社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

ユビキタスクラウドにおけるユーザコンテキスト管理サービスの一考察 ～適応型通知サービスの実装～

江上　公一†　　柗本　真佑†　　中村　匡秀†

† 神戸大学　〒 657–8531 兵庫県神戸市灘区六甲台町 1–1
E-mail: †egami@ws.cs.kobe-u.ac.jp, ††{shinsuke,masa-n}@cs.kobe-u.ac.jp

あらまし　「いまだけ・ここだけ・あなただけ」を目的とする個人に即した適応型ユビキタスサービスが人と情報環境の円滑な関係構築のために求められている．我々は，クラウドのコンセプトに基づいて，ユーザの要求や状況に応じて資源をサービスとして調達する適応型ユビキタスサービス基盤（ユビキタスクラウド）を提案・実装している．本論文では，ユーザコンテキストに応じてサービスの振る舞いを切り替えるためのモデルを提案し，ユーザコンテキストを管理するサービス「コンテキストマネージャ」を実装する．また，ケーススタディとしてユーザコンテキストに応じて通知手段を切り替える適応型メッセージ通知サービスを実装し，コンテキストマネージャの有効性を評価する．
キーワード　適応型ユビキタスサービス，クラウドコンピューティング，ユーザコンテキスト

A Management Service of User Context for Ubiquitous Cloud

Koichi EGAMI†, Shinsuke MATSUMOTO†, and Masahide NAKAMURA†

† Kobe University　Rokkoudai-cho 1–1, Nada-ku, Kobe, Hyogo, 657–8531 Japan
E-mail: †egami@ws.cs.kobe-u.ac.jp, ††{shinsuke,masa-n}@cs.kobe-u.ac.jp

Abstract　The adaptive ubiquitous services, which dynamically adapt behaviors to requirements and contexts, are one of the major challenges in the ubiquitous computing. To facilitate the management of ubiquitous service resources, we are currently implementing a novel platform called ubiquitous cloud, borrowing the concept of the cloud computing. In this paper, we present models for changing service behavior to adapt user context. And we implement "Context Manager" which manages user context. In the end of this paper, we evaluate our model by implementing actual ubiquitous service which changes methods of messaging as to user context.
Key words　ubiquitous services, cloud computing, user context

1. はじめに

社会的な要求の多様化に伴い，ユビキタス環境における様々なサービス（ユビキタスサービス）は，より一層ユーザの嗜好や周辺環境の状況に適応できるように求められている．このような**適応型ユビキタスサービス**の実現のために，現在では AI プランニング [1] や，サービス再構築 [2]，サービス資源の動的発見 [3] などの技術が提案されている．しかし現状では，制御対象となる機器やデバイス，外部環境の状況を捉えるためのセンサといった，**ユビキタスサービス資源**の相互運用性を保証する標準的なプラットフォームが確立されていない．そのため，上記の適応型ユビキタスサービス実現のための技術を，実用的なサービス上で実装することは容易ではない．

我々は先行研究 [4] において，ユビキタスサービス資源を効率的に一元管理し，サービス資源の利用や適応型ユビキタスサービスの開発を支援するプラットフォーム，**ユビキタスクラウド**を提案している．ユビキタスクラウドは以下の 2 点の要求から，クラウドコンピューティング [5] のコンセプトに基づいて設計されている．

- 多種多様なサービス資源を扱えること
- 多種多様なユーザが利用できること

これらの要求に応えるため，ユビキタスクラウドは以下の 4 つのキーコンポーネントを持つ．

- **サービス資源レジストリ：**ウェブサービスとして利用可能な様々なユビキタスサービス資源を一元提供する．
- **適応型資源ファインダ：**現在のユーザや環境のコンテキスト，及び与えられた要求に基づいて最適なサービスオペレーションを発見する．
- **コンテキストマネージャ：**下層のセンササービスを用いて，実世界の様々な環境情報（コンテキスト）を集める．

- **サービスコンシェルジュ：**エンドユーザ向けに公開されている適応型アプリケーションを推薦・実行する．

先行研究 [4] では，「サービス資源レジストリ」及び「適応型資源ファインダ」について議論した．本稿では，特に「コンテキストマネージャ」について議論する．

ユビキタス環境においては，ユビキタスサービスのユーザやその周辺環境は，その時々に応じて動的に変化することが想定される．一例として，ユーザへメッセージを送信する適応型メッセージ通知サービスを考える．この場合，通常はメールなどの非同期の通信手段を利用すれば良い．一方，内容に緊急性がある場合は，電話などの同期通信手段を用いて直接伝える必要がある．ところが，メッセージの受け取り手が事情により手を離せない状況にあれば，電話に出ることができない．その場合には，室内のスピーカーを利用してメッセージを音声として再生したり，電話以外のデバイスで受け取り手にメッセージを伝えられたり出来ることが望まれる．従って，適応型ユビキタスサービスでは，このような「ユーザ」や「周辺環境」のコンテキストに応じて動的に振る舞いを変えることが重要となる．

上記のコンテキスト推定に係る要素のうち，本稿では特に「ユーザ」に関するコンテキストに対してモデル化を行う．そのためのアイデアとして，ユーザの身体に関する制約を考慮したユーザモデルを提案する．提案手法は以下の 2 つのモデルから構成される．

- **ユーザ状態モデル：**ユーザの状態を表すモデル．ユーザの現在位置や利用できる身体部位（手や耳など）の情報を持つ．
- **オペレーション制約モデル：**サービス資源の機能（オペレーション）が，ユーザの身体部位に対してどのような制約を課すのかについての情報を持つ．

これらのモデルを用いることで，適応型ユビキタスサービスは現在のユーザの状態に合わた最適なオペレーションの選択が可能となる．例えば，台所で料理をしているユーザにメッセージを送信する場合，このユーザは「手」が占有された状態にあるため，「手」に対して制約を課す電話オペレーションは利用不可能となる．一方，台所に配置されたスピーカーの再生オペレーションは「耳」のみに対して制約を課すため，利用可能である．これらのユーザモデルを利用することで，適応型ユビキタスサービスが動的に振る舞いを切り替えることが可能となる．

本稿では，ユーザのコンテキストモデルを基盤としたコンテキストマネージャを提案し，ケーススタディとして上記の例題として示した適応型メッセージ通知サービスを実装する．

2. 準　　備

2.1 ユビキタスクラウド

我々は先行研究において，ユビキタスクラウドを提案している．ユビキタスクラウドは，サービス資源やユビキタスサービスそのものを効率的に管理・運用することで，適応型ユビキタスサービスの開発・利用を支援する．適応型ユビキタスサービスでは，一般的な計算機以外にもネットワーク家電や各種センサデバイスなどが利用されるため，クラウドは多種多様なユビキタスデバイスを扱えることが求められている．また，サービス資源の開発者やサービスの開発者，及びそれらの利用者など，クラウドに関するステークホルダには様々なものが存在すると想定できる．これらの事から，ユビキタスクラウドではクラウドコンピューティングのコンセプトを借りて設計を行っている．

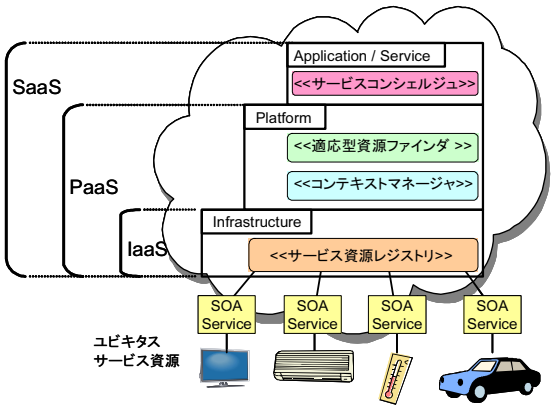

図 1　ユビキタスクラウドのアーキテクチャ

ユビキタスクラウドのアーキテクチャを図 1 に示す．ユビキタスクラウドは以下の 3 つの階層を持っており，それぞれクラウドの機能を果たすための主要なコンポーネントを持つ．

- **インフラストラクチャ（IaaS）レイヤ：**ネットワーク上で利用可能なサービス資源として実装された様々なユビキタスデバイスを，効率的に管理するための基本的な機能を提供する．「サービス資源レジストリ」を内部コンポーネントとして持つ．
- **プラットフォーム（PaaS）レイヤ：**適応型ユビキタスサービスの開発を容易にする開発用プラットフォームを提供する．「適応型資源ファインダ」「コンテキストマネージャ」を内部コンポーネントとして持つ．
- **アプリケーション/サービス（SaaS）レイヤ：**開発/公開された適応型サービスの一覧を提供する．「サービスコンシェルジュ」を内部コンポーネントとして持つ．

各階層に収められた 4 つのコンポーネントはユビキタスクラウドのキーコンポーネントであるが，これで十分というわけではない．ユビキタスクラウドはアプリケーションや目的によって，より多くのコンポーネントを追加することによって進化していかなければならない．

先行研究においては，「サービス資源レジストリ」「適応型資源ファインダ」について議論した．本稿では，「コンテキストマネージャ」について議論するため，これらの 3 つについて以下に簡単に述べる．

2.2 サービス資源レジストリ

サービス資源レジストリは，多様なサービス資源を説明するメタデータ（デバイスクラス，オペレーションタイプ，物理的場所，目的，ユーザ等）を保存している．サービスの開発者は，レジストリに登録されているメタデータから任意のサービス資源やオペレーションを特定できる．

2.3 適応型資源ファインダ

適応型資源ファインダは要求やコンテキストに対して最も適切なサービスオペレーション（以下，オペレーションとする）を推薦する．これは要求を解釈して，サービス資源レジストリ

に対する問い合わせの，より具体的なクエリに変換する．問い合わせの結果はコンテキストマネージャが提供するコンテキストでフィルタリングする．そのため，適応型資源ファインダは，適応型サービスの開発者に対して強力なツールとなる．

2.4 コンテキストマネージャ

下層のセンサデバイスを用いて，実世界からコンテキストに関する情報を集める．集められたセンサ情報はユーザの状態や要求，嗜好，環境情報を特徴付ける，より抽象的なコンテキストとして解釈される．このコンテキストは適応型ユビキタスサービスや適応型資源ファインダなどに幅広く利用される．コンテキストマネージャを利用することでサービス開発者は容易にコンテキスト情報を得ることが可能となる．

ユビキタスクラウドは，これらの 3 つのコンポーネントを活用することで，適応型ユビキタスサービスの開発を支援する．具体的には，サービス開発者は適応型資源ファインダにサービス利用者の現在の状態に応じたオペレーションを問い合わせ，必要であればコンテキストマネージャからコンテキストを取得し,「その場，その時，そのユーザ」に合わせたサービスの振る舞いを定義することが出来るようになる．

3. ユーザコンテキスト管理サービス

3.1 ユーザコンテキストと適応型ユビキタスサービス

適応型ユビキタスサービスでは，ユーザの状態に合わせて振る舞いを変えられることが求められる．このユーザの状態に関する情報を，本稿では「ユーザコンテキスト」と呼ぶ．ユーザコンテキストには，ユーザの現在の状況を表すあらゆる情報が含まれている．例えば，ユーザが「今どこにいて」,「どのような状態で」,「どのような環境にいるか」という情報である．また,「どのようなものが好きか」といった嗜好に関する情報も含まれる．これらの情報を基に，適応型ユビキタスサービスは「その場所」で「その状態，環境」に応じた「そのユーザの好む」振る舞いを選択する．

個人に合わせたサービスを提供するためには，ユーザコンテキスト情報のみでは不十分である．サービスが利用するオペレーションが，ユーザに対して与える制約についても考慮しなければならない．オペレーション利用における制約には，ユーザの状態にどのような影響を与えるかという情報が含まれる．例えば，電話を利用するオペレーションを考えた場合，ユーザは他のことに耳を傾けたり，手を使ったりするのが困難になるため，電話オペレーションは「耳」と「手」に対する制約を課すと見なすことができる．このオペレーション利用における制約を考慮することで，現在のユーザコンテキストに応じたオペレーションの選別が可能になる．

本節では，我々が提案するユーザコンテキストとオペレーション利用における制約のモデルについて説明し，モデル化されたデータを運用するための管理サービスについて述べる．

3.2 ユーザ状態モデル

ユーザコンテキストには，ユーザの現在の状況を表すあらゆる情報が含まれている．このユーザの現在の状況を表す情報としては「ユーザ自身の状態」と「ユーザ周辺の環境」，および「個々のユーザの嗜好」の 3 つが考えられる．これらの情報を利用することで，サービスは「ユーザ自身の状態」及び「ユーザ周辺の環境」に合わせて振る舞いを選択し,「個々のユーザの嗜好」を考慮することで個々のユーザに合った振る舞いを優先的に決定することが出来る．本稿では，これらの 3 つの情報のうち「ユーザ自身の状態」についてのモデル化を行う．

ユーザ自身の状態を表すモデルを「ユーザ状態モデル」と呼ぶ．適応型ユビキタスサービスを利用するユーザは，あらゆる場所，あらゆる状況下にあることが想定される．従って，ユーザ状態モデルに必要な情報として，以下のものを定義した．

- **現在位置：**ユーザが現在どの場所にいるかを表す．
- **身体部位の状態：**ユーザの各身体部位の現在の状態（使用可，不可等）を表す．

「現在位置」はユーザの現在状態を表す情報として，重要なコンテキストの一つである．例えば，メッセージ通知サービスの場合，ユーザが屋外にいればモバイル端末を選択し，宅内のある部屋にいればその部屋に設置された情報端末を選択するといった，サービスオペレーションの適切な選択が可能となる．

ユーザの現在状態を表すコンテキストとして,「身体部位の状態」の使用可/不可という情報に着目する．ユーザの現在状態を表現する手段としては，会議中や休憩中といった「状況」に着目した分類分けも可能であるが，その想定できる「状況」の全てを事前に定義しておくことは困難である．一方，身体部位は全ての人間に共通であり．数も有限なので，事前に定義することが容易となる．

3.3 オペレーション制約モデル

適切なオペレーションを選択する際には，ユーザ側のコンテキスト情報だけでは不十分である．選択されるオペレーション自体が，ユーザに対してどのような制約を課すのかについても考慮しなければならない．我々は，各オペレーションがユーザに対して課す制約をオペレーション制約モデルとして提案する．オペレーション制約モデルは，以下の情報を持つ．

- **同期性：**同期的に利用できるか，非同期的に利用できるかを表す．
- **身体制約：**ユーザの各身体部位に与える制約を表す．身体部位はユーザ状態モデルと対応し，各部位に対する制約あり/なしの情報を持つ．

「同期性」とは，オペレーションが同期的に利用できるか，非同期的に利用できるかを表す．例えば，電話は呼び出し側と受信側が同時に受話器を取らないと成立しない，同期オペレーションである．一方，メールは受信側がメール受信時にメールを確認できなくてもよい，非同期オペレーションである．同期的なオペレーションではユーザの現在状態を考慮する必要があるが，非同期的なオペレーションでは必ずしもユーザの現在状態を考慮する必要は無い．この情報は，確実に相手と連絡を取りたい場合に同期的に利用できるオペレーションを選択するといったように活用できる．

「身体制約」とは，そのオペレーションが，ユーザのどの身体部位に対し制約を課すかを表す．例えば，電話を利用する時には，ユーザは「手」「耳」「口」を使うので，電話オペレーショ

表 1　ユーザコンテキスト

現在位置	耳	目	手	口
台所	○	○	×	○

表 2　オペレーションの例

サービス資源	サービスオペレーション	利用可能場所	耳	目	手	口	同期性
電話	電話呼び出し	台所	×	○	×	×	同期
携帯電話	メール送信	どこでも	○	×	×	○	非同期
スピーカー	音声通知	台所	×	○	○	○	非同期

ンは「手」「耳」と「口」に制約を与えると定義できる．これは，ユーザ状態モデルにおける「身体部位の状態」と対応し，両者のデータを照合することでその時点のユーザが利用可能なオペレーションの一覧を取得することが出来る．

3.4　コンテキストマネージャ

これらのモデルを用い，効率的に最適なサービスの振る舞いを決定するために，我々はコンテキスト管理サービス「コンテキストマネージャ」を実装した．コンテキストマネージャは，以下の用途で使われると考えられる．

（1）　特定のユーザの状態を取得する

（2）　特定のオペレーションの制約を取得する

これらに対応する，コンテキストマネージャの API として以下のものを実装した．

（1）　getUserStatus(String userName)

（2）　getOperationAffect(String classID, String method)

getUserStatus() では，引数にユーザ名を指定することで対象ユーザの状態を取得することが出来る．getOperationAffect() では，引数にデバイスを一意に同定するクラス ID と，そのデバイスで使用する機能（メソッド）を指定することで，その機能を利用する際にユーザに課す制約を取得することが出来る．

3.5　使　用　例

コンテキストマネージャの使用例として，台所で洗い物をしているユーザに連絡をとるケースについて考える．このユーザの持っているコンテキストと，その時点で利用可能なオペレーションの一覧を，表 1 と表 2 に示す．なお，これ以降は表中の○を「制約なし」，×を「制約あり」として扱う．

また，サービス側が行う処理は以下の通りである．

ステップ 1：通知先の対象となるユーザの状態を取得する

ステップ 2：対象のユーザの現在位置で利用できるオペレーションの一覧を取得する

ステップ 3：取得した各オペレーションにおける，ユーザへの制約の一覧を取得する

ステップ 4：ユーザの状態とオペレーションの制約を照合し，選択すべきオペレーションを絞り込む

以上 4 つのステップの内，コンテキストマネージャが関与するのはステップ 1 とステップ 3 である．それぞれにおいて，前述した getUserStatus()，getOperationAffect() メソッドが利用される．

今回の例においては，ユーザは「手」が使用不可であるため，選択されるべきオペレーションは，「手」に対する制約を持たないものになる．従って，上記の 3 つの内ではスピーカーによる「音声通知」が最適なオペレーションとなる．ただし，連絡の内容が緊急であり，確実に連絡を取りたいという送信者の要求があった場合には，同期的に利用できる「電話呼び出し」が選択される可能性もある．

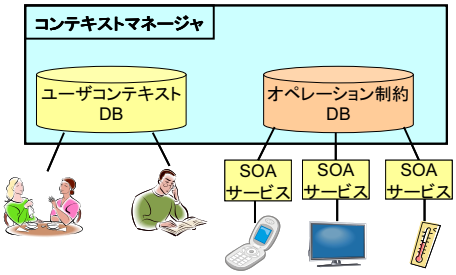

図 2　コンテキストマネージャ

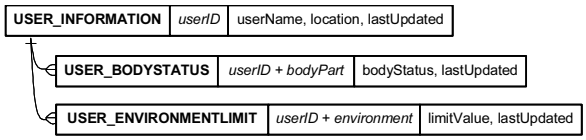

図 3　ユーザコンテキスト DB データモデル

以上のようにして，ユーザ状態とオペレーションの制約を考慮することで，ユーザの現在の状態に合わせたオペレーションの選択が可能となり，サービスは最適な振る舞いを決定することが出来るようになる．

4.　プロトタイプ

前述したモデルを基に実際にコンテキストマネージャを実装した．コンテキストマネージャのアーキテクチャを図 2 に示す．コンテキストマネージャは 2 つのデータベースを持つ．それぞれ前章で述べたユーザ状態モデルとオペレーション制約モデルに対応しており，適応型資源ファインダや他の適応型ユビキタスサービスは，コンテキストマネージャを通してこれらのデータベースにあるデータを利用する事が出来る．

4.1　ユーザコンテキスト DB

ユーザコンテキスト DB には，ユーザ状態モデルに関するデータが格納されている．この DB におけるデータモデルは図 3 に示す ER 図の通りである．

ER 図では，1 つの箱はテーブルなどのエンティティを表している．エンティティには主キーと属性が含まれている．図 3 では，「ユーザ情報」「ユーザの身体状態」「ユーザの環境制約」の 3 つのエンティティがある．「ユーザ情報」と線で結ばれている「ユーザの身体状態」「ユーザの環境制約」は，それぞれ親子関係であることを示す．

「ユーザ情報」には，ユーザ名と位置情報，そして最終更新時のデータが格納されている．「ユーザの身体状態」には，身体部位とその部位の状態，そして最終更新時のデータが記録される．「ユーザの環境制約」には，環境要素，制限値，そして最終更新時のデータが記録される．

4.2　オペレーション制約 DB

オペレーション制約 DB には，オペレーション制約モデルに関するデータが格納されている．この DB におけるデータモデ

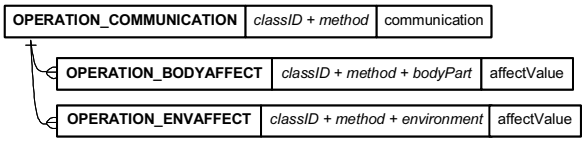

図 4　オペレーション制約 DB データモデル

表 3　サービス資源一覧

番号	サービス資源	サービスオペレーション	利用可能場所	耳	目	手	口	同期性
1	電話	電話呼び出し	リビング	×	○	×	×	同期
2	携帯電話	携帯電話呼び出し	どこでも	×	○	×	×	同期
3	携帯電話	メール送信	どこでも	○	×	×	○	非同期
4	スピーカー	音声通知	和室	×	○	○	○	非同期
5	テレビ	テロップ	リビング	○	×	○	○	非同期

ルは図 4 に示す ER 図の通りである．

オペレーション制約 DB には,「オペレーションの同期性」「オペレーションの身体部位への影響」「オペレーションの環境要素への影響」の三つのエンティティが存在する.「オペレーションの同期性」では，オペレーションを意味するクラス ID とメソッドを主キーとし，同期性に関するデータを格納している.「オペレーションの身体部位への影響」には，対象の身体部位と影響度合いに関するデータが格納されている.「オペレーションの環境要素への影響」には，対象の環境要素と影響度合いに関するデータが格納されている．

4.3　実 装 技 術

これらの DB を構築し，コンテキストマネージャを実装した．実装にあたって利用した技術は以下のとおりである．

- **データベース：**MySQL + MyBatis
- **サービス化：**Apache Tomcat 5.5 + Axis2

5.　ケーススタディ

我々はコンテキストマネージャの有用性を確認するために，コンテキストマネージャを利用した適応型ユビキタスサービスを実際に実装する評価実験を行った．実験では，実際にユーザの状態を切り替えて，サービスがユーザの状態に合わせたサービスの振る舞いを選択するかを確認した．

5.1　適応型メッセージ通知サービス

実装するサービスは，ユーザの状態に合わせてメッセージの通知手段を切り替える「適応型メッセージ通知サービス」である．サービスは，通知先となる対象のユーザの現在位置や身体部位の状態を考慮し，その時点で最適な通知手段を選択する．

この実験は，我々の構築するホームネットワークシステム「CS27-HNS」[6] 上で行った．CS27-HNS にはリビングと和室があり，それぞれの部屋内にある家電機器の一部はネットワークに繋がれており，SOA 化 [7] されたサービス資源として利用できる．適応型メッセージ通知サービスで利用するサービス資源は，表 3 に示すとおりである．ただし，表 3 中の 1，2，5 のオペレーションは，本実験のための架空のサービス資源として設定したもので，実際には実装されていない．

5.2　サービスシナリオ

本実験では，リビング又は和室にユーザがいると仮定し，そのユーザの様々な状況（身体部位の状態）において，サービス

表 4　オペレーションの推薦結果

番号	現在位置	耳	目	手	口	推薦オペレーション
1	和室	○	○	○	○	携帯電話呼び出し，メール送信，音声通知
2	和室	○	○	○	×	メール送信，音声通知
3	和室	○	○	×	○	音声通知
4	和室	○	×	○	○	携帯電話呼び出し，音声通知
5	和室	×	○	○	○	メール送信
6	和室	○	○	×	×	音声通知
7	和室	○	×	○	×	音声通知
8	和室	×	○	○	×	メール送信
9	和室	○	×	×	○	音声通知
10	和室	×	○	×	○	(メール送信)(音声通知)
11	和室	×	×	○	○	(メール送信)(音声通知)
12	和室	○	×	×	×	音声通知
13	和室	×	○	×	×	(メール送信)(音声通知)
14	和室	×	×	○	×	(メール送信)(音声通知)
15	和室	×	×	×	○	(メール送信)(音声通知)
16	和室	×	×	×	×	(メール送信)(音声通知)
17	リビング	○	○	○	○	電話呼び出し，携帯電話呼び出し，メール送信，テロップ
18	リビング	○	○	○	×	メール送信，テロップ
19	リビング	○	○	×	○	テロップ
20	リビング	○	×	○	○	電話呼び出し，携帯電話呼び出し
21	リビング	×	○	○	○	メール送信，テロップ
22	リビング	○	○	×	×	テロップ
23	リビング	○	×	○	×	(メール送信)(テロップ)
24	リビング	×	○	○	×	メール送信，テロップ
25	リビング	○	×	×	○	(メール送信)(テロップ)
26	リビング	×	○	×	○	テロップ
27	リビング	×	×	○	○	(メール送信)(テロップ)
28	リビング	○	×	×	×	(メール送信)(テロップ)
29	リビング	×	○	×	×	テロップ
30	リビング	×	×	○	×	(メール送信)(テロップ)
31	リビング	×	×	×	○	(メール送信)(テロップ)
32	リビング	×	×	×	×	(メール送信)(テロップ)

がどのオペレーションを選択するかを確認した．実装した適応型メッセージ通知サービスの処理の流れは，以下に示すとおりである．なお，サービス資源レジストリについては先行研究を参考されたい．

ステップ 1：コンテキストマネージャからユーザの状態 US（現在位置，身体部位の状態）を取得する

ステップ 2：サービス資源レジストリからユーザの現在位置で利用できるオペレーションの一覧 SOL_1 を取得する

ステップ 3：SOL_1 に対して，コンテキストマネージャからそれぞれの持つ制約 OA（同期性，身体部位への影響）を取得する

ステップ 4：US と OA を照合し，選択すべきオペレーションの一覧 SOL_2 を作成する

ステップ 5：SOL_2 が空の時，US を無視して「非同期」に利用できるオペレーションの一覧を新たな SOL_2 とする

また，全ての組み合わせに対する結果は，表 4 中に示した．この実験中では，ユーザの身体部位の状態はセンサやユーザ自身の入力などによって，取得可能であると仮定している．

表 4 の「番号」はユーザの番号を表す.「現在位置」は，そのユーザの現在位置を表し，和室又はリビングにいることを示している.「耳」「目」「手」「口」はそれぞれ，そのユーザの身体部位の状態を表す.「推薦オペレーション」は，対象のユーザに対して，サービスが最終的に選択したオペレーションの一覧を表している．丸括弧が付いているオペレーションは，その時点ではユーザの身体部位の状態から利用できないが，非同期に利用できるために推薦候補となっていることを意味する．

5.3　推 薦 結 果

ユーザの身体部位の状態を考慮してオペレーションを選択することは，表 3 と表 4 において，両方共が同じ身体部位に制約

又は制限がかからないように選択することである．例えば，4番のユーザは目が使用不可能なので，オペレーションは目に制約のないものを選択するべきとなる．従って，目に制約のない（目の項目に×が入っていない）携帯電話呼び出しと音声通知が選択される．表 3 と表 4 を比べると，全ての組み合わせに対して，同じ身体部位に制約又は制限がかからないように選択されていることが分かる．このことから，この適応型メッセージ通知サービスは機能的な要求は満たしていると言える．

次に，実際に選択されたオペレーションが，その時のユーザにとって好ましいものかどうかを考える．例えば，1 番のユーザは現在和室にいて，身体に対して特に制限がないことが分かる．よって，和室で利用できるオペレーションの一覧（携帯電話呼び出し，メール送信，音声通知）がそのまま選択・推薦される．一方，17 番のユーザも同様にして，リビングで利用できるオペレーションが推薦されている．

3 番のユーザは和室にいて「手」が使用不可能なので，状況の一例として洗濯物を畳んでいる状態を想定できる．このユーザに対しては音声通知が推薦されているが，音声通知は洗濯物をたたむ手を止める必要がない一方で，その他の携帯電話を使ったオペレーション等では手を止める必要があるため，一々家事を中断したくないユーザにとっては良い推薦と言える．

25 番のユーザはリビングにいて「目」と「手」が使用不可能なので，状況の一例として読書中であると想定できる．このユーザには最適なオペレーションが存在しないため，次善策としてメール送信とテロップが推薦されている．しかし，実際にはテロップはテレビの電源がついていないと使用不能なので，この 2 つのオペレーションのうちメール送信が優先的に推薦されることが望ましい．

6. 考　　察

6.1 特　　長

表 4 の結果から，本稿で我々が提案するモデルを利用することで，サービスが容易にユーザの状態を考慮してオペレーションを動的に切り替えるられるようになったことが分かる．また，身体部位という普遍的なものをパラメータとすることで，あらゆるユーザに対して共通の運用が可能である．これは，どのユーザによって利用されるかが想定できないユビキタスサービスにおいては，非常に有用な特長と言える．

6.2 課　　題

一方で，各身体部位の状態をどのように取得するかという点は，現時点での課題でもある．例えば，ユーザの現在位置と周辺デバイスの使用状況からユーザの状況を推定できるようにテンプレートを用意するといった方法を検討中である．

さらに，本稿ではユーザコンテキストの表現方法として，ユーザ自身の状態のみを扱ったが，ユーザの周辺状況のモデル化も課題の一つである．周辺状況を利用することで，音の騒がしい環境にあれば携帯電話よりもメールを優先するといった振る舞いが可能となる．

また，本稿で提案したモデルだけでは，同じ身体部位の状態を持ったユーザならば誰に対しても共通の推薦結果を返してしまうので，よりユーザ個々人に適した振る舞いをするために，ユーザ毎に異なるデバイス利用の傾向を反映させるための情報も必要である．これに対しては，ユーザのデバイス利用の履歴などを参考にしてマイニングを行うといったアプローチを考えている．

7. ま と め

本稿では，我々はユーザコンテキストを適応型ユビキタスサービス開発に利用するためのモデルを提案した．また，ユーザコンテキストを管理するコンテキストマネージャを実装し，ケーススタディとして適応型メッセージ通知サービスを例に有効性を評価した．結果として，提案モデルを利用することでユーザの状態を考慮して容易にオペレーションを動的に切り替える事が可能となった．

本稿で述べたコンテキストマネージャは，ユビキタスクラウドの一コンポーネントである．我々は，コンテキストマネージャを拡張することで，よりユーザコンテキストに即した適応型ユビキタスサービスの実現を目指すと共に，ユビキタスクラウドの他コンポーネントの拡充をすることで，容易に適応型ユビキタスサービスを開発できるようにする予定である．

謝辞　この研究の一部は，科学技術研究費（基盤研究 B 23300009，若手研究 B 21700077，研究活動スタート支援 22800042），および，ひょうご科学技術協会の助成を受けて行われている．

文　　献

[1] A. Ranaganathan and R. Campbell, “Autonomic Pervasive Computing based on Planning,” Proceedings of the International Conference on Autonomic Computing, IEEE, 2004.

[2] K. Carey, D. Lewis, S. Higel and V. Wade, “Adaptive Composite Service Plans for Ubiquitous Computing,” 2nd International Workshop on Managing Ubiquitous Communications and Services (MUCS 2004), December 2004.

[3] R. Harbird, S. Hailes and C. Mascolo, “Adaptive Resource Discovery for Ubiquitous Computing,” 2nd Workshop on Middleware for Pervasive and AdHoc Computing, 2004.

[4] K. Egami, S. Matsumoto and M. Nakamura,“Ubiquitous Cloud: Managing Service Resources for Adaptive Ubiquitous Computing,” 1st IEEE PerCom Workshop on Pervasive Communities and Service Clouds (PerCoSC 2011).

[5] I. Foster, Z. Yong, I. Raicu and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared,” Grid Computing Environments Workshop (GCE 2008), pp. 60–69, 2008.

[6] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada and K. Matsumoto, “Constructing home network systems and integrated services using legacy home appliances and web services,” International Journal of Web Services Research, vol. 5, no. 1, pp. 82–98, 2008.

[7] M. P. Papazoglow and D. Georgakopoulos, “Service Oriented Computing,” Communications of the ACM, vol. 46, no. 10, pp.25–28, 2003.