

# Wapplet フレームワーク： ユビキタス環境のパーソナライゼーションを実現 するアプリケーションフレームワーク

岩本 健嗣（慶應義塾大学 政策・メディア研究科）

近年、ユビキタス環境に対する関心は、研究領域、ビジネス領域を問わず高まっている。ユビキタス環境は一般的に機器やセンサが埋め込まれた部屋や空間によってユーザを支援する計算機環境である。このユビキタス環境を実現するためには、ハードウェア、ソフトウェアを問わずさまざまな要素技術が必要となる。その中で本研究では、環境のパーソナライゼーションに注目する。ユビキタス環境が今後発展すれば、情報キオスクのような場が、誰でも利用できる計算機環境として提供されることが考えられる。パーソナライゼーションとは、このような公共の機器やセンサを、利用者がその場で、アドホック（場当たりの）に占有するという概念である。ユーザは、パーソナライゼーションされた機器や環境を、あたかも自分専用の計算機環境として利用する事が可能となる。

このようなパーソナライゼーションを実現するためには、アプリケーション自体と、公共の機器で動作するサービスソフトウェアの両方を含む、新たなフレームワークの構築が必要となる。このようなパーソナライゼーションを実現するフレームワークをWappletフレームワークと名付け、これを実現する。

## 1 研究背景

背景として、機器のパーソナライゼーションに着目した新しいユビキタス環境である Personalized Public Spaceを提案する。

近年、ユビキタス環境が研究分野やビジネス領域で盛んに取り上げられている。ユビキタス環境では、部屋や環境に埋め込まれた機器やセンサが、特にその存在を意識することなくユーザを支援する計算機環境であると捉えられる。しかし、それらの利便性を既存の計算機環境と同様に高めることを考えた場合、ユビキタス環境における機器が、個々のユーザの違いにどれだけ適応できるかが、重要なファクターとなる。本研究では、個々のユーザに最適な計算機環境を即興的に構築するという概念として、パーソナライゼーションという考え方を導入する。ユビキタス環境において、環境に存在する機器は不特定多数のユーザを対象とするために、特定ユーザに即したサービスを提供することができず、利便性を損なう場合がある。パーソナライゼーションとは、逆に、個人情報や嗜好、要求を積極的に利用して、特定ユーザに特化した計算機環境をその場で構築し、提供するという概念である。この概念に基づいて構築されるユビキタス環境として Personalized Public

Spaceを提案する。

Personalized Public Spaceでは、ユーザはウェアラブルコンピュータや携帯電話など、非常に小型で持ち歩く労力の小さな機器のみを保持し、実際の作業は、公共の機器を一時的に占有して利用する。つまり、Personalized Public Spaceは、今までのように、自分の計算機を持ち歩いたり、自分の作業場所に移動したりするのではなく、行く先々で公共の機器や計算機を利用したいときに占有して利用する計算機環境である。

Personalized Public Spaceの特徴を以下にあげる。

- 公共環境の個人占有

機器やセンサが公共に提供されていることを想定し、これらをユーザが利用したい時に一時的に占有し、あたかも自らの所有物であるかのように利用することが可能である。その際はユーザの個人情報や、好み、要求が機器に反映される。利用後はまたこれらの機器を解放し、他のユーザが利用することも可能である。

- 柔軟な機器利用方法の提供

公共の空間において、ユーザが利用できる機器は場合によって異なる。また、同じ空間であっても、他のユーザによって機器が占有されることが考えられるため、あらかじめ利用できる機器を想定することはできない。そのため、ユーザの要求に対して、複数の機器を組み合わせたリ、必要な機器が利用できない時に、他に代替的に利用できる機器を探索するなど、柔軟にサービスを提供する必要がある。

Personalized Public Spaceの概要を図1に示す。



図 1: Personalized Public Spaceの概要

Personalized Public Spaceでは、ユーザが常に持ち歩ける小型の機器をウェアラブルセンタと呼ぶ。実際の作業は、公共機器（パブリックデバイス）を用いて行う。その際、機器のパーソナライゼーション、認証、ユーザ設定や作業状態の反映はすべてウェアラブル

センタによって集中的に行われる。

## 1.1 パブリックデバイス

オフィスや街頭などの公共の空間に設置され、不特定多数のユーザから利用されることを前提とした機器である。例えばディスプレイ、プリンタ、スピーカなどの機器があげられる。不特定多数のユーザに利用されるため、いわゆる一般の計算機環境における、特定ユーザ、特定アプリケーションのためのサービスソフトウェアと異なり、以下の機能を実現する必要がある。

- アプリケーションを限定しない利用方法の提供

機器の利用方法はユーザによって異なる。そのため、利用毎に異なるアプリケーションによって利用される可能性がある。例えば、ディスプレイはビデオ再生に利用するか、テレビ電話に利用されるかはユーザの目的によって異なる。これを実現するためには、特定のアプリケーションを想定せずに、抽象化されたインタフェースを提供する必要がある。

- 個人の設定や嗜好の反映

不特定多数のユーザに利用されることを前提としているため、特にスケーラビリティやセキュリティの点から、個人の設定をパブリックデバイス自体に保存する事はできない。そのため、ユーザの個人設定を利用時に動的に反映する必要がある。

## 1.2 ウェアラブルセンタ

ウェアラブルセンタとは、ユーザが常に身に付けたり、持ち歩くことが可能で、計算機としての機能を持つデバイスである。具体的には、PDAや携帯電話、ウェアラブルコンピュータがウェアラブルセンタの候補となりうる。ウェアラブルセンタは、環境の制御機構としての役割を担い、公共空間に存在する機器をユーザの目的に合わせて制御する機能を持つ。

Personalized Public Spaceでは、ユーザは最低限このウェアラブルセンタのみを持ち歩き、行く先々で環境に存在するパブリックデバイスを占有して利用する。ウェアラブルセンタが実現すべき機能を以下に示す。

- 環境の情報の収集

現在ユーザがいる環境の状態を把握する必要がある。具体的にはその環境において利用可能なパブリックデバイスを把握し、ユーザの目的に合わせてパーソナライゼーションを行なう。また、ユーザ状況に適應するために、センサを用いて環境情報を得て、それをアプリケーションの動作やパーソナライゼーションに反映する必要がある。

- 個人情報、作業内容の保持と反映

パーソナライゼーションに必要な個人情報や、作業内容は、セキュリティの点が

らパブリックデバイスでは保持できない．これらの情報はウェアラブルセンタで集中的に管理し，パーソナライゼーションの際に必要な情報をパブリックデバイスに反映する．特に作業の継続性を維持するために，作業内容を保存し，別の環境で反映する機構が必要となる．その際，異なる環境では同様のパブリックデバイスが利用できるとは限らないため，なんらかの作業状態の抽象化が必要となる．また，同種のパブリックデバイスが存在する場合，実際に利用する機器はユーザの好みや状況によって選択されるべきである．そのためウェアラブルセンタはユーザの嗜好や状況に適應する必要がある．

## 2 Wapplet フレームワーク

前章で述べたように，Personalized Public Spaceを実現するためには，いくつかの実現しなくてはならない要件がある．これらを実現するためには，パブリックデバイス，ウェアラブルセンタのミドルウェアを含むようなアプリケーションフレームワークの構築が必須となる．

本研究では，そのアプリケーションフレームワークをWapplet フレームワークと名付け，これを実装する．

### 2.1 Wappletフレームワークの課題

Personalized Public Spaceを実現するために，Wapplet フレームワークの構築にあたって取り組む課題を以下にあげる．

- 機器の組み合わせ

ひとつのパブリックデバイスのみでユーザの要求を満たすことが出来るとは限らないため，複数のパブリックデバイスをユーザのひとつの目的に合わせて，組み合わせて利用できるようにする必要がある．

- 機器とアプリケーションの分割

複数機器の機能を組み合わせて利用する場合，機器はあらかじめ特定の機能を実現するアプリケーションを提供することはできない．例えば，ディスプレイをビデオ再生として利用するか，GUIの画面として利用するかはユーザの目的に委ねられるため，ビデオ再生アプリケーションやGUIクライアントなどは，機器にあらかじめ用意しておくことはできない．

- 作業内容，状態の保存，反映

Personalized Public Spaceでは，公共の機器を一時的に占有して利用するため，その作業状態は機器上に保存することができない．特に一時的に作業を中断し，その後別の移動先などで作業を再開することを考慮すると，なんらかの作業状態の保存やその復元方法が必要となる．またその際，実際に利用できる機器が変わっても，出来るだけ中断前と同じ作業環境を維持するような工夫が必要となる．

- ユーザの嗜好，状況の反映

例えば，同じテレビ電話アプリケーションであっても，ユーザが自動車を運転中であれば，ビデオのようなアプリケーションの利用は適当ではなく，音声のみのメディアが優先されるべきである．このように，センサを用いて，ユーザの状況をウェアラブルセンタが取得し，これに従いアプリケーションの挙動を変化させる機構が必要となる．

また，このような状況に応じた機器選択には，ユーザの嗜好も密接に関係する．これらの嗜好の記述方法や反映機構の構築が必要である．

## 2.2 Wappletフレームワークの実現手法

前節で述べたような課題を解決し，Wappletフレームワークを実現するために実現手法について述べる．

### 2.2.1 機器の組み合わせの実現

環境に存在する複数のパブリックデバイスを，ユーザの要求に応じて組み合わせるという観点では，複数の実現手法が存在する．大別すると，自律分散的手法と，集中制御による手法，またそれらの複合型である．

自律分散的手法では，各パブリックデバイス上に分散されたアプリケーション同士が自律的に協調動作を行なうことで機器の組み合わせを実現する．この手法では，各アプリケーションをユーザの個人情報に従って集中的に制御する機構が存在しないため，パーソナライゼーションは困難になる．

集中制御による手法では，PDAなどの情報端末上のアプリケーションやGUIから，複数の機器の制御を集中的に行なうことで，機器の組み合わせを実現する．この手法では，PDAなどでユーザの情報を一括管理することで，パーソナライゼーションは容易になる．しかし，機器同士のトラフィックを制御することが困難であったり，トラフィック自体が制御機構に集中する可能性がある．

本提案では，これら2つのアプローチを複合した形を採用する．このアプローチでは，複数のアプリケーションをパブリックデバイス上で動作させ，これらの協調動作によって機器の組み合わせを実現する．各アプリケーションは，アプリケーション同士でプログラマが規定する独自のデータ交換を行い，協調動作を実現する．また各アプリケーションは，ウェアラブルセンタからの集中制御によって個々のユーザに適応した動作を行なったり，環境の変化に適応する．

### 2.2.2 パーソナライゼーションの実現

組み合わせの実現のために，分散アプリケーションを集中制御する手法を採用することを上述した．これを踏まえてパーソナライゼーションの実現手法を考察する．

この手法において、アプリケーションのモジュールは機器を一時的に占有し、そのユーザの目的を遂行する。このように、アプリケーションのモジュールをパブリックデバイス上で動作させ、ユーザの嗜好や設定をもとに制御することは、まさに機器をパーソナライズすることであると言える。

Wappletフレームワークにおいて、パーソナライゼーションは、アプリケーションモジュールをパブリックデバイスに移送し、実行することによって実現する。この際、アプリケーションはユーザの嗜好や作業状態、個人情報ウェアラブルセンタから受け取る。これらの情報に則してサービスプロバイダの提供する機能を利用することで、実際に個々のユーザに特化した計算機環境を提供する。また、ユーザインタフェースなどを通して行なわれるユーザとのインタラクションなど、インタラクティブな動作はすべてウェアラブルセンタからの指示で行なわれる。

### 2.3 Wappletフレームワークにおいて必要となる基盤ソフトウェア

Wapplet フレームワークにおける機器を組み合わせる機能を実現するためにセンサを利用した状況の取得は不可欠であるが、その中でユーザおよびパブリックデバイスの位置情報把握は状況把握の基盤となる情報である。ユーザを中心としたパブリックデバイスの存在、位置情報を把握することで、Wapplet フレームワークは機器を組み合わせる最低限の尺度を取得し、その上で個人の嗜好や、位置情報以外の環境情報を尺度として追加する。位置情報の管理、提供を行うソフトウェアの存在が必要となる。

ユビキタス環境において、このような位置情報ミドルウェアは、環境を隠蔽し、位置情報を利用する対象に対して環境に依存せず統一的に提供できる必要がある。

## 3 Wapplet フレームワークの設計

Wapplet フレームワークは公共空間のパーソナライゼーションの実現を目的としたアプリケーションフレームワークである。本ソフトウェアは大きく分けて3つの部分から成る。

1. Wapplet (分散アプリケーション)
2. サービスプロバイダ (機器上のサービスソフトウェア)
3. ミッション機構 (ウェアラブルセンタ上の制御機構)

各コンポーネントの関係を図2に示す。

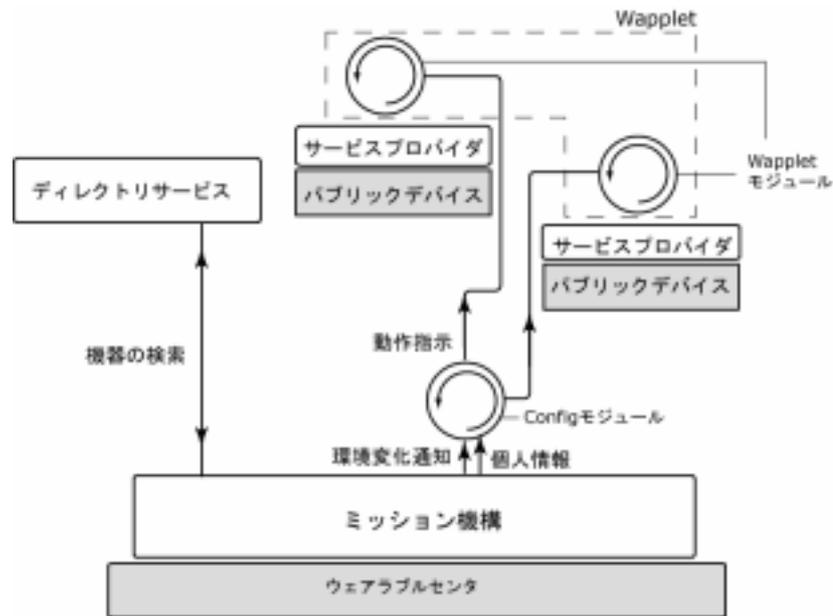


図 2: Wappletフレームワークの設計

Wapplet は本フレームワークにおけるアプリケーションであり、ユーザの目的に合わせた機能を持ち、必要な機器を用いてその作業を遂行する。サービスプロバイダは環境上のパブリックデバイスで実行されているソフトウェアで、機器の機能を Wapplet に提供する。

ミッション機構はウェアラブルセンタ上で実行される制御機構で、個人情報や作業状態を管理し、また機器のパーソナライゼーションを行なう。

### 3.1 Wapplet

本フレームワークにおけるアプリケーションは、機器の組み合わせを実現するために、複数の機器上のアプリケーションが互いに連携を取らなくてはならない。各機器上では 1 つの分散アプリケーションのコンポーネントを実行し、この機器上で動作する部分を Wapplet モジュールと呼ぶ。そして 1 つまたは複数の Wapplet モジュールを統合する 1 つのアプリケーションを Wapplet と呼ぶ。

利用できる機器が変化した場合、Wapplet モジュールを新たに利用する機器に移動することによってアプリケーションレベルでの機器の切り替えを実現する。実際にアプリケーションによる機器の機能の利用は、Wapplet モジュールがサービスプロバイダのインタフェースを利用することで行なう。

### 3.2 サービスプロバイダ

各機器上で分散して実行されるアプリケーションの実行環境として、Wapplet モジュー

ルを実行するためのミドルウェアを提供する必要がある。Wapplet フレームワークでは、このミドルウェアをサービスプロバイダと呼ぶ。サービスプロバイダは、Wapplet モジュールに対して、機器の機能を利用するためのインタフェースや実行環境を提供する。

またサービスプロバイダは、ウェアラブルセンタ上の制御機構からの検索を可能にするために、機器の機能を登録する必要がある。登録は、機器の機能を提供するインタフェースをディレクトリサービスに登録することで行なう。

### 3.3 ミッション機構

ミッション機構は、環境のパーソナライゼーションを行なうために、ユーザの目的に添った Wapplet アプリケーションを機器上に移送する。

また、一つの Wapplet アプリケーションに対して 1 つの Config モジュールを実行する。Config モジュールは一つの作業を抽象化したもので、一般的には一つのアプリケーション毎に存在する。Config モジュールは Wapplet モジュールの操作（環境変化への対応など）や、作業状態の保持などを行なう。また、ミッション機構はユーザの個人情報や嗜好を保持し、これを Config モジュールに反映する。Config モジュールはその個人の情報に応じて Wapplet モジュールを制御する。

ミッション機構では機器の検索機能を実現する必要がある。機器の検索機能では、ユーザの目的に応じてパーソナライゼーションに必要なサービスプロバイダを検索する。

## 4 実装

本章では、Wapplet フレームワークの実装として、Wapplet、サービスプロバイダ、およびミッション機構の実装について述べる。実装は Java 言語 (JDK1.3) を用いて行い、メディアデータ処理には JMF2.1 を用いた。

### 4.1 Wapplet

Wapplet の基本機能として、機器を切り替えるための移動機能と、ミッション機構との通信機能を実装した。また、サービスプロバイダを利用したメディアアクセスの提供と、Wapplet 間の協調動作を記述するためののための API を用意した。

### 4.2 ミッション機構

Wapplet プログラマが Config モジュールを作成するために、Wapplet が利用する機器の種類を登録するための API と、登録した機器の種類が変化した際のミッション機構からの通知を受けとるためのイベント駆動型の API を構築した。

また、ミッション機構からのサービス検索結果を受け取り、Wapplet モジュールに対する移動の通知を行うための機能を実装した。これらの通知の受け取りや Wapplet モジュールへの移動通知は、クラスライブラリの内部で処理されるため、Wapplet プログラマが実際に利用する必要はない。

### 4.3 サービスプロバイダ

サービスプロバイダの機能として、メディア処理機能、機器登録機能を実装した。

基本機能としては、Wapplet モジュールを受付け、実行する機能を実現した。サービスプロバイダはシリアルライズされた Wapplet モジュールを、RMI を介して送受信し、スレッドを生成して実行する。

また、メディア処理を行う機能として、RTP とファイルによって提供されるメディアデータを処理する機能と、Wapplet モジュールから操作を行う機能を提供した。

## 5 まとめ

本論文では、ユビキタスコンピューティング環境においてパーソナライゼーションに注目した計算機環境として Personalized Public Space を提案した。パーソナライゼーションを実現するために、アプリケーション自体と、公共の機器で動作するサービスソフトウェアの両方を含む、新たなフレームワークとして Wapplet フレームワークを実現した。

Wapplet フレームワークを利用することで、ユビキタスコンピューティング環境において、ユーザは即興的に機器を組合せて、パーソナライズされた計算機環境を利用することが可能である。また、Wapplet フレームワークは頻繁に起り得る環境変化に対して、利用する機器の切り替えや、作業状態の保存によって適応することができる。