一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
ISEC2013-70 LOIS2013-36 (2013-11)

異種分散Webサービスに基づくコンテキストアウェアサービスの管理プラットフォームの実装

髙塚　広貴†　　佐伯　幸郎†　　柗本　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りつつある．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アウェアサービスの実現が期待されている．我々は先行研究において，様々な種類の分散した Web サービス (センササービス，情報システム，家電サービスなど) を利用したコンテキストア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の提案を行っている．提案フレームワークは 5 つのレイヤから構成され，既存の Web サービスから取得されるデータに基づいてコンテキストを定義し，コンテキストアウェアサービスを Event, Condition, Action の組である ECA 規則で定義する．本稿では，提案されたフレームワークに基づき，各レイヤの要素の作成，削除，取得，編集及びコンテキストアウェアサービスを実行する機能を持つ基盤の実装を行う．これらの機能を Java を用いて実装し，API を Web サービスとして公開する．

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

Implementing Execution Platform 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. We have previously proposed 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 services and so on). The proposed framework consists of five layers and we define the context based on the data of the existing Web services, and then define the context-aware service by the ECA rule that is a set of event, condition and action. In this paper, we implement a platform based on the framework. The platform has the functions to register, delete, get and edit elements in each layer and to run the services. We implement these using Java, and publish APIs as the Web service.

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 サービスや 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**コンテキストアウェアサービス**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ア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ア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アウェアサービス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，コンテキストア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．本件については先行研究において，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] が提案されている．

このようにコンテキストア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，コンテキストア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．これらのコンテキストを用いたサービスを作成・再利用できる統一的な枠組みが存在すれば，従来のコンテキストア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，より高度なコンテキストアウェアサービスの作成が可能となる．

我々は先行研究において，異種分散 Web サービスを利用したコンテキストアウェアサービスの開発支援及び，体系的管理を可能とするフレームワーク (**RuCAS**, Rule based management framework for Context-Aware Services) [3] を提案している．提案フレームワークは 5 つのレイヤ (WebServiceLayer, AdapterLayer, ContextLayer, ActionLayer, ECARuleLayer) から構成され，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アウェアサービスを作成する．提案フレームワークでは，入出力をセンサなどの特定のデバイスに限定せず，異種分散 Web サービスを組み合わせたコンテキストアウェアサービスを作成・管理できるようになる．

そこで本稿では，RuCAS の実行基盤である，**RuCAS Platform** の実装を行い，コンテキストアウェアサービスの開発支援，体系的管理及び実行を可能とし，これらの機能を Web サービスとして使用可能な API の提供を実現する．また，RuCAS Platform を利用したコンテキストアウェアサービスの動作確認のため，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いて，RuCAS Platform の利用によるコンテキストアウェアサービスの作成及び実行を行う．

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 サービス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となった．

Web サービスとは，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 サービスでは，サービスの 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 サービスを利用することで，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 サービス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ア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 や照明，エアコン，カーテン，扇風機などの家電及び，温度センサや湿度センサ，照度センサなどの各種センサをはじめとした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・センサの機能を Web サービスとして利用できる HNS 環境 **CS27-HNS** [4] 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 にアクセスすることで実現できる．

2.3 コンテキストアウェアサービス

コンテキストとは，人や物，環境など現実世界の状況のことであり，様々なセンサやシステムから得た情報をもとに定義される．**コンテキストアウェアサービス**とは，コンテキストの変化を自動的に検知し，その変化に対応した機能や情報を提供するサービスのことを指す．

例えば,「部屋の温度センサの値が 28 ℃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．

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

本節では，先行研究において提案したフレームワーク (**RuCAS**) の説明を行う．ただし，RuCAS 内の要素の表現には英語表記を用いる．

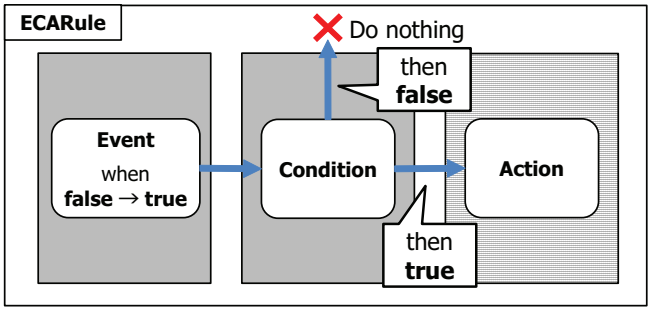

図 1　ECA 規則の流れ

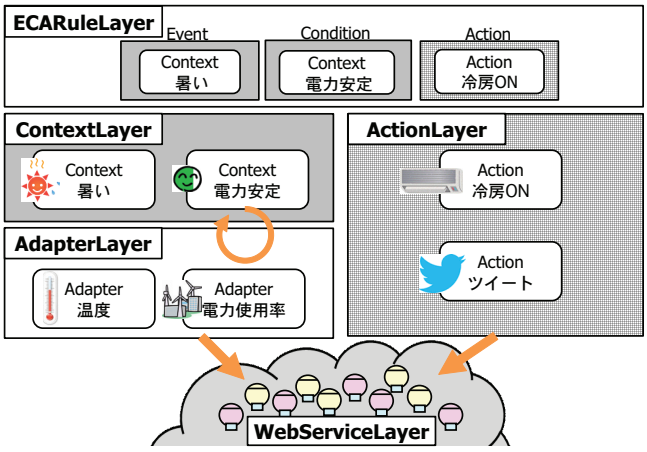

図 2　RuCAS のアーキテクチャ

3.1　ECA 規則 (Event-Condition-Action Rule)

ECA 規則とは，トリガとなる Event が発生した際，条件部となる Condition が満たされていた場合，Action を実行する，という規則である．ECA 規則の流れを図 1 に示す．RuCAS では，以下のようにコンテキストアウェアサービスを ECA 規則の形で定義する．

Event: サービスのトリガとなる Context

Condition: 必要条件となる一つ以上の Context

Action: 実行対象の Action

3.2　RuCAS のアーキテクチャ

図 2 に RuCAS のアーキテクチャを示す．RuCAS は ECA 規則を定義する上での役割により分割された 5 つのレイヤ (WebServiceLayer, AdapterLayer, ContextLayer, ActionLayer, ECARuleLayer) から構成されている．各レイヤにおいて，下層の機能を利用することで要素を作成し，ECARuleLayer において，作成した要素を組み合わせて ECA 規則を定義することでコンテキス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

既存の Web サービスのレイヤであり，コンテキストアウェアサービスの入力となる数値や真偽値など，何らかの値が取得できる Web サービスと，コンテキストアウェアサービスの出力となる機器制御などを行う Web サービスがこのレイヤに含まれる．

3.2.2　AdapterLayer

コンテキストアウェアサービスの入力となる Web サービスの API を統一するレイヤであり，サービスの API を getValue() に統一する **Adapter** をそれぞれのサービスについて登録する．

3.2.3　ContextLayer

条件式や判定間隔をもとに定義された **Context** を一元管理するレイヤである．Context は判定間隔ごとに指定された Adapter を使用することで値を取得し，条件式が成立しているか否かの判定を行う．条件式は「value ≧ 20」のような比較演算で表現され，“value” 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) であるとして，次回の判定までこの真偽値を保持する．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 && Humid」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

コンテキストアウェアサービスの出力となる Web サービスを管理するレイヤであり，Web サービスのエンドポイントやメソッド名，引数をもとに **Action** を登録する．

3.2.5　ECARuleLayer

ContextLayer で登録した Context と，ActionLayer で登録した Action を使用し ECA 規則を定義することで，コンテキストアウェアサービスを作成するレイヤである．

Event に指定された Context が「false から true」になった際に，Condition に指定された Context が「全て true」である場合，Action に指定された Web サービスが実行される．

3.3　コンテキストアウェアサービス作成の概要

RuCAS では，入力となる 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などをもとにを Context を登録する．また，出力となる Web サービスを Action として登録する．登録した Context と Action を組み合わせて ECA 規則を定義することでコンテキストアウェアサービスを作成する．

4.　RuCAS Platform

本節では，RuCAS の実行基盤である **RuCAS Platform** における各クラスの機能について示す．RuCAS はコンテキストアウェアサービスの体系的管理，作成支援及び実行を行う基盤である．図 3 に RuCAS Platform のクラス図を示す．

4.1　Adapter

Adapter は，RuCAS における AdapterLayer のクラスである．endpoint, method をもとに getValue() により Web サービスから値を取得する．Web サービスの返り値が複数ある場合は property でそれらの名称を保持し，いずれかを getValue() の引数にすることで取得する値を選択する．Adapter の持つ主要な属性とメソッドを以下に示す．

`adapterid:`　Adapter を一意に決定する ID

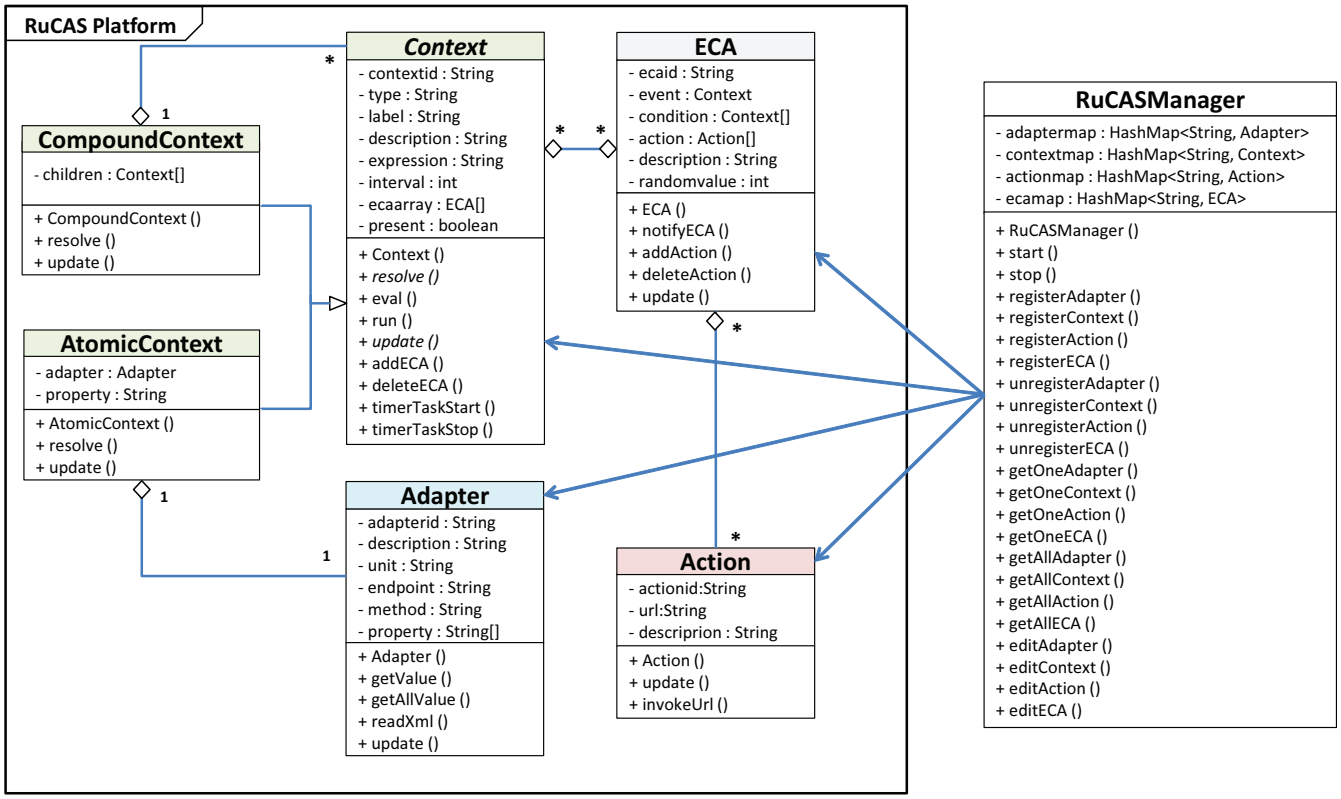

図 3　RuCAS Platform のクラス図

`endpoint:`　使用する Web サービスのエンドポイント
`method:`　使用する Web サービスの API
`getValue():`　Web サービスを実行し指定した値を取得する

4.2　Context

Context は，RuCAS における ContextLayer のクラスである．また，Context は抽象クラスであり，後述の AtomicContext と CompoundContext により継承されている．Context では，interval で設定された間隔ごとに run() を実行する．run() 内では，Web サービスから取得した値を条件式である expression に代入する resolve()，代入された条件式の評価を行う eval() が順に実行され，次回の run() 実行まで評価の結果を present に保持する．また，present の値が “false” から “true” になった場合は，Context が Event に指定されている ECA のインスタンスの notifyECA() を実行する．Context の持つ主要な属性とメソッドを以下に示す．

`contextid:`　Context を一意に決定する ID
`type:`　単一 (A) または複合 (C) かの種類
`expression:`　Context の条件式
`interval:`　Context の成立判定を行う間隔 (ミリ秒)
`ecaarray:`　自信が Event に指定されている ECA の配列
`present:`　Context の現在の成立状態
`eval():`　条件式の評価を行う
`run():`　判定間隔ごとに代入・評価を行う

4.2.1　AtomicContext

AtomicContext は，RuCAS における AtomicContext のクラスであり，Context を継承している．条件式の評価に使用する Adapter を持ち，resolve() においてその Adapter の getValue() を呼び出し，取得した値を条件式の “value” 部分に代入する．AtomicContext の主要な属性とメソッドを以下に示す．

`adapter:`　値を取得する Adapter
`resolve():`　条件式の “value” 部分に取得した値を代入する

4.2.2　CompoundContext

CompoundContext は，RuCAS における CompoundContext のクラスであり，Context を継承している．条件式である expression に含まれている Context を children に持ち，resolve() において children のそれぞれの Context について resolve() を再帰的に呼び出し，条件式の contextid 部分に代入する．ConmpundContext の主要な属性とメソッドを以下に示す．

`children:`　条件式に含まれている Context の配列
`resolve():`　条件式の contextid 部分に取得した値を代入する

4.3　Action

Action は，RuCAS における ActionLayer のクラスである．Web サービスのエンドポイント，メソッド，引数をまとめて url として持つ．Action の主要な属性とメソッドを以下に示す．

`actionid:`　Action を一意に決定する ID
`url:`　メソッドと引数を含む Web サービスのアドレス
`involeUrl():`　Web サービスを実行する

4.4　ECA

ECA は，RuCAS における ECA 規則によるコンテキストアウェアサービスのクラスである．event に指定されている Context が “false” から “true” になった際に，その Context から notifyECA() が呼び出される．notifyECA() では，condition

に指定されている Context の成立状態を確認し，全て “true” ならば，action に指定されている Action の invokeUrl() を呼び出し，Web サービスを実行する．また，ECA クラスのインスタンスが作成された際，event に指定された Context の ecaarray に作成された ECA を追加する．ECA の主要な属性とメソッドを以下に示す．

`ecaid:` 作成したサービスを一意に決定する ID
`event:` Event となる Context
`condition:` Condition となる Context の配列
`action:` ECA 規則の Action となる Action の配列
`notifyECA():` ECA 規則にあたるメソッド

4.5 RuCASManager

RuCASManager は RuCAS Platform の各オブジェクトを操作するインターフェースを提供するクラスである．各クラスのインスタンスの生成，削除，取得，編集を行うメソッドや Context の成立判定の開始と停止，つまりコンテキストアウェアサービスの実行と停止を一括して行うメソッドを持つ．また，RuCASManager のメソッドは RuCAS Platform の API として外部に提供される．例えば，温度センササービスをもとにした Adapter「Temp」を登録する API を実行する場合，以下のような URL にアクセスすればよい．

```
http://hns/RuCASPlatform/registerAdapter?adapterid=
Temp&description=室内温度の取得&unit=℃&endpoint=http
://hns/TemperatureSensorService&method=getValue&pro
perty=return
```

なお，Web サービスとして使用可能にするために，RuCASManager で Context や ECA を扱う際には属性の変換を行っている．

4.6 実　　装

RuCAS Platform では各インスタンスの持っている属性を BSON 形式に変換し，MongoDB に保存している．また，開発工数は 3 人月，行数は 3296 行であった．RuCAS Platform の実装に使用した技術を以下に示す．

開発言語： Java 1.7.0_21
`DB:` MongoDB 2.4.3
`Web` サーバ： Apache Tomcat 7.0.39
`Web` サービスエンジン： Apache Axis2 1.6.2

5. ケーススタディ

本節では，ケーススタディとして CS27-HNS における RuCAS Platform を利用したコンテキストアウェアサービス「CoolService」の作成及び実行を行う．「CoolService」は，「蒸し暑くなった際，関西の電力使用量に余裕があれば，冷房を ON にする」といったサービスである．サービス作成には，以下の既存の Web サービスを使用する．

- **温度・湿度センササービス:** API を通じて，温度・湿度センサの値を取得するサービスである．
- **電力使用状況 API [5]:** 電力会社の電力使用状況を取得する API である．
- **家電リモコンサービス:** Web サービス経由で家電を操作するサービスである．

RuCAS Platform の API の実行系列を以下に示す．

```
/* Adapter 登録 */
registerAdapter("Temp", "室内温度の取得", "℃",
"http://.../TemperatureSensorService", "getValue",
"return");
registerAdapter("Humidity", "室内湿度の取得", "%",
"http://.../HumiditySensorService", "getValue",
"return");
registerAdapter("PowerUsageOfKansai", "関西の電力使用
状況", "kW", "http://...Setsuden", "latestPowerUsage
?appid=...&area=kansai", {"Usage", "Capacity"});
/* Context 登録 */
registerContxt("Hot", "A", "環境", "室内温度が 28 ℃以
上", "value>=28", 5000, "Temp", "return");
registerContxt("Humid", "A", "環境", "室内湿度が 75%
以上", "value>=75", 5000, "Humidity", "return");
registerContxt("Muggy", "C", "環境", "蒸し暑い",
"Hot&&Humid", 5000, "", "");
registerContxt("PowerStable", "A", "電力", "関西の
電力使用量に余裕がある", "value<=18000000", 1800000,
"PowerUsageOfKansai", "Usage");
/* Action 登録 */
registerAction("CoolingOn",
"http://.../IRemoconService/sendIRSignal?...",
"冷房を ON にする");
/* ECA 登録 = コンテキストアウェアサービス作成*/
registerECA("CoolService", "Muggy", "PowerStable",
"CoolingOn", "蒸し暑くなった際に電力使用量に余裕があれ
ば冷房を ON にする", 0);
```

API の実行により，室内の温度が 28 ℃以上かつ室内の湿度が 75%以上になった際，関西の電力使用量が 1800 万 kW 以下ならば，冷房を ON にするコンテキストアウェアサービスを RuCAS Platform の使用により作成した．なお，作成に要した時間は約 20 分であった．

次に，コンテキストアウェアサービス「CoolService」を実行する．実行に際し，Context の成立を容易にするため，“Hot” の条件を「25 ℃以上」，“Humid” の条件を「27%以上」に変更した．図 4 にサービスの実行を行った 2013 年 11 月 2 日の 8 時から 20 時の研究室内の温度・湿度，関西の電力使用状況を示す．Event である “Muggy” の成立は，“Hot” と “Humid” が両方成立することが必要である．“Humid” については常に成立しており，“Hot” については 12 時ごろに室内温度が 25 ℃を上回り，Event である “Muggy” が成立した．この時，Condition である “PowerStable” が成立していたため，Action である “CoolingOn” が実行され，冷房が ON になった．

以上により，RuCAS Platform の API の実行によりコンテキストアウェアサービスである「CoolService」作成した．また，サービスを実行した際，実際に定義したコンテキストに応じて動作が行われることを確認した．

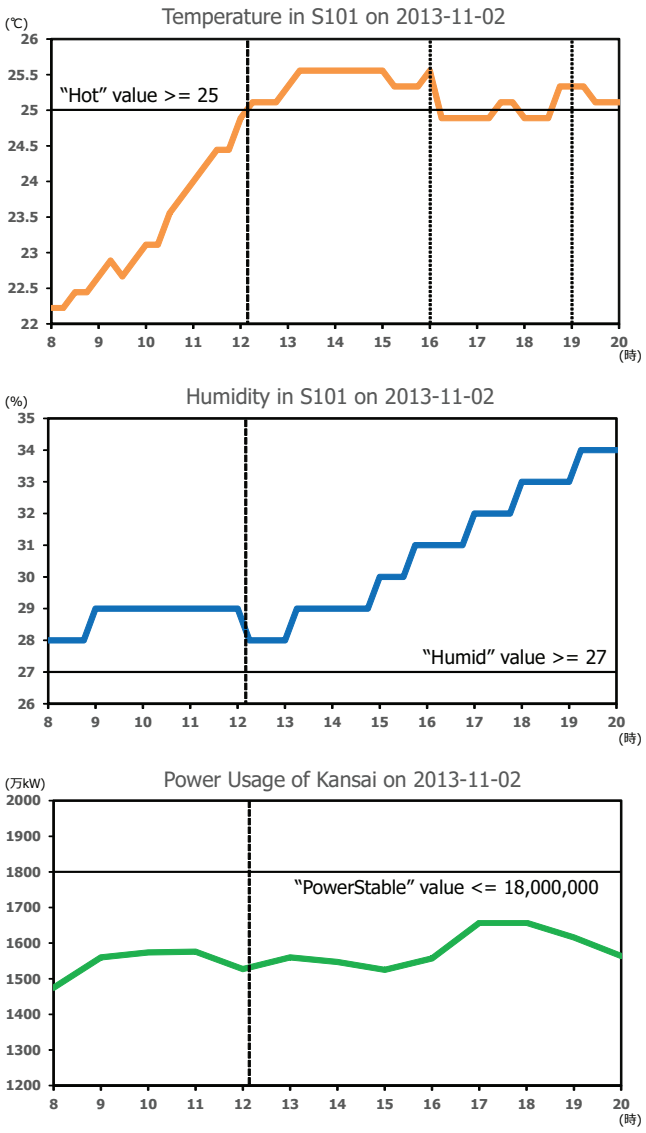

図 4　2013 年 11 月 2 日の研究室内の温度・湿度，関西の電力使用状況

6. 考　　察

一般に，気温などの値は単調に増加・減少するのではなく，増減を繰り返しながら増加・減少していく．そのため図 4 の温度グラフにおける 16 時から 19 時の間のように，Event にあたる Context が成立するしきい値付近において値が増減する場合，“true” と “false” 間の状態遷移を繰り返し，結果として Action を何度も実行する．この問題に対し，Context が “false” から “true” になった際に，一時的に条件式のしきい値を安定補償値 α だけ変化させ，再び “false” になった際に元の値に戻す，といった実装を行った．例えば，“Hot” ならば，“false” から “true” になった際に，条件式を「value≧25」から「value≧25 + α」に一時的に変更する．

また，作成されたコンテキストアウェアサービスの数が増加するにしたがって，複数のサービスが互いに干渉・衝突を起こすサービス競合が発生する．例えば,「外出する際，冷房を OFF にする」というコンテキストアウェアサービス「おでかけサービス」と 5. で作成した「CoolService」が存在しているとする．ここで,「CoolService」により冷房が ON になっている状態で誰かが外出すると,「おでかけサービス」により冷房が OFF になる，という競合が発生する．サービス競合問題については，先行研究 [6] において検出・解消策が考案されているため，これらをフレームワークで利用することで解決できる．

今後の課題としては，RuCAS Platform を使用してコンテキストアウェアサービスを作成する場合と RuCAS Platform を使用せずコンテキストアウェアサービスを作成する場合の工数の比較や，コンテキスト及びコンテキストアウェアサービスの数が増加した際の動作の安定性などといった RuCAS Platform の評価を行うこと．また，RuCAS Platform の使用を支援するサービスの提供を行うことである．

7. おわりに

本稿では，異種分散 Web サービスを使用したコンテキストアウェアサービスの開発支援と体系的管理及び実行を行う基盤である RuCAS Platform の実装を行い，RuCAS Platform の API を使用して，コンテキストアウェアサービスの作成及び実行の確認を行った．RuCAS Platform によりコンテキストアウェアサービスの体系的管理を行うことで，従来のコンテキストアウェアサービスよりも多くの現実世界の情報に基づいた，より高度なコンテキストアウェアサービスの作成が可能となる．また，ケーススタディを通して，RuCAS Platform の API を使用して実際に動作するコンテキストアウェアサービスを作成した．

今後の課題としては，RuCAS Platform の評価及び，RuCAS Platform の使用を支援するサービスの実装を行うことである．

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ティング方式,” 電子情報通信学会技術研究報告，vol.108, no.290，pp.15–20，Nov. 2008.

[2] 坂本寛幸，井垣　宏，中村匡秀，“コンテキストア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] 高塚広貴，佐伯幸郎，まつ本真佑，中村匡秀，“異種分散 web サービスに基づくコンテキストアウェアサービスの管理フレームワークの提案,” 電子情報通信学会技術報告，vol.113，no.245，pp.1–6，Oct. 2013.

[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, Jan. 2008.

[5] “Yahoo JAPAN Web API 電力使用状況 API,” `http://developer.yahoo.co.jp/webapi/shinsai`.

[6] 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，vol.108，no.462，pp.007–012，March 2009.