特集論文

KODAMA と VPC によるユビキタス環境のためのフレームワークの構築と評価

KODAMA and VPC based Framework for Ubiquitous Systems and its Experiment

高橋 健一
Kenichi Takahashi

九州大学大学院システム情報科学研究院
Department of Intelligent Systems, Kyushu University
tkenichi@al.is.kyushu-u.ac.jp

雨宮 聡史
Satoshi Amamiya

(同 上)
roger@al.is.kyushu-u.ac.jp

岩尾 忠重
Tadashige Iwao

富士通研究所
Fujitsu Kawasaki Laboratories
iwao@flab.fujitsu.co.jp

鍾 国強
Guoqiang Zhong

九州大学ベンチャービジネスラボラトリー
Venture Business Laboratory, Kyushu University
zhong@al.is.kyushu-u.ac.jp

貝沼 達也
Tatsuya Kainuma

富士通プライムソフトウェアテクノロジー
Fujitsu Prime Software Technologies
kainuma@pst.fujitsu.com

雨宮 真人
Makoto Amamiya

九州大学大学院システム情報科学研究院
Department of Intelligent Systems, Kyushu University
amamiya@al.is.kyushu-u.ac.jp, <http://www-al.is.kyushu-u.ac.jp/~amamiya/japanese/index.html>

keywords: ubiquitous computing, multi-agent systems, mobile computing

Summary

Recently, agent technologies have attracted a lot of interest as an emerging programming paradigm. With such agent technologies, services are provided through collaboration among agents. At the same time, the spread of mobile technologies and communication infrastructures has made it possible to access the network anytime and from anywhere. Using agents and mobile technologies to realize ubiquitous computing systems, we propose a new framework based on KODAMA and VPC. KODAMA provides distributed management mechanisms by using the concept of community and communication infrastructure to deliver messages among agents without agents being aware of the physical network. VPC provides a method of defining peer-to-peer services based on agent communication with policy packages.

By merging the characteristics of both KODAMA and VPC functions, we propose a new framework for ubiquitous computing environments. It provides distributed management functions according to the concept of agent communities, agent communications which are abstracted from the physical environment, and agent collaboration with policy packages.

Using our new framework, we conducted a large-scale experiment in shopping malls in Nagoya, which sent advertisement e-mails to users' cellular phones according to user location and attributes. The empirical results showed that our new framework worked effectively for sales in shopping malls.

1. は じ め に

今日、携帯電話を利用したインターネットへの接続は、喫茶店や路上など、場所を選ばずにあらゆる場所で行われている。また、無線 LAN を利用した Hot Spot 接続サービスも空港や飲食店を始めとして、急速に広がりを見せている。このように、携帯電話や無線 LAN を利用することで、時間や場所を選ばずに必要な情報やサービスにアクセスできるというユビキタス環境 [Weiser 91, Ubiquitous

02] が現実のものとなりつつある。このようなユビキタス環境を実現するためには、様々な機器を相互通信させ、その上で必要な情報の交換やサービスの相互利用を行わせることが必要であり、次のような機能をもった枠組みが要求される。

サービスや情報の分散管理 ユビキタス環境において、あらゆる機器がネットワークに接続され、サービスや情報を提供する。これらのサービスや情報を集中管理するには限界がある。このため、これらを分散管

理するための仕組みが必要となる．

サービスの利用 サービスを利用するためには，サービスを利用するための手順を獲得する必要がある．例えば，電話の利用では，1．受話器を取る，2．ダイヤルする，といった電話を利用するための手順が存在する．このようなサービスを利用するための手順を獲得し，その手順に従ってサービスを利用する．

物理的な通信のサポート ユビキタス環境において，サービスや情報は物理的に分散して存在する．このため，これらの間での通信をサポートする環境が必要となる．

情報の保護 個人情報を公開することは，プライバシー保護の上で好ましくない．

そこで我々は，KODAMA (Kyushu university Open & Distributed Autonomous Multi-Agent) [Zhong 02] と VPC (Virtual Private Community) [Iwao 01, Iwao 03] と呼ばれる 2 つのエージェント技術を利用し，これらの機能を統合することでユビキタス環境を実現するのに必要な機能を備えた枠組みを実現した．

KODAMA は，分散管理を容易とするためのコミュニティ構造と物理的な位置に依存しないエージェント間通信環境を提供する．VPC は，各エージェントへサービスを割り当てるための仕組みとユーザプロファイルの管理のための仕組みを与える．このため，KODAMA と VPC を統合することによって，提供されるサービスをコミュニティごとに独立に管理し，エージェントがコミュニティに加わった際に，そのコミュニティで提供されているサービスをエージェントに提供できるようになる．

我々は，このことを検証するために，ユーザの属性と位置に適合した広告を携帯電話に向けて e-mail で発信するシステムを構築し，名古屋の商店街で実験を行った．実験は，2003 年 3 月 15 日から 4 月 30 日の期間に約 900 人のユーザの協力を得て行った．この実験で，ユーザは，自分の属性と位置に基づいた広告を携帯電話への e-mail として得ることができた．

2. KODAMA

KODAMA は，分散管理を容易とするためのコミュニティ構造と物理的な位置に依存しないエージェント間通信環境を提供する．図 1 に KODAMA の概要を示す．

KODAMA におけるエージェントは，カーネルユニットとアプリケーションユニットで構成される．カーネルユニットは，KODAMA エージェントの共通機能を与える．アプリケーションユニットは，個々のエージェントが持つ独自の機能で，アプリケーションの開発者によって作成される．これらのエージェントは，コミュニティと呼ばれるエージェントの集まりを構成する．各コミュニティには，それを代表するポータルエージェントが存在し，ポータルエージェントによって独立に管理される．

図 1 KODAMA の概要

また，KODAMA では，物理的通信環境として ACZ (Agent Communication Zone) を準備する．ACZ は，論理的なエージェントの構造を物理的なアドレス空間にマッピングする機構であり，エージェント間通信を実現するための物理的通信機能を提供する．このため，エージェントは，物理的通信環境を意識せず通信することができる．

2.1 KODAMA エージェント

KODAMA におけるエージェントは，カーネルユニットとアプリケーションユニットによって構成される．カーネルユニットは，コミュニティを構成するための機能やエージェント間協調を実現するための機能を持つ．アプリケーションユニットは，個々のエージェントによって異なる機能であり，plug-in モジュールの集まりとして構成される．plug-in モジュールは，適用ルールと状態，plug-in プログラムで構成される．適用ルールは，優先度と解釈ポリシーで構成される．エージェントは，メッセージを受信すると，適用ルールによって受信したメッセージを評価して一つの plug-in モジュールを選択し，その plug-in プログラムを呼び出す．

2.2 コミュニティ

KODAMA におけるエージェントは，エージェント単体では解決できない問題も，エージェント間の協調で解決しようとする．このため，KODAMA によるシステムは，複数のエージェントによって実現される．KODAMA では，この複数のエージェントの集合体をコミュニティと呼び，そのコミュニティを一つのエージェントとして扱うことで，階層的なシステム構築を可能としている．一般に，システムは，モジュールの階層的な組み合わせで構成される．人間社会 (e.g. 会社) も，階層的な組織構造で機能している．このような階層構造の実現において，モジュールの組み合わせで作られたシステムには，そのシステムを利用するためのインタフェースが必要とな

```
// if true, then filter out msg
public boolean incoming_msg_filter(KMessage msg){
    boolean flag = true;
    if( msg.getSender().equals("tel agent") ){
        flag = false;
    }
    return flag;
}
```

図 2 incoming-msg-filter の例．"tel-agent" 以外のエージェントが送信したメッセージをフィルタアウトする．

る．また，人間社会においては，例えば，各部署や各課に部長や課長などの代表者が存在する．これらの例と同様に，エージェントの集合体であるコミュニティにもその代表（インタフェース）となるエージェントが必要となる．そこで，KODAMA では，コミュニティを代表するエージェントとしてポータルエージェントを導入する．ポータルエージェントは，facilitator[Cutkosky 93] とは異なり，エージェントの初期化やエージェントの動作の管理を行わない．また，コミュニティ内のエージェント同士の通信にも制限を与えない．

このようなコミュニティの階層構造によって，KODAMA では，以下の機能を実現する．

コミュニティ構造に従ったエージェント間通信の制御

コミュニティ構造上遠くのエージェントとの通信より，近くのエージェントとの通信を優先する．また，コミュニティごとにメッセージフィルターを設定することで，不要なメッセージの流通やコミュニティ内の情報の漏洩を防ぐ．

エージェントの管理 すべてのエージェントを集中的に管理するのではなく，コミュニティごとに分散して管理する．

エージェントへの論理アドレスの付加 エージェントには，コミュニティ構造に従った一意な論理アドレス（LADR: Logical ADdRes）が与えられる．LADR を利用することで，エージェントを一意に特定する．

§1 メッセージフィルター

ポータルエージェントは，メッセージフィルターによって，コミュニティ内外に渡って送信されるメッセージをフィルタリングする．メッセージフィルターは，incoming-msg-filter，outgoing-msg-filter，join-filter で構成される．incoming-msg-filter は，コミュニティに入ってくるメッセージをフィルタリングする．outgoing-msg-filter は，コミュニティから出て行くメッセージをフィルタリングする．join-filter は，エージェントのコミュニティへの参加を制御する．各フィルターの機能は，それぞれのプログラムによって実現される．図 2 に *tel agent* 以外のエージェントが送信したメッセージをフィルタアウトする incoming-msg-filter の例を示す．

§2 論理アドレス

エージェント間で peer-to-peer 通信を実現するためには，相手のエージェントを特定できなければならない．そ

こで，KODAMA では，コミュニティ構造に従った一意なアドレス（LADR）をエージェントに与える．LADR は，エージェントが属するコミュニティ名とそのコミュニティでの一意な識別子によって，

<LADR> ::= <community-name> "." <UID>

<community-name> ::= <LADR> | "root"

<UID> ::= <string>

として定義される^{*1}．例えば，*root.x* コミュニティに属するエージェントには，*root.x.a*，*root.x.b* のような LADR が与えられる．ここで，<UID> は，次の手順で与えられる．

- (1) エージェント *A* は，コミュニティ *C* に参加するときに，コミュニティ *C* のポータルエージェントに JOIN メッセージを送信する．
- (2) コミュニティ *C* のポータルエージェントは，コミュニティ *C* でユニークな ID（<UID>）を生成し，エージェント *A* に返す．

2.3 Agent Communication Zone (ACZ)

ACZ は KODAMA エージェントが動作する全てのマシン上に配置される．これらの ACZ は，peer-to-peer なネットワークを構成する．ACZ は，APP (Agent Platform Protocol) [高橋 03] を利用し，エージェントの位置情報を交換することで LADR から，そのエージェントが存在する ACZ のアドレス（PADR: Physical ADdRes）を求める．このことを実現するために，ACZ は，LAT (Local Agent Table)，RAT (Remote Agent Table) の 2 つのテーブルを持つ．LAT は，その ACZ が存在するマシン上に存在するすべてのエージェントの LADR を保持するテーブルである．RAT は，他のマシンに存在するエージェントの LADR とそのエージェントの PADR を対としたデータを保持するテーブルである．

ACZ によるメッセージの配送は，LAT，RAT の状態によって次の 3 つの場合がある．

Case1 LAT に目的のエージェントが存在する場合．LAT を参照し，直ちにメッセージを配送する．

Case2 LAT に目的のエージェントが存在しないが，RAT に目的のエージェントが存在する場合．RAT を参照し，目的のエージェントの PADR に対してメッセージを送信する．ここで，メッセージの送信が失敗すれば，RAT の情報を修正し Case3 の手段を利用する．

Case3 Case1, 2 に当てはまらない場合．各 ACZ は，検索要求を受け取ると，図 3 の手順によって目的のエージェントを見つけ，メッセージを配送する．

この仕組みによって，エージェントはお互いの物理的な位置（e.g. IP address）を意識せずに通信することができる．

*1 最上位のコミュニティのポータルエージェントに *root* という LADR が与えられる．

```

void searchPADR(PADR source, LADR target){
    agt = search for the nearest agent from target
           on the community structure from RAT;
    if(agt == target){
        send the PADR of agt to source;
    }else{
        send search-request of target to PADR of agt;
    }
}

```

図 3 エージェントの検索プログラム概要．source は検索要求を出した ACZ の PADR で，target は目的のエージェントの LADR である．

3. Virtual Private Community(VPC)

VPC は，各エージェントへのサービスの割り当てとユーザプロファイルの管理のための仕組みを与える．図 4 に VPC の概要を示す．VPC において，各エージェントへのサービスの割り当ては，ポリシーパッケージとして定義される．VPC-P (VPC-Platform) は，ユーザプロファイルに従ってポリシーパッケージを評価し，ユーザエージェントに適切な role を割り当てる．ユーザエージェントは，割り当てられた role に従って動作することで，サービスを利用する．

3.1 ポリシーパッケージ

図 5 にポリシーパッケージの構造を示す．ポリシーパッケージは，role とその role を割り当てるための rule からなる．Rule は，condition と role 名からなる．Condition は，rule を満足するための属性条件を示す．属性は，属性名と属性値で構成される．Role は，role 名とそのプログラムで構成される．

3.2 ユーザプロファイル

ユーザプロファイルは，属性名とその属性値で構成され，VPC-P の Profile Manager によって評価される．Profile Manager は，それぞれのユーザプロファイルに対する評価モジュールを持つ．評価モジュールは，属性名を受け取ると，その属性値を返す．Profile Manager は，ユー

図 4 VPC の概要

```

<policy-package>::=<rules><roles><contents>
<rules>::=<rule>|<rule><rules>
<rule>::=<condition><role-names>
<role-names>::=<role-name>|<role-name><role-names>
<condition>::="TRUE" | "and" <condition><condition>
               | "not" <condition>
               | "eq" <attribute>
               | "<" <attribute>
<attribute>::=<variable-name><value>
<roles>::=<role>|<role><roles>
<role>::=<role-name><program>

```

図 5 ポリシーパッケージの構造

ザプロファイル名と condition を受け取ると，そのユーザプロファイルに対する評価モジュールから属性値を受け取り，その condition を評価する．condition を評価した結果として TRUE，あるいは，FALSE を返す．このように，Profile Manager によって，ユーザプロファイルが condition を満足するかを調べることができる．

3.3 ポリシーパッケージの評価

Policy Evaluator は，ポリシーパッケージの各 rule をユーザプロファイルに従って評価することで，適切な role をユーザエージェントに割り当てるための機能を持つ．Policy Evaluator は，Profile Manager に各 rule の condition の評価を要求する．Profile Manager は，condition を評価し，その結果を TRUE，あるいは，FALSE で返す．Policy Evaluator は，その結果から適当な role を選択し，ユーザエージェントに割り当てる．

3.4 エージェント間の協調

エージェント間の協調は，Community Manager によって実現される．Community Manager は，Dataflow Computing[Amamiya 84]に基づいた FRM (Field Reactor Model) [岩尾 02] の機能を備え，パターンマッチングによる柔軟なエージェント間の協調を実現する．Community Manager にメッセージが送られると，FRM のパターンマッチングの仕組みによって適当なエージェントを見つけ出し，そのエージェントにメッセージを送る．

4. KODAMA と VPC の統合

KODAMA は，コミュニティの概念によるサービスや情報の分散管理とエージェントに物理的なネットワーク環境を意識させない通信環境を与える．一方，VPC は，各エージェントへサービスを割り当てるための仕組みとユーザプロファイルの管理のための仕組みを与える．そこで，KODAMA と VPC を統合し，表 1 に示す方法によって，ユビキタス環境を実現するための枠組みを構築する．

表 1 KODAMA と VPC の統合によるユビキタス環境を実現するために必要な機能の実現方法

機能	実現方法
サービスや情報の分散管理	コミュニティごとにポリシーパッケージを管理
サービスの利用	エージェントが role を獲得し、その role に従って動作することで、サービスを利用
物理的な通信のサポート	ACZ の働きによって、エージェントはお互いの物理的な位置を意識することなく通信
情報の保護	・メッセージフィルタによってコミュニティ内外に隔てるメッセージをフィルタリング ・ public, private プロファイルによって、エージェント間で交換される情報を制限

図 6 ポリシーパッケージの登録（左）とユーザエージェントへの role の割り当て手順（右）

4.1 ポリシーパッケージとユーザプロファイルの管理

KODAMA では、ポータルエージェントによってコミュニティが独立に管理される。そこで、KODAMA と VPC の統合システムでは、ポリシーパッケージの管理をポータルエージェントのアプリケーションユニットの機能として、ユーザプロファイルの管理をすべてのエージェントのアプリケーションユニットの機能として実現する。ここで、ユーザエージェントに role を割り当てるためには、ポリシーパッケージをユーザプロファイルで評価することが必要となる。しかし、すべてのユーザプロファイルをポータルエージェントに対して公開することは、プライバシー保護の上で好ましくない。そこで、public と private の 2 つのユーザプロファイルを用意する。public プロファイルは他のエージェントに公開してもよいプロファイルであり、private プロファイルは外部に公開されないプロファイルである。また、ポリシーパッケージの condition に secure 属性の指定が可能であり、secure 属性が指定された condition はポータルエージェントによってのみチェックされる。

ポータルエージェントへのポリシーパッケージの登録とユーザエージェントへの role の割り当ては図 6 の手順によって行われる。

ポータルエージェントへのポリシーパッケージの登録は、サービスを提供するエージェント（サービス提供エージェント）によって行われる（図 6 左）。サービス提供エージェントは、コミュニティに加わるために、そのコミュニティのポータルエージェントに JOIN メッセージを送信する。JOIN メッセージを受け取ったポータルエージェントは、join-filter によって、そのエージェントのコミュニティの参加を許可するかどうかを判断し、その結果、コミュニティへの参加を許可すれば UID を生成して返す。コミュニティへの参加が許可されれば、サービ

ス提供エージェントは、提供するサービスについてのポリシーパッケージをポータルエージェントに送信する*2。ポータルエージェントは、受け取ったポリシーパッケージを自分のコミュニティで提供可能なサービスとして登録する。

また、ユーザエージェントへの role の割り当ては、図 6 右のように行われる。ユーザエージェントのコミュニティへの参加動作は、サービス提供エージェントと同様なので省略する。ユーザエージェントは、コミュニティに加わると、加わったコミュニティのポータルエージェントに public プロファイルを送信する。ポータルエージェントは、登録されたポリシーパッケージの中から、secure 属性が指定されたすべての condition を満たし、かつ、public プロファイルに適合する rule を返す。ここで、ユーザエージェントに返される rule は、public プロファイルですでにチェックされた condition を除いた rule となる。ユーザエージェントは、返された rule を private プロファイルで評価し、適合した role をポータルエージェントに対して要求する。ポータルエージェントは、要求された role を返す。ユーザエージェントは、返された role に従って動作することでサービスを利用する。

このように、ユーザプロファイルを public と private の 2 つに分割することで、ユーザエージェントは、private プロファイルを公開することなく必要なサービスを利用できる。また、サービス提供エージェントは、condition に secure 属性を指定することで、ユーザエージェントに提示されることのない属性値条件を定義できる。

4.2 VPC-P の実装

VPC-P は、Profile Manager と Policy Evaluator、Community Manager で構成される。

Profile Manager は、ユーザプロファイルの管理機能を提供する。KODAMA と VPC の統合モデルにおいて、Profile Manager の機能は、ユーザエージェントのアプリケーションユニットの機能として実現される。

Policy Evaluator の機能は、ユーザエージェント、ポータルエージェントの双方によって実現される。ポータルエージェントは、自分が管理するポリシーパッケージをユーザエージェントからの public プロファイルによって評価し、適合した rule をユーザエージェントに返す。ユー

*2 コミュニティへの参加が許可されなければ、サービス提供エージェントの例外プログラムが呼び出される。

図7 アプリケーション例

エージェントは、ポータルエージェントから返された rule を private プロファイルによって評価し、ポータルエージェントから role を獲得する。

エージェント間の協調を管理する Community Manager の機能は、plug-in モジュールの適用ルールを FRM のパターンマッチングメカニズムに対応させることで実現する。

4.3 アプリケーション例

KODAMA と VPC の統合システムの例として、ホテルに設置された情報家電機器を利用するためのアプリケーションを示す。図7に示すように、あるホテルを表す Hotel コミュニティが存在する。Hotel コミュニティには、Front エージェントが存在し、電話の利用権を発行するためのサービス (auth:authorize) を提供する。また、Hotel コミュニティには、ホテルの部屋を表す Room-A コミュニティが存在する。Room-A コミュニティには電話やテレビの機能を提供するエージェント (Tel エージェント、TV エージェント) が存在する。また、ホテルには、センサが取り付けられ、そのセンサによってユーザの位置を特定できるものとする。

このとき、誰でもテレビを利用することができるとすると、そのためのポリシーパッケージの rule は次のようになる^{*3}。

```
<rules><rule>
  <condition>true</condition>
  <role-name>tv</role-name>
</rule></rules>
```

また、電話の利用権を持つユーザだけが電話を利用することができるようにすると、そのためのポリシーパッケージの rule は次のようになる。

```
<rules><rule>
  <condition secure=true>
    authority of phone = true
  </condition>
  <role-name>tel</role-name>
</rule></rules>
```

フロントで電話を利用するための料金を支払ったユーザだけが電話を利用するための権利を得ることができる

とすると、そのためのポリシーパッケージの rule は次のようになる。

```
<rules><rule>
  <condition secure=true>
    paid = true
  </condition>
  <role-name>authorize</role-name>
</rule></rules>
```

ユーザ X がフロントで電話を利用するための料金を支払っているとすると、ユーザ X のユーザエージェント (userX) が管理するユーザプロファイルは次のようになる。

```
<profile>
  <public>
    paid = true
  </public>
  <private>
    name = userX
  </private>
</profile>
```

また、ユーザ Y がフロントで電話を利用するための料金を支払っていないとすると、ユーザ Y のユーザエージェント (userY) が管理するユーザプロファイルは次のようになる。

```
<profile>
  <public></public>
  <private>
    name = userY
  </private>
</profile>
```

§1 Room-A コミュニティへのユーザエージェントの参加

ユーザエージェントの Room-A コミュニティへの参加は、図6の手順に従う。userX が Room-A コミュニティに加わると public プロファイルを portalA に送信する。portalA は、受け取った public プロファイルでポリシーパッケージを評価する。その結果、電話の利用権発行サービスとテレビの rule が public プロファイルに適合する。このため、portalA は、電話の利用権発行サービスとテレビの rule を userX に返す。

```
<rules>
  <rule>
    <condition>true</condition>
    <role-name>authorize</role-name>
  </rule>
  <rule>
    <condition>true</condition>
    <role-name>tv</role-name>
  </rule>
</rules>
```

userX は、受け取った rule を private プロファイルで評価する。その結果、private プロファイルに適合した authorize role と tv role を portalA から獲得する。userX は、tv role に従って TV エージェントに操作要求を出すことでテレビを利用する。また、authorize role に従って電話を利用するための権利を獲得し、再び、図6の手順を行うことで tel role を得ることができる。

一方、userY は、Room-A コミュニティに加わったとしても、paid 属性を持たないため、電話の利用権発行サー

*3 ポリシーパッケージとユーザプロファイルの表記を一部省略する。

ビスを獲得することができない．結果として，userY は，電話を利用できない．このため，userY は，テレビだけしか利用できないこととなる．

§2 新しいコミュニティの作成

コミュニティの作成は，まず，ポータルエージェントの生成によって行われる．ポータルエージェントは生成されると，上位コミュニティのポータルエージェントからポリシーパッケージを受け取り，登録する．次に，サービス提供エージェントが生成される．サービス提供エージェントは，図 6 の手順で自分が提供するサービスのためのポリシーパッケージをポータルエージェントに登録する．

例えば，図 7 では，Room-B コミュニティを作成する場合，まず，portalB が生成される．portalB は，Hotel コミュニティのポータルエージェントから，電話の利用権発行サービスのためのポリシーパッケージを獲得する．次に，Room-B の情報家電機器の構成に基づいてサービス提供エージェント（tel エージェント）が生成される．Tel エージェントは，電話を利用するためのポリシーパッケージを portalB に登録する．このことで，Room-B コミュニティに加わるユーザエージェントに対して，電話の利用権発行サービスと電話を利用するためのサービスを提供できるようになる．

§3 コミュニティ間通信

コミュニティ間の通信は，2・2・1 節で述べたメッセージフィルターによって制限される．例えば，portalA，portalB のメッセージフィルターが tel role によって送信されたメッセージしか通さないとすると，tel role 以外から送信されたメッセージは，コミュニティ内（外）からコミュニティ外（内）へ送信されることはない．よって，Room-A コミュニティ内のエージェントと Room-B コミュニティ内のエージェントとの通信は，tel role を利用したものだけに限られる．

また，これらのエージェントは，ACZ のサポートによって，お互いの物理的な位置を意識せず，ネットワーク透過的に通信することができる．

5. 広告配信実験システム

我々は，KODAMA と VPC の統合システムを利用し，名古屋の大須商店街（図 8）で実験を行った．実験では，商店街に約 100 個の受信機（図 9 右）を設置し，携帯電話を持つユーザに発信機（図 9 左）^{*4}を貸し出した．ユーザが発信機を持って商店街を歩くと，商店街に設置した受信機によってユーザの位置を検知し，ユーザのプロファイルと位置に基づいた広告情報をユーザの携帯電話に向けてメールで配信する．このように，ユーザは，行く先々の情報を携帯電話でリアルタイムに得ることができる．

図 8 大須商店街マップ

図 9 発信機（左）と受信機（右）

5・1 システムの構成

図 10 に実験システムの構成を示す．実験では，ユーザコミュニティと各商店街のコミュニティを準備する．

ユーザコミュニティには，ポータルエージェントと，ユーザエージェントを生成するエージェント（以下，ユーザ生成エージェントと呼ぶ）が存在する．ユーザ生成エージェントは，アプリケーションユニットの機能として，ユーザ登録のための Web ユーザインタフェースを持つ．ユーザが興味や携帯電話のメールアドレス，年齢などを登録すると，ユーザ生成エージェントは登録された情報を基にユーザエージェントを生成する．生成されたユーザエージェントは，ユーザの位置に従ってコミュニティを移動^{*5}し，移動先のコミュニティで提供されているサービスを利用する．ここで，ユーザエージェントのコミュニティの移動は，次の手順で行われる．

- (1) 移動前に参加しているコミュニティからの脱退．脱退通知（LEAVE メッセージ）を脱退するコミュニティのポータルエージェントに送信する．
- (2) 移動先のコミュニティへの参加．コミュニティへの参加は，図 6 に示した手順で行う．

各商店街のコミュニティには，はじめ，ポータルエージェントだけが存在する．ポータルエージェントは，サービス提供エージェントからユーザに配信する広告についてのポリシーパッケージを受け取り登録する．

^{*4} 7 秒に 1 回受信機に向けて信号を発信する．

^{*5} ユーザエージェントはコミュニティを移動するだけで，マシン間を移動するわけではない．

図 10 実験システムの概要

5.2 システムの動作

本実験では、まず、ユーザ登録のためのインタフェースを通して、ユーザの情報（興味や性別、携帯電話のメールアドレスなど）を登録する。ユーザ生成エージェントは、登録されたユーザの情報からユーザプロフィールを生成^{*6}し、そのユーザプロフィールを持つユーザエージェントを生成する。ユーザが商店街を歩くと受信機が発信機からの信号を受信し、そのユーザの位置情報をユーザエージェントに知らせる。ユーザエージェントは、位置情報を public プロファイルとして登録する。ユーザが同一商店街を移動していれば、ユーザエージェントはポータルエージェントに public プロファイルと共に位置更新メッセージを送信する。ポータルエージェントは、public プロファイルをポリシーパッケージで評価し、適合した rule を返す。ユーザエージェントは、受け取った rule を private プロファイルで評価して role を獲得する。ユーザエージェントは、獲得した role に従って動作することで、サービスを利用する。ユーザが異なる商店街に移動していれば、ユーザエージェントはコミュニティを移動し、移動先のコミュニティのポータルエージェントに public プロファイルを送信する。ポータルエージェントは、public プロファイルをポリシーパッケージで評価し、適合した rule を返す。ユーザエージェントは、受け取った rule を private プロファイルで評価して、role を獲得する。ユーザエージェントは、獲得した role に従って動作することで、サービスを利用する。

5.3 実験で利用したポリシーパッケージ

本実験で用いたポリシーパッケージの一例として、店の近くにいる、家具に興味があるユーザに対して情報を配信する場合のポリシーパッケージを示す。

```
<policy-package>
  <rules>
    <rule>
      <condition>
        interest = furniture
        gender = male
        :
        location = area(x1, y1, x2, y2)
```

*6 興味や年齢などの個人の特定が難しいと思われる情報を public プロファイルとして設定し、名前や携帯電話のメールアドレスなどの個人の特定が容易と思われる情報を private プロファイルとして設定した。

```
String mail_server = "server address";
String subject = "大須@得！家具の...";
String body = "見ているだけで楽しくなっちゃう家具が...";
void invoke(){
  String mail_addr = getPrivateProfile("mail_address");
  Mailer.sendMail(mail_server, mail_addr, subject, body);
}
```

図 11 furniture-ad role のプログラム概要

図 12 操作性（左）と配信されるメールの量（右）

```
</condition>
<role-name>furniture-ad</role-name>
<rule>
<roles>
  <role>
    <role-name>furniture-ad</role-name>
    <program>instance of furniture-ad role</program>
  </role>
</roles>
</policy-package>
```

このポリシーパッケージによって、家具に興味があり、area(x1, y1, x2, y2) の範囲のユーザエージェントは、furniture-ad role を獲得する。ここで、furniture-ad role は、携帯電話への広告メール配信サービスであり、図 11 のプログラムによって実現される。

ここで、getPrivateProfile は、指定された private プロファイルの情報を取り出すためのメソッドである。Mailer.sendMail は、mail_addr へメールを送信するためのメソッドである。ユーザエージェントは、図 6 の手順に従って role を獲得し実行することで、行った先々にある店の情報をユーザの携帯電話に向けてメールで送信する。商店街を歩くユーザは、携帯電話で受け取ったメールを読むことで、行った先々にある店の情報を得ることができる。メールの本文には、広告の概要と URL を含んでおり、ユーザは、URL にアクセスすることでその詳細な情報も知ることができる。

5.4 実験結果と考察

実験は、2003 年 3 月 15 日から 4 月 30 日の期間に行われ、901 人の実験参加者と 56 店舗からの 107 種類の広告メールを得て行った。

実験の結果、10579 通のメールがユーザに送信され、URL が示す Web ページは、1625 回参照された。アンケートの結果では、図 12、図 13 の結果が得られた。また、693 人中 200 人がメールを見て、平均 15256 円の商品を購入したと回答した。

本実験システムでは、情報を見るためのユーザインタ

図 13 メールを受け取るタイミング (左) とメールの配信間隔 (右)

フェースとして携帯電話を利用している。図 12 左の結果と約 50%の人がネット対応の携帯電話を利用している状況 [総務省 02] を考えると、ユビキタス環境で携帯電話をユーザインタフェースとして利用することは有効であると思われる。配信されるメールの量 (図 12 右) では、37%の人が送られてくるメールの量が多すぎると答えた一方、18%の人が少なすぎると答えた。これは、システムを積極的に利用してくれた店に興味があるユーザは多くのメールを受け取ったが、システムの利用に消極的だった店にしか興味がないユーザはあまりメールを受け取らなかったためだと考えられる。

ユーザがメールを受け取るタイミングは、図 13 左で示されるような結果が得られた。ユーザが広告が配信されるエリアに入ったあとその広告を受け取るまでの時間は、(ユーザの位置を検出するまでの時間) + (本システムの処理時間) + (メールを送信してからユーザの携帯電話にメールが届くまでの時間) である。携帯電話の電波状況によってメールを送信してからユーザのメールが携帯電話に届くまでの時間が異なるが、本実験のような商店街での情報配信システムでは、7 秒に 1 回の位置検出で、タイミング良くユーザに広告メールが届けられたことがわかる。また、メールが配信される間隔は、図 13 右で見られるように 63%のユーザがちょうどいいと答えた一方、29%のユーザが短すぎると感じている。この問題は、ユーザプロフィールを細かく設定し、配信されるメールを限定することや一定時間あたりに配信されるメール数を制限することで解決できると考えられる。

ユーザの感想としては、1) 色々な情報や知らないお店を知ることができて、すごく楽しかった。2) もう少したくさんのお店の中が見たい。3) たくさんのメールを受け取ることは嫌だ。などが、店の感想としては、4) 今まで来なかった客が来るようになった。5) 売り上げが上がった。などが得られた。感想 1, 2, 4, 5 で見られるように、我々の実験システムがユーザや店に役立ったことがわかる。一方、感想 3 で示されるような、否定的な感想も受け取った。この問題は、public プロファイルや private プロファイルの情報の変更をユーザの携帯端末上でできるようにすることで解決できると考えられる。

5.5 実験システムの機能評価

ユビキタス環境を実現するために必要な機能として、表 1 に示した、サービスや情報の分散管理、サービスの利用、物理的な通信のサポート、情報の保護について評価する。

§ 1 サービスや情報の分散管理

本システムでは、サービスや情報をコミュニティごとに管理する仕組みを提供する。実験システムでは、商店街ごとにコミュニティを定義し、商店街ごとに配信される情報を管理している。このことによって、本システムでは、サービスや情報をコミュニティごとに分散して管理し、負荷の集中を防ぐ。

§ 2 サービスの利用

ユーザエージェントは、public プロファイルをポータルエージェントに送信し、その結果として role を獲得する。role は、サービスを利用するための手順を記したプログラムである。ユーザエージェントは、獲得した role に従って動作することで、サービスを利用する。実験システムにおいて、role はユーザの携帯電話に向けてメールを送信するプログラムであり、獲得した role に従ってユーザエージェントが動作することにより、ユーザエージェントは、ユーザの属性と位置に基づいたメールをユーザの携帯電話に向けて送信することができる。

一方、private プロファイルを公開しない結果、利用できるサービスが必ずしもユーザの満足にいかないものとなることも考えられる。このことについては、public, private プロファイルをコミュニティや通信相手のエージェントに応じて詳細にすることによって、個人情報保護しつつより満足のいくサービスをユーザに提供できるようになると考えられる。

§ 3 物理的な通信のサポート

本システムでは、物理的通信環境として ACZ を準備している。ACZ は、論理的なエージェントの構造を物理的なアドレス空間にマッピングする機構である。このことによって、エージェントは物理的通信環境を意識せず通信することができる。すなわち、エージェントの構成がシステムを実現しているマシンの構成に依存しないため、エージェントの構成を変更することなく、マシンの増設による負荷分散を実現できる。例えば、商店街のコミュニティごとにそれぞれ別のマシン上で動作させたとしても、一つのマシン上で動作させたとしても、エージェントの構成や実装は変わらない。エージェントは、システムの物理的構成を意識することなく論理的なコミュニティ空間上で動作する。

§ 4 個人情報の保護

本システムでは、public と private の 2 つのユーザプロフィールとメッセージフィルタによって、個人情報を保護するための枠組みを実現している [雨宮 03]。ユーザエージェントは、private プロファイルを公開せずに public プロファイルだけを公開して、サービスや情報を

獲得する。また、メッセージフィルタによって、コミュニティ内外に渡って送信されるメッセージをフィルタリングする。実験では、興味、年齢、名前、メールアドレスを属性として用いた比較的簡単な個人情報の保護を行ったが、さらにきめ細かな保護属性記述方法に関しては今後の課題である。

6. 関 連 研 究

Jini[Jini] は、マシンやプログラムをグループ化して連合体を形成し、互いのリソースを必要に応じて利用しあえる環境を実現するために開発された Java ベースの分散システムである。Jini において、新しい機器がネットワークに接続されると、Discovery プロトコルを利用して Lookup サービスに提供するサービスを登録する。クライアントは、Java プログラミング言語の型(クラス)を手がかりとして、Lookup サービスから目的のサービスを見つける。Universal Plug and Play (UPnP) [UPnP] は、パソコンや周辺機器、AV 機器、家電製品などをネットワークを通じて接続し、相互に機能を提供しあうための技術である。UPnP では、XML で記述された情報を用いてネットワークに接続された機器が互いに認識し通信する。しかし、これらは、本システムのように public, private の 2 つのユーザプロファイルによってユーザの個人情報を守りつつ、目的のサービスを利用するための枠組みを提供しない。また、本システムのサービス提供エージェントとユーザエージェントは、お互いの物理的なアドレスが変わる環境においても、そのことを意識せずに通信することができる。

C-MAP[角 98, 角 00] は、博物館や博覧会などの展示会場を想定し、展示会場において各見学者に展示ガイド情報を提供することを目的とする。C-MAP では、エージェントサーバ上で各ユーザのガイドエージェントが起動し、各ユーザの現在位置や興味に合わせて個人化されたガイド情報を携帯情報端末に送る。しかし、C-MAP では、位置や興味に合わせたガイド情報を携帯情報端末に送ることを目的としており、我々のシステムのようにサービスを利用するための手順(role)を獲得し、獲得した role に従って動作することでサービスを利用することを対象としていない。我々の実験システムでは、ユーザの携帯情報端末への情報配信を他のエージェントとの通信を必要としない role として実現している。

ユビキタスパersonalizeエージェント[服部 03] は、ユビキタス環境でユーザに関する各種データを自動収集することでユーザの様々な状況に対応するとともに、パーソナライズされたコンテンツやサービスをユーザに対して効率的に提供することを目的とする。しかし、本システムのように各ユーザの位置や興味に合わせてユーザエージェントがサービスを利用するための手順を獲得し、その手順によってサービスを利用することを対象としてい

ない。

7. お わ り に

本稿では、KODAMA と VPC を統合したユビキタス環境実現のための枠組みを提案した。KODAMA はコミュニティによる分散管理と物理的な位置に依存しないエージェント間通信を与える。VPC は各エージェントへサービスを割り当てるための仕組みとユーザプロファイルの管理のための仕組みを与える。このため、KODAMA と VPC を統合することで、提供されるサービスをコミュニティごとに独立に管理し、エージェントがコミュニティに加わった際に、そのコミュニティで提供されているサービスを利用できるようにするための枠組みを実現した。我々は、KODAMA と VPC の統合システムを利用し、情報家電機器を利用するための応用例を示した。応用例では、サービスがコミュニティごとに分散して管理され、ユーザがコミュニティに加わることで、そのコミュニティで提供されているサービスを利用できることを示した。

また、ユーザの属性と位置に基づいた情報配信システムを実現し、名古屋の商店街で実験を行った。実験でユーザは携帯電話と発信機を持って商店街を歩くことで、行く先々にある店の情報を携帯電話へのメールとして受け取る。約 900 人のユーザの協力が得て実験を行った結果、本システムがユビキタス環境において効果的に働くことを確認した。

今後の課題として、public, private プロファイルをコミュニティや通信相手のエージェントに応じて詳細に定義できるようにすることや [Damianou 01, Karjoth 97] などで論じられているような個人情報保護の詳細を定義することが挙げられる。

謝 辞

本研究は通信・放送機構(TAO)からの委託研究「相互接続時のセキュリティポリシーの管理技術に関する研究開発」の一環として実施した。

◇ 参 考 文 献 ◇

- [Amamiya 84] M. Amamiya, R. Hasegawa: Dataflow Computing and Eager and Lazy Evaluation, New Generation Computing, 2, pp. 105-129 (1984)
- [雨宮 03] 雨宮聡史, 岩尾忠重, 雨宮真人: マルチエージェント方式によるセキュアでオープンな情報流通ネットワークの構築, 情報処理学会研究報告, ユビキタスコンピューティングシステム, No. 2, pp. 1-6 (2003)
- [Cutkosky 93] Mark R. Cutkosky, Robert S. Englemore, Richard E. Fikes, Michael R. Genesereth, Thomas R. Gruber: PACT: An Experiment in Integrating Concurrent Engineering Systems, IEEE Computer 26(1), pp. 28-38 (1993)
- [Damianou 01] N. Damianou, N. Dulay, E. Lupu, M. Sloman: The Ponder Policy Specification Language, Policy 2001, pp. 18-38 (2001)
- [服部 03] 服部正典, 長健太, 大須賀昭彦, 一色正男, 本位田真一: ユビキタス環境における Context-Aware なパーソナルエー

ジェントの構築とその実証実験, 電子情報通信学会論文誌, Vol. J86-D-I, No. 8, pp. 543-552 (2003)

[Iwao 01] T. Iwao, Y. Wada, M. Okada, M. Amamiya: A Framework for the Exchange and Installation of Protocols in a Multi-Agent System, CIA-2001, pp. 211-222 (2001)

[岩尾 02] 岩尾忠重, 岡田誠, 雨宮真人: マルチエージェント連携基盤: Field Reactor Model, 電子情報通信学会論文誌, Vol. J85-A, No. 8, pp. 895-907 (2002)

[Iwao 03] T. Iwao, S. Amamiya, G. Zhong, and M. Amamiya: Ubiquitous Computing with Service Adaptation using Peer-to-Peer Communication Framework, FTDCS 2003, pp. 240-248 (2003)

[Jini] Jim Waldo: Jini アーキテクチャの概要, <http://jp.sun.com/products/wp/java/jiniarch.pdf>

[Karjoth 97] G. Karjoth, D. B. Lange, M. Oshima: A Security Model for Aglets, IEEE Internet Computing, Vol. 1, No. 4, pp. 68-77 (1997)

[総務省 02] 総務省統計局: IT 関連統計資料集 (平成 15 年版), <http://www.stat.go.jp/data/it/index.htm>

[角 98] 角康之, 江谷為之, Sidney Fels, Nicolas Simonet, 小林薫, 間瀬健二: C-MAP: context-aware な展示ガイドシステムの試作, 情報処理学会論文誌, Vol. 39, No. 10, pp. 2866-2878 (1998)

[角 00] 角康之, 間瀬健二: 実世界コンテキストに埋め込まれたコミュニティウェア, 情報処理学会論文誌, Vol. 41, No. 10, pp. 2679-2688 (2000)

[高橋 03] 高橋健一, 雨宮聡史, 鍾国強, 松野大輔, 峯恒憲, 雨宮真人: エージェント・プラットフォームを相互運用するためのメッセージ通信プロトコル, 電気学会論文誌 C, Vol.123-C, No.8, pp. 1503-1511 (2003)

[Ubiquitous 02] 特集 コピキタスコンピューティング世界を実現する革新的ネットワーク技術, 情報処理, Vol. 43, No. 6, (2002)

[UPnP] Universal Plug and Play, <http://www.upnp.org/>

[Weiser 91] Mark Weiser: The Computer for the Twenty-First Century, Scientific American, pp. 94-104 (1991)

[Zhong 02] G. Zhong, S. Amamiya, K. Takahashi, T. Mine and M. Amamiya: The Design and Implementation of KODAMA System, IEICE Transactions INF.& SYST., Vol. E85-D, No 4, pp. 637-646 (2002)

〔担当委員: 沼尾正行〕

2003 年 10 月 31 日 受理

著者紹介

高橋 健一

1976 年生。1999 年九州大学工学部情報工学科卒業。2001 年同大学院システム情報科学研究科修士課程修了。2004 年同大学院システム情報科学府博士課程修了。博士 (工学)。現在, 九州大学大学院システム情報科学研究院学術研究員。マルチエージェントシステムの研究に従事。情報処理学会, 電子情報通信学会, 電気学会 各会員。

雨宮 聡史

1973 年生。2004 年九州大学大学院システム情報科学府博士課程入学。現在に至る。マルチエージェントシステム, 分散情報検索システム, 並列アーキテクチャの研究に従事。

岩尾 忠重

1989 年大分大学電子工学科卒業。1991 年九州大学大学院総合理工学研究科修士課程修了。1991 年 (株) 富士通研究所入社。グループウェアシステム, エージェントシステムの研究に従事。情報処理学会, ソフトウェア科学会 各会員。

鍾 国強

1971 年生。1993 年中国福州大学卒業。1996 年中国四川大学修士課程修了。2002 年九州大学大学院システム情報科学研究科博士課程修了。現在, 九州大学ベンチャー ビジネスラボラトリーポスドク研究員。工博。マルチエージェントシステムの研究に従事。IEEE, ACM, 電子情報通信学会 各会員。

貝沼 達也

1956 年生。1983 年東京工業大学数学科卒業。同年 (株) パナファコム入社, 言語プロセッサ・UNIX OS 開発に従事。1992 年 (株) 富士通プライムソフトテクノロジ入社, 分散コンポーネントアーキテクチャ, モバイルエージェント, 仲介エージェントの研究開発に従事。

雨宮 真人 (正会員)

1942 年生。1967 年九州大学工学部電子工学科卒業。1969 年同大学大学院工学研究科修士課程修了。同年日本電信電話公社武蔵野電気通信研究所入所。以来, プログラミング言語・処理系, 自然言語理解, データフロー・アーキテクチャ, 並列分散処理, 関数型/論理型言語, 知能処理アーキテクチャ, マルチエージェントシステム等の研究に従事。現在, 九州大学大学院システム情報科学研究院知能システム学部門教授。電子情報通信学会, ソフトウェア科学会, 人工知能学会, IEEE, ACM, AAAI 各会員, 電子情報通信学会フェロー, 情報処理学会フェロー。工学博士。