

異種分散 Web サービスに基づくコンテキストウェアサービスの 管理フレームワークの提案

高塚 広貴[†] 佐伯 幸郎[†] 梶本 真佑[†] 中村 匡秀[†]

[†] 神戸大学 〒 657-8501 神戸市灘区六甲台町 1-1

E-mail: †tktk@ws.cs.kobe-u.ac.jp, ††sachio@carp.kobe-u.ac.jp, †††{shinsuke,masa-n}@cs.kobe-u.ac.jp

あらまし M2M 技術やクラウドサービスの普及に伴い、多種多様なデータが Web サービスを介して利用可能となってきている。これらのデータから現実世界の状況（コンテキスト）を判断し、自律的な制御を行うコンテキストウェアサービスの実現が期待されている。しかしながら従来研究においては、Web を介してコンテキストやアクションを連携したサービスを、体系的に作成・管理する有効な方法は存在しない。本研究では、様々な種類の分散した Web サービス（センサ、情報システム、家電の状態など）を利用したコンテキストウェアサービスを統一的に作成・管理できるフレームワークを提案する。提案手法では、まず既存 Web サービスのデータに基づいてコンテキストを定義する。次に、コンテキストウェアサービスを (Event, Condition, Action) の組で定義する。Event はサービスの実行契機となるコンテキストの成立、Condition はサービスの実行可能条件、Action は制御に相当する Web サービスである。提案フレームワークはクラウドシステムに配備され、作成したコンテキストやサービスは Web サービスとして公開される。したがって他のサービスで再利用したり他のシステムやアプリケーションとも連携可能となる。ケーススタディとして、実際のホームネットワークシステムにおけるコンテキストウェアサービスへの適用の考察を行う。

キーワード Web サービス, コンテキスト, ECA 規則, ホームネットワークシステム, センササービス

A Framework for Managing Context-Aware Services Based on Heterogeneous and Distributed Web Services

Hiroki TAKATSUKA[†], Sachio SAIKI[†], Shinsuke MATSUMOTO[†], and Masahide NAKAMURA[†]

[†] Kobe University Rokko-dai-cho 1-1, Nada-ku, Kobe, Hyogo, 657-8501 Japan

E-mail: †tktk@ws.cs.kobe-u.ac.jp, ††sachio@carp.kobe-u.ac.jp, †††{shinsuke,masa-n}@cs.kobe-u.ac.jp

Abstract With the spread of Machine-to-Machine (M2M) and cloud services, we have become to be able to use heterogeneous and distributed data. Implementation of the service which judges the context and controls autonomous from these data is expected. However, in existing research, method of systematically creating and managing for service which works together the context and the action across the web does not exist. In this research, we propose the framework which can create and manage in a unified manner context-aware services using heterogeneous and distributed Web services (sensor services, information system, state of appliance and so on) In the proposed method, first, we define the context based on the data of the existing Web services. Then we define the context-aware service as a set of event, condition and action. The event is the establishment of the context which is execution opportunity of the service. The condition is the executable terms of service. The action is Web services which correspond to the control. The proposed framework is deployed in the cloud system, and created contexts and services are published as Web services. Therefore we can reuse these in other services and cooperate with other systems and applications. As case studies, this framework is applied to context-aware services in the real home network system.

Key words Web services, context, event-condition-action rule, home network system, sensor services

1. はじめに

近年、計算資源をネットワーク経由でユーザがサービスとして利用するクラウド技術の発展、並びに人間の介在なしに機器同士が相互に通信を行い動作が行われる M2M 技術の発展により、気温などの情報をはじめとした現実世界に関する多種多様なデータを、Web サービス (Web-API) 経由で取得できるようになっている。これらのデータから現実世界の状況 (コンテキスト) を判断し、自律的な制御を行うコンテキストウェアサービスの実現により、環境の変化や人間の行動に対して、“気の利いた” サービスの作成が可能となる。

高度なコンテキストウェアサービスの実現に向けた、センサ情報からのコンテキスト推定及びその利用方法については、ユビキタス分野においてコンテキストウェア技術としてこれまでに報告されている [1]。これに対し、センサに限らず異種分散 Web サービス経由で取得できるデータ (センサ、情報システム、家電の状態など) をコンテキスト推定に利用すれば、更に高度なコンテキストウェアの実現が可能となるが、入力される各種データの提供元やフォーマットなどの管理が複雑となるため、実際の報告例は少ない。また、コンテキストウェアサービスとして Web サービスの呼び出しを行うことで、より複雑なサービスの提供が可能となる。これについては、物理センサから取得したコンテキストをもとに Web サービスの呼び出しを行うセンサ駆動サービス [2] が過去提案されている。

このようにコンテキストウェアサービスの入出力双方を Web サービス化することにより、コンテキストウェアサービスをより柔軟に・使いやすく提供することが可能となる。しかしながら、コンテキスト推定に用いるデータ、コンテキストの推定、コンテキストに起因した動作それぞれの関係が複雑になるため、体系的な管理を行わなければサービスの提供が困難となるが、このような研究はこれまでに報告されていない。これらのコンテキストを用いたサービスを作成・再利用できる枠組みが存在すれば、従来のコンテキストウェアサービスよりも多くの現実世界のデータに基づいた、より複雑で精度の高いコンテキストウェアサービスの作成が可能となる。

そこで本研究では、異種分散 Web サービスを利用したコンテキストウェアサービスを統一的に作成・管理できるフレームワークを提案する。提案フレームワークは 5 つのレイヤ (ECARuleLayer, ContextLayer, ActionLayer, AdapterLayer, WebServiceLayer) から成り、各レイヤの要素を使用して、ECARuleLayer において事象 (Event)、条件 (Condition)、動作 (Action) の組である **ECA 規則** を定義することでコンテキストウェアサービスを作成する。提案フレームワークでは、データを取得する Web サービスを AdapterLayer に登録し、それをもとにコンテキスト推定を行う。同様に、動作にあたる Web サービスを ActionLayer に登録するため、センサなどに限定された範囲だけでなく、異種分散 Web サービスを組み合わせたコンテキストウェアサービスを作成・管理できるようになる。各レイヤの要素は、他のサービスからも利用することができるため、将来的な拡張にも対応できる。

異種分散 Web サービスを利用したコンテキストウェアサービスが作成可能であることの確認のため、ケーススタディとして、実際のホームネットワークシステムにおける提案フレームワークの適用によるコンテキストウェアサービス作成に関する考察を行う。

2. 準備

2.1 Web・クラウド時代におけるコンテキスト推定のための技術

情報通信技術の発達により、最近では Web から様々な情報が取得できる。特に M2M (Machine to Machine) や Web-API といった技術は重要な役割を担っている。

M2M とは、センサや通信モジュールを内蔵した様々な機器が、人の手を介さずに、現実世界の状況を発信する技術である。M2M 発達の背景には、通信の高速化や、センサ技術の進化がある。加えて、クラウド技術・ビッグデータ技術の発達により、センシングし取得したデータを機器側で処理することなく全てサーバに伝達可能となったことにより、時差なく状態を取得し、リアルタイムにデータを処理することが可能になった。

Web-API とは、ネットワーク経由で異なるアプリケーションやサービスの機能を共有する技術である。Web-API では、API を公開する側と利用する側で通信を行う技術として、SOAP や REST と呼ばれる HTTP などのプロトコルを使用し、XML 形式のデータを送受信する仕組みを利用する。Web-API を利用することで、個人では作ることが困難な機能や、データベースを利用した開発が可能となる。

M2M の活用によりリアルタイムに処理されたデータを使用したサービスと、Web-API を組み合わせることで、センシングされたデータをネットワーク経由で使用したコンテキストウェアサービスの実現が可能となる。

2.2 ホームネットワークシステム (HNS)

宅内の家電や設備機器をネットワークに収容して、付加価値サービスを実現するシステムを **ホームネットワークシステム (HNS)** という。TV や DVD、照明、エアコン、カーテン、扇風機、空気清浄機などの機器がネットワークに接続され、様々なサービス・アプリケーションが実現される。この HNS の実現法として、我々の研究室では、サービス指向アーキテクチャ (SOA) を HNS に適用し、各家電の機能を Web サービスとして利用できる HNS 環境 **CS27-HNS** [3] を開発している。CS27-HNS では機器依存の制御方法や通信プロトコルを Web サービスでラップしており、全ての機器の機能を SOAP または REST 形式の Web-API として利用できる。例えば、テレビのチャンネルを 6ch にするには、`http://hns/TVService/setChannel?channel=6` といった URL にアクセスするだけで良い。

2.3 コンテキストウェアサービス

コンテキストとは、人や物、周辺環境など現実世界の状況を表すものである。コンテキストウェアサービスとは、様々なシステムやセンサなどより得られる情報からコンテキストの変化を認識し、その変化に対応した機能や情報をユーザに提供するサービスのことを指す。

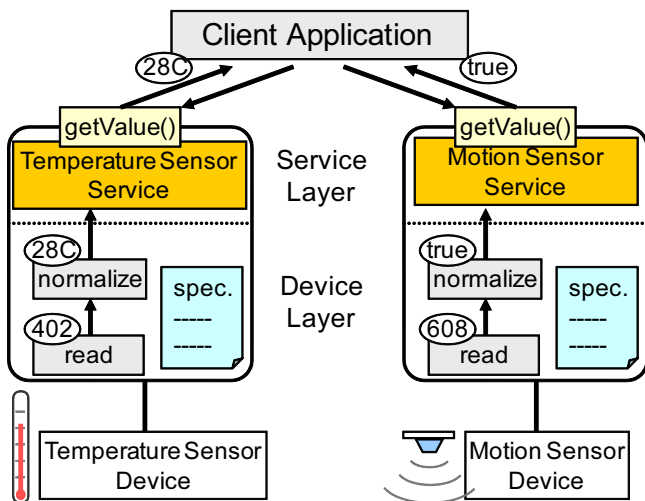


図 1 SSF の標準的なインターフェースによるセンサ値の取得

例えば、「部屋の温度センサの値が 28 度以上」という情報からは「暑い」というコンテキストが推定される。システムはこの「暑い」というコンテキストをもとに「エアコンの冷房運転を開始する」といったコンテキストに応じたサービスを提供する。

2.4 先行研究

Web サービスを意識したコンテキストウェアサービス実現に向けた先行研究として、温度センサや照度センサなどの環境センサを、Web サービスとして容易に配備可能とするセンササービスフレームワーク (SSF) [4] や、複数のセンササービスを組み合わせるより高度なセンササービスを構築可能にするセンサマッシュアッププラットフォーム (SMuP) [5]、センサを利用したコンテキストウェアサービスをエンドユーザが開発できるセンササービスバインダー (SSB) [6] が提案されている。

SSF の利用イメージを図 1, 図 2 に示す。SSF において、各センサは自身が測定可能なプロパティを持っている。例えば、温度センサであれば temperature という摂氏温度をプロパティとして持っている。ユーザは getValue メソッドを利用することで、各センサのプロパティの値を取得することができる。また、各センサは自身のプロパティ値の変動を監視しており、登録された条件式 (コンテキスト条件) に基づいてコンテキストの検出を行うことができる。コンテキスト条件のセンササービスへの登録は、register メソッドによって行うことができる。例えば、温度センサに Hot というコンテキスト名で temperature >= 28°C というコンテキスト条件を登録できる。登録されたコンテキストは subscribe メソッドによって、任意の Web サービス呼び出しと関連付けることができ、センササービスはコンテキスト条件の値が真になったとき、notify メソッドにより関連付けられた Web サービスを呼び出す。このようにして、コンテキストウェアサービスを作成できる。

SMuP は、既存のセンササービスのプロパティやコンテキスト条件を算術演算や論理演算で結合した仮想的なセンササービスを作成できる。例えば、温度センサと湿度センサを組み合わせる不快指数を算出する仮想センサなど、単一のセンサでは不可能だったコンテキストの検出が可能となる。SMuP で作成さ

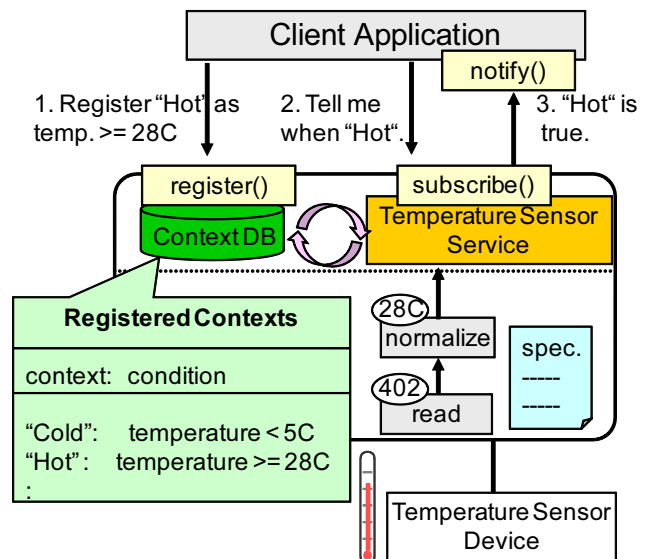


図 2 SSF によるコンテキストウェアサービス

れた仮想センサは SSF のセンササービスと同様に使用することができるので、センサを組み合わせるコンテキストウェアサービスを作成することができる。

SSB は、専門知識を持たないエンドユーザを対象としたセンサ駆動サービス構築支援環境であり、専門知識の必要な Phase と専門知識が不要な Phase が分離されており、後者の Phase では、表示されたコンテキスト名や機器機能名をエンドユーザが選択するだけでセンサ駆動サービスの構築が可能となる。

2.5 コンテキストウェアサービス作成の課題

SSF のように推定したコンテキストからコンテキストウェアサービスを作成する研究では、センサなどのデバイスとサービスの結びつきが強く、他のデバイスやサービスとの連携が取れない状態にある。SMuP においても、複数のセンササービスの連携は可能であるが、センササービス以外のサービスとの連携は不可能である。ソフトウェアの機能をサービスという単位で捉え、それらのサービスを連携・統合して新たなサービスを開発する、というサービス指向アーキテクチャ (SOA) の考え方にに基づき、センサだけでなく様々な種類の複数のサービスを組み合わせることによる高機能なサービスの実現と、そのサービス自身も他のサービスと組み合わせることが可能である、ということがクラウド時代においては必要である。

そのため、異種分散 Web サービスに基づいたコンテキストを組み合わせるコンテキストウェアサービスを作成及び統合的に管理するフレームワークを作成し、センササービスに限らず複数の異なるサービスを連携したコンテキストウェアサービスを作成可能にすること、また、フレームワークに登録されたコンテキストなどの要素が他のサービスからも使用可能にすることが必要である。

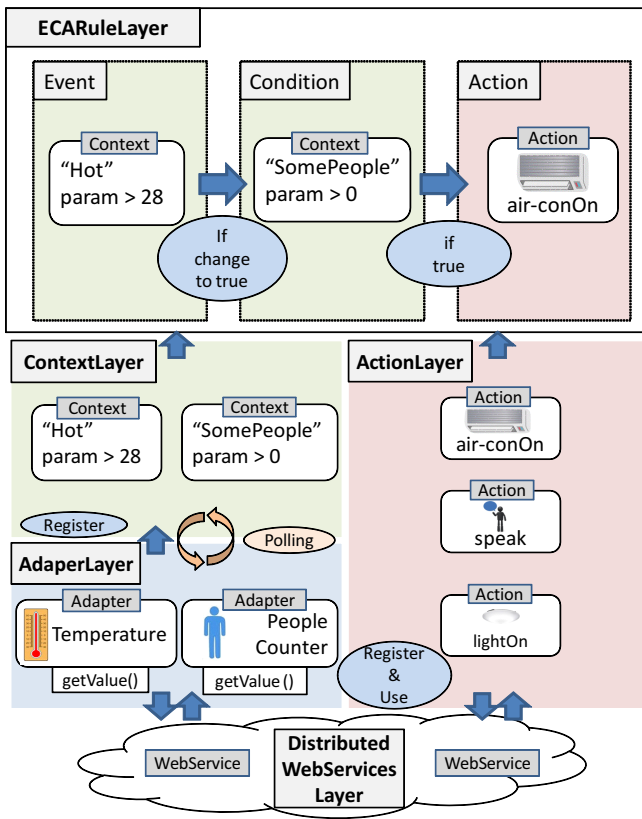


図3 提案フレームワークのアーキテクチャ

3. 異種分散 Web サービスに基づくコンテキストトアウェアサービスの管理フレームワーク

3.1 システム要件

本節では異種分散 Web サービスを連携したコンテキストトアウェアサービスを作成可能にするために、以下の R1-R4 の要件を満たすフレームワークを提案する。以降、提案フレームワーク内の要素の表現には英語表記を用いる。

R1: Context として異種分散 Web サービスの情報を登録可能である

R2: Action として異種分散 Web サービスを登録可能である

R3: Context と Action を組み合わせたコンテキストトアウェアサービスが作成可能である

R4: 登録された各要素の利用・管理が可能である

3.2 アーキテクチャ

図3に提案フレームワークのアーキテクチャを示す。提案フレームワークは5つのレイヤ (ECARuleLayer, ContextLayer, ActionLayer, AdapterLayer, WebServiceLayer) から成る。各レイヤを相互に利用することでレイヤの要素を作成し、図3中の ECARuleLayer のように、作成した要素を利用して ECA 規則を定義することでコンテキストトアウェアサービスを作成する。以下に各レイヤの機能について示す。

3.2.1 WebServiceLayer

WebServiceLayer は、既存の Web サービスのレイヤであり、数値や真偽値など何らかの値が返り値として取得できる Web サービスならば、全て提案フレームワークで使用すること

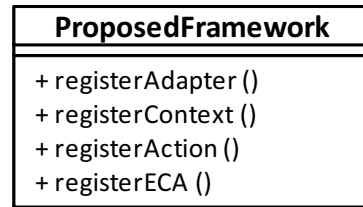


図4 提案フレームワークが実行可能なメソッド

が可能であり、このレイヤに含まれる。

3.2.2 AdapterLayer

WebServiceLayer 内の Web サービスを AdapterLayer に登録したものを提案フレームワークにおける **Adapter** と定義する。

SSF のようなセンササービスを利用する場合と比較し、一つのサービスに依存しないため値を取得するメソッドを利用し、判定などの処理はフレームワーク内で行う。

3.2.3 ContextLayer

設定された時間ごとに Adapter から値を取得し、条件式が成立しているか否かの判定を行うものを提案フレームワークにおける **Context** と定義し、ContextLayer 内で登録・管理される。条件式は「param >= 20」のような比較演算で表現され、「param」の部分に Adapter から取得した値が代入される。成立していればその Context が真 (true) であるとし、成立していなければ偽 (false) とする。Context には **Atomic** (単一) と **Compound** (複合) の二種類が存在する。上記の例は AtomicContext の条件式であり、CompoundContext の場合は「Hot && NoPeople」のように、ContextLayer に存在する複数の Context を論理演算子 (!, &&, ||) で繋ぐことで条件式を表現する。

3.2.4 ActionLayer

家電操作など現実世界の機器を制御する Web サービスを ActionLayer に登録したものを提案フレームワークにおける **Action** と定義する。

3.2.5 ECARuleLayer

各レイヤに登録された Context と Action を使用し ECA 規則を定義してコンテキストトアウェアサービスを作成する。**ECA 規則** (Event-Condition-Action Rule) とは、サービスの起点となる Event が発生した際、条件である Condition が満たされていた場合、処理である Action を実行する、という規則のことである。

提案フレームワークでは、以下のように Context と Action を規則に定義する。

Event: コンテキストトアウェアサービスの起点となる Context

Condition: 必要条件となる一つ以上の Context

Action: 実行する Web サービスにあたる Action

Event に定義された Context が「false から true」になった際に、Condition に定義された Context が「全て true」である場合、Action に定義された Web サービスが実行される。

3.3 フレームワークが実行可能なメソッド

図 4 に提案フレームワークが実行可能なメソッドを示す。各レイヤの要素の登録と、ECA 規則の定義を行いコンテキストウェアサービスを作成するメソッドである。提案フレームワークは Web サービスとして提供されるため、メソッドを実行するためには以下のような URL にアクセスすればよい。

http://hns/PropFW/registerAdapter?adapterid=Temperature&endpoint=http://hns/TemperatureSensorService&method=getValue

registerAdapter の引数は、Adapter を一意に決定するための *adapterid*, 値を取得する Web サービスのアドレスを示す *endpoint*, 値を取得するためのメソッドを示す *method* である。

registerContext の引数は、Context を一意に決定するための *contextid*, Context が Atomic(A) か Compound(C) かを示す *type*, 条件式を示す *expression*, Adapter から値を取得する間隔をミリ秒で示す *sensitivity*, 値を取得する Adapter を一意に決定するための *adapterid* である。

registerAction の引数は、Action を一意に決定するための *actionid*, 実行される Web サービスのアドレスとなる *url* である。

registerECA の引数は、コンテキストウェアサービスを一意に決定するための *ecaid*, Event となる Context の id である *eventid*, Condition となる Context の id である *conditionid*, Action となる *actionid* である。

3.4 提案フレームワークによるコンテキストウェアサービスの作成の概要

提案フレームワークでは、値を取得する Web サービスを Adapter として登録し、登録した Adapter から取得する値に対する条件式を定義し、Context を作成する。また、家電操作など制御を行う Web サービスを Action として登録し、登録した Context と Action を図 3.2 のように組み合わせて ECA 規則を定義することでコンテキストウェアサービスを作成する。

4. ケーススタディ

本節では、ケーススタディとして本研究室の HNS における提案フレームワークを利用したコンテキストウェアサービス「クールサービス」の作成の考察を行う。

Event に「気温と湿度が高い」という CompoundContext を、Condition に「人が在室している」と「明るい」という AtomicContext を定義し、冷房を ON にする旨の音声出力後に冷房を ON にする Action を持つ「クールサービス」を作成する。図 5 にクールサービスの概要を示し、表 1-4 に作成にあたり実行するメソッドとその引数の一覧を示す。メソッドの実行には 3.3 に示したような URL にアクセスすればよい。

WebServiceLayer:

- **各センササービス:** 気温、湿度、照度センサの値を取得するサービスである。
- **入退室管理サービス (InOutUserManageService):** 入退室時に入室または退室に相当するメソッドを Web ページ経由で実行することで、在室状況を管理するサービスである。こ

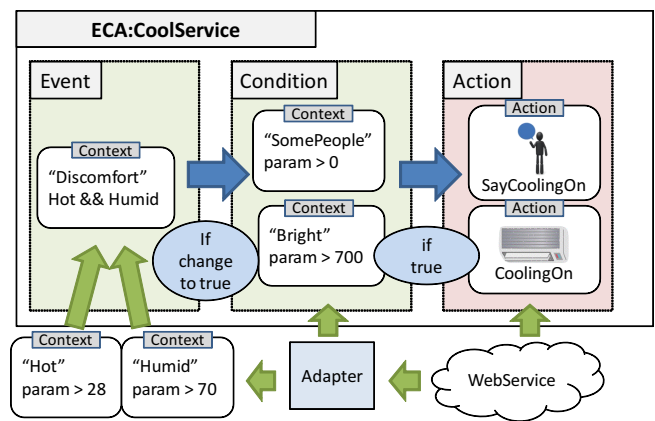


図 5 CoolService の概要

のサービスは現在の在室者の数を取得するメソッド (getHeads) を持つ。

- **棒読みサービス (BoyomiService):** 入力されたテキストを音声出力するサービスである。
- **エアコン操作サービス (AirConditionerCeilingService):** エアコンの操作を行うサービスである。

AdapterLayer: registerAdapter メソッドにより、Adapter 「Temperature」, 「Humidity」, 「Light」, 「PeopleCounter」を登録する。

ContextLayer: registerContext メソッドにより、Atomic-Context 「Hot」, 「Humid」, 「SomePeople」, 「Bright」を登録し、「Hot」と「Humid」より CompoundContext 「Discomfort」を登録する。

ActionLayer: registerAction メソッドにより、Action 「SayCoolingOn」, 「CoolingOn」を登録する。

ECARuleLayer: これまで登録した要素を組み合わせる ECA 規則を定義し「クールサービス」を作成する。Event に「Discomfort」、Condition に「SomePeople」と「Bright」、Action に「SayCoolingOn」と「CoolingOn」を定義したコンテキストウェアサービス「CoolService」を作成する。

以上により、気温が 28℃より大きい、かつ湿度が 70%より大きいという条件が成立したとき、人がいて、かつ照度が 700より大きければ、「冷房をつけます」と音声出力した後、冷房の電源を ON にするコンテキストウェアサービスが作成できることが確認できた。

5. 考察

提案フレームワークにより、様々な種類の分散した Web サービスを利用したコンテキストウェアサービスを実現することが可能であることが確認できた。しかし、コンテキストウェアサービス作成の際、メソッドを実行するために入力により逐一 URL にアクセスしていたのでは、作成するコンテキストウェアサービスの数の増加に比例して、メソッド実行のための入力量及びアクセス回数が大幅に増加してしまう。

また、提案フレームワークでは、複数の Context を複合した CompoundContext の作成や、コンテキストウェアサービスの作成は可能であるが、複数の Web サービスから取得した複

表 1 registerAdapter メソッドの引数

adapterid	endpoint	method	概要
Temperature	http://...TemperatureSensorService	getValue	気温を取得する
Humidity	http://...HumiditySensorService	getValue	湿度を取得する
Light	http://...LightSensorService	getValue	照度を取得する
PeopleCounter	http://...InOutUserManageService	getHeads	在室者数を取得する

表 2 registerContext メソッドの引数

contextid	type	expression	sensitivity	adapterid	概要
Hot	A	param>28	7,000	Temperature	気温が 28 度を越えている
Humid	A	param>70	7,000	Humidity	湿度が 70%を越えている
SomePeople	A	param>0	5,000	PeopleCounter	在室者が存在する
Bright	A	param>700	7,000	Light	照度が 700lx を越えている
Discomfort	C	Hot&&Humid	10,000	empty	Hot と Humid が同時に真である

表 3 registerAction メソッドの引数

actionid	url	概要
SayCoolingOn	http://...BoyomiService?speak=冷房をつけます	音声「冷房をつけます」
CoolingOn	http://...AirConditionerCeilingService?on	冷房をつける

表 4 registerECA メソッドの引数

ecaaid	eventid	conditionid	actionid
CoolService	Discomfort	SomePeople, Bright	SayCoolingOn, CoolingOn

数の値を利用しての Context は登録できない。例えば、「不快度指数が 80 以上」といった Context を登録したいとする。不快度指数の式は温度を T °C, 湿度を H % とすると, 以下の式で表される。

$$0.81T + 0.01H(0.99T - 14.3) + 46.3 \geq 80$$

この式を条件式にする場合, 温度と湿度を一つの Context で同時に取得し計算しなければならないが, 現在のフレームワークではこのような Context は登録することができない。そのため, 計算するレイヤを追加する必要がある。

次に, 作成されたコンテキストウェアサービスの数が増加するにしたがって, サービス競合が発生することが予測される。サービス競合問題については, 先行研究 [7] において検出・解消策が考案されているため, フレームワークに組み込むことを視野に入れていく。

今後の課題は, 提案フレームワークの実装を行ったうえで, 実際にコンテキストウェアサービスの作成を行い, 評価を行うことが必要である。また, 提案フレームワークの各レイヤの要素の登録及びコンテキストウェアサービスの作成を容易に行うことができ, かつ各要素を利用しやすくするためのサービスを提供することが必要である。

6. おわりに

本論文では, コンテキストウェアサービスのデバイスとサービスの結びつきが強く, 他のデバイスやサービスとの連携が取れない問題に対し, 異種分散 Web サービスを使用したコンテキストウェアサービスを作成可能な提案フレームワークを適用することで, この問題を解決できることを確認した。また, ケーススタディを通して, センササービス以外のサービスを含む複数の異なる Web サービスを連携したコンテキストウェアサービスの作成が, 従来のセンサ駆動サービスより容易に可能であることが確認できた。

今後の課題としては, 提案フレームワークを実装し, 実際に稼働させた上でコンテキストウェアサービスを作成すること。また, 各要素及びコンテキストウェアサービスの登録・作成及び管理を容易にするサービスを提供することである。

謝辞 この研究の一部は, 科学技術研究費 (基盤研究 C 24500079, 基盤研究 B 23300009), 及び, 積水ハウスの研究助成を受けて行われている。

文 献

- [1] 辻賢太郎, 上岡英史, “センサ情報に基づいたユーザへのアラート方式 (ホームネットワーク, ユビキタスネットワーク, コンテキストウェア, e コマース及び一般),” 電子情報通信学会技術研究報告. MoMuC, モバイルマルチメディア通信, vol.108, no.290, pp.15–20, nov 2008.
- [2] 坂本寛幸, 井垣 宏, 中村匡秀, “コンテキストウェアアプリケーションの開発を容易化するセンササービス基盤,” 電子情報通信学会技術研究報告, vol.108, no.458, pp.381–386, March 2009.
- [3] 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, Jan. 2008.
- [4] M. Nakamura, S. Matsuo, S. Matsumoto, H. Sakamoto, and H. Igaki, “Application framework for efficient development of sensor as a service for home network system,” the 8th IEEE 2011 International Conference on Services Computing (SCC 2011), pp.576–583, July 2011.
- [5] 坂本寛幸, 井垣 宏, 中村匡秀, “SMuP: センササービスのマッシュアップを実現するサービス指向基盤,” ウィンターワークショップ 2010・イン・倉敷 論文集, vol.2010, no.3, pp.73–74, Jan. 2010.
- [6] M. Nakamura, S. Matsuo, and S. Matsumoto, “Supporting end-user development of context-aware services in home network system,” Studies in Computational Intelligence, ed. by R. Lee, pp.159–170, Springer, Nov. 2012.
- [7] 池上弘祐, 吉村悠平, 井垣宏, 中村匡秀, “サービス期間を考慮したホームネットワークサービス競合検出・解消システムの実装,” 電子情報通信学会 OIS 研究会, 第 IEICE-108 巻, pp.007–012, March 2009.