE(加速度センサ)	頭上	頭を傾ける
パワーチャージ	uPart(照度センサ)	手のひら	手を握る
攻撃	uPart(ボールスイッチセンサ)	腕	腕を前に突き出す

自に設定した。図 3.2 は「パワーチャージ」状態から「攻撃」状態にコンテキストが移行する際のモーションを、図 3.3 はゲームスクリーンショットを示している。

3.3.3 WWSD 方式の問題点

本節では、UBI-SHOOT!!-ver1.0-から評価した WWSD 手法の問題点について述べる。

コンテキスト定義における問題

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の定義は開発者の試行錯誤の上設計したものであった。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の定義は、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 3.4 の手順を踏まなければならない。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



図 3.2: UBI-SHOOT!!-ver1.0-: 「パワーチャージ」から「攻撃」へ変化時のアクション
(左: 「パワーチャージ」 右: 「攻撃」)

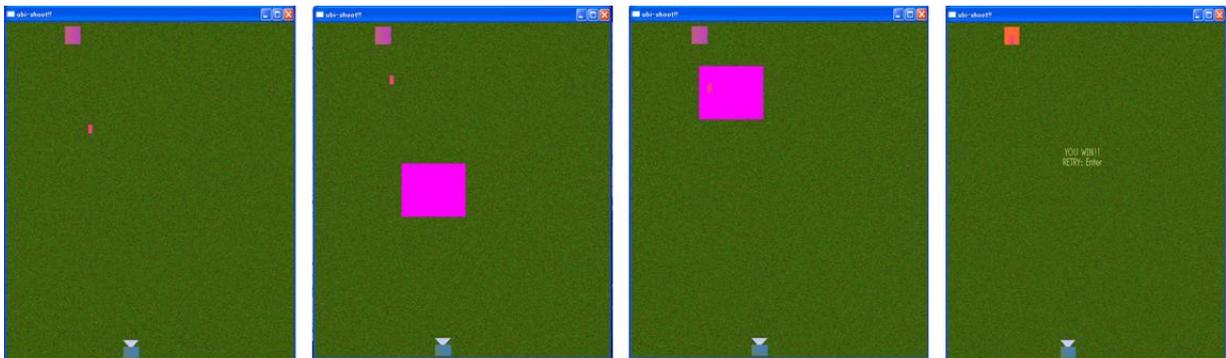


図 3.3: UBI-SHOOT!!-ver1.0-: 「パワーチャージ」から「攻撃」へ変化時のスクリーン
ショット (左: 「パワーチャージ」 右: 「攻撃」が敵機に命中した瞬間)

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、期待どおりの動作がゲームで起これば、コンテキストの定義が終了となる。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ため、開発は困難である。

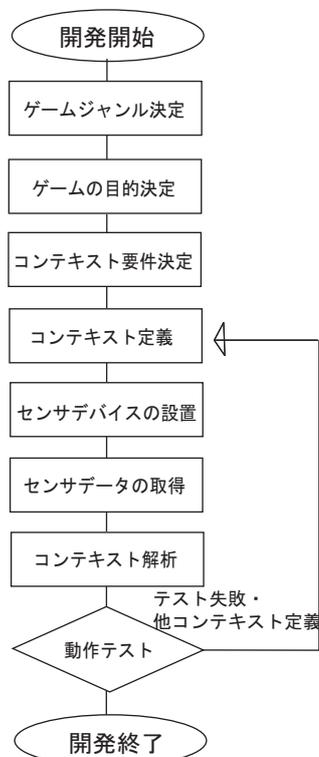


図 3.4: WWSD 方式における開発者の開発ステップの流れ

コンテキスト解析における問題

UBI-SHOOT!!-ver1.0-のシステムは、無線センサデバイスからの生データを直接ゲームシステムで処理していた。しかしこの設計では問題がある。

それはゲームシステムの中で、数多くの無線センサデバイスから送られてくる生データを識別し、加えてコンテキスト解析の作業を行わなければならないという点である。UBI-SHOOT!!-ver1.0-では使用した無線センサデバイスは2種類3個だけであったのでまだ実装可能であったが、より多くの無線センサデバイスを実世界指向ゲームに組み込みたい場合は、実装が困難となる。また、ゲーム開発者が無線センサデバイスからのセンサデータを直接扱うことになるので、無線センサデバイスのハードウェア面、ソフトウェア面やセンサネットワーク等の幅広い知識が要求される。このことは、WWSD手法を用いる実世界指向ゲーム開発の敷居を高くする原因である。

よってこれら問題を解決するためには、2章2節で説明したようなセンサネットワーク用のミドルウェアシステムが必要となる。しかし、ゲームにおいて必要なコンテキストは、必ずしも実生活で頻繁に使用するものとは限らないため、既存のミドルウェアでは WWSD 方式での実世界指向ゲームの開発に適しているものがない、という問題点がある。

3.4 WearMiddle の提案

本研究では、コンテキスト定義における問題とコンテキスト解析における問題を解決する、実空間指向ゲームに特化したミドルウェアシステムを提案する。ゲーム開発者は本機構を用いることにより、WWSD 手法を用いた実空間指向ゲームの開発が可能である。

3.4.1 コンテキスト定義における問題解決

本機構では、コンテキストの定義をゲーム開発者が試行錯誤の上行わなければならない手順を取り除くために、ゲーム開発において必要なコンテキストの定義を決めるコンテキストエディタ機能を用意している。コンテキストエディタ機能を用いることにより、ゲーム開発者が WWSD 手法を用いた実空間指向ゲームの開発を行うためのステップは減少する。具体的には、図 3.4 の繰り返し部分のコンテキスト定義から動作テストまでの作業を WearMiddle は開発者の代わりに行う。

ゲーム開発者は、コンテキストエディタ機能に用意されてるコンテキスト一覧から、ゲームに使用したいコンテキストを選択しコンテキスト定義の詳細を設定するだけでよく、コンテキスト定義をスムーズに行うことができる。

3.4.2 コンテキスト解析における問題解決

本機構では、ゲーム開発者がコンテキストの解析システムを構築する手順を取り除くために、コンテキスト解析機能を有する。

コンテキスト解析機能を用いることにより、ゲーム開発者は本機構から送信されるコンテキストデータのみに着目するだけでゲーム開発ができる。

3.5 センサネットワークにおけるミドルウェアの要件

センサネットワーク研究におけるミドルウェアは、無数の無線センサデバイスからのデータをアプリケーションに適した形で提供し、集約的に管理・制御できるソフトウェアであると考えられる。よってミドルウェアの目的はセンサネットワーク複雑性をユーザに隠蔽することであるといえる。

「センサネットワークのためのミドルウェア技術」[27]には、センサネットワークにおけるミドルウェアの要件がいくつか挙げられている。その中でも、特に必要な要件を以下にまとめた。

- Generic
異なるアプリケーションに対して適用可能な一般性を持ったインタフェースの実現が必要である。
- Adaptive
適応的に自らの構成を変更する機能が必要である。
- Reflective
適応的に自らの振る舞いを変更する機能が必要である。
- Event-based data-centric communication
RPC のようなものとは異なる、イベント駆動の通信プロトコル、および WWW のようなコンテンツ指向の通信機能が必要である。
- Holistic view on the Internet and sensor-networks
センサノードの情報を集約するだけでなく、外部のネットワークへとつなげられる必要がある。
- Application knowledge in nodes
センサネットワークの適用先に関する知識を組み込めるような仕組みが必要である。
- Adaptive fidelity algorithm
利用可能なリソースにあわせて、適応的に、最適な観測精度を実現するような機能が必要である。
- Automatic configuration
センサネットワークと同様に、管理者の関与は最低限で済むような仕組みが必要である。
- Localized Algorithms
スケーラビリティとロバストネスを考慮して、センサノード間の局所的インタラクションで実行されるようなしくみが必要である。

3.6 関連研究

本節では、関連研究としてセンサネットワークを用いたアプリケーション開発に用いられるミドルウェアについて概説する。前節で述べたセンサネットワークにおけるミドルウェアの要件をふまえたミドルウェアの研究例を述べる。

ミドルウェアの種類

以上のようなセンサネットワークにおけるミドルウェアの要件を満たすために、同じく「センサネットワークのためのミドルウェア技術」において5つのタイプのミドルウェアのアプローチがあると述べられている。以下にその5つを述べる。

- イベント処理指向ミドルウェア

逐次的に変化してゆくアプリケーションにおいて、無線センサデバイスが、異常値の検出などのイベントを感知したときに、即座にアプリケーションへ通知するイベントドリブン型のミドルウェア。

- データ管理指向ミドルウェア

無線センサデバイスを実空間に無数に偏在させた場合、それらのデータを如何に管理するかが重要となってくる。

センサデータから、アプリケーションに必要なデータを選択し取得するデータ管理に特化したミドルウェア。

- メッセージ型ミドルウェア

無線センサデバイスからは、一度に複数のデータが、絶え間なく高速に流れていくため、これらのデータをすべて管理することは困難である。これらの問題を解決するデータ抽象化手法の一つに、Publish/Subscribe モデルがある。このモデルでは、Subscribe は購読したい特定の種類のメッセージを事前に予約し、Publish はだれに対してメッセージを送るかは考慮せず、トピックに対してメッセージを発行する。

膨大で高速に流れてくるセンサデータから必要なものを即座に選択するために、高水準な Publish/Subscribe ベースのメッセージ送受信を行うためのミドルウェア。

- QoS 指向ミドルウェア

無線センサデバイスからのデータのみを頼りに実空間の情報を解析するセンサネットワークにおいて、センサデータの品質（信頼性）は重要な問題である。

アプリケーションとの相互関係を保証するための、品質保証に特化したミドルウェア

- インターネット指向ミドルウェア

従来の無線センサデバイスを用いたアプリケーションは、閉じているセンサネットワーク上に構築されているものが多い。

様々なクライアントからアクセス可能なアプリケーションを作るため、インターネットとセンサネットワークを融合させるためのミドルウェア

実空間指向ゲームに用いるミドルウェアの重要な役割は、イベント処理にあるといえる。よって特にイベント処理型のミドルウェアについて研究例をあげる。

DSWare

DSWare[20] は、センサノードの協調によるサービスとそのデータを中心としたセンサネットワークアプリケーションのためのミドルウェアである。DSWare は以下の6つのコンポーネントを持っている。

- DataStorage コンポーネント DSWare は、特定のセンサデバイスのデータをを複数のセンサデバイスが集団で保持することで、負荷の分散や信頼性の向上を図る機能を持つ。
- Data Caching コンポーネント頻繁に呼び出されるデータをキャッシングしておく機能を持つ。
- Group Management コンポーネント近接するセンサのセンサデータの信頼性を相互に監視したり、異常なセンサデバイスをデータ集約機能のための協調処理から排除したりし、センサデバイス群の局所的協調を提供する。
- Event Detection コンポーネント検知するイベントを定義できる。定義できるイベントは「原子イベント (Atomic Event)」と「複合イベント (Compound Event)」の2種類がある。
- Data Subscription 隣接するデバイスへあらかじめデータをコピーしておくことで、転送パスを共通化し、転送に関与するノードを削減する機能を提供する。
- Scheduling コンポーネント他のコンポーネントのスケジューリングを行う。スケジューリングは、イベントに対する実時間性、消費電力を考慮して行われる。

3.7 本章のまとめ

本章では既存の実空間指向ゲームの問題点をあげ、それらを解決する手法として、WWSD 手法を提案した。また、WWSD 手法を用いて実装したプロトタイプについての説明をした。次に、本手法に関する関連研究・開発例やプロトタイプの評価結果を

挙げ、現状では本手法を用いた実空間指向ゲームの開発が困難であることを示し、その問題を解決する機構である WearMiddle を提案した。最後に、センサネットワークに用いられるミドルウェアの関連研究について述べた。

第4章 WearMiddleの設計

本章では、前章で提案した WearMiddle の設計について述べる。まず WearMiddle の概要について述べ、本研究の想定環境について述べる。次に WearMiddle の設計方針を述べ、全体モジュールの概要、各モジュールごとの詳細という流れで述べる。

4.1 WearMiddleの概要

WearMiddle は、複数の無線センサデバイスを対象としたミドルウェアシステムである。WearMiddle は、複数種類の無線センサデバイスからのデータを処理できるセンサデータ統一機能がある。センサデータ統一機能は、無線センサデバイスの種類によって構成が異なるセンサデータを、汎用的に扱える。またコンテキスト定義における問題とコンテキスト解析における問題を解決するため、コンテキストエディタ機能とコンテキスト解析機能も備えている。コンテキストエディタ機能は、コンテキストの解析方法を定義しているコンテキスト定義表を所持しており、本機能を使用してゲーム開発者はゲームプログラムの開発を行う。またコンテキスト解析機能は、センサデータからコンテキストデータへの解析や、ゲーム開発者が作るゲームプログラミングとのコンテキストデータのやり取りを行っている。

4.2 想定環境

本機構が想定している環境について述べる。本機構を利用する上での前提条件として以下の要素をあげる。

- 無線センサデバイスの種類

WearMiddle は本論文で実装するプロトタイプにおいては使用する無線センサデバイスの種類を限定するが、基本的に使用できる無線センサデバイスの種類を限定しないことを想定している。

- 無線センサデバイスの使用個数

使用する無線センサデバイスの個数も基本的に限定しない。ただ、多数の無線センサデバイスをプレイヤーが装着することはプレイヤーの動きやすさの観点から見て現実的ではない。そのため、WearMiddle はプレイヤーが実空間指向ゲームをプレイする上で、必要最低限に動き回ることができる個数として、10 個程度を想定している。

- 無線センサデバイスの設定

無線センサデバイスからの WearMiddle へのデータ送信の際、使用する無線センサデバイスの種類別にポート番号を振り分けることを想定している。無線センサデバイスごとに異なるポートを使用することによって、システムがデータを送信した無線センサデバイスを判別できる。

- ネットワーク環境

WearMiddle は無線センサデバイスで構成されるセンサネットワーク環境を想定している。よって、各無線センサデバイスからのデータを受信するためのシンクノード、そしてシンクノードから本機構へと情報を送信するための IP ネットワークを必要とする。

- 各機能の対象ユーザ

本機構の設計は、ゲーム開発者側とプレイヤー側の 2 つのユーザを対象に別々に行う。コンテキストエディタ機能は、ゲーム開発者側が開発の段階で利用する際に用いられる。センサデータ統一機能とコンテキスト解析機能はプレイヤー側を対象としており、プレイヤーが本機構を用いて開発された実空間指向ゲームを利用する際に用いられる。

4.3 機能要件

前述した設計方針と想定環境に基づき、本機構の設計について詳細を述べる。WearMiddle には大きくわけて、コンテキスト定義表という 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 つの機能とに分割できる。

コンテキスト管理表

WearMiddle は、コンテキスト管理表というデータテーブルを備えている。データテーブルにあらかじめ定義してあるゲームコンテキスト等のデータを、コンテキストエディタ機能とコンテキスト解析機能が参照する。コンテキスト管理表の中身の定義は、ゲームに多く用いられるものから設定していく。また、データテーブル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データを使用する。

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、ゲーム開発者のコンテキスト要求に応答する役割をもつ。従来、ゲーム開発者が WWSD 方式を用いた実空間指向ゲームの開発を行う際、

必要なコンテキストデータをどのように取得するかが問題であった。ゲーム開発者が必要とするコンテキストデータを取得するためには、以下の情報が必要である。

- どの種類の無線センサデバイスを使用するか
- どの ID の無線センサデバイスを使用するか
- どこに無線センサデバイスを配置するか
- どのような動作を行えばいいか
- どのような形でコンテキストデータを取得できるのか

WearMiddle のコンテキストエディタ機能は、開発者のコンテキスト要求に対して以上の 5 つの要素を開発者に提供することによって、コンテキスト定義における問題を解決している。

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の形に統一する役割をもつ。WearMiddle は、何種類もの無線センサデバイスに対応するように想定されているため、それらの規格の違うセンサデータを同等に扱う。よって本機能が必要である。

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、WWSD 方式のコンテキスト解析における問題を解決している。

4.3.1 全体構成

各 3 つの機能から成り立つ WearMiddle の、全体構成図は図 4.1 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。

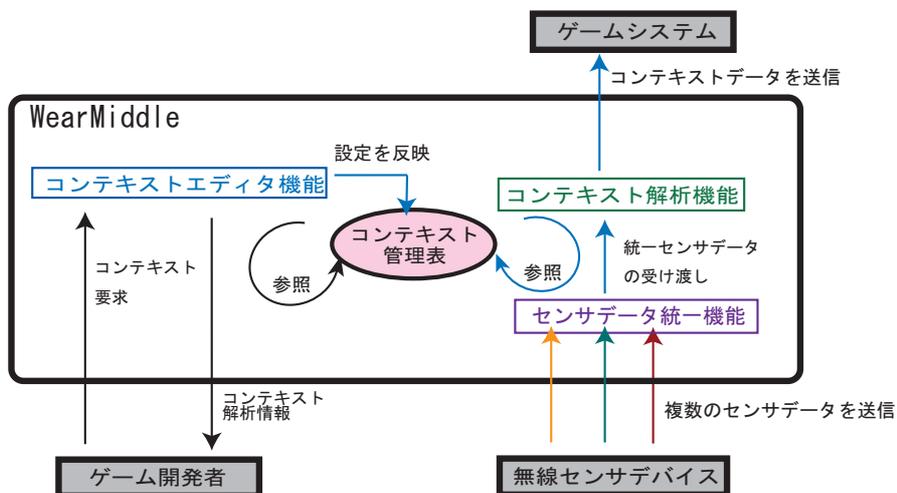


図 4.1: WearMiddle の全体構成図

4.4 モジュールの設計

本節では、前節で述べた WearMiddle を構成する 1 つのデータテーブルと、3 つの機能における詳細なモジュールの設計について説明する

4.4.1 コンテキスト管理表

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 4.1 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単一、複数センサー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。

4.4.2 コンテキストエディタ機能

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。モジュールは図 4.2 に示すように、コンテキスト選択モジュールとコンテキスト詳細設定モジュールによって構成されている。

表 4.1: コンテキスト管理表格納データ構成

項目	説明
ゲームコンテキスト	ゲームに使用できるコンテキスト
デバイスの種類	プレイヤーに設置するデバイスの種類
デバイスのID	プレイヤーに設置するデバイスのID
デバイスの設置位置	プレイヤーがデバイスを設置する位置
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

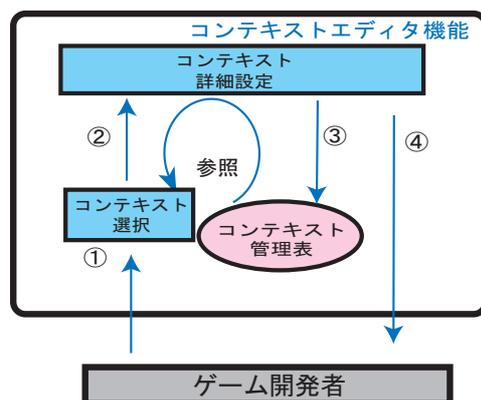


図 4.2: コンテキストエディタ機能のモジュール図

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、WearMiddle を用いて実空間指向ゲーム開発に使用できるゲームコンテキストを、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ゲームコンテキストを選択する。

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 4.1 のように格納されているデータの中には、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、コンテキストの定義を自由に行える。コ

ンテキスト詳細設定モジュールで設定したゲーム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。

4.4.3 センサデータ統一機能

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。モジュール図は、図 4.3 に示す通り受信モジュールとセンサデータ統一モジュールによって構成されている。

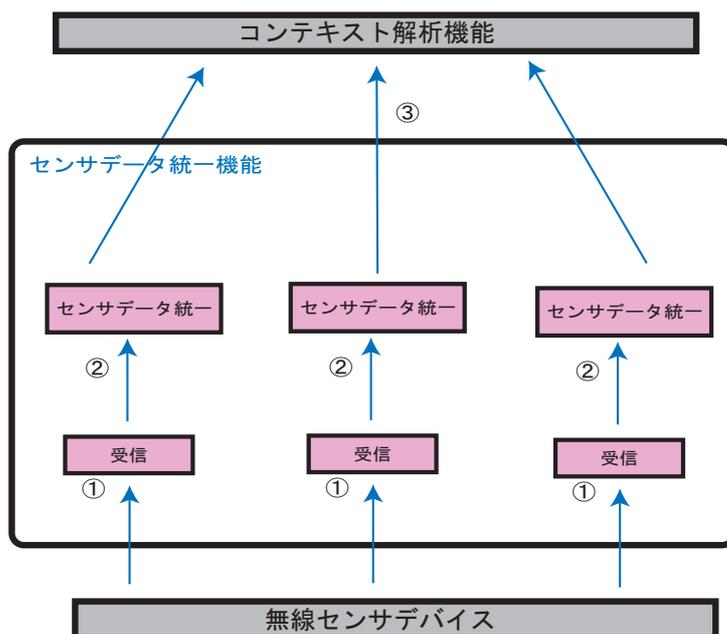


図 4.3: センサデータ統一機能におけるモジュール図

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。図 4.4、図 4.5 は、uPart から送信されるセンサデータの構成で、図 4.6 は、MICA MOTE から送信されるセンサデータの構成である。このように、無線センサデバイスによってセンサデータの構成が異なっているため、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。

設計方針において、あらかじめ無線センサデバイスからの送信データは、種類別にそれぞれ異なるポートに送られるようになっている。センサデータ受信モジュールは、受信ポート別にそれぞれの無線センサデバイスからのセンサデータの構成に合わせて、使用する無線センサデバイスの数だけ実装を行う。

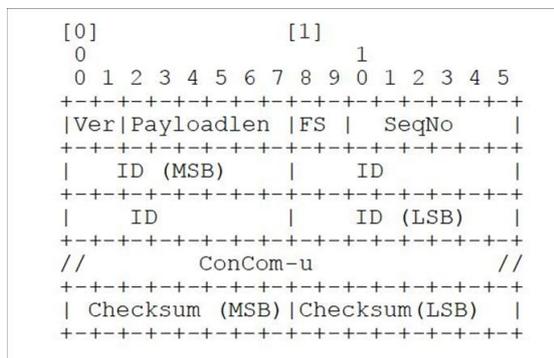


図 4.4: uPart の送信データの構造

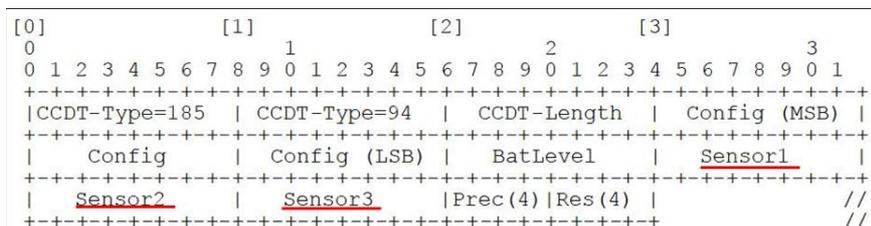


図 4.5: uPart の送信データの concom-u の中身

adress (16bit)	type (8bit)	group (8bit)	length (8bit)	data	CRC (16bit)
----------------	-------------	--------------	---------------	------	-------------

図 4.6: MICA MOTE の送信データの構造

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。

4.4.4 コンテキスト解析機能

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詳細に説明する。モジュール図は図4.7に示す通り、センサデータ解析モジュールと送信モジュールの2つで構成されている。

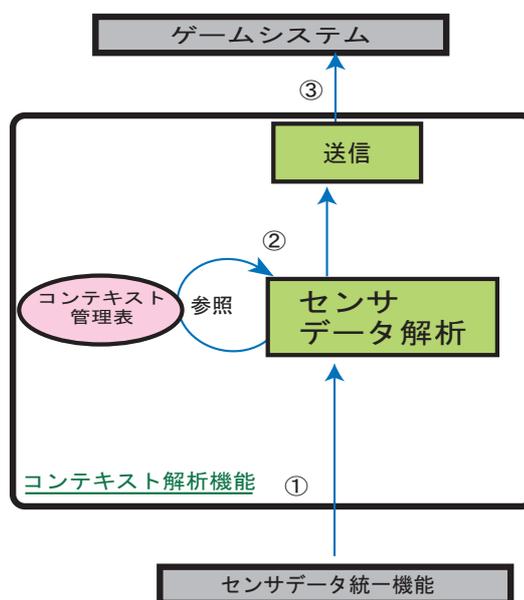


図 4.7: コンテキスト解析機能におけるモジュール図

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し、プレイヤーのコンテキストデータを解析する。そして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、単一センサデータ解析モジュールと複数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。

4.4.5 動作手順

WearMiddle の動作手順を説明する。まず、ゲーム開発者は WearMiddle を起動し、コンテキストエディタ機能を用いてゲーム開発に必要となるコンテキストを用意されている中から選択し、そのコンテキストを解析するのに必要な情報を得て、その情報をもとにゲーム開発を行う。その際、コンテキスト解析システムから送られてくるコンテキストデータを受け取るモジュールをゲームシステムに導入する必要がある。

次に、WearMiddle から得たコンテキスト解析情報を元に、プレイヤーが無線センサデバイスを体に設置し、無線センサデバイスを起動させる。後は、WearMiddle が起動していることを確認した上で、ゲームシステムを起動させることによって、プレイヤーは WWSD 手法を用いた開発した実空間指向ゲームを遊ぶことができる。

4.5 本章のまとめ

本章では、WearMiddle の設計について述べた。まず、WearMiddle の概要について述べ、本研究が想定する環境について述べた。次に本研究にて提案する機構の具体的な設計方針を述べ、全体モジュールの概要、各モジュールごとの詳細を説明した。

第5章 WearMiddleの実装

5.1 実装の概要

本章では、本機構の実装について詳細に述べる。本論文では、WearMiddleのプロトタイプを実装した。まず実装環境について説明し、次に無線センサデバイスの事前実装について述べる。そして次に、4章で設計したモジュールの実装について述べる。

5.1.1 実装環境

WearMiddle は、Java 言語を用いて実装された。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。

5.1.2 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からのセンサデータをゲームシステムに組み混むためには、プレイヤーが行動を起こしてから、直ぐにセンサデータが WearMiddle に送信されなければならないからである。よって、無線センサデバイスのデータ送信間隔はできる限り速く設定した。

各無線センサデバイスにおいての、送信頻度の変更の詳細について述べる。

- MICAz MOTE

MICAz MOTE の TinyOS[6] 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。送信頻度変更部分の疑似コードを以下に載せる。

表 5.1: MICAz MOTE センサデータ送信頻度変更疑似コード

```

int timerrate = 50;
command result StdControl.start(){

    call Timer.start(TIMERREPEAT, timerrate);
    return SUCCESS;
}

```

- uPart

uPart のセンサデータの送信間隔は、図 5.1 に示す TecO の Web サイトのコンフィギュレーションページにおいて変更できる。コンフィギュレーションページで、データ送信間隔の設定変更後に、configure ボタンを押すことで、図 5.1 の黒い部分が点滅する。その当該部分に照度センサを近づけ、感知させることでデータ送信間隔の設定を行う。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。

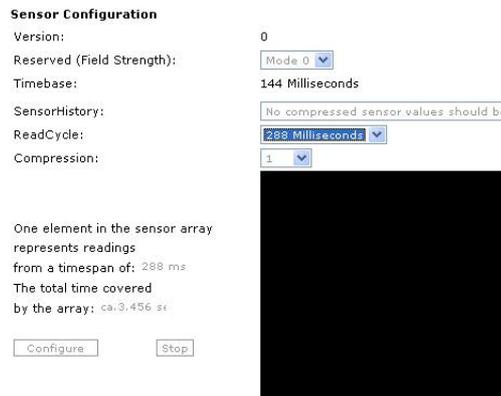


図 5.1: uPart Sensor Configuration ページ ([24] より引用)

5.2 モジュールの実装

本節では、WearMiddle の各モジュールの実装手法を述べる。

5.2.1 コンテキスト管理表

本節は、コンテキスト管理表の実装について説明する。コンテキスト管理表は、表 5.2 のような記述方法をとった。

表 5.2: コンテキスト管理表

左移動,mote,1 2 3, 頭上 肩, 加速度,125, 頭を左に傾ける,"Left"
 右移動,mote,1 2 3, 頭上 肩, 加速度,145, 頭を左に傾ける,"Right"
 ジャンプ,mote,2 1 3, 足の甲, 加速度,140, 踵をあげる,"Jump"
 しゃがむ,mote,4 2 3, 腿の裏, 加速度,140, かがむ,"Squat"

左から順に、ゲームコンテキスト、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。

5.2.2 コンテキストエディタ機能

コンテキストエディタ機能の実装について説明する。コンテキストエディタの起動時のスクリーンショットを図 5.2 に示す。図 5.2 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で、下半分がコンテキスト詳細設定モジュールである。



図 5.2: 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

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、ゲームコンテキストの一覧を表示する。コンテキスト選択モジュールの疑似コードを図 ?? に示す。

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、ゲーム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。このハッシュテーブルは、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ゲーム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ゲーム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、その中で下記項目が表示される。

- 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
- プレイヤの動作
- ゲームシステムに送られる文字列

使用センサと閾値は、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されない。

図 5.3 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に選択肢があり、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。

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、開発者が選択したゲーム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。

5.2.3 センサデータ統一機能

センサデータ統一機能の実装手法について説明する。

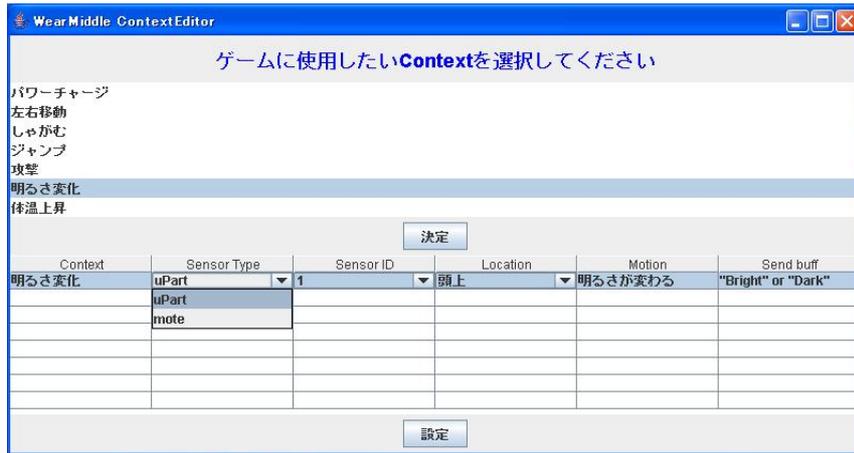


図 5.3: Context Editor コンテキスト選択時スクリーンショット

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される必要がある。MICAz MOTE と uPart のセンサデータを受信するために、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、4.5.2 において述べている送信データの構造に基づいて行った。例として、Mica Mote のセンサデータ統一モジュールの処理を表 5.3 の疑似コードに記す。

表 5.3: センサデータ統一モジュール疑似コード (Mica Mote 用)

```
void moteMessageUnification(Message m, char NodeType){
    MultihopMsg msg = new MultihopMsg(m.dataGet());
    HihorzMsg hm = new HihorzMsg(msg.dataGet());
    String value = String.valueOf(hm);
    int ID = msg.getOriginaddr();

    String buffer = NodeType + ID + value;
```

```
SendtoAnalyzmodule(buffer);
```

```
}
```

まず、受信モジュールから受け取った Mote のセンサデータから、NesC のライブラリを使用して Mote の ID(ID) と数値データ (value) を取得する。そして、無線センサデバイスの種類情報 (NodeType) と Mote の 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 に関しても同様に、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。

5.2.4 コンテキスト解析機能

コンテキスト解析機能の実装手法について説明する。

コンテキスト解析機能は、コンテキストエディタ機能で設定したハッシュテーブルを用いる。

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、統一センサデータに含まれる NodeType と ID、value の値からコンテキストデータを解析する。

センサデータ解析モジュールの一部を疑似コードで図 5.4 に記す。

表 5.4: センサデータ解析モジュール疑似コードの一部

```
for(int i=0; i < ARRAYSIZE; i++){
    Object[] Table = (Object[])hashtable.get(contextList[i]);

    //ノードタイプが一致
    if( Table[1] == NodeType){
        //センサ ID が一致
        if(Table[2] == ID){
            //センサの数値データが閾値以内
            if(value <= Tnale[5] || value >= Table[5]){
                singleContext = Table[0]
                toSendmodule(Table[7]);
            }
        }
    }
}
```

```
} else {  
/* 以下省略 */
```

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、NodeType 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設定した。ただし、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が出てくる。

5.3 本章のまとめ

本章では、WearMiddle の実装について詳細に述べた。まず実装環境について説明し、事前実装について述べた。次に本論文で実装したプロトタイプの全体の概要と、各モジュールの実装について詳細に述べた。

第6章 WearMiddleの評価

WearMiddle の評価において、WearMiddle で実装した ECN-Slider という実空間指向ゲームを用いた。本章ではまず、ECN-Slider の実装について述べ、次にで評価の概要について述べる。

6.1 ECN-Slider の実装

ENC-Slider は、WearMiddle のプロトタイプで提供している、「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用いた、実空間指向ゲームである。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。

そのスクリーンショットを図 6.1 に示す。



図 6.1: ECN-Slider スクリーンショット

ECN-Slider 自体のプログラムは、キーボード操作やマウス操作を想定したものとほぼ変わらない。ゲーム開発者が特別に追加する作業は、WearMiddle のコンテキスト解析機能の送信モジュールから送られてくるコンテキストデータのソケット通信による受信と、受信したコンテキストデータによる条件分岐でゲーム内のイベント判定をさせる作業だけである。

ECN-Slider は Macromedia Flash Professional 8[1] を用いて実装した。

6.2 評価概要

WearMiddle の評価として、WearMiddle で実装した実空間指向ゲームである ECN-Slider を用いて、定量的評価と定性的評価を行う。

WearMiddle の定量的評価は、ECN-Slider 実行時の WearMiddle の処理遅延を計測し評価する。また、WearMiddle の定量的評価は、他の WWSD 方式を用いた実空間指向ゲームとの開発行程の比較を行う。

6.3 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 PC (CPU: Pentium M 1.4GHz, Memory 512MByte, OS: Windows XP) 上で起動する。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 PC と接続されている。

評価手法

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 上でコンテキストデータが反映されるまでの時間を計測した。また、WearMiddle が処理する無線センサデバイスを 1 個から 4 個に一つずつ増やして計測を行った。

評価結果

無線センサデバイスを 1 種類だけ用いたときの、使用個数別処理遅延時間を図 6.2 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実空間指向ゲーム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。

Mote を使用した場合、1 個から 3 個の場合の処理遅延は特に問題なかったが、4 個の場合は多少遅くなった。Mote を 4 個使用した時の Wearmiddle の処理遅延時間は 200ms であり、デバイス自体のデータ送信間隔が 50ms に設定してあることから、プレイヤーの動きが実空間指向ゲームに反映されるまでの遅延が最大で約 250ms あるということに

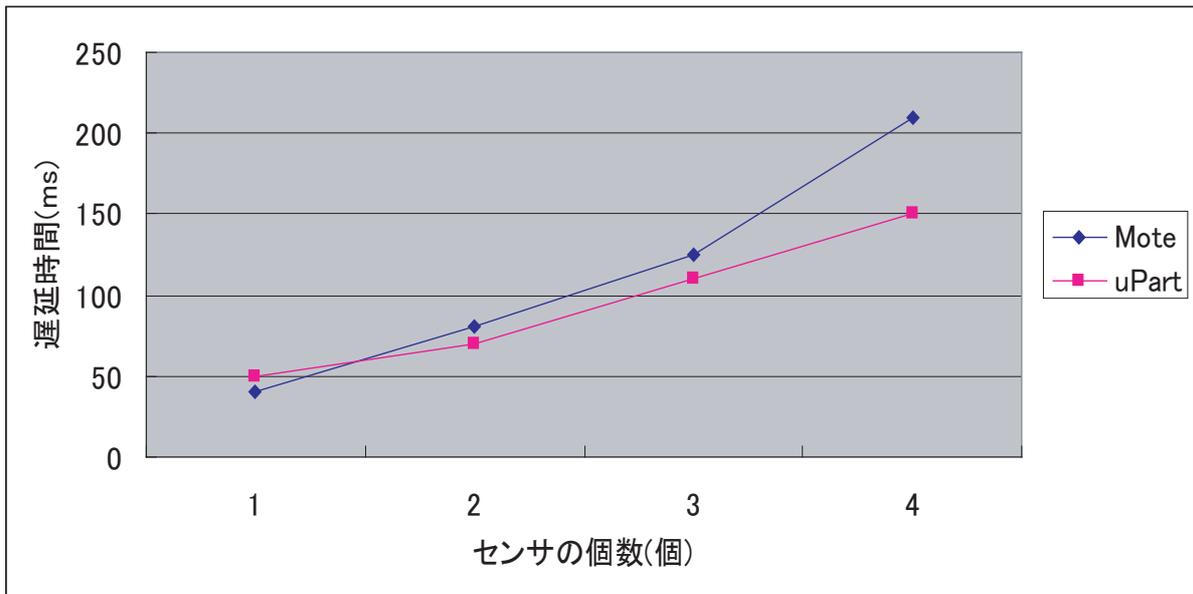


図 6.2: 無線センサデバイスの使用個数別処理遅延時間 (1 種類)

なる。これは、ECN-Slider のように遅延の影響が大きいゲームにおいては、許容範囲限界の遅延であるといえる。WearMiddle で Mote を使用する際は、最大で 3 個までは遅延をほぼ気にせずに実空間指向ゲームの開発が可能である。

また uPart のデバイスの送信間隔が最短で 288ms であることから、1 個の場合でも最大で 330ms 程の遅延時間が発生する。よって WearMiddle 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な遅延がユーザに大きく影響するゲームコンテキストではなく、照度センサによってゲーム内の昼夜が変わる昼夜変更のような、遅延の影響が少ないゲームコンテキストに使用する等の工夫が、現時点では必要であるといえる。

6.4 定性的評価

定性的評価では、関連研究との比較を開発行程の観点から行う。

6.4.1 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、以下の 3 つである。

- 開発者の作業ステップ数

WearMiddle の目的の一つに、WWSD 手法における開発者の作業ステップ数の減少があった。よって、Wearmiddle を使用しない場合との比較を行う。作業ス

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。多いものは×、少ないものは とした。

- 開発者の無線センサデバイスの知識の必要性

無線センサデバイス知識の必要性は、ゲーム開発者が無線センサデバイスについての深い知識がなくとも開発が行えるか、ということである。WWSD手法を用いた実空間指向ゲーム開発の敷居を低くするための重要な要素であるため、評価基準とした。知識が必須な場合は×を、ほぼ必要としないは とした。

- 開発者のネットワークへの知識の必要性

ネットワーク知識の必要性は、ゲーム開発者がネットワークやネットワークプログラムの知識がない場合の開発が困難かどうか、ということである。これも、WWSD手法を用いた実空間指向ゲーム開発の敷居を低くするために重要な要素であるため、評価基準とした。知識が必須な場合は×を、ほとんどいらぬものは を、多少の知識が必要とされるものは とした。

比較結果を、表 6.1 に記す。

表 6.1: 他 WWSD 手法実空間指向ゲームとの比較評価

	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
ユビキタスマンスター	×	×	×
UBI-SHOOT!!-ver1.0-	×	×	×
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のための試行錯誤を開発者は行っておらず、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意識をしないでよいため、表 6.1 のような評価結果となった。WearMiddle を用いたゲームの開発では、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、簡単な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、他の実世界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。

6.5 本章のまとめ

本章では、本論文にて提案する WearMiddle の評価を行った。その結果、WearMiddle は WWSD 手法を用いた実空間指向ゲームの開発を容易にできることが分かった。ま

た、WearMiddle の処理遅延時間から、現状では 4 個程度の無線センサデバイスの使用には十分耐えうることが分かった。

第7章 結論

最後に本研究の結論を述べる。まず今後の課題を述べ、次に本論文のまとめを述べる。

7.1 今後の課題

本節では、本研究の今後の課題点を列挙しその詳細について述べる。

遅延の減少

本研究で行った WearMiddle の実装では、WearMiddle で無線センサデバイスを4個使用した状態では、プレイヤー動作から、実世界指向ゲームの反応までに250msから300msまでの遅延が生じることが分かった。これらの原因として、無線センサデバイスのデータ送信頻度の限界や、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れや、コンテキストの解析における遅延が挙げられる。今後の課題は WearMiddle の設計を見直し、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や、コンテキストの解析時の探索を高速化するアルゴリズムを導入することである。

汎用性の向上

本研究で実装した WearMiddle は、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、また実装コンテキスト数も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。今後の課題は、コンテキスト管理表の内容をより汎用的にすることである。

開発者の自由度向上

本研究では、無線センサデバイスをゲーム開発者になるべく意識させない事が目的の一つだったため、ゲーム開発者による無線センサデバイスを用いた独自のコンテキスト作成の自由度が低かった。よって今後の課題は、開発者側が自由にコンテキストの定義を行い、コンテキスト管理表を編集可能にし、ゲーム開発者が利用可能なコンテキストを自由に設定できるようにする。

7.2 まとめ

本研究では、実空間指向ゲームの開発において、無線センサデバイスを体に装着する WWSD 手法を提案した。そして WWSD 手法を用いたゲーム開発を支援する機構として WearMiddle を提案した。まず、現状の実空間指向ゲームの開発に用いられるセンシングデバイスの問題点をあげ、無線センサデバイスを体に装着する WWSD 手法の必要性について述べた。また、WWSD 手法特有のコンテキスト定義問題とコンテキスト解析問題について述べた。これらの問題定義に基づき、WWSD 手法を用いた実空間ゲームの開発に特化したミドルウェアシステムである WearMiddle を提案し、その設計を行った。コンテキスト管理表というデータテーブルと、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つの機能を設計し、実装した。評価は、WearMiddle を使用して実装した ECN-Slider を使い、アンケートによる定量的評価と関連研究との比較による定性的評価を行った。WearMiddle を用いることで WWSD 手法による実空間指向ゲームの開発ステップが少なくなり、開発への敷居は低くなるため有用である。今後の課題として、遅延の減少、汎用性の向上、開発者への自由度を向上をし、WearMiddle の改良を進める。

謝辞

本研究を進めるにあたり、御指導を頂きました、慶應義塾大学環境情報学部教授徳田英幸博士に深く感謝致します。

また、貴重な御助言を頂きました間 博人氏、斉藤 匡人氏、高橋 ひとみ氏、青柳 禎矩氏、金田 裕剛氏、榊原 寛氏をはじめとする慶應義塾大学徳田研究室 ECN 研究グループの諸氏、また徳田・村井・楠本・中村・高汐・湧川研究室の方々には、研究会の活動を通じて多くの御意見、御助言を頂きましたこと拝謝します。

そして大学生活の中で、常に苦楽を共に過ごした横山 宗幸氏、伊藤 佑奈氏、飯塚直氏、大森 裕介氏、また中学時代から数々の助言と協力を頂いた、山岸 勇太氏、坂本 義和氏、その他多くの友人に心から感謝します。

最後に、私が望んだ道に進むことをあらゆる面で支え、今日まで不自由なく生きる環境を与えてくれた家族に心から深謝と敬愛を表し、謝辞と致します。

2007年1月31日

渡邊 裕一郎

参照論文

- Yuichiro Watanabe, Sadanori Aoyagi, Hitomi Takahashi, Hiroto Aida, Hideyuki Tokuda.
“UBI-SHOOT!! -ver1.0- : A Context-Aware Game Using Wearable Sensor-Networks”
INSS2006 :Third International Conference on Networked Sensing Systems,
May. 2006.
- 渡邊 裕一郎, 青柳 禎矩, 高橋 ひとみ, 間 博人, 徳田 英幸.
“UBI-SHOOT!! -ver1.0- : A Context-Aware Game Using Wearable Sensor-Networks”
日本ソフトウェア学会 SPA2006,
2006年3月.

参考文献

- [1] Adobe. *Macromedia Flash Professional 8*: <http://www.adobe.com/products/flash/flashpro/>, 2005.
- [2] Akihiko Shirai. Gpuviewer : <http://akihiko.shirai.as/projects/gpuvision/>.
- [3] ZigBee Alliance. <http://www.zigbee.org/>.
- [4] Adrian David Cheok, Siew Wan Fong, Kok Hwee Goh, Xubo Yang, Wei Liu, Farzam Farzbiz, and Yu Li. Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal And Ubiquitous Computing*, Vol. 8, No. 2, 2004.
- [5] Crossbow. <http://www.xbow.com/>.
- [6] D.Gay and P.Levis and R.Brewer and D.Culler. The nesC Langage :A Holistic Approach to Network Embedded System. In *Proceedings of the International Conference on Management of Data*, 2003.
- [7] 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.
- [8] EnterBrain. *Weekly Famitu*, 2006.
- [9] 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.
- [10] J.M.Kahn, Kazu R.H, and K.S.J.Pister. Mobile Networking for Smart Dust. In *Proc. of the 4th International Conference on Mobile Computing and Networking*, 1999.
- [11] Mark Weiser. "The Computer for the Twenty-First Century". In *Scientific American*, pp. 94–104, September 2001.
- [12] Micheal Beigl, Christian Decker, Albert Krohn, Till Riebel, Tobias Zimmer. upart: Low Cost Sensor Networks at Scale. In *UBICOMP 2005*, 2005.

- [13] Micro Soft. *XBOX360*: <http://www.xbox.com/ja-JP/hardware/xbox360/>, 2006.
- [14] Millennial Net 社. <http://www.millennial.net/>.
- [15] Newt Games. *モギィ アイテムハント* :<http://www.mogimogi.com/>.
- [16] Nintendo. <http://www.nintendo.co.jp/>.
- [17] Nintendo. The legend of zelda -twilight princess- :<http://www.nintendo.co.jp/wii/rzdj/index.html>.
- [18] NINTENDO. *Wii*: <http://www.nintendo.co.jp/wii/>, 2006.
- [19] RainGraph. *Aceslider2* :<http://www.raingraph.com/>.
- [20] S. Li and S. Son and J. Stankovic. Event Detection Services Using Data Service Middleware in Distributed Sensor Networks. In *In Proc. of the 2nd International Workshop on Information Processing in Sensor Networks*, pp. 502–517, 2003.
- [21] Ando shigeru, Tamura Yosuke, Tobe Yoshito, and Minami Masateru. *Sensor Network Technology*. Tokyo Denki University Publication Service, 2005.
- [22] Sony Computer Entertainment Inc. *EyeToy*: <http://www.eyetoy.jp/>, 2004.
- [23] Sony Computer Entertainment Inc. *Play Station 3*: <http://www.jp.playstation.com/ps3/>, 2006.
- [24] TecO. <http://particle.teco.edu/>.
- [25] T.Okoshi and S.Wakayama and Y.Sugita and S.Aoki and T.Iwamoto and J.Nakazawa and T.Nagata and D.Firusaka and M.Iwai and A.Kusumoto and N.Harashima and J.Yura and Y.Ikeda and H.Tokuda. Smart Space Laboratory Project : Toward the Next Generation Computing Environment. In *IEEE International Workshop on Networked Appliances*, 2001.
- [26] 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.
- [27] 寺島明男. *無線センサネットワークの研究動向 - センサネットワークのためのミドルウェア技術を中心として*, 2004.
- [28] 川西直, 川原圭博, 森川博之, 青山友紀. *実空間センサ情報を用いたモンスター収集ゲームの試作*. 電子情報通信学会通信ソサイエティ大会, 2004.

- [29] 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.
- [30] 川西直, 川原圭博, 板生知子, 森川博之, 青山友紀. 実空間センサ情報を用いたエンターテインメントアプリケーションの実現機構. 電子情報通信学会通信ソサイエティ大会, Aug 2003.