

クラウドと Web of Things を活用した サービス指向ホームネットワークシステム

高塚 広貴¹ 鷹取 敏志¹ 堀内 大祥¹ 佐伯 幸郎¹ 松本 真佑¹ 中村 匡秀¹

概要: ホームネットワークシステム (HNS) とは, 様々な家電やセンサを宅内のネットワークに接続し付加価値を提供するサービスである. HNS はこれまで多くの研究開発が進められ, 最近では商用のシステムも登場している. しかしながら, これらの従来システムの多くは, 家電やセンサ等のデバイス, デバイスを利用するサービス, サービスを管理するホームサーバ, サービスを操作するためのユーザ・インターフェースの互いの依存度が高い. そのため, HNS の規模が大きくなるにつれて, 拡張や再利用が困難になるという課題がある.

我々の研究グループでは, HNS の構成要素をすべて「サービス」とみなし, サービス指向アーキテクチャ (SOA) の考え方を利用してこれらのサービスを連携する, という形で HNS を構成する研究を行っている. 本稿では, 我々の最新の研究成果を組み合わせた, 環境センサといくつかの家電から構成されるポータブルホームネットワークシステムを構築し, 様々なコンテキストを用いたサービスをその場で作成・実行するデモンストレーションを行う. また, バーチャルエージェントを用いた対話型操作システムを利用した家電操作やサービス実行も行う.

Service-Oriented Home Network System Using Cloud Technology and Web of Things

HIROKI TAKATSUKA¹ SATOSHI TAKATORI¹ HIROYASU HORIUCHI¹ SACHIO SAIKI¹
SHINSUKE MATSUMOTO¹ MASAHIDE NAKAMURA¹

1. はじめに

家電やセンサなどを家庭内のネットワークに接続することで, 多種多様なサービスを提供する**ホームネットワークシステム (HNS)** に注目が集まっている. HNS では, ネットワークに接続されたテレビやエアコン, 扇風機, 照明機器, 温度計などの機器を使用することで, 複数機器の連携制御 [1] や, 状況に応じた機器の制御を行うコンテキストウェアサービス [2] 等の付加価値サービスを提供できる. 最近では, HNS を利用したアプリケーションが社会に登場している. しかしながら, これらの従来システムの多くは, 家電やセンサ等のデバイス, デバイスを利用して機能を提供するサービス, サービスを管理するホームサーバ, サービスを操作するためのユーザ・インターフェースが互

いに密接に結びつき, 特定のベンダに依存している. そのため, HNS の拡張や, 他ベンダのサービスやシステムとの連携が困難になるという課題がある.

そこで我々の研究グループでは, **サービス指向アーキテクチャ (SOA)** の考え方を適用し, HNS を構成するすべての要素を「サービス」とみなし, サービスの連携という形で HNS を構成する研究を行っている. 本稿では, 我々の最新の研究成果から, 以下の3つの成果を組み合わせたデモンストレーション発表を行う.

(1) マルチベンダサービスを実現するクラウド型ホームネットワークシステム

従来の HNS では, 宅内に設置される HNS と HNS 上で動作するサービスが密結合している. そのため, ユーザは自分の好きな機器やサービスを自由に選ぶことが難しい. 本稿では, 複数ベンダの HNS を収容できるクラウド型ホームネットワークシステムのアーキテクチャについて

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

紹介する。提案アーキテクチャでは、HNS の操作やデータを標準的なサービスとしてクラウド上で管理する。ベンダの提供するサービスはクラウドを介してのみ HNS にアクセスできるようにする。これにより、HNS とサービスの疎結合が実現され、ユーザやベンダにとってより選択肢が多く柔軟な HNS が実現される。

(2) Web of Things を利用したコンテキストウェアサービスの作成・管理

我々は、様々な Web of Things (情報システム、センサ、家電など) を利用したコンテキストウェアサービスを、体系的に作成・管理するためのフレームワーク **RuCAS(Rule-based management framework for Context-Aware Service)** を提案している。提案フレームワークは、各コンテキストをネットワーク上の既存の Web サービスから取得されるデータに基づいて定義する。また全てのコンテキストウェアサービスを (イベント、コンディション、アクション) の組である **ECA 規則** で定義する。ここで、イベントはサービスの実行契機となるコンテキストの成立、コンディションはサービスの実行可能条件、アクションは制御に相当する Web サービスである。このように、全てのコンテキストウェアサービスは、シンプルで統一的な規則で管理される。

(3) HNS におけるバーチャルエージェント・サービスフレームワーク

HNS では、従来の専用リモコンに代わる、新たな操作インタフェースの開発が求められている。バーチャルエージェントを HNS の操作インターフェイスに用いる HNS-バーチャルエージェントユーザインタフェース (HNS-VAUI) はその中の一つである。HNS-VAUI では、ユーザはバーチャルエージェントとの自然なインタラクションを通じて、家電操作等を簡単に実行することができる。我々の研究室では、HNS-VAUI を実現するために、MMDAgent を用いた HNS-VAUI を開発・運用している。しかしながら、MMDAgent を用いた従来の実装は仕様に強く依存しているため、ユーザとの高度なインタラクションが実現できない。そこで、新たに HNS 操作のためのサービス指向 MMDAgent 制御フレームワークを提案し、提案フレームワークに基づく HNS-VAUI を実際の HNS に実装する。提案フレームワークでは、HNS-VAUI システムの制御をキャラクターの動作などの制御機能と HNS 操作に必要な外部 Web サービスの実行ロジックの制御機能に分離する。これらを Web サービス化することにより、外部からの入力に応じた高度でステートフルなインタラクションを実現できる。

クラウド型 HNS に提供するサービスとして、RuCAS によるコンテキストウェアサービス作成・管理サービスと、HNS-VAUI による HNS の操作インタフェースを提供することで、3つの技術の連携を行う。これにより、コンテキス

トウェアサービスのアクションに HNS-VAUI を適用して、コンテキストに応じた家電制御をバーチャルエージェントに行ってもらおう、といったことが可能となる。

デモンストレーションでは、環境センサといくつかの家電から構成されるポータブルホームネットワークシステムを構築し、デモンストレーション会場から取得される様々なコンテキストを用いたサービスをその場で作成・実行する。また、「初音ミク」をバーチャルエージェントとした対話型操作システムを利用した家電操作やサービス実行も行う予定である。

2. 準備

2.1 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收容して、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)**[1][4] 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をネットワークに接続し、宅内外から遠隔制御・監視したり、連携制御するといったサービスを実現するシステムである。HNS はスマートシティ、エネルギー、防犯、医療・介護等様々な分野で応用が期待されており、いくつかのサービスは市場に登場している。スマートフォンを用いてシャッターを遠隔開閉したり [5]、エアコンを自動的に節電運転するサービス [6] が存在する。

2.2 HNS の課題

現状では、各サービスベンダが提供サービスに適合した機器を用いて、ユーザの家庭に HNS を構築する仕組みになっている。つまり、機器とサービスが密に結合しているということである。特定の家電構成と密に結合した HNS サービスは再利用性が低いため、サービス開発・運用のコストが高いという問題がある。

また、一般に HNS サービスは主に人間が単一の機器のみを制御することを目的としており、他のアプリケーションから呼び出したり、ネットワークを介した他機器との連携・統合は非常に限定されており、相互接続性が低い。

2.3 CS27-HNS

上記の課題を解決するため、我々の研究室では、サービス指向アーキテクチャ (SOA) の考え方を HNS に適用し、各家電・センサの機能を Web サービス [7] として利用できる HNS 環境 **CS27-HNS** [1] を開発している。CS27-HNS では機種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 REST または SOAP 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://cs27-hns/TVService/setChannel?channel=6>

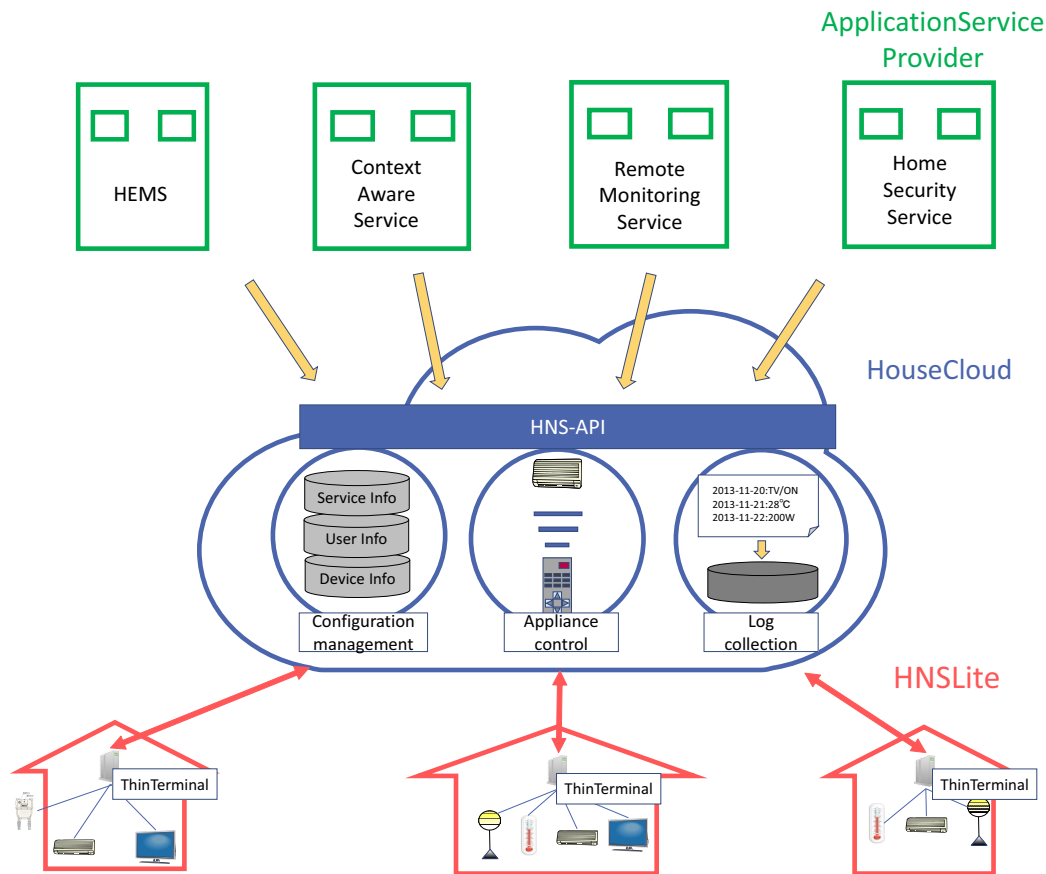


図 1 クラウド型 HNS の全体アーキテクチャ

といった URL にアクセスすることで実現できる。

CS27-HNS では、SOA を適用することにより、どのようなベンダの家電または旧来の家電であっても HNS に組み込むことが可能である。また、相互接続性も確保されており、PC やスマートフォンなど、どのようなデバイスからでも家電操作やセンサ値の取得を行うことができる。

本稿では、この CS27-HNS に基づいて行われた研究の中から、3つの研究を取り上げ、その連携を行う。

3. クラウドと Web of Things を活用した HNS

3.1 マルチベンダサービスを実現するクラウド型ホームネットワークシステム

HNS の実現に対する研究開発はこれまで多くの報告がある。しかしながら、一般家庭への幅広い普及には至っていない。その一要因として、現状の HNS では、HNS の構成要素（家電機器やセンサ、ホームサーバ、通信プロトコルなど）と、ベンダが提供する HNS サービスとが密結合していることが挙げられる。密結合した HNS では、単一宅内において複数のベンダが提供する HNS サービスを同時に利用することは容易ではない。一方、HNS 環境を制御するホームサーバを複数設置することで、マルチベンダ HNS 環境を実現することも可能ではあるが、この方法は導入コ

ストが高く、また異なるベンダの HNS サービス間で特定の家電機器を共有することもできない。これらの問題により、HNS を導入するとユーザは購入家電の選択肢が限られるのみならず、利用する HNS サービスの選択肢も特定のベンダが提供しているものに限られてしまう。その結果、ユーザは HNS の導入をためらうため、普及が進まない。

我々は現状の HNS の問題を解決するために、先行研究 [8] において、クラウド型 HNS を提案している。クラウド型 HNS とはクラウドのコンセプトに基づいた HNS の新たな形態である。クラウド型 HNS とは、従来 HNS の課題を解決するための新たな HNS の形態である。HNS が提供する基本的な機能をサービスとして抽象化し、クラウド上で管理することで HNS の構成要素とサービスとの疎結合を図る。これにより、ユーザは手軽にマルチベンダの HNS サービスを利用できるようになるほか、HNS 導入コストや管理コストの低減が見込める。

クラウド型 HNS の全体アーキテクチャを図 1 に示す。クラウド型 HNS は HNSLite、HouseCloud、ApplicationServiceProvider の 3 つ要素から構成されている。

HNSLite

HNSLite (図下段) は従来の HNS 環境からホームサーバを取り除き、代わりに ThinTerminal が設置された軽

量の HNS 環境である。ThinTerminal は HouseCloud からの命令に基づき、各家庭のログデータの取得や家電機器の実行を行う。ThinTerminal はインターネットと宅内ネットワークとのゲートウェイであり、宅内機器と HouseCloud を接続するための最低限の機能だけを持つ。これにより、宅内には物理空間との境界になる低レベルなデバイスしか存在しないことになる。従来のホームサーバで行われていた高級なサービスは、全てクラウド上で実行される。

HouseCloud

HouseCloud (図中央) はクラウド型 HNS の中心的な役割を果たすレイヤである。HouseCloud は HNS サービスを実現するために、構成情報の管理、家電機器の制御、住宅ログの収集の 3 つの機能を持つ。

構成情報の管理とは、HouseCloud に接続される HNS 環境の静的な構成情報を管理する機能である。管理対象となる構成情報には、宅内の機器一覧や住民構成、利用しているサービスなど様々なマスタ情報が含まれる。**家電機器の制御**とは、HNSLite に設置された宅内機器の操作を、抽象化したサービスとして ASP に提供する機能である。HNS サービスからの機器操作リクエストを受信した場合に、操作対象の機器を特定し、具体的な操作コマンドに変換して HNSLite に指示する。**住宅ログの収集**とは、HNSLite に設置されたセンサの情報や機器の状態ログを収集し、蓄積する機能である。膨大な量のログ情報を一元管理し、HNS-API 経由で HNS サービスにログ情報を提供する。

ApplicationServiceProvider(ASP)

ASP (図上段) とは、HNSLite 用の HNS サービスの開発・提供を行うプロバイダのことである。HNS サービスに必要な情報や操作は、HouseCloud の公開している HNS-API を通じて実行する。HNS ベンダに強く依存していた HNS 固有の機能や処理は、全て HouseCloud 上でサービスとして抽象化されるため、ベンダに依存せず HNS サービスの開発が可能となる。ユーザは HouseCloud に登録された様々なサービスから、自宅の HNS に適したものを自由に選ぶことが出来る。

3.2 Web of Things を利用したコンテキストウェアサービスの作成・管理

Web of Things とは、日常的に使用するあらゆる機器やモノが Web に繋がり、API を通じて利用可能になる、という構想である。Web of Things 実現に向けて、近年、様々な計算資源をネットワーク上のサービスとして利用するクラウド技術 [9] や、機器同士が人間の介在なしに通信し動作する M2M 技術 [10] が発展してきている。これにより、現実世界に関する多種多様なデータを Web サービス [7] や Web-API 経由で取得することが技術的に可能になっ

てきている。現実世界のデータの例として、HNS を利用した気温や湿度、人の存在、家電の状態といった宅内情報が挙げられる。これらのデータから現実世界の状況情報である **コンテキスト** を判断し、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、**コンテキストウェアサービス** [11] が実現可能である。HNS においてもコンテキストウェアサービスを利用することで、周囲の状況に応じたより付加価値の高い HNS サービスを提供することができる。しかしながら、従来研究では、センサなどコンテキストの情報源と、作成されたコンテキスト、さらにそのコンテキストを用いたサービス (家電操作サービスなど) との結びつきが強いため、他のコンテキストやサービスとの連携を行うことが困難である。

この問題に対して、Web サービスの利用で得られる多種多様なデータを利用することで、Web を跨いだ大域的なコンテキストウェアサービスを実現できる。しかし、連携する Web サービスが増えるほど、サービス間の関係が複雑になるため、コンテキストの情報源となる Web サービス、それに基づくコンテキスト、さらにコンテキストウェアサービスを体系的に作成・管理できる仕組みが必要である。

そこで、我々は [12] において、Web サービスに基づいたコンテキストウェアサービスを体系的に作成・管理するフレームワーク **RuCAS (Rule-based management framework for Context-Aware Service)** を提案している。

RuCAS は、Web 上に分散する様々な Web サービスから情報を取得し、その情報に基づいたコンテキストを容易に定義・管理する手段を提供する。またコンテキストウェアサービスを、(イベント、コンディション、アクション) の組である **ECA 規則** で定義する。ECA 規則とは、(イベント、コンディション、アクション) の組で任意のコンテキストウェアサービスを定義する規則であり、RuCAS の重要な設計思想を反映したものである。

RuCAS においては、コンテキスト、イベント、コンディション、アクションは以下のように定義される。

コンテキスト：異種分散 Web サービスから取得した情報に基づいて定義される状況情報。あるいは、既存のコンテキストを組み合わせで定義できる状況情報。

イベント：コンテキストウェアサービスの実行契機となるコンテキスト。

コンディション：コンテキストウェアサービスの実行に必要な条件。1 つ以上のコンテキストで定義される。

アクション：コンテキストウェアサービスで実行される制御。1 つ以上の Web サービスで定義される。

ECA 規則は次のように定義される。

ECA 規則： c_1, c_2, \dots を任意のコンテキスト、 a_1, a_2, \dots を任意の Web サービス呼び出しとする。このとき、1 つの ECA 規則は、 $r = [E : c_i, C : \{c_{j_1}, c_{j_2}, \dots, c_{j_m}\}, A :$

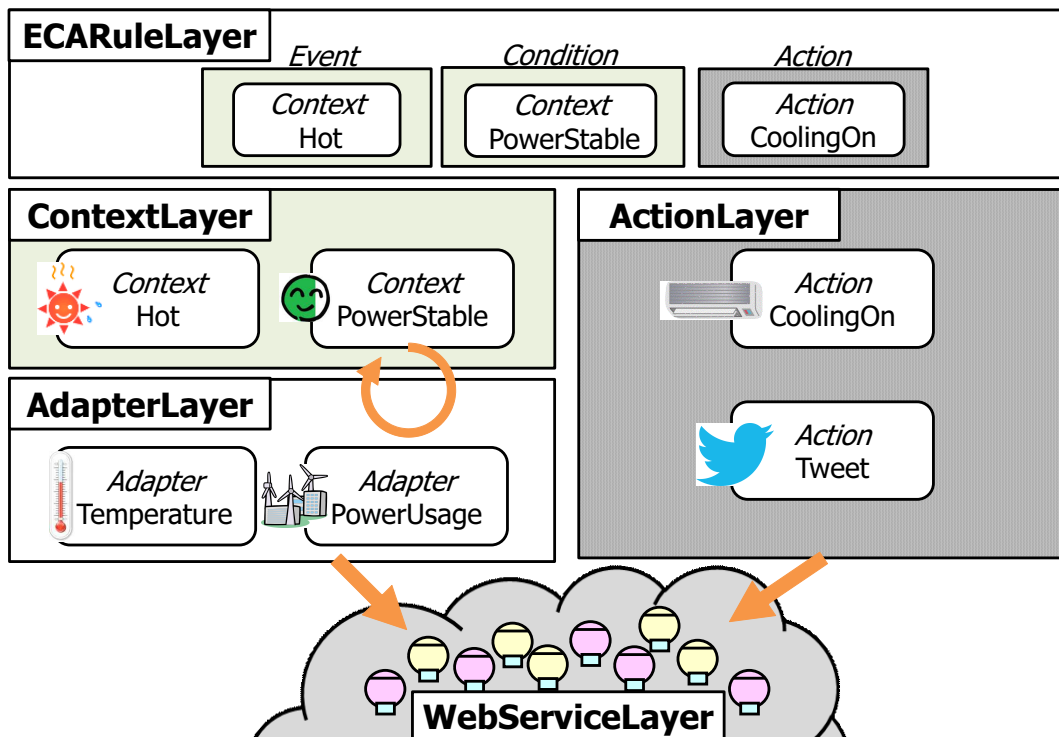


図 2 RuCAS のアーキテクチャ

$\{a_{k_1}, a_{k_2}, \dots, a_{k_n}\}$ で定義される．ここで、 E はイベント、 C はコンディション、 A はアクションである． r のイベント E の c_i が偽 (false) から真 (true) になったとき「イベント E が発生する」という． E が発生した際、コンディション C のコンテキスト $c_{j_1}, c_{j_2}, \dots, c_{j_m}$ が全て真の時、 r は実行される． r が実行されると、アクション A の全ての Web サービス $a_{k_1}, a_{k_2}, \dots, a_{k_n}$ が実行される．

図 2 に RuCAS のアーキテクチャを示す．RuCAS は、ECA 規則を既存部品の組み合わせで効果的に組み立てられるように、責務を分担した 5 つのレイヤ (WebService Layer, Adapter Layer, Context Layer, Action Layer, ECARule Layer) で構成している．以下に各レイヤについて示す．

WebService Layer : コンテキストアウェアサービスの入出力として利用する既存の Web サービスのレイヤである．

Adapter Layer : 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 Web-API を統一するアダプタを作成・管理するレイヤである．

Context Layer : アダプタを介して得られた Web サービスの情報に基づいてコンテキストを定義し、一元管理するレイヤである．

Action Layer : コンテキストアウェアサービスの出力となる Web サービスをラップし、アクションとして管理するレイヤである．

ECARule Layer : Context Layer で登録したコンテキストと、Action Layer で登録したアクションを利用して、ECA 規則を定義し、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．各 ECA 規則の作成は次のような手順で行われる．(1) Context Layer からコンテキストを 1 つ選んで、イベントを定義する．(2) Context Layer からコンテキストを 1 つ以上選んで、コンディションを定義する．(3) Action Layer からアクションを 1 つ以上選んで、アクションを定義する．RuCAS を用いたコンテキストアウェアサービスの作成の流れを以下に示す．

アダプタの作成 : 既存の Web サービスを利用して、必要なアダプタを作成する．

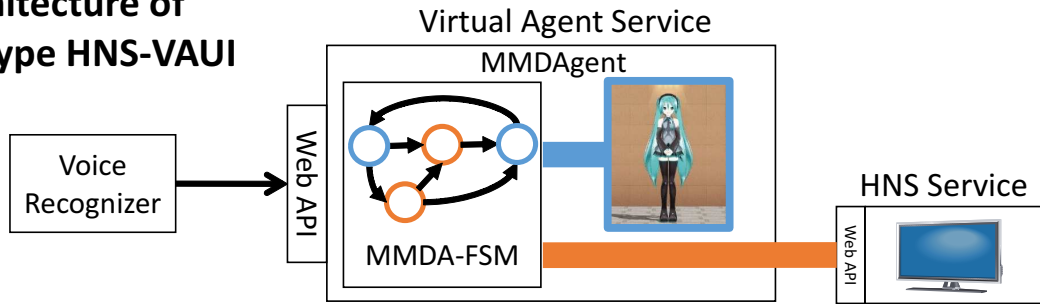
コンテキストの作成 : 作成したアダプタを利用して、必要なコンテキストを定義する．

アクションの作成 : コンテキストアウェアサービスで実行したい Web サービスの API を利用して、アクションを作成する．

ECA 規則の作成 : 作成したコンテキストとアクションを利用して、コンテキストアウェアサービスを ECA 規則として作成する．

RuCAS では、アダプタ、コンテキスト、アクション、ECA 規則のそれぞれは独立に作成、利用できるため、作成済みの要素を再利用して、効率よく、コンテキストアウェアサービスを作成できる．

Architecture of Prototype HNS-VAUI



Architecture of Proposed HNS-VAUI

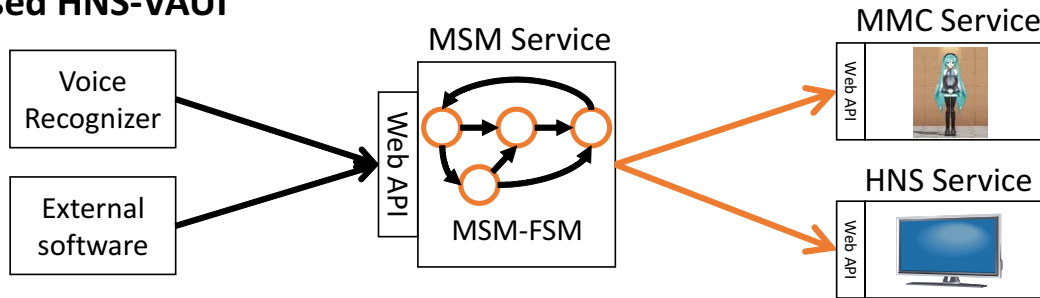


図 3 HNS 操作のためのサービス指向 MMDAgent 制御フレームワークのアーキテクチャ

3.3 HNS におけるバーチャルエージェント・サービスフレームワーク

HNS における重要な課題として、ユーザインタフェース (UI) のユーザビリティの向上があげられる。HNS では日常的に操作するという特性上、ユーザにとって学習コストが低く操作しやすい、かつ簡単で便利であるユーザフレンドリな UI がより強く求められている。HNS に適したこのような UI として、**バーチャルエージェント (VA)** を UI に用いた VAUI がある。VA は、ユーザと画面内のキャラクターとのインタラクションを通じてシステムの操作を行う UI である。親近感や対話の現実感から、無機質で情報量の少ない従来の音声インタフェースに代わる、次世代の UI として注目を集めている [13][14]。HNS における機器の操作に VAUI を用いることで、人と会話しているような自然なインタラクションを通じて HNS 機器を操作することができる。また、機器の種類や操作が多数ありシステムからユーザへの情報量が増える場合において、音声フィードバックだけではなく操作可能な機器や操作内容を画面に表示することで、フィードバックの長大化を防ぎ、ユーザに優しいフィードバックを行うことが可能である。

我々は、HNS の UI として VAUI を適用した、**HNS-バーチャルエージェントユーザインタフェース** (以下、HNS-VAUI) を提案し、CS27-HNS において、MMDAgent[15] を利用したプロトタイプを開発・運用している [16]。しかしながら、プロトタイプでは、VA の制御は MMDAgent 独自の実装仕様に強く依存しており、キャラクターの制御とシステムのロジックの制御が分離できないことや、実行でき

る機能が少ないという問題点があった。そのため、ユーザの要求を満たす高度なインタラクションを行うことができず、ユーザビリティが高い便利な HNS-VAUI を実現するには至っていない。

そこで、我々は [17] において、HNS 操作のためのサービス指向 MMDAgent 制御フレームワークを提案し、実装した。提案フレームワークでは、サービス指向アーキテクチャに基づき、キャラクターの動作の制御と外部 Web サービスの実行の制御をそれぞれ別の機能として捉え、キャラクターの動作の制御を行う **MMCSERVICE** (Miku Miku Command Service) と、外部 Web サービスを実行し、ロジックの制御を行う **MSMSERVICE** (Miku State Machine Service) を新たに作成し、これらの機能を分割する。図 3 にアーキテクチャを示す。MMCSERVICE は、キャラクターの動作 (発話・モーション) の実行を管理する Web サービスである。他のアプリケーションは API を通じ MMCSERVICE を呼び出すことで、キャラクターに動作をさせることができる。MSMSERVICE は、記述能力を高めた新たなオートマトンを管理し、入力に応じて、外部 Web サービスの実行をステートフルに制御する Web サービスである。MSMSERVICE が管理するオートマトンには、プロトタイプでは実現できなかったインタラクションを実行する機能を追加し、また、外部からオートマトンの編集を可能にする。MSMSERVICE から、ユーザからの入力に対して MMCSERVICE を用いたキャラクターの動作や HNS Service を用いた家電操作を実行することでインタラクションを構成し、HNS-VAUI を実現する。また、外部アプリケーションから MSMSERVICE を呼

び出すことで、ユーザからの入力以外をトリガとしたインタラクションも実行することができる。

以下に、この2つのWebサービスの機能について述べる。

MMCSERVICE

MMCSERVICEは、MMDAgentにおけるキャラクタの発話・モーション機能をサービスとして捉え、プロトタイプで実装したVirtual Agent ServiceをラップしたWebサービスである。キャラクタの発話とモーションをWeb APIとして外部から制御することが可能なWebサービスである。MMCSERVICEは内部でVirtual Agent Serviceを呼び出し、MMDAgentと通信することで、画面内のキャラクタの発話とモーションの制御を行う。MMCSERVICEは、内部でキャラクタの発話やモーションの状態を管理しているため、外部サービスはMMCSERVICE内部の状態を考慮する必要はなく、任意のタイミングで各APIを実行することでキャラクタに動作させることができる。

MSMSERVICE

MSMSERVICEはオートマトンを持っており、状態に応じたステートフルなインタラクションを構成するWebサービスである。入力に対し、ロジックの制御のもと、ステートフルにWebサービスを実行することで、ユーザとシステムのインタラクティブなインタフェースを実現する。MSMSERVICEのWeb APIを用いて、外部からMSMSERVICEのオートマトンへの入力を行うことができる。MSMSERVICEは入力を受けると遷移を行い、同時にその遷移に登録されているシステムの動作(MMCSERVICEを用いたキャラクタの制御やHNS Serviceを用いた家電操作)を実行する。また、オートマトンを編集するWeb APIを持ち、インタラクションを変更することも可能である。

提案フレームワークに基づいて実装することで、HNSをより便利で快適に操作可能なインタフェースとしてのHNS-VAUIを実現できる。また、インタラクションを好みに合わせてカスタマイズし、各家庭や各個人に合わせた適応型のインタフェースを実現することも可能である。さらに、3.2であげたRuCASと連携したサービスも実行可能である。

3.4 三研究の連携

これまでに紹介した3つの新しい技術を連携し、新しい付加価値サービスを実現する。クラウド型HNSに提供するサービスとして、RuCASによるコンテキストウェアサービス作成・管理サービスと、HNS-VAUIによるHNSの操作インタフェースを提供することで、HNSとRuCASのみを組み合わせた場合や、HNSとHNS-VAUIのみを組み合わせた場合よりも高度なサービスの提供を行うことができる。例えば、コンテキストウェアサービスのアク

ションにHNS-VAUIを適用して、コンテキストに応じた家電制御をバーチャルエージェントに行ってもらおう、といったことが可能となる。

4. デモンストレーション

本節では、デモセッション内で行うデモンストレーションについて説明する。我々は、展示ブースを簡易HNSとし、以下の5つのシナリオのデモンストレーションを行う。

HNS Controllerを用いた家電操作

HNS Controllerと呼ばれるWebアプリケーションを用いて家電を操作する。HNS Controllerは1つの端末からHNS上のあらゆる家電を操作することができる。

HNS-VAUIを用いた家電操作

画面上のキャラクタとの対話を通じて家電を操作する。従来のリモコンによる家電操作に比べ、ユーザにとって直感的な操作が可能である。

HNS-VAUIを用いた外部情報取得

画面上のキャラクタとの対話を通じて、外部の情報を取得し、ユーザにフィードバックする。HNS内のサービスのみでなく、外部で公開されているサービスとの連携が可能である。

RuCASによるコンテキストウェアサービス

RuCASを用いて実際にECAに基づいたコンテキストウェアサービスを作成する。GUI上で容易にサービスを作成することができる。

RuCASとHNS-VAUIとの連携

実際にコンテキストウェアサービスを発火させ、HNS-VAUIを用いたアクションを実行する。コンテキストの変化に応じた付加価値サービスをキャラクタを通じてユーザに提供することができる。

これらのデモンストレーションを通じ、これからの新しいHNSのあり方を示すとともに、未来の住居についての一つの可能性を示す。

5. おわりに

本稿では、我々の研究室においてサービス指向アーキテクチャを利用して構築したCS27-HNSに基づいて行われた最新の研究成果から、(1) マルチベンダサービスを実現するクラウド型ホームネットワークシステム、(2) Web of Thingsを利用したコンテキストウェアサービスの作成・管理、(3) HNSにおけるバーチャルエージェント・サービスフレームワークの3つの成果を紹介した。3つ研究の連携により、それぞれを個別に利用した場合に比べて、より高度なサービスの提供を行うことができる。また、デモンストレーション発表で行う予定である3つの成果を組み合わせた5つのシナリオについて述べた。

6. 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 26280115, 基盤研究 C 24500258, 若手研究 B 26730155）の研究助成を受けて行われている。

参考文献

- [1] Nakamura, M., Tanaka, A., Igaki, H., Tamada, H. and Matsumoto, K.: 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).
- [2] Gouin-Vallerand, C., Abdulrazak, B., Giroux, S. and Dey, A. K.: A Context-aware Service Provision System for Smart Environments Based on the User Interaction Modalities, *J. Ambient Intell. Smart Environ.*, Vol. 5, No. 1, pp. 47–64 (2013).
- [3] 石田建一, 佐藤 誠: HEMS による家庭用冷暖房・照明エネルギーの削減効果の検証, 日本建築学会環境系論文集, No. 595, pp. 57–64 (2005).
- [4] Li, X. and Zhang, W.: The design and implementation of home network system using OSGi compliant middleware, *Transactions on Consumer Electronics*, Vol. 50, No. 2, pp. 528–534 (2004).
- [5] iTamaHome, <http://www.tamahome.jp/i-tamahome>.
- [6] スマート HEMS, <http://www2.panasonic.biz/es/densetsu/aiseg/features/index.html>.
- [7] Alonso, G., Casati, F., Kuno, H. and Machiraju, V.: *Web Services, Data-Centric Systems and Applications*, Springer Berlin Heidelberg (2004).
- [8] 鷹取敏志, 枳本真佑, 佐伯幸郎, 中村匡秀: マルチベンダサービスを実現するクラウド型ホームネットワークシステムの提案, 電子情報通信学会技術報告, Vol. 113, No. 303, 電子情報通信学会, pp. 53–58 (2013).
- [9] Velte, T., Velte, A. and Elsenpeter, R.: *Cloud Computing, A Practical Approach*, McGraw-Hill, Inc., 1st edition (2010).
- [10] Wu, G., Talwar, S., Johnsson, K., Himayat, N. and Johnson, K.: M2M: From mobile to embedded internet, *IEEE Communications Magazine*, Vol. 49, No. 4, pp. 36–43 (2011).
- [11] Cohen, N., Black, J., Castro, P., Ebling, M., Leiba, B., Misra, A. and Segmuller, W.: Building context-aware applications with context weaver, IBM Research Division (2004).
- [12] 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 (2013).
- [13] Ochs, M., Pelachaud, C. and Sadek, D.: An empathic virtual dialog agent to improve human-machine interaction, *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, Vol. 1, pp. 89–96 (2008).
- [14] Cassell, J.: Embodied conversational interface agents, *Communications of the ACM*, Vol. 43, No. 4, pp. 70–78 (2000).
- [15] MMDAgent Project Team: MMDAgent - Toolkit for Building Voice Interaction Systems, Nagoya Institute of Technology.
- [16] Soda, S., Nakamura, M., Matsumoto, S., Izumi, S., Kawaguchi, H. and Yoshimoto, M.: Implementing Vir-

- tual Agent as an Interface for Smart Home Voice Control, *In Asia-Pacific Software Engineering Conference (APSEC2012)*, pp. 342–345 (2012).
- [17] Horiuchi, H., Saiki, S., Matsumoto, S. and Nakamura, M.: Implementing Service Framework for agent-based interactive interface in Home Network System, *IEICE Technical Report*, pp. 61–66 (2014).