

## Flash コンテンツクリエイタのための 体感型コンテンツ作成支援フレームワーク

田中宏平<sup>†1</sup> 岸野泰恵<sup>†2</sup> 寺田 努<sup>†3</sup>  
宮前雅一<sup>†4</sup> 西尾章治郎<sup>†1</sup>

センサ技術を活用し、実空間のモノを操作することでコンピュータを扱う体感型コンテンツが注目を集めている。しかし、センサ技術を用いたコンテンツ制作のためには、センサ接続に関する知識やセンサデータの扱い方に関する知識が必要となる。また、一度作成したコンテンツに対しても、用いるセンサを変更したり、新たにセンサを追加することが困難であった。そこで本研究では、センサの扱いに関する知識に乏しいコンテンツ作成者が、体感型コンテンツを制作することを支援するフレームワークを実現した。提案フレームワークでは、スクリプト制御が可能で多くの Web コンテンツに活用されている Flash を用いてコンテンツを作成することを想定し、コンテンツ作成者が入出力として利用したいユーザの動作や状況を選択するだけで、体感型コンテンツに必要な Flash スクリプト、装着型センサに関する設定を行えるフレームワークを実現する。提案フレームワークのプロトタイプ実装を用いて複数のコンテンツを作成し、体感型コンテンツが容易に作成できることを確認した。

### A Framework for Flash Contents Creators to Construct Flash Contents

KOHEI TANAKA,<sup>†1</sup> YASUE KISHINO,<sup>†2</sup> TSUTOMU TERADA,<sup>†3</sup>  
MASAKAZU MIYAMAE<sup>†4</sup> and SHOJIRO NISHIO<sup>†1</sup>

Recently, multimedia interactive contents that can be dealt intuitively by motions attract a great deal of attention. The systems realizing such contents detect user motion by using various sensors. However it is difficult for contents creators to learn how to deal with sensors and their raw sensing data. In addition, it is also difficult to modify completed contents because of the change/add sensors. In this paper, we propose a framework that helps content creators who do not have enough knowledge of sensor management to deal with sensors. Our framework assumes that the contents are created by using Flash that is popular for creating multimedia contents for Internet distribution. Using our tool that supports complex transactions of interactive contents creation, the things that the creator does are selection of contexts and gestures that use for input. We confirmed that creators can create contents easier through creations of various contents by using a prototype system.

#### 1. はじめに

従来のコンピュータは、キーボード、マウスやジョイスティックなどの入力機器を用いて操作することが一般的であったが、近年、センサ技術を用いた直観的なインタフェースが注目を集めている。特にゲーム業界において、2006年冬に任天堂から発売された Wii は、加速度センサ、CMOS センサを備えたコントローラをラケットや剣に見立ててゲームを直観的に楽しむことを可能とし、新たなゲーム市場の獲得に成功した。このような直観的なインタフェースは、複雑化する操作の敷居を下げるだけでなく、実際に行動することによる運動不足

の解消など、新たなコンピュータの側面を形成している。しかし、このようなセンサ技術を用いたコンテンツを制作するためには、センサの接続に関する知識やセンサデータの処理方法に関する知識が必要となり、従来の開発者にとって、その敷居は高い。また、一度作成したコンテンツに対して、利用するセンサの変更や新たなセンサの追加を行うことは、コンテンツのプログラムを書き換える必要がある。

一方で、近年の Web の普及に伴い、Web ブラウザ上でアニメーションやゲームを楽しむための Flash<sup>1)</sup> が普及している。Flash は、内容に対してのファイルサイズが小さいことや、幅広いプラットフォームに対応しているため、インターネットとの親和性が高い。2008年6月現在、Flash を再生するための Flash Player は、アメリカ、日本などでインターネットに接続できるパソコンの 99%以上で利用されており、ほとんどのユーザが Flash コンテンツを利用できる環境にある。さらに Flash は、ActionScript と呼ばれるスクリプト言語を利用して、ユーザの入力を受け付けたり、画像を操作するなどのプログラミングを容易に行えることから、プロ・アマチュアを問わずインタラクティブなコンテンツ制作によく

<sup>†1</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology, Osaka University

<sup>†2</sup> NTT コミュニケーション科学基礎研究所  
NTT Communication Science Laboratories

<sup>†3</sup> 神戸大学大学院工学研究科  
Graduate School of Engineering, Kobe University

<sup>†4</sup> ウェストユニティス株式会社  
WESTUNITIS Co., LTD.

```

WHEN GPS_MOVE
IF CURRENT.Pos == 'station'
THEN DO BROWSER_OPEN('URL')

```

図 1 ECA ルールの例

用いられている。

そこで本研究では、センサの扱いに関する知識が乏しい Flash コンテンツ制作者が、センサやアクチュエータを用いた体感型コンテンツを作成することを支援するフレームワークを提案する。本研究で実現したコンテンツ作成の支援ツールを用いることで、コンテンツ作成者が入出力として用いたいユーザの動作や状況を選択するだけで、体感型ゲームに必要な Flash スクリプトの出力や、装着型センサの利用に必要な設定を行える。

以下、2章で本研究の想定環境を定義し、3章で関連研究を述べる。4章で提案フレームワークの詳細について説明し、5章で提案機構を用いたコンテンツ制作について説明する。6章で考察を行い、最後に7章でまとめる。

## 2. 想定環境

本研究で想定しているユーザは、センサの接続に関する知識やセンサデータの処理方法に関する知識は十分でないが、Flash コンテンツは制作できる知識があるユーザである。近年、Flash を用いた Web コンテンツが数多く作成されており、このようなユーザも多数存在する。

また本研究では、ユーザの動作やコンテキストを取得・定義するために筆者らの研究グループで開発した Wearable Toolkit<sup>4)</sup>を用いる。Wearable Toolkit は、イベント駆動型のルール処理エンジンとその関連ツールで構成される状況依存システム作成支援キットである。ルール処理エンジンで処理されるルールは、発生する事象 (Event)、実行条件 (Condition)、実行する動作 (Action) の3つを一組とした ECA ルールにより記述する。例えば、「ユーザが駅に到着した時に、時刻表を表示する」システムを作成する場合には、図1に示すように、GPS 電波を受信したときに (WHEN) その値が駅付近であれば (IF)、時刻表の Web ページを表示する (THEN DO)、という一つのルールで実現できる。さらに、Wearable Toolkit には図2に示すような、ユーザの動作や状況を定義するコンテキスト定義ツールが付属している。このツールは、簡単にジェスチャや状態をイベントとして定義するためのツールである。ツールでは、コンテキストとして定義したい動きを実際に行うことで、半自動的にコンテキストを定義できる。また、システム利用時にはこのツールが状況認識を行い、状況の変化をイベントとしてルール処理エンジンに伝えている。現在、コンテキスト定義ツールでは、3軸加速度センサ、温度センサ、GPS、RFID タグリーダなどのデバイスをサポートしている。

## 3. 関連研究

センサなどの様々なデバイスを自作プログラムの中で容易に利用するための機構はこれまでも数多く存在している。

Phidgets<sup>3)</sup> はボタンやスライド、ミニジョイスティック、タッ

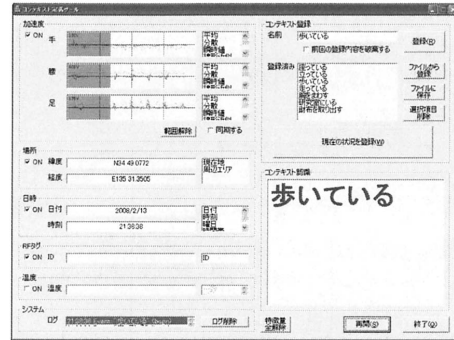


図 2 コンテキスト定義ツール

チセンサ、光センサ、RFID リーダなどの各種センサや、サーボモータや LED などのアクチュエータを統一したインタフェースで扱えるようにしたパッケージである。一般発売されており、Java, Visual Basic, C++, Delphi など数多くのプログラミング言語用のライブラリが公開されているため、プログラミングの知識があればセンサを用いた任意のアプリケーションを開発できる。

Teleo<sup>2)</sup> は MakingThings 社の製品で、Phidgets と同様いくつかのモジュールを組合せることで、各種センサやモータを入出力としたアプリケーション開発を支援するパッケージである。C や C++ などで作成したアプリケーションから直接 Teleo を利用するための開発キット、音楽/マルチメディア向けの統合開発環境としてメディアアート分野で多用されている Max/MSP 用のオブジェクトも提供されており、簡単に利用できる。

Gainer<sup>5)</sup> は、既存の各種センサやモータを自由に接続し、活用するためのプラットフォームである。Gainer を用いることで、インタフェースやメディア・インストールの開発を行える。Gainer では、Phidgets や Teleo と異なり、専用のセンサを用いるのではなく、ハードウェアの設計を自分で行うための機構もあるため、自由にセンサやモータを利用できる。Max/MSP や Processing, Flash との連携を行うためのライブラリも提供されており、容易にセンサを用いたアプリケーション開発を行えるようになる。

これらの機構を用いることでセンサを用いた幅広いアプリケーション開発が行えるが、どの機構もアプリケーション作成時のセンサの接続を容易にするためのものであり、アプリケーション開発にはセンサデータの扱い方に関する知識が必要となる。本研究で提案する機構では、Flash コンテンツ制作者は、センサの接続だけではなくセンサデータの処理方法に関する知識も十分ではないと考え、センサデータを用いたユーザの状況や動作の認識までを提案機構で扱うことで、アプリケーション開発時にセンサデータを意識する必要がない。

## 4. 提案機構

本章では、提案するフレームワークの設計について述べる。

Flash を用いた体感ゲームは、図3に示すような構成で動作する。Flash 上から直接利用できないセンサなどの様々な

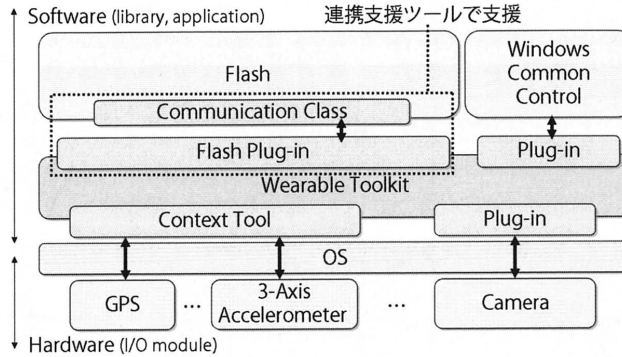


図 3 提案機構の位置づけ

ハードウェアを Wearable Toolkit が仲介することで、体感型ゲームなどの Flash コンテンツ作成が容易になる。一方、Wearable Toolkit には Flash の状態を取得する機能はなく、Flash も Wearable Toolkit と通信することは容易ではない。そこで、図中における Flash Plug-in と通信を行うクラス、および連携を支援するツールを開発することでフレームワークを実現する。

#### 4.1 処理の流れ

Flash 上でセンサデータから算出されるユーザの状況や動作を利用するための Wearable Toolkit との連携は、次の手順で行われる。

- (1) Wearable Toolkit を起動
- (2) Wearable Toolkit から Flash ゲームを起動  
Wearable Toolkit は、Flash Plug-in を介して Flash の通信クラスと通信し、Flash ゲームに記述されたイベント・アクション定義情報を取得する
- (3) 取得した情報をもとに、ECA ルールのイベント・アクションを定義
- (4) ユーザが何らかの動作を行ったときに、その動作が Context Tool により認識され、3. で定義したアクションに基づいて関連情報とともに Flash ゲームに送信される
- (5) Flash ゲームの通信クラスは受信した情報をもとに、プログラマが記述した ActionScript を呼び出し処理を実行
- (6) Flash 側から Wearable Toolkit のイベントを発生させる場合、Flash コンテンツ内で、2. で定義したイベントに対して、RaiseEvent コマンドを実行する。イベント発生情報は、通信クラスを介して Flash Plug-in に送信され、Flash Plug-in が該当イベントを発生させる

以下で、2. から 6. の処理についての詳細を説明する。

##### 4.1.1 Flash 処理の追加

Wearable Toolkit で認識したコンテキストを Flash で利用するためには、Wearable Toolkit における ECA ルールのアクション部で Flash コンテンツ内の処理を指定できるようにする必要がある。ただし、Flash コンテンツが行える処理はコンテンツごとに異なるため、処理 2. の段階で、Flash コ

```
RegisterAction(
    "ACTION_NAME", OnAction, 説明, 引数の型);
function OnAction(Arg:引数の型){
    //ここに ACTION_NAME の処理を記述
}
```

図 4 アクションの登録

```
RegisterEvent(
    "EVENT_NAME", 説明, 引数の型);
```

図 5 イベントの登録

ンテンツが行える処理をアクションとして Toolkit に登録する。この処理は、図 4 に示すようなスクリプトを Flash コンテンツのプリアンブル部に記述しておくことで実行される。

処理 4. で示した、ユーザの動作に応じて Flash 処理を実行する仕組みは、Toolkit から Flash コンテンツを再生しているコントロールに対して SetVariable() 関数を実行することで実現する。SetVariable() 関数は、Flash コンテンツ内の特定の 변수を書き換える関数であり、その変数が変化した時に呼ばれるコールバック関数を、プログラマが記述した特定の ActionScript に関連付けることで処理を実現する。

##### 4.1.2 Flash イベントの登録

Flash の状態変化を Wearable Toolkit に伝えるためには、状態変化が発生した際に通信を行う fscommand() 関数を利用する。Flash コンテンツが発生させるイベントは、処理 2. の段階であらかじめ Toolkit に送信して登録する。この処理も、図 5 に示すようなスクリプトを Flash コンテンツに記述しておく。

また、イベントの発生を Toolkit に送信するためには、Flash コンテンツ内で RaiseEvent を呼び出す。

#### 4.2 連携支援ツール

Flash と Wearable Toolkit を連携させるためには、前節で述べたスクリプトの記述が必要であるが、そのためには Wearable Toolkit と Flash を連携させる仕組みについて知る必要がある。そこで、スクリプトの記述や必要なルールを



図 6 連携支援ツール

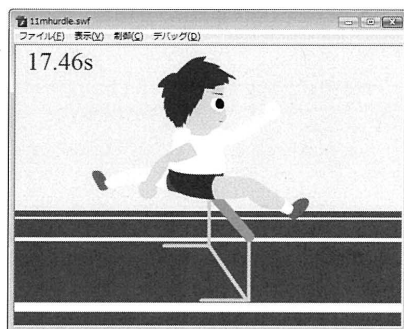


図 7 110m ハードル競技ゲーム

自動生成できる連携支援ツールを実装した。連携支援ツールは、図 6 に示すようなインタフェースで、コンテンツ設計時に考えた、どのようなコンテキストを認識したときにどのような処理を行うか、Flash がどのような状況になった際に Toolkit でどのような処理を行うかをコンボボックスから選択するだけでスクリプトやルールを自動生成できる。

図中の出力ボタンを押すことで、Wearable Toolkit と Flash コンテンツの連携のために必要なスクリプトが記述された状態の Flash ファイルを出力できる。

## 5. アプリケーション実装例

本章では、提案機構を用いて具体的にどのような手順でコンテンツ開発を行うかを図 7 に示す、110m ハードル競技の体感ゲームを作成した例を交えて説明する。

体感ゲーム開発における主な流れは、以下の通りである。

**Step1** ゲームの設計

**Step2** コンテキスト・動作の登録

**Step3** Wearable Toolkit のルール定義

**Step4** Flash ゲームの作成

**Step5** デバッグ、テスト

提案機構は、Flash コンテンツ制作者にとって敷居の高い Step2, Step3, Step4 について支援する機構である。各ステップについて、以下で詳細に説明する。

### Step1 ゲームの設計

このステップでは、作成するゲームの全体像を考える。ゲームの目的やターゲットなど、通常のゲーム制作で考える点に加えて、体感ゲーム制作においては、特にコンテキスト・動作をゲームにどのように利用するのか、Flash コンテンツで発生したイベントを Toolkit でどのように活用するかを明確にする必要がある。

例えばハードル競技ゲームの場合、画面の合図によってゲームがスタートし、110mの距離を走りきるまでのタイムを競うという、ゲームの趣旨や目的に加え、プレイヤーが「走る」、「止まる」時にはゲームキャラクターも「走る」、「止まる」、プレイヤーの「ジャンプ」時には、キャラクターを「ジャンプ」させるという、ユーザ動作とゲーム動作の関連付けを行う。また、ゲームキャラクターがハードルに引っかかった際に、振動子にフィードバックする。

### Step2 コンテキスト・動作の登録

ゲームに利用するコンテキストを Wearable Toolkit のコンテキスト定義ツールを用いて登録する。

まず、コンテキスト定義ツールを起動し、実際にゲーム中で利用したいプレイヤーの動作を行う。登録ボタンを押していくつかの簡単な質問に回答すればコンテキストが登録される<sup>6)</sup>。さらに、同ツールを用いて、目的の動作が正しく認識できているかを確認する。

ハードル競技ゲームの例では、この段階で「走る」、「止まる」、「ジャンプ」のコンテキストをコンテキスト定義ツールを用いて登録し、正しく認識できていることを確認しておく。

### Step3 Wearable Toolkit のルール定義

Wearable Toolkit のルール定義は、連携支援ツールを用いることで容易に行える。Step1 で定義した連携を連携支援ツールを用いて入力する。

具体的にハードル競技ゲームの場合は、「走る」コンテキストに変化したというイベントが発生した際に、ゲームキャラクターを「走る」に変更するため、図 8 に示すように、イベントにコンボボックスから、走るコンテキストを認識を選択し、FLASH\_GAME\_RUN というアクションを実行するように記述する。このような作業を行うことで、対応する ActionScript、および Wearable Toolkit で用いるルールが生成される。同様に、「止まる」、「ジャンプ」も定義する。

また逆に、Flash コンテンツの内容をイベントとする連携も定義する。具体的には、「ハードルにつまづく」イベントを定義する場合、イベント名は自由に設定し、アクションとしてコンボボックスから振動子を振動するアクションを選択する。

### Step4 Flash ゲームの作成

ここで、Flash のゲーム作成に着手する。連携支援ツールの出力ボタンを押すことで、連携に必要な基本的な記述の入ったプロジェクトファイルが出力され、それをもとに Flash コンテンツの作成を行う。Flash ゲーム作成にあたり、Wearable Toolkit との連携が完全でない間は、Wearable Toolkit で認識したイベントをキー入力などで代用するなどして、開発を行うとスムーズである。

Flash で発生するイベントの定義は、イベントの発火を通知したい場所で RaiseEvent() 関数を呼び出すように、連携支援ツールから挿入する。ハードル競技ゲームの例では、ゲー



図 8 走るイベントのアクションの定義

ムキャラクターがハードルにぶつかった時にイベントが発火するため、Flash コンテンツ上で、ハードルとの当たり判定を行っているところにスクリプトを挿入する。

### Step5 デバッグ, テスト

最後に、デバッグとテストを行う。デバッグは、Wearable Toolkit に付属のデバッグを用いることで、Flash コンテンツを起動した際にアクション、イベントが登録されているかを確認できる。また、ゲームにおいては難易度調節も重要なポイントとなる。コンテキストの認識精度はコンテキスト定義ツールを用いて調節が可能である。

作成したゲームの拡張などでイベントやアクションを追加で定義する際にも、連携支援ツールを用いることで、アクション・イベント一つ一つに対してルールや ActionScript を出力できるため、コピーアンドペーストするだけで拡張が行える。

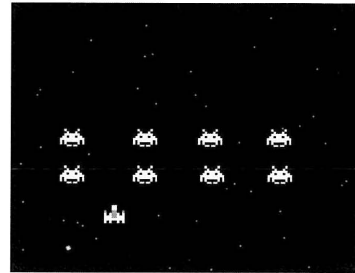
同様の手順で、ジェスチャーで戦闘機を操作し、ジェスチャーをきっかけにミサイルを発射して敵を撃ち落とし、その得点を競うジェスチャーシューティングゲームや、ユーザが飛んでくるボールに合わせて加速度センサつきのバットをスイングし、ホームランの数を競うホームラン競争、右手と左手を交互に揺くと前に進み、ゴールまでの時間を競う水泳ゲームなどいくつかのゲームを作成した。それぞれのゲーム画面を図 9 に示す。

## 6. 考 察

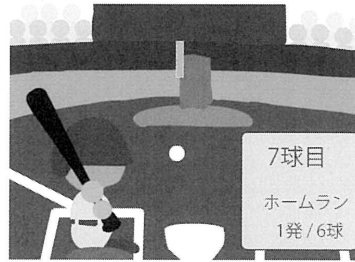
### 6.1 評 価

本提案機構を用いることで、どれだけ簡単に体感コンテンツが制作できるかについて議論する。評価では、既にキーボード入力で行うゲームとして作成されている Flash ゲームに対して、2 種類のコンテキストを操作に割り当てたところ、1 時間以下で体感ゲームとして動作させられた。ただし、キーボードで行うゲームを体感ゲームとすることで、認識精度、認識に要する遅延の問題などから、難易度がかなり上がることも分かった。そのため、難易度調節を時間をかけて行う必要がある。

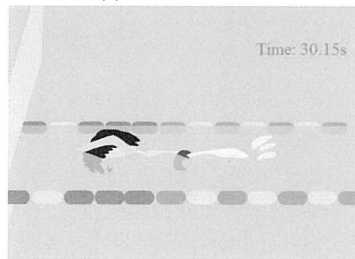
またゲーム以外のコンテンツとして、神戸ルミナリエのイルミネーション募金箱のディスプレイ部に本システムで作成したコンテンツをインストールした。この募金箱コンテンツは、募金の呼びかけを行うユーザの行動を認識し、ディスプレイ上の募金を呼びかけるキャラクターの動きに反映させるというもの



(a) ジェスチャーシューティング



(b) ホームラン競争



(c) 水泳ゲーム

図 9 作成した体感型ゲーム

である。例えば、ユーザのお辞儀を認識すると、図 10 に示すように募金箱内のキャラクターもお辞儀をする。本システムのディスプレイ部は、Flash 部の作成も含めて 3 時間程度で制作が完了するなど、難易度の調整の必要のない体感コンテンツの作成は容易であることがわか。

たとえ Flash コンテンツが既に完成している場合でも、提案機構を用いずにセンサを活用した体感コンテンツを作成するには、大変な実装時間を要すると考えられるため、提案機構を用いることで、体感コンテンツの作成が容易になることは明らかである。

### 6.2 幅広い Flash コンテンツへの対応

現在は Flash のプロジェクトファイルである fla ファイルを修正することで、Wearable Toolkit との連携を行っている。しかし、Flash コンテンツファイルである swf ファイルは仕様が開示されているため、swf ファイル内に含まれる ActionScript を直接操作することも可能である。swf ファイルの ActionScript を直接編集できると、プロジェクトファイルの存在しない既存の数多くの Flash コンテンツを体感ゲームに



図 10 募金箱コンテンツ

変換でき、Flash コンテンツ制作に関する知識がないユーザでも既存のゲームを体感ゲームに変換することが可能となる。

### 6.3 Flash と Wearable Toolkit の連携方法

現在は、`fscommand()` 関数と `SetVariable()` 関数を用いて、連携を実現している。しかし、最新の ActionScript3.0 および、Flash Player9 を用いることで、直接 ActionScript で定義された関数を呼び出すことが可能である。ActionScript3.0 は本格的なプログラミングが可能である反面、デザイナーにとっては敷居が高いため、ActionScript2.0 でも利用可能な提案機構の方法には、大きなアドバンテージがあると言える。ただし、処理のオーバーヘッドなどは調査する必要がある。

## 7. ま と め

本研究では、センサ技術を簡単に扱う機能を有する Wearable Toolkit と Flash コンテンツを連携させる機構を実現することで、Flash コンテンツ制作者が容易にセンサ技術を活用したコンテンツを作成できることを確認した。Wearable Toolkit と連携させることで、センサによって認識されたコンテキストを活用するだけでなく、振動子などの他のデバイスとの連携も可能になるなど、Flash コンテンツ制作者にとって幅広い選択肢を提供できるものとなった。さらに、連携を支援するツールを作成することで、より手軽に Flash と Wearable Toolkit の連携を行えるようになった。

今後は、Flash の状態をルールの Condition として取得できるような拡張や、swf ファイルを直接編集できるツールの開発を行う予定である。

## 謝 辞

本研究の一部は、文部科学省科学研究費補助金基盤研究 (A) (20240009)、特定領域研究 (19024046)、特別研究員奨励費 (19-55371) によるものである。ここに記して謝意を表す。

## 参 考 文 献

- 1) Adobe Flash CS3 ホームページ。  
<http://www.adobe.com/jp/products/flash/>.
- 2) MakingThings 社ホームページ。  
<http://www.makingthings.com/>.
- 3) Phidgets ホームページ。  
<http://www.phidgets.com/>.
- 4) Wearable Toolkit ホームページ。  
<http://wearable-toolkit.com/>.
- 5) S. Kobayashi, T. Endo, K. Harada, and S. Oishi. GAINER: A reconfigurable I/O module and software libraries for education. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06)*, pp. 346–351, 2006.
- 6) 寺田努, 宮前雅一, 山下雅史. その場プログラミング環境実現のための状況定義ツールおよび状況処理エンジンの開発. 情報処理学会シンポジウムシリーズマルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO2008) 論文集, pp. 1487–1495, 2008.