

## W4: ウェラブルサーバによる個人情報発信型アーキテクチャ

井上 亮文\* 高久 宗史\* 柴 貞行\*  
加藤 淳也† 重野 寛† 岡田 謙一†

e-mail: akifumi@mos.ics.keio.ac.jp

本研究では、ウェアラブル端末を Web サーバにみたてて様々な個人情報を発信する環境を想定する。この環境下では、ウェアラブルサーバが取得する生体情報や、周囲に存在する他のユーザの情報など、動的な情報がコンテンツとなる。しかし既存の WWW はアクセスに必要な識別子を発信側が事前に用意すること、識別子が1つの静的なリソースしか示せない、という問題がある。提案アーキテクチャでは、動的コンテンツへのアクセスに用いる識別子をインフラ側で定義している。また、属性に応じたサーバのクラスタ構成機構により、他のウェアラブルサーバと連携したコンテンツの生成を可能にする。本稿では提案アーキテクチャW4のコンテンツアクセス方式とそのプロトタイプ実装について述べる。

## W4: Personal Information Transmissive Architecture with Wearable Computers

Akifumi Inoue\*, Hirofumi Takaku\*, Sadayuki Shiba\*,  
Junya Kato†, Hiroshi Shigeno† and Ken-ichi Okada†

In this research, an environment where anyone publishes a variety of personal information by wearable servers is supposed. In such an environment, biological and peripheral information of users would be the contents. However, typical publishing architecture WWW is not suitable because the identifier of WWW is defined by publishing users itself and can indicate only static resources. Proposed architecture "W4" defines some identifiers as an infrastructure. In addition, server clustering mechanism is adopted for generating multi-server response. This paper describes a contents access method and prototype system "Wapache".

### 1 はじめに

従来のウェアラブルコンピューティングに関する研究では、ウェアラブルデバイスの「個人専用端末」という側面を積極的に利用したサービスの提供に主眼が置かれてきた。メディア変換やセンサ情報を用いたデータの受信・表示方法など、ウェアラブルデバイスから見れば情報の流れは主として受信・Downlinkであったといえる。

しかし、ウェアラブルデバイスが情報の発信源とな

ることで得られる利点も大きい。常にセンシングをしながらユーザと共に移動するウェアラブルコンピュータが取得する情報量は膨大である。このようなリアルタイムな情報は、DBに蓄積されたデータとは異なる形で他の人に役立つ情報を含んでいる可能性が非常に高い[1]。

我々はこの点に注目し、従来捨てられることの多かったウェアラブルデバイスから取得される情報を積極的に活用することに主眼を置く。そしてウェアラブルデバイスユーザをコンテンツの受信者から生成者へ方向付けることを目指す。本稿ではこのウェアラブルコンピュータを用いた個人情報発信におけるコンテンツアクセス方式に注目し、その問題点とプロトタイプシステムについて述べる。

\* 慶應義塾大学大学院 理工学研究科 開放環境科学専攻  
School for Open and Environmental Systems,  
Graduate School of Science and Technology,  
Keio Univ.

† 慶應義塾大学 理工学部  
Faculty of Science and Technology, Keio Univ.

## 2 Wearable World Wide Web

### 2.1 概念

W4 の概念を図 1 に示す。従来のウェアラブルデバイス (WD) は、外部のデータベース上のデータの表示先であり、WD を保持するユーザからみれば Downlink の情報の流れが主流であった。これに対して W4 では、ユーザと共に移動することによって外部の情報に多く接する WDs の特徴に注目し、WD を保持するユーザを人間サーバと見なし、その情報へとアクセスする。WD を保持するユーザから見れば、これは Uplink の情報の流れになる。W4 の世界では、相互に無線通信可能な WDs を保持したユーザが多数存在する。そして、あるユーザが保持する WDs のうち 1 台は、何らかの手段で外部からネットワークアクセス可能であり、WDs の情報を統括する W4 サーバ (W4S) として機能する。そして W4S が取得・生成する情報は、W4 クライアントに受信・閲覧される。

ここで Rhodes らにおける WD の定義 [2] では、(1) ユーザと共に移動、(2) 常時電源 ON、(3) 環境認識可能なセンサーを装備、という特徴があることを踏まえると、W4 で扱うべき情報は、WWW における画像や文書といったファイルリソースよりも、W4S となったユーザからリアルタイムに取得・生成される情報に意義があると考えられる。そこで、以下の 4 つを W4S で取り扱うウェアラブルインフォメーション (WI) として定義する。

- 五感情報  
視覚・聴覚・触覚・嗅覚・味覚情報
- 位置情報  
現在の物理的位置情報
- 生体情報  
自律神経系生体情報。心拍数や体温、呼吸数など。
- 個人プロフィール情報  
名前、所属組織等。

W4 を実現するには、様々な技術的課題を解決する必要がある。以降はその課題について述べる。

### 2.2 識別子

WWW コンテンツを構成する全てのリソースは発信側が自由に用意する。よって A のページに存在する自己紹介が B のページでは無い場合もある。同様にリ

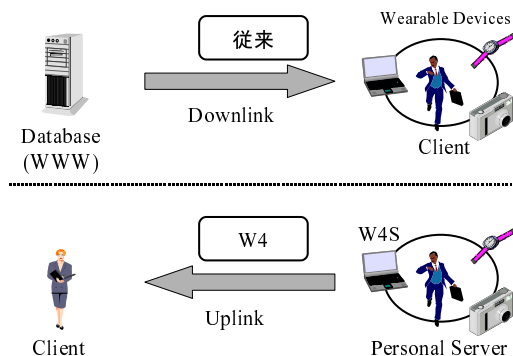


図 1: W4 概念図

ソースへ確実に到達するための識別子 URL も発信側で自由に付加可能である。この URL 文字列とリソースの内容の間には基本的に関連性はない。profile.html は自己紹介に関するものだと予想はできるが、これは発信側が社会的慣習に沿って命名しただけであり、実際の内容が天気予報であってもかまわない。つまり、基本的に WWW リソースはその存在と識別子は未知であり、存在を事前に知っている発信側や、アクセス経験のあるユーザが用意した Hyper Link 経由で間接的にアクセスするしかない。

これに対し W4 のリソースとなる WI は、2.1 にあるように、W4S を持つ全ユーザに対して、その存在と内容が明確に定義されている。こういった既知のリソースには、あらゆるユーザから直接的にアクセス可能であることが望ましい。ここで WWW と同じように発信者毎に識別子が独自に付加されてしまうと、直接アクセスは困難となってしまふ。

### 2.3 サーバ間連携レスポンス

Web ページの閲覧では、クライアントが Web サーバ上の HTML ドキュメントをリクエストする。サーバはそのレスポンスとしてリクエストされたドキュメントを返す。こういった Web ページは、ユーザがあらかじめ用意した複数のリソースで構成されるため、ドキュメント上のハイパーリンクは他の場所に用意されたリソースを参照する静的なものである。また、WWW で用いる識別子 URL は各々 1 つのリソースしか指し示さない。よって画像と文書が同時に表示されているように見えても、実際はクライアントが別々にリクエストしている (図 2 上)。

これに対し W4 では「A の近くにいる人」というリクエストがありうる。この場合、WD がユーザと共に移

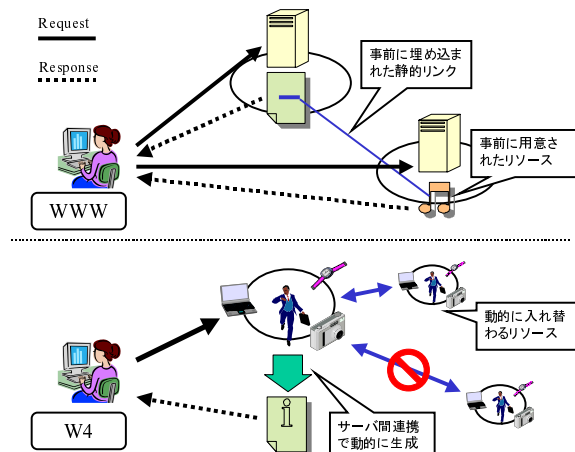


図 2:

動可能という特性により，A の近くの W4S は時々刻々と変化する (図 2 下)．W4 で用いるプロトコルは，このような 1 つの識別子で動的に変化する複数のリソースを扱うために，複数 W4S 間連携によるレスポンスの実現が必要になる．

### 3 W4 コンテンツアクセス方式

2.2 で述べた，識別子の設定が発信側依存の問題に対し，W4 では WI リソースの識別子をインフラ側で定義したアクセス方式を採用する．この W4 コンテンツアクセス方式は，従来の URL 表記を参考に以下のように構成をとる．

<スキーム部>://<アドレス部>//<WI 識別子>

スキーム部には通信プロトコル名が入る．W4 では，WI をやりとりするプロトコル Wearable Information Protocol(WIP) がこれにあたる (後述)．

アドレス部にはリクエストを受け付ける W4S のサーバ名，もしくは IP アドレスが入る．WWW ではホスト名とアドレスの解決に DNS が用いられてきた．W4 では，W4S 個人という関係が成り立つことを考えると，ユーザの名前と IP アドレスの解決機構が望ましい．本提案ではこれら名前解決は P2P[4] やオーバーレイネットワーク技術に譲り考慮しない．

最後に，各 WI に対応する WI 識別子が続く．ここで 2.1 の定義によれば，WI の中には五感情報を含むが，このうち嗅覚，味覚及び触覚を扱うことは現状では困難なため，視覚と聴覚のみ扱うことにする．

### 3.1 WI 識別子

定義した WI 識別子を表 1 に示す．このようにインフラ側で識別子名を定義しておくことで，W4 クライアントが全ての WI へ確実にアクセス可能になる．

視覚情報へのアクセスは，その人が見ている映像情報であり，ユーザが装備するカメラが撮影している映像へのアクセスとなる．同様に聴覚情報へのアクセスは，その人の聞いている音声情報であり，マイクへのアクセスである．位置及び生体アクセスは，それぞれ GPS と生体センサがリアルタイムに取得する値へのアクセスとなる．Profile アクセスは，従来の WWW と同様にユーザが事前に用意し，XML で記述された個人情報へのアクセスである．この個人情報は生年月日のように変化しない静的なものと，年齢や所属組織のように変化する動的なもので構成されるが，今回は更新に関しては考慮しない．最後に，2.3 で挙げた複数 W4S 連携の例として，アクセスされたユーザの周囲に存在する W4S の情報を取得するための識別子 (近隣アクセス) も定義した．

表 1: WI 識別子の定義

名称	識別子表記	アクセス先
視覚アクセス	//vision	カメラ
聴覚アクセス	//hearing	マイク
位置アクセス	//location	GPS 受信機
生体アクセス	//bio	生体センサ
Profile アクセス	//profile	プロフィール情報
近隣アクセス	//neighbor	近くの W4S

### 3.2 W4S クラスタリング

2.3 に述べたように，W4 では，1 つのリクエストに対して複数の W4S が連携してレスポンスを返すコンテンツが存在し得る．この実現のため，W4 では，各 W4S が WI の状態を元に相互に関連付けて管理する．クラスタは，プロフィール情報の所属といった各属性でも構成できるが，ここでは簡単のため位置情報によるクラスタのみを考慮する．

各 W4S は位置情報を基にクラスタを構成することで，自身に物理的に隣接した W4S の位置情報を常に把握可能となる．自身に対して周辺の W4S に対するリクエストがきた場合，このクラスタの結合をたどることでの確かなレスポンスを生成できる．クラスタは，

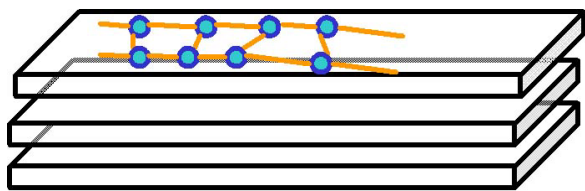


図 3: 位置情報クラスタリング

W4S の新規参入や移動などの変更に対して柔軟に対応し、自己組織化機能を備えている。

### 3.3 WIP: Wearable Information Protocol

WWW における HTTP に代わって、W4 で用いられるプロトコルが WIP である。HTTP は WWW コンテンツへのアクセスが目的であった。WIP では、W4 コンテンツへのアクセスに加えて、3.2 で述べた W4S 連携レスポンスも処理できる必要がある。そこで HTTP を踏襲し、WI 識別子の取り扱いと、W4S クラスタリングを同時に実現可能なリクエスト-レスポンス型のプロトコルとして実装する。以降、WIP の詳細について述べる。

#### 3.3.1 Request

W4 クライアントから W4S へのリクエストメッセージは BNF を用いて表すと以下の形式をとる。HTTP/1.1[3] と同様に、1 行目に Request-Line、2 行目以降はヘッダ群、CRLF を挟んで、もし存在すれば message-body が続く。

```
Request = Request-Line
        *(( general-header
          | request-header ) CRLF)
        CRLF
        [message-body]
```

この Request-Line も、HTTP/1.1 と同様に以下のように定義される。なお、ここでは表示の都合上改行をしている。

```
Request-Line = Method SP
              [Request-URI | Location] SP
              WIP-Version CRLF
```

まず Method トークンに始まり、Request-URI と Protocol Version が後に続き、CRLF で終了する。現在の WIP バージョンは WIP/0.1 である。HTTP との違いは Method の種類とバージョンの他に、セカンドトークンに位置情報 (Location 形式) が挿入可能な点である。この Location 形式のデータは、GPS 受信機から得られる各緯度経度情報の時・分・秒形式のデータをデリミタ”.”によって区切ったものである。

WIP Method には、WWW コンテンツのリクエストに用いる GET/POST 等以外に、8 つのメソッドが独自に存在する (表 2)。このうち 6 つは、3.1 で定義された各 WI 識別子で指示されるリソースのリクエストに用いられ (VISION, HEARING, LOCATION, BIO, PROFILE, NEIGHBOR)、残りの 2 つは位置情報クラスタの構成に用いられるメソッドである (INIT, GPS)。WIP リクエストを生成する WIP クライアントは、WI 識別子からどのメソッドを用いるかを決定する。

VISION メソッドにおける Request-URI は、仮にユーザが複数のカメラを装備している場合、その ID となる。指定されなかった場合や、該当カメラ ID が存在しなかった場合には、W4S 側で設定された視覚情報が返される。同様に HEARING メソッドではマイク ID が Request-URI となる。

NEIGHBOR メソッドは、リクエストを受信したユーザの近くに誰がいるのかを要求するメソッドである。ここで Request-URI には、何人分の情報を要求するかが設定され、数値形式でないなど判定が不可能の場合のデフォルトは 4 である。

INIT メソッドは、新たにクラスタに参入する W4S を、位置情報クラスタの適切な位置へ接続するために用いる。INIT メソッドを受信した W4S は、Request-Line 中の位置情報を調べ、該当位置に近い隣接 W4S へと INIT メソッドをフォワードしていく。最終的な隣接 W4S は、自身が隣接する W4S であることを通知できるように、INIT の Request-Line 中には新規 W4S の位置情報と IP アドレスが含まれている。この時の隣接 W4S の位置情報通知に用いられるのが GPS メソッドであり、言わば INIT メソッドの応答的な役割も果たしている。

他のメソッドの Request-URI は、現実装では無視されている。

#### 3.3.2 Response

W4S はクライアントからのリクエストを受信・解釈した後、WIP レスポンスメッセージを返す必要がある。

表 2: WIP メソッド

メソッド名	Request	Request-Line	Response
VISION	視覚情報	Camera ID	カメラの映像ストリーム
HEARING	聴覚情報	Microphone ID	マイクの音声ストリーム
LOCATION	位置情報	-	Location 形式の位置データ
BIO	生体情報	-	規定せず
PROFILE	プロフィール情報	-	XML 形式のプロファイル
NEIGHBOR	近隣 W4S の情報	要求する W4S の数	近隣 W4S の距離情報
INIT	クラスタ構成用	クラスタ加入 W4S の位置情報とアドレス	隣接 W4S の GPS メソッド
GPS	位置情報の通知	自身の位置情報	-

このレスポンスの形式は以下のようになっている。

```
Response = Status-Line
          *(( general-header
            | request-header ) CRLF)
          CRLF
          [message-body]
```

WIP は HTTP の拡張のため、Status-Line には HTTP で用いられるようなステータスコード番号が用いられる (例:200(ok))。ここでは表 2 の WIP 独自のメソッドに対するステータスコードが 200(ok) の場合のレスポンスについて述べる (表 2 参照)。

VISION 及び HEARING メソッドのレスポンスは、ステータスコード送信の後、リクエスト元のクライアントとコネクションを確立し、カメラもしくはマイクによって取得した視聴覚情報のストリーム送信を開始する。

LOCATION メソッドのレスポンスは、ステータスコードに続けて自身の位置情報を先述の Location 形式で返す。

BIO メソッドのレスポンスは、様々な生体情報を message-body として送信する。同様に PROFILE メソッドも XML 形式のプロファイル情報を message-body として返すが、双方とも表示やデータの利用方法などは規定しておらず、クライアント側の実装に委ねることとする。

NEIGHBOR メソッドのレスポンスは、Request-Line 中の数だけ、近くの W4S の情報を返す。自身が保持する数以上の W4S 情報が要求された場合は、NEIGHBOR メソッドを再帰的に送信して対応する。現段階では簡単のため、近くの W4S の位置情報を返すのみである。

3.3.1 の最後にも述べたように、INIT メソッドは新たにクラスタに参入する W4S に物理的に隣接する W4S

までフォワードされる。最初及び途中で INIT メソッドを受信したノードはステータスコードをリクエスト相手に返すのみで、最終的な隣接 W4S のみが、レスポンスとして WIP リクエストの GPS メソッドを参入ノードに対して実行する。GPS メソッドは、自身の位置情報を WIP リクエスト時に伝達するため、応答はステータスコードのみである。

## 4 実装

### 4.1 システム構成

前章までに、ウェアラブルデバイスを身につけたユーザの情報発信アーキテクチャである W4 を提案し、実現上の問題点とその解決手法を提案した。それらの提案の有効性を確認するにあたり、実際のウェアラブルデバイスを用いてプロトタイプシステム Wapache の実装を行った。システム構成を図 4 に示す。

Wapache のコンポーネント構成を示す。W4 クライアントは、W4S とクライアント-サーバ関係の WIP 通信を行う。W4S 上では WIP を処理するデーモンプロセス (WIPD) が常にリクエストを監視しており、WD インターフェース (ウェアラブルデバイスインターフェース) を通してカメラ、マイク、GPS、プロフィールを操作する。このインターフェースは各ウェアラブルデバイスを抽象化するために実装され、デバイスの仕様に極力依存せずに制御することを可能にしている。W4S は、3.2 で述べた位置情報クラスタによって隣接する他の W4S を把握しており、これら複数の W4S で WIP レスポンスが生成可能である。

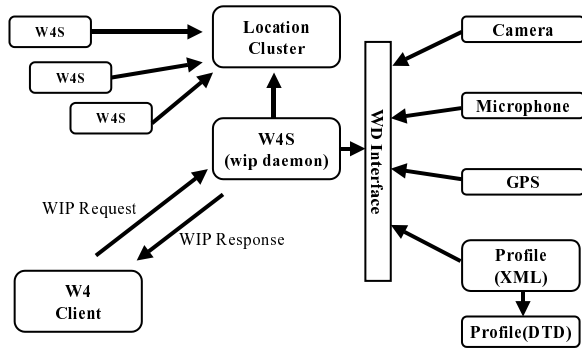


図 4: Wapache コンポーネント構成

## 4.2 実装環境

Wapache の各プログラムモジュールは、Windows XP が動作するノート PC 上 (IBM ThinkPad X24) に、Java 1.4 を用いて実装された。動画及び音声ストリームの送受信部分には Java Media Framework 2.1.1b を、GPS データは Java Communication API を用いて RS232-C 経由で取得した。通信ネットワークは無線 LAN である。各 WD としては、GPS 受信機は Garmin 社 e-Trex、カメラとマイクは Elecom 社製のものをを用いた。

## 4.3 実装画面

Wapache を装備したユーザの視覚情報にアクセスした様子を示す。図 5 の視覚情報が図 6 となる。デモンストレーションの結果、Wapache が発信する各 WI に対して、提案した WI 識別子を用いた WIP でアクセスできることが確認できた。また、近隣アクセスにより近くの W4S の位置情報が取得できることも確認できた。



図 5: 視覚情報アクセス 1

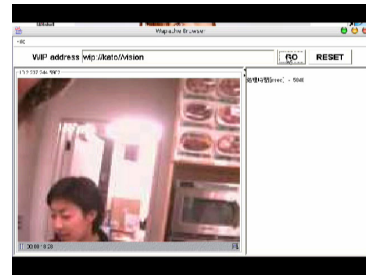


図 6: 視覚情報アクセス 2

## 5 おわりに

本研究では、ウェアラブルコンピュータを用いた個人情報発信の実現に注目した。従来の WWW は、発信側が識別子を自由に設定できてしまうこと、識別子が 1 つの静的なリソースしか指せないことから、動的に変動する WI を扱うには適さないことを指摘した。そこでインフラ側で WI 識別子を定義することで、誰でも確実に WI にアクセスできるようにした。また、1 つの識別子で W4S を連携させるレスポンスを実現するため、W4S の管理にクラスタリング機構を用いた。

現段階は、W4 の全体像を提示しただけであり、この要素技術は改善の余地が大きい。今後はこれらの改善と共に、WIP メソッドの拡充などを進めていく。

## 謝辞

本研究は 21 世紀 COE プログラム研究拠点形成費補助金のもとに行われた。ここに記して謝意を表す。

## 参考文献

- [1] 高久 宗史, 井上 亮文, 柴 貞行, 加藤 淳也, 重野 寛, 岡田 謙一: ウェアラブルネットワーク環境での Uplink サービスフレームワーク. 情報処理学会研究報告 2002-DPS-110, pp. 61-66, 2002.
- [2] Bradley J. Rhodes: The Wearable Remembrance Agent: A System for Augmented Memory. Proc. of ISWC'97, pp. 123-128, 1997.
- [3] RFC2616: <http://www.ietf.org/rfc/rfc2616.txt>.
- [4] S. Ratnasamy, P. Francis, M. Handley, R. M. Karp, S. Schenker: A Scalable Content-addressable Network. Proc. of ACM SIGCOMM, pp. 161-172, 2001.