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.

適応型ユビキタスサービスの開発を支援するユビキタスクラウドの提案

江上　公一†　　柗本　真佑†　　中村　匡秀†　　井垣　　宏††

† 神戸大学　〒 657–8501 神戸市灘区六甲台町 1–1
†† 東京工科大学　〒 192–0982 東京都八王子市片倉町 1404-1
E-mail: †{egami,shinsuke,masa-n,igaki}@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) を提案する．ユーザの要求に応じて適切なセンサ・インタフェース・Web アプリケーション等を動的に調達することで，多様なユーザのための適応型ユビキタスサービスを実現する．また，我々の構築したホームネットワーク環境におけるケーススタディに対して実験を行い，その有効性を評価する．
キーワード　ユビキタスサービス，サービス動的発見，PaaS，クラウドコンピューティング，レジストリ

Ubiquitous Cloud: Development Platform of Adaptive Ubiquitous Services

Kouichi EGAMI†, Shinsuke MATSUMOTO†, Masahide NAKAMURA†, and Hiroshi IGAKI††

† Kobe University　Rokkoudai-chou 1–1, Nada-ku, Kobe, Hyogo, 657–8501 Japan
†† Tokyo University of Technology
E-mail: †{egami,shinsuke,masa-n,igaki}@cs.kobe-u.ac.jp

Abstract　The *adaptive ubiquitous services*, which are dynamically tailored for individual users based on preferences and contexts, are one of the major challenges in the ubiquitous computing. To achieve the challenge, this paper presents a novel platform called *ubiquitous cloud*, borrowing the concept of the cloud computing. For a given request, the ubiquitous cloud understands user's context and preference. Then, the cloud dynamically finds appropriate *service resources* of sensors, interfaces and applications, each of which is exhibited as a Web service. The resources are finally integrated to achieve the tailor-made ubiquitous service matching the request. To demonstrate the feasibility, we conduct a case study with a prototype system implemented on an actual home network system.
Key words　ubiquitous service, dynamic service finding, PaaS, cloud computing, service registry

1. はじめに

現在，ネットワーク環境や IT 技術の進歩に伴い,「いつでも・どこでも・誰にでも」を目的としたユビキタスサービスが実現されつつある．例えば，子供の通学・通塾の様子を保護者が携帯電話から見守る「あんしんグーパス」[1] や，宅内の家電製品をネットワークに接続し，より便利な環境を提供する「ホームネットワークシステム」などが研究・開発されている．これらのユビキタスサービスは，大多数のユーザが持つ興味やニーズ，利用環境を想定して開発されている．

既存のユビキタスサービスの問題点として，サービスを一旦開始すると，その後にサービスの利用環境の変更やユーザ要求そのものの変更に対応することが困難になるという点が挙げられる．これは，既存のユビキタスサービスでは，サービスが利用する「モノ」の組み合わせや，取得する実世界情報，サービスの提供インタフェースやサービスロジック（以後，全て含めてサービス資源と呼ぶ）を開発時に決定する事に由来する．この問題点を示す例として，前述の「あんしんグーパス」サービスでは携帯電話に駅の通過情報が通知されるが，ある保護者が自宅にいるときはテレビやスピーカーで通知して欲しいという要求を持ったとする．この場合，通過情報を携帯電話用だけでなくテレビやスピーカー用に加工するためのロジックや，保護

者の位置を検知して通知手段を切り替えるといったロジックが必要となる．しかし，既に実装を終えた「あんしんグーパス」サービスに修正を加えるのは，大きなコストが発生してしまう．

そこで本研究では，クラウドのコンセプトに基づいてサービス資源を個人に合わせて動的に調達する，適応型ユビキタスサービスの構築基盤（ユビキタスクラウド）の実現に取り組む．ユビキタスクラウドにより,「いまだけ・ここだけ・あなただけ」を目的とする個人の状況や利用環境に適応したユビキタスサービス（適応型ユビキタスサービスと呼ぶ）を構築できるようになり，人と情報環境の円滑な関係を築くことが可能となる．

クラウドとは，ネットワーク上に存在する様々な IT 資源を，その実体を意識せずにサービスとしてユーザがオンデマンドに調達できる新しいコンピューティング形態である [2]．ユビキタスクラウドでは，このクラウドコンピューティング形態を応用して，ユーザからの調達要求に応じて適切なサービス資源の調達を動的に行い，調達結果を返却する．ユーザは，その調達結果を基にサービス資源を組み合わせることで，個々の要求に即したサービスを享受することが出来る．

本研究で提案するユビキタスクラウドは，以下に示す要素から構成される．

- サービス資源プール：ネットワーク上から利用できる様々なサービス資源を管理しているプール
- インタフェースクラウド：ユーザのサービス要求に対応して，サービス提供のための最適なインタフェースデバイスの選定を行うクラウド
- センサクラウド：ユーザのサービス要求に対応して，適切なセンサデバイスを用いて必要な実世界情報を収集し，標準的な形式に加工・提供するクラウド
- クラウドマネージャ：ユビキタスクラウドを利用するためのマネージャ

これらを連携することで，ユーザ要求やコンテキストに応じた適切なサービス資源の調達を実現し，ユーザ要求やコンテキストに基づくオンデマンドなサービス構築や情報提示を行う．さらに，このユビキタスクラウドを利用したアプリケーションを実際に開発し，その有用性を確認した．

2. 準　　備

2.1 ユビキタスサービス

近年，ユビキタスコンピューティング [3] に関する研究が盛んに行われている．ユビキタスコンピューティング環境では，組み込み型コンピュータが実世界に偏在しており，端末機能やネットワークへのアクセス機能の向上により，どのような場所からでも組み込み型以外の通常のコンピュータ上の機能を利用できるという特徴がある．この環境においてコンピュータにより実現される機能を，本稿ではユビキタスサービスと呼ぶ．

ユビキタスサービスは，インターネット上にある各サービスや情報資源などの他に，センサやアクチュエータなどの実世界に偏在する「モノ」を利用して，様々な機能をユーザに提供する．代表的なものには，ホームネットワークシステム（HNS）における家電連携サービスがある [4]．HNS は，家庭内にある家電や設備，各種センサを家庭内ネットワークに接続したシステムである．これらの機器をネットワーク越しに連携し，付加価値機能を実現したものが HNS 家電連携サービスである．例えば，TV，DVD，カーテン，照明，サラウンドスピーカーを連携して，ワンタッチで映画館の雰囲気で映画を視聴できる「DVD シアターサービス」や，ユーザの帰宅をドアセンサで検知して，部屋照明をつける「お帰りサービス」などがある．

こうしたユビキタスサービスは，通常，サービスが利用するサービス資源は開発時に決定する．そのため，サービス開始後に，ユーザの利用環境の変更やユーザ要求そのものの変更に対応することは困難である．従って，ユーザ個々人，又はその時々によって変化するユーザの利用環境や要求に対して，柔軟に対応することが，これからのユビキタスサービスには求められる．

2.2 適応型ユビキタスサービス

適応型ユビキタスサービスは，個々のユーザに合わせてサービス資源を動的に調達し，その機能を提供するサービスである．このサービスでは，サービスの実行時に利用対象となるユーザの状況や，利用するインタフェースが現在使用できるかといった情報から，その時に最適なサービス資源を選択する．

例として，ユーザが部屋の間を移動する際に移動元の部屋の状態を移動先の部屋に引き継ぐ「状態引継ぎサービス」を考える．このサービスでは，部屋 A にいたユーザが部屋 B に移動するとき，部屋 A の各家電の状態（目標状態とする）を取得し，それに沿うように部屋 B の各家電を操作する必要がある．

これを従来の方法で実装する場合，既知の部屋及び家電全てに対して，ソースコード又は外部ファイルに機器情報を記述しておく必要がある．そのため，実装後に部屋や家電が新たに追加された場合，その都度ソースコード又は外部ファイルの修正が必要となる．

一方，適応型ユビキタスサービスではサービス資源を動的に調達可能であるため，機器情報の用意や修正が不要である．上記の状態引継ぎサービスを実現するためには，まず部屋 A に存在する家電の一覧とその目標状態を動的に取得し，次いで取得した部屋 B に存在する各家電に対して，目標状態に沿うように家電操作を実行する．このように実装することで，既知の部屋や家電の数にコストが左右されず，また，実装後に部屋や家電が追加されても影響を受けない．

2.3 クラウドコンピューティング

クラウドとは，ネットワーク上に存在する様々な IT 資源を，その実体を意識せずにサービスとしてユーザがオンデマンドに調達できる新しいコンピューティング形態である [2]．例えば，ソフトウェア環境やネットワーク環境などのインフラ設備そのものを仮想サーバといった形でユーザに提供する IaaS（Infrastructure as a Service）や，アプリケーションソフトが稼動するためのハードウェアや OS などの基盤一式をインターネット上のサービスとして遠隔から利用できるようにした PaaS（Platform as a Service）などがある．

2.4 先行研究「ユビレジ」

ユビキタスサービスが利用するサービス資源を検索する手法として，Web サービスを検索する UDDI（Universal Descrip-

tion, Discovery and Integration) [5] が知られている．UDDI は Web サービスに関する情報（名称や機能，サービスへのアクセスポイントなど）をレジストリに登録することで目的のサービスを検索可能とするが，ユビキタスコンピューティング環境においては，サービス資源が物理空間に密接に結びついていることから，IT 空間に閉じた従来の Web サービス検索では限界がある．例えば家の 1 階のリビングにあるスピーカを利用したい場合，宅内に複数存在するスピーカサービスの中から目的の場所に設置されたスピーカサービスを正しく検索する必要がある．

この問題を解決するべく，我々は先行研究において，ユビキタスサービスが利用するサービス資源を効率的に検索するためのサービスレジストリ「ユビレジ」[6] を提案している．ユビレジでは，サービス資源を 3 つのサービスタイプに分類し，物理ロケーション及び目的キーワードの付加することで，サービス資源を物理的な場所や目的に応じて検索できる．

3. ユビキタスクラウド

3.1 目的とアプローチ

これからのユビキタスサービスには，ユーザ一人一人の要求や利用環境の変化に柔軟に適応し，サービスを提供することが求められる．我々が提案するユビキタスクラウドは，そのような適応型ユビキタスサービスの創出・合成を可能とする適応型ユビキタスサービス基盤である．クラウドのコンセプトをユビキタスコンピューティング環境に応用し，ユーザ要求に応じて必要なサービス資源を調達することで，適応型ユビキタスサービスの実現を目指す．

3.2 アーキテクチャ

ユビキタスクラウドは，図 1 に示す以下の 4 つの要素によって構成される．

- サービス資源プール：ネットワーク上から利用できる様々なサービス資源を管理しているプール
- インタフェースクラウド：ユーザのサービス要求に対応して，サービス提供のための最適なインタフェースデバイスの選定を行うクラウド
- センサクラウド：ユーザのサービス要求に対応して，適切なセンサデバイスを用いて必要な実世界情報を収集し，標準的な形式に加工・提供するクラウド

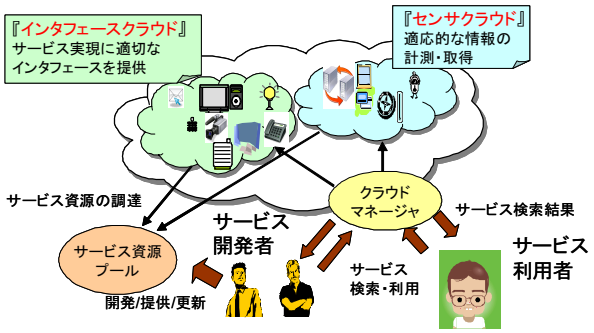

図 1　ユビキタスクラウド

- クラウドマネージャ：ユビキタスクラウドを利用するためのマネージャ

これらを連携することで，個々のユーザの要求や状況に応じた適切なサービス資源の調達を実現し，ユーザごとにオンデマンドなサービス構築や情報提示を行う．

ただし，本論文では，ユビキタスサービス開発時に様々なサービス資源をどのように組み込むかについて注目するため，センサに特化しているセンサクラウドについては，紙面の都合から説明を省略する．

3.3 サービス資源プール

サービス資源プールには，これまでに開発されネットワークに公開されている様々なサービス資源が蓄積されている．ここには，新たに開発されたサービス資源などを自由に登録できる．ユビキタスクラウドでは，サービス資源プールに蓄えられたサービス資源の中から，その時のユーザ要求に即したモノを検索・発見し，ユーザに提供するユビキタスサービスに組み込む．従って，様々なユーザ要求に対して，効率的にサービス資源を検索・発見するためのフレームワークが求められる．

本研究では，このサービス資源プールに「ユビレジ」[6] を採用している．「ユビレジ」の詳細については参考文献 [6] を参照されたい．

3.4 インタフェースクラウド

インタフェースクラウドは，ユーザの要求に即したインタフェースを検索・発見し，最適な形でユーザに提供する機構である．ここで言うインタフェースとは，ユーザに対して何らかの機能を提供するサービス資源のことを指す．

インタフェースクラウドが持つ機能は以下の通りである．

- サービス資源の検索
- サービス資源の利用
- サービス資源の実行用 URL の取得
- 過去の検索履歴の確認
- 検索履歴の削除

「サービス資源の検索」では，インタフェースクラウドがユーザから要求を受けた時，サービス資源プールに当該要求に即したサービス資源を問い合わせる．インタフェースクラウドは，その結果をそのままユーザに提供することもあれば，各サービス資源がどのような機能を持っているかという情報だけを抽出してユーザに提示することもある．例えば，場所は「リビング」で，目的は「空調機能」という要求をインタフェースクラウドが受けると，その結果は次のようなものになる．

- リクエスト ID ＝ 1
- インタフェース ID ＝ 1
- インタフェースの場所＝リビング
- インタフェースの概要＝エアコン電源オン

ここで，リクエスト ID は一つのユーザ要求に対して割り当てられる識別番号である．インタフェース ID は，複数のインタフェースが発見された時に各インタフェースに割り当てられる識別番号である．この例の場合，「リビング」にある「空調機能」を備えたものには，エアコンの一つだけが該当することを示す．この結果をアプリケーション中で解析することでその時

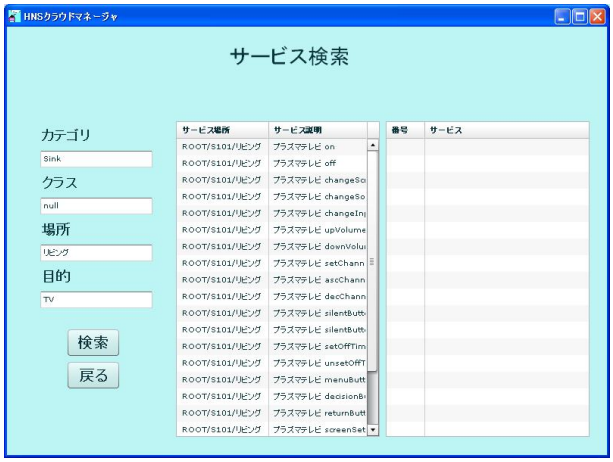

図 2　クラウドマネージャ画面 1

図 3　クラウドマネージャ画面 2

点での要求に沿ったサービスを動的に取得できるので，サービス資源の追加・変更に対してアプリケーション側で修正を施す必要がなくなる．

「サービス資源の利用」では，検索時に提供された情報を基にサービス資源の機能を実行できる．上述の例では，リビングにおいて空調機能のあるサービス資源はエアコンであり，リクエスト ID とインタフェース ID は共に 1 である．従って，リクエスト ID に 1 を指定し，インタフェース ID にも 1 を指定してサービス資源の実行要求を行うと，ユビキタスクラウド側で当該エアコンのオン操作を実行してくれる．これにより，ユーザはサービス資源の利用方法を知らなくても，サービス資源を利用することが出来る．

「サービス資源の実行用 URL の取得」では,「サービス資源の利用」と同様に，リクエスト ID とインタフェース ID を指定することで，該当するサービス資源を実行する際の URL が取得できる．上述の例に従うと，リクエスト ID とインタフェース ID に 1 を指定することで「http://CS27HNS/Airconditioner/on」といった URL が得られる．

「検索履歴の確認」では，過去にユビキタスクラウドに与えた要求や，それらの要求に対しての検索結果が確認できる．

「検索履歴の削除」では，ユビキタスクラウドが保持している過去のユーザ要求とその検索結果を，削除することが出来る．

3.5　クラウドマネージャ

本論文で提案するインタフェースクラウドやセンサクラウドを利用するにあたっては，ある程度の専門的な知識を必要とする．例えば，それぞれのクラウドからの返り値は XML 形式のため，XML の知識が必要となる．また，各データが何を意味するのかについての知識も必要であり，これらは直接ユビキタスクラウドを利用する際のハードルを上げている．そのため，専門的な知識を持たない一般ユーザでも，ユビキタスクラウドを利用して各々が望むようなサービスを構築するための環境があれば望ましい．

クラウドマネージャは，このような考えを基に実装されている．具体的には，ボタンのクリック作業でユビキタスクラウドを利用したサービスの作成が出来る．引数などが必要となった場合には，適宜テキスト入力も要求される．

クラウドマネージャを利用したサービス作成の例として，TV 鑑賞サービスの作成の流れを追ってみる．TV 鑑賞サービスは，TV の電源をつける，空調設備を起動する，照明をつける，という 3 つのサービスを自動的に実行するサービスである．従って，まずは TV の電源をつけるサービスを検索する．図 2 がその時の画面であり，一番上にあるサービスが「プラズマテレビ on」であるので，このサービスを選択し，追加する．同様にして，次に空調設備を検索すると,「エアコン on」が発見されたのでこれを追加する．最後に照明をつけるサービスを検索し,「天井ライト on」を追加する．図 3 が，サービスを 3 つ追加した後の画面である．ここで左下の作成ボタンを押すと，所定の場所に perl 実行ファイルが書き出されるので，このファイルを実行することで，3 つのサービスが自動的に実行される．

3.6　実　　装

以上の議論に基づいて，ユビキタスクラウドの実装を行った．具体的には，ユビキタスクラウド本体は Java による Web サービスで実装した．また，クラウドマネージャは Flash で実装した．実装に利用した技術の諸元を以下に示す．

【ユビキタスクラウド】
開発環境: Eclipse 3.4.0
API 実装: Java
Web サーバ: Apache Tomcat 5.5
Web サービスエンジン: Apache Axis 2.1.3
【クラウドマネージャ】
開発環境: Flex Builder 3
実装言語: Action Script 3.0

4.　ケーススタディ

4.1　概　　要

本節では，我々が提案するユビキタスクラウドの有用性を確認するために行った比較実験について述べる．

この比較実験では，ユーザが部屋を移動する際に，部屋の家電機器の状態を移動先の部屋に引き継ぐ「状態引継ぎサービス」を対象に行う．状態引継ぎサービスは，例えばリビングで TV

を鑑賞していたユーザが来客のために部屋を移動した際，移動先の部屋で改めて家電機器を操作しなくとも，サービスが自動的に移動元の部屋と同じ状態にしてくれるサービスである．

この「状態引継ぎサービス」を Java アプリケーションとして実装する．実装に際しては，各部屋の各家電機器を操作するためのアクセス情報などが必要となる．そこで，アクセス情報を予めソースコードや外部ファイルに記述しておく従来の方法と，我々が提案するユビキタスクラウドを利用した場合とで両者を比較する．定量的に比較するための指標として，プログラムの規模を表すステップ数を利用する．

4.2 状態引継ぎサービス

実験は，我々が構築している HNS 環境，CS27HNS [4] 上で行った．実験に用いた部屋，及び設置家電機器は表 1 の通りである．

表 1　CS27HNS の構成

リビング	和室
天井エアコン	エアコン
天井ライト	卓上ライト
プラズマテレビ	アナログテレビ
	空気清浄機

状態引継ぎサービスでは引継ぎ元と引継ぎ先の部屋名という 2 つの引数を取る．今回の場合，表 1 にあるリビングと和室の間で，引継ぎ元の部屋にある各家電機器の状態を引継ぎ先の部屋で再現する．この時，2 つの部屋に共通して存在するエアコン，ライト，テレビについては状態の再現が行われるが，和室にのみ存在する空気清浄機については無視される．

状態引継ぎサービスの処理フローについては，以下のようになる．

（1） 引継ぎ元の部屋にある家電機器の取得
（2） 取得した各家電機器の現在状態の取得
（3） 引継ぎ先の部屋にある家電機器の取得
（4） 引継ぎ元と引継ぎ先の部屋に共通する家電機器が存在する場合，引継ぎ元の状態に合わせて引継ぎ先の家電機器を操作する

これらの各処理について，各部屋の各家電機器の情報をソースコード中にハードコードして取得する場合（ケース A とする）と，ユビキタスクラウドを利用して取得する場合（ケース B とする）の 2 通りの実装を行った．各処理をそれぞれのケースでどのように実装の差異が生じたかを次に示す．

まず（1）の処理では，ケース A の場合，各部屋の各家電機器の情報をソースコード中に記述し，部屋名を if-else 文によって比較することで，対象の部屋の家電機器の情報を取得する．一方，ケース B の場合，ユビキタスクラウドに指定された部屋名を要求として与えることで，ユビキタスクラウドから利用可能な家電機器の一覧を取得する．

（2）の処理では，ケース A とケース B で実装上の差は特に生まれなかった．従って，今回の比較実験の対象外とした．

（3）の処理は，対象とする部屋が異なるだけで（1）と共通の処理を行っているため，説明は割愛する．

（4）の処理では，両ケース共に引継ぎ元の家電機器の電源がオフであれば引継ぎ先でもオフとし，逆にオンであればオンにするという操作を実行するように実装をした．この際，ケース A の場合は，操作を実行するための URL はソースコード中に記述しているが，ケース B の場合は，ユビキタスクラウドから取得した情報を元に動的に URL を生成し，操作を実行する形で実装している．

このように 2 つのケースで実装したところ，それぞれのステップ数の対比結果は表 2 となった．

表 2　ステップ数の対比（2 部屋 7 家電）

	ケース A	ケース B
（1）の処理	53	69
（3）の処理	53	69
（4）の処理	56	16

4.3 考　　察

表 2 の結果から，(4）の処理ではユビキタスクラウドを利用するケース B の方が必要なステップ数が少なくなっていることが分かる．これは，ケース A では全ての部屋の全ての家電機器について，予め使われるであろう機器操作のための実行用 URL をハードコードする必要があるのに対して，ケース B の場合は，ユビキタスクラウドから取得した情報から動的に実行用 URL を生成できることが大きく影響しているからである．また，今回は家電操作に特化して実験を行ったため，アプリケーション側で動的に実行用 URL を生成しているが，実際は家電操作以外の様々なサービス資源を扱うことも考えられる．その場合，個別に実行用 URL を生成するための処理を記述する方法もあるが，ユビキタスクラウドの機能である「サービス資源の実行用 URL の取得」を利用することで，冗長な記述を避けることが可能と考えられる．

一方，(1）と（3）の処理では，ユビキタスクラウドを利用するよりもソースコード中にハードコードするほうがステップ数が少なくなっている．しかし，これはユビキタスクラウドからの結果を解析するための諸設定に要するステップ数がかさんだ結果である．それを示すため，今回の実験で扱った部屋，及び家電の数を 2 倍にした場合のステップ数をシミュレートした．このシミュレートでは，全く同じ家電構成のリビングと和室が 2 部屋ずつ存在すると仮定し，それらの家電の情報を，ケース A ではソースコード上に記述し，ケース B ではサービス資源プールに追加した．この時のステップ数の比較結果は表 3 の通りである．

表 3　ステップ数の対比（4 部屋 14 家電）

	ケース A	ケース B
（1）の処理	105	69
（3）の処理	105	69
（4）の処理	102	16

表 3 から分かるように，いずれの場合もケース B の方がステップ数で下回っている．さらに，ケース A はいずれもステッ

プ数が増加したのに対し，ケース B ではステップ数の変化は無かった．これは，部屋や家電機器の数に変化が生じたときは，ユビキタスクラウドのサービス資源プールにそれらの情報が追加されるためである．この結果から，3.4 で述べたように，処理に必要な家電機器の情報などをユビキタスクラウドから取得することで，アプリケーション側で新たに修正・追加をする必要が無いことが分かる．従って，部屋や家電機器の数が増えた場合でも，ユビキタスクラウドを利用することでアプリケーションの開発コストを抑えることが可能であると同時に，アプリケーション実装後に追加されるであろう未知の部屋や家電に対して，柔軟に対応できると考えられる．

また，前述したようにユビキタスクラウドを利用する場合，ユビキタスクラウドからの返り値である XML を解析するための準備が必要となるが，これらは定型であることが多く，雛形を用意しやすい利点が挙げられる．

5. 関 連 研 究

ユビキタスコンピューティング環境における，個人の要求に即したユビキタスサービスの提供に関連する他の研究として，2005 年に提案された USON アーキテクチャ [7] や，2008 年から実施されている CUBIQ プロジェクト [8] などがある．

USON アーキテクチャでは，ユビキタスサービスをサービステンプレート（ST と呼ぶ）として定義し，ユビキタスサービスの実行時に ST に従って各サービス部品（SE と呼ぶ）を発見・実行している．各 ST，SE は語彙情報という，USON アーキテクチャにおいて共有される情報を持ち，この情報を基に SE の発見や機能の実行などが行われる．しかし，この語彙情報には，その機器が実世界のどこに存在するかと言う場所情報や，どのような目的で利用されるかと言った目的の情報が明示的に含まれていない．そのため，例えば，「テレビ機能を提供する」モノとして，1 階にあるテレビと 2 階にあるテレビが発見された場合，1 階と 2 階の違いを無視して両者を同等と評価する可能性がある．これは，もしもユーザが 1 階にいた場合，ユーザは 1 階のテレビを使いたいはずであり，2 階のテレビを使う必要は無いにもかかわらず 2 階のテレビを推薦してしまうことに繋がる．一方，ユビキタスクラウドの場合，クラウドへのサービス資源要求に場所や目的が含まれるため，場所別，目的別にサービス資源を調達することが出来る．

CUBIQ プロジェクトは広範なスコープを持っているが，その中でも本研究と関連する部分として，状況情報サービス連携技術が挙げられる．そこでは，写真画像を利用して直感的に情報閲覧やアプリケーションの操作などを可能にする u-Photo [9] に関する研究もある．u-Photo は物理空間の視覚情報の上に仮想空間の情報を重ねるために使われるデジタル写真メディアで，ユーザは専用の Viewer を利用することで，画像上の情報家電からその時点での状態を取得したり，アプリケーションの起動と言った操作を行うことが出来る．u-Photo とユビキタスクラウドの相違点は，u-Photo は実世界上の情報家電及びセンサが対象となっているのに対し，ユビキタスクラウドでは，実世界上の情報家電やセンサに加え，仮想空間上の様々なサービスも取り扱える点にある．

6. ま と め

本論文では，クラウドのコンセプトに基づき，適応型ユビキタスサービスの開発を支援するためのユビキタスクラウドの提案を行った．ユビキタスクラウドを構成する 4 つの要素を定義し，その中でも，インタフェースクラウド・クラウドマネージャについて述べた．さらに，我々の構築した HNS 環境において実験を行い，その有効性を確認した．具体的には，部屋や家電の数が多くなった場合でも，アプリケーション開発者側で各家電機器に関する情報を用意する必要が無く，開発コストを抑えることが出来ることや，アプリケーション側で家電機器の増加などの影響を受けないことが確認された．

今後の課題としては，まずはクラウドマネージャでサービスを作成する際に，条件分岐などの複雑な処理も実現できるようにすることを考えている．また，インタフェースクラウドへの要求についても，さらに使い勝手を良くする方法を検討していきたい．例えば，場所や目的について個別に指定する現在の仕様に加え，「リビングにある空調機能を備えたもの」のように自然言語に近い形で受け付けられるようにしたり，一覧が定まっている要求項目についてはクラウドマネージャがリスト表示するなど，システム側や UI 側からの妥当なアプローチは無いかを検討していきたい．

謝辞 この研究の一部は，科学技術研究費 (若手研究 B21700077，20700027) の助成を受けて行われている．

文 献

[1] アイテック阪急阪神社「あんしんグーパス」 http://anshin-gp.jp/index.html

[2] 湯川 抗, 前川 徹, “大企業のクラウドコンピューティングへの取り組みに向けた考察,” 富士通総研（FRI）経済研究所研究レポート, No.337, 2009.

[3] M. Weiser, “The Computer for the 21st Century,” Scientic American, Vol.265, No.3, pp.94–104, September 1991.

[4] 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, January 2008.

[5] UDDI Version 2.0, http://msdn.microsoft.com/ja-jp/library/cc465684.aspx

[6] 瀬戸 英晴, 江上 公一, 松尾 周平, 井垣 宏, 中村 匡秀, “ユビキタスネットワークにおけるサービス資源検索のためのサービスレジストリの考察,” 電子情報通信学会技術研究報告, Vol.109, No.456, pp.19–24, March 2010.

[7] 武本 充治, 大石 哲矢, 岩田 哲弥, 山登 庸次, 田中 洋平, 徳元 誠一, 島本 憲夫, 黒川 章, 須永 宏, 小柳 恵一, “ユビキタスコンピューティング環境に適したサービス提供アーキテクチャにおけるサービス合成方式とその実装,” 情報処理学会論文誌, Vol.46, No.2, pp.418–433, February 2005.

[8] CUBIQ プロジェクト http://www.soumu.go.jp/menu_seisaku/ictseisaku/ictR-D/jigyou_ichiran_h20_1.html

[9] S. Aoki , M. Ito, J. Yura, J. Nakazawa, K. Takashio, and H. Tokuda, “u-Photo Mobile: Interacting with Smart Environments via Clickable Photos on Mobile Phones,” International Conference on Intelligent Environment, Vol.2, pp.327–334, July 2009.