

## ホームネットワークシステムにおけるプレゼンスセンシングのための サービスフレームワークの提案

檜尾 勇樹<sup>†</sup> 渡邊 雄一<sup>†</sup> まつ本真佑<sup>†</sup> 佐伯 幸郎<sup>†</sup> 中村 匡秀<sup>†</sup>

<sup>†</sup> 神戸大学 〒657-8501 神戸市灘区六甲台町1-1

E-mail: <sup>†</sup>{kashio,nabe}@ws.cs.kobe-u.ac.jp, <sup>††</sup>{shinsuke,masa-n}@cs.kobe-u.ac.jp,

<sup>†††</sup>sachio@carp.kobe-u.ac.jp

**あらまし** ホームネットワークシステム (HNS) における効率的なプレゼンスセンシングを実現するためには、様々なアプリケーションが共通に利用できる経済的で拡張可能なシステムが必要である。本論文では、複数の赤外線センサを利用して宅内の様々な場所における人のプレゼンスを取得・提供するプレゼンスセンサ・サービスフレームワーク (PSS フレームワーク) を提案する。提案フレームワークは、プレゼンスセンサ・デバイス (PSD) とプレゼンスセンサ・ターミナル (PST), さらに、プレゼンスセンサ・アグリゲータ (PSA) から構成される。PST は接続された PSD の値変化を監視し PSA に通知する。PSA は複数の PST からの通知を管理し、各センサの近傍のプレゼンスを推定する。この推定処理はプレゼンスセンサ・サービス (PSS) として公開され、様々な HNS アプリケーションに利用される。PSS フレームワークでは、PST を必要に応じて増やすことでセンシング位置の動的な追加に対応できる。また、PSA をクラウドに置くことで、複数 HNS のプレゼンスを統合管理することも可能となる。本稿では、提案フレームワークを Phidget センサおよび Java Web サービスを利用して試作した。また、プレゼンス可視化するアプリケーションを開発し動作試験を行った。

**キーワード** プレゼンスセンシング, ホームネットワークシステム, コンテキスト推定, Web サービス, 赤外線センサ

## A Service Framework for Presence Sensing in Home Network System

Yuki KASHIO<sup>†</sup>, Yuichi WATANABE<sup>†</sup>, Shinsuke MATSUMOTO<sup>†</sup>, Sachio SAIKI<sup>†</sup>, and Masahide  
NAKAMURA<sup>†</sup>

<sup>†</sup> Kobe University Rokkoudaityou 1-1, Nada-ku, Kobe, Hyogo, 657-8501 Japan

E-mail: <sup>†</sup>{kashio,nabe}@ws.cs.kobe-u.ac.jp, <sup>††</sup>{shinsuke,masa-n}@cs.kobe-u.ac.jp,

<sup>†††</sup>sachio@carp.kobe-u.ac.jp

**Abstract** To achieve efficient presence sensing within home network system (HNS), an inexpensive and elastic system that can be shared by various HNS applications is required. This paper presents *Presence Sensor Service Framework*, which manages human presence information in various places using multiple infrared sensors. The framework consists of Presence Sensor Device (PSD), Presence Sensor Terminal (PST) and Presence Sensor Aggregator (PSA). A PST monitors PSDs and notifies a PSA of the status update. Upon the notifications from PSTs, the PSA estimates human presence around each PSD. The estimation process is deployed as Presence Sensor Service (PSS), which are used by various HNS applications. The proposed framework can add or change sensing places dynamically by adding PST as necessary. It can also integrate presences of multiple HNSs by deploying PSA on an external cloud. We prototype the proposed framework with Phidgets Motion Sensor and Java Web services. We conduct a preliminary evaluation by developing a presence visualization application.

**Key words** presence sensing, home network system, context estimation, Web services, infrared sensors

## 1. はじめに

特定の位置に人やモノが居るかどうかを判断する**プレゼンスセンシング**は、機器の自動制御やスマートサービスに幅広く応用される技術である。自動ドアや自動照明をはじめとして、家電の節電制御 [1] や、オフィスにおける機器や空調の効率運転 [2]、ビルの廊下の照明制御 [3] など様々な局面で利用されている。我々の研究グループでは**ホームネットワークシステム (Home Network System, HNS)** [4] の研究を続けており、一般住宅内の HNS に適したプレゼンスセンシング手法について検討を行っている。

屋内におけるプレゼンスセンシングのソリューションとして、様々なセンサを用いたシステムが提案されている。例えば、赤外線センサを用いるもの [5] や磁気センサを用いるもの [6]、圧力センサ [7]、超音波センサ [8]、無線 LAN [9]、シート型センサ [10] などが存在する。

しかしながら従来のシステムは、プレゼンスセンサが 1 つの機器やサービス内で閉じていたり、高価なデバイスや大掛かりなインフラを必要としたりするため、一般住宅内の HNS に導入するには敷居が高い。さらに、プレゼンスの検知位置を後から動的に追加・削除したり、複数の HNS のプレゼンスを連携したスマートシティサービスのようなものは想定していない。

そこで本論文では、比較的安価な赤外線センサを複数連携して、HNS 内の様々な場所における人のプレゼンスを取得・提供する枠組み**プレゼンスセンサ・サービスフレームワーク (Presence Sensor Service Framework, PSS フレームワーク)**を提案する。

PSS フレームワークは、物理的なセンサデバイスである**プレゼンスセンサ・デバイス (Presence Sensor Device, PSD)**と、PSD を束ねる**プレゼンスセンサ・ターミナル (Presence Sensor Terminal, PST)**、さらに、センサデータを集約してプレゼンスを管理する**プレゼンスセンサ・アグリゲータ (Presence Sensor Aggregator, PSA)**から構成される。PST は接続された PSD の値を監視し、値の変化がある閾値を超えた際に PSA に通知する。PSA は PST からセンサ値変化の通知を受け取ると、当該 PSD の ID とその値、通知時刻を記録する。PSA はそれぞれの PSD の周辺に人がいるかどうかを、最終通知時刻から推定・計算するアルゴリズムを実装しており、これを Web サービス (**プレゼンスセンサ・サービス (Presence Sensor Service, PSS)**と呼ぶ)として公開する。

これによって、様々な HNS サービス、アプリケーションは、PSS を介して宅内のプレゼンス情報を取得・利用できるようになる。PSS フレームワークでは、PST および PSD を必要に応じて追加・削除することで、プレゼンスの検知位置を動的に構成することができる。また、PSA をカスケードしてクラウド上に置くことで、複数の HNS におけるプレゼンス情報を連携したスマートシティサービスを実現することも可能である。

また本論文では、PSS フレームワークのプロトタイプを作成した。PST は Phidgets 社の赤外線センサおよびインターフェースキットを利用して実装した。PSA は Java で実装し、PSS を

Apache Axis2 Web サービスとしてデプロイした。HNS 側のアプリケーションとして、Google Chart API を用いた簡単な可視化アプリケーション作成し、開発したシステムが正常に動作していることを確認した。

## 2. 準備

### 2.1 ホームネットワークシステム (HNS)

**ホームネットワークシステム (HNS)** は、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である。TV や DVD、照明、エアコン、カーテン、扇風機、空気清浄機等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。

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** を開発している [11]。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://cs27-hns/TVService/setChannel?channel=6> といった URL にアクセスするだけで良い。

CS27-HNS を用いた付加価値サービスの例として、宅外からの機器制御、消費電力振り返り [12]、電力ピークカット [13]、複数機器の連携サービス [14]、音声操作 [15]、パーソナルリモコン [16] 等を開発している。また、温度センサや照度センサ等の環境センサを、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) [17] や、センサを利用したコンテキスト・ア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) [18] も提案している。

### 2.2 HNS におけるプレゼンスセンシング

プレゼンスセンシングとは、特定の位置に人やモノが存在するかどうかを判断する技術であり、様々な分野で応用されている [19] [20]。本稿では特に **HNS 環境における人のプレゼンスセンシング**に焦点を絞る。HNS 環境における人のプレゼンスセンシングは、**在室判定** (人が部屋にいる/いない) や**住人位置推定** (住人がどこにいるか)、**到来イベント検知** (ドアに近づいた、玄関に入った、機器の前に来た) 等、宅内の人の動きや居場所を感知する有効な手段を与える。こうした人の動きや居場所の情報は、より人目線に立った HNS サービス・アプリケーションの実現に役立てられる。

HNS 用途のプレゼンスセンシング・システムを実現するにあたっては、一般住宅に導入することや、HNS サービスでの利用形態や将来に向けた拡張を考慮しなければならない。そこで本稿では特に、下記の 4 つの要件を洗い出した。

**要件 R1:** 比較的安価なデバイスで実現できること。

**要件 R2:** 宅内でセンシングする場所を動的に追加、削除できること。

**要件 R3:** 様々な HNS サービス・アプリケーションからセンサ情報を共有できること。

**要件 R4:** 一軒の家に閉じない拡張が可能であること。

ここで、要件 R1 は一般住宅に導入するためコストを低くした

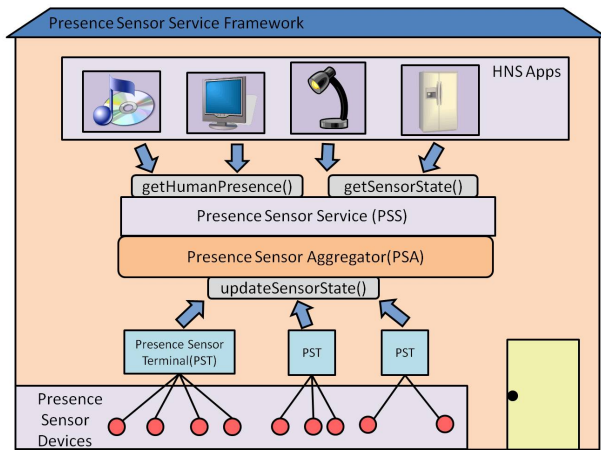


図1 提案フレームワークのアーキテクチャ

いこと、要件 R2 は家毎に異なる間取りや機器に柔軟に対応できること、要件 R3 はプレゼンスセンシングの利用効率・再利用性をあげること、要件 R4 は将来的にスマートシティサービス等への対応を考えること、を反映したものとなっている。

これまでに様々なプレゼンスセンシング・システムが提案されているが、上記の要件 R1~R4 をすべて満たすシステムは我々の知る限り存在しない。多くのシステムは、オフィス環境や公共施設向けのものであったり、プレゼンスセンサが特定の機器やサービスのみで閉じているものがほとんどである。

そこで、我々は要件 R1~R4 を満たす HNS 向けの新しいプレゼンスセンシング・システムの研究・開発を進めている。

### 3. HNS におけるプレゼンスセンシングのためのサービスフレームワーク

本稿では HNS 内のプレゼンスセンシングを効果的に行う枠組みプレゼンスセンサ・サービスフレームワークを提案する。

#### 3.1 システムアーキテクチャ

図1に提案するプレゼンスセンサ・サービスフレームワーク (PSS フレームワーク) の全体像を示す。提案フレームワークは、人のプレゼンスを検知するための**プレゼンスセンサ・デバイス (PSD)**、複数の PSD を束ねて監視する**プレゼンスセンサ・ターミナル (PST)**、宅内の全ての PST を集約して管理する**プレゼンスセンサ・アグリゲータ (PSA)** から構成される。

PSD は宅内でプレゼンスを取得したい場所に設置される。HNS 内の近隣に設置された複数の PSD は1つの PST に接続される。PST は接続された複数の PSD の値変化を監視する。ある PSD ( $d$  とする) の値変化が閾値を超えた場合、PST は  $d$  の ID とセンサ値、接続情報を PSA に通知する。通知を受け取った PSA は、PST から送られた  $d$  に関するセンサ情報を更新し、最終更新時刻を記録する。PSA は各 PSD の情報の最終更新時刻から、その PSD の近傍に人がいるかどうかを推定するアルゴリズムを実装している。このアルゴリズムは Web サービスとして公開され、様々な HNS サービス、アプリケーションから Web-API を通して利用される。公開された Web サービスを**プレゼンスセンサ・サービス (PSS)** と呼ぶ。

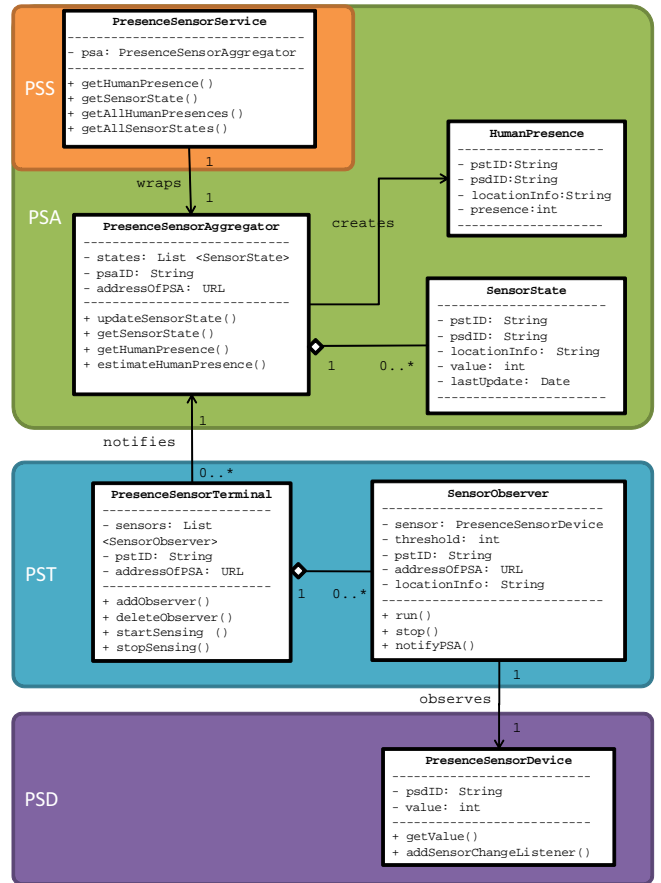


図2 提案フレームワークのクラス図

ビスを**プレゼンスセンサ・サービス (PSS)** と呼ぶ。

図2にシステム全体の UML クラス図を示す。以降、各構成要素について詳細を説明する。

#### 3.2 プレゼンスセンサ・デバイス (PSD)

PSD は HNS 内の人のプレゼンスを取得するための物理的なセンサデバイスである。宅内でプレゼンスを取得したい場所に設置される。人のプレゼンスを取得できるデバイスとしては、様々なセンサが考えられるが、要件 R1 を満たすためにはなるべく安価で宅内に設置しやすいものが望ましい。今回の研究では、赤外線モーションセンサを使用することとした。図2に示す通り、PSD は自らの ID 番号 (psdID) とセンサ値をとる。今回用いた赤外線モーションセンサの場合は、熱を持った物体の移動による赤外線の変化がセンサ値となる。値の変化が起こったときに通知するリスナを加えることもできる。

複数の PSD を近隣に配置することで、プレゼンスセンシングの精度を上げることができる。例えば、部屋の天井中央に1つだけセンサを設置するよりも、格子状に複数設置するほうが、その部屋の在室判定をより正確に行うことができる。センサ追加による精度の向上は、コストとのトレードオフになる。

#### 3.3 プレゼンスセンサ・ターミナル (PST)

PST は複数の PSD を束ねて監視し、センサ値に変化が起こった場合 PSA に通知する役割を持つ。図2に示す通り、PST は接続される PSD 毎にオブザーバを持っている。PST はさらに自身の ID (pstID) と通知先 PSA の URL を持っており、オ

ブザーバを作成するときにこれらの情報を設定する。

オブザーバはさらに監視対象の PSD と閾値、その PSD がカバーする場所の情報 (locationInfo) を持っている。PST によってオブザーバが作成されると、PSD の値変化をリスナを用いて監視する。PSD の値変化がある閾値 (threshold) を超えた場合に、人のプレゼンスを検出し、PSA に通知を行う。通知する情報は、[PST の ID, PSD の ID, センサ値, 場所情報] の組である。ここで重要なことは、PSA への通知は、PST の各オブザーバが自律的に判断して行っているということである。また通知の方向は PST → PSA であり、センサの値が閾値以上に振れた場合だけ通信が起こるイベント駆動型通信であることにも注意されたい。これによって PST や PSD の増加に対する PSA への負荷や通信量の増加を抑制することができる。

その一方で、人がいても PSD に十分な値変化が起きないケースも存在する。そのようなケースでのプレゼンスの判断は PSD では行わず、PSA に任せている。すなわち、PST では振幅の大きい明らかなプレゼンスだけをイベントとして検出し、より詳細なプレゼンス推定は通知先の PSA で行うという役割分担になっている。

### 3.4 プレゼンスセンサ・アグリゲータ (PSA)

PSA は宅内の PST から送られてくる通知を集約して管理し、各 PSD 近傍のプレゼンスを推定する役割を持つ。図 2 に示すとおり、PSA は各センサの状態、自らの ID(psaID) とアドレスを持っている。センサの状態 (SensorState) は PSD 毎に存在し、[PST の ID, PSD の ID, 場所情報, センサ値, 最終更新時刻] の情報から構成される。

それぞれの PST は、PSA の `updateSensorState()` メソッドに [PST の ID, PSD の ID, センサ値, 場所情報] を渡して実行することで、通知を行う (実際には Web-API 経由で実行する)。メソッドが実行されると、PSA は該当する PSD のセンサ状態 (SensorState) を更新 (存在しなければ新規作成) する。その際、最終更新時刻を記録しておく。各センサの状態は、`getSensorState()` メソッドに PSD キーを渡すことで取得できる。ここで PSD キーとは、PST の ID と PSD の ID を接続したもので、宅内の PSD を一意に指定する文字列である。

さらに PSA は人のプレゼンスを推定するメソッド `getHumanPresence()` を実装している。このメソッドは、PSD キーが与えられると、その PSD のセンサ状態の最終更新時刻から、その近傍に人がいる可能性 (プレゼンス尤度, **Presence Likelihood** と呼ぶ) を -1(不在)~100(必ず居る) の数値で返す。

いま、センサ状態の最終更新時刻から、プレゼンスのリクエストがあった時刻までの時間 (経過時間と呼ぶ) を  $t$  とする。このとき直感的に、 $t$  が小さいほどプレゼンス尤度は大きい。つまり、センサが反応してからあまり時間が経過していないのでまだ近くに人がいる可能性が高いといえる。逆に  $t$  が大きくなればなるほど可能性が低くなる。

プレゼンス尤度のモデル化には様々な方法が考えられるが、本稿では経過時間  $t$  についての関数  $PL(t)$  で定義する。

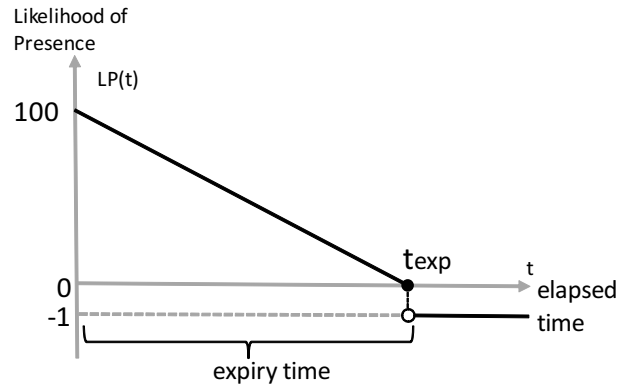


図 3 プレゼンス尤度のグラフ

$$PL(t) = \begin{cases} 100 (1 - t/t_{exp}) & (0 \leq t \leq t_{exp}) \\ -1 & (t > t_{exp}) \end{cases}$$

ここで  $t_{exp}$  は満了時間をあらわし、センサがこの時間を超えて更新されない場合には不在 (-1) とする閾値としてはたらく。LP(t) のグラフを図 3 に示す。LP(t) に基づくプレゼンス尤度の計算は、図 2 中の `estimateHumanPresence()` で行われる。その結果を利用して、`getHumanPresence()` はプレゼンス情報 (HumanPresence) を作成しリクエスト元に返す。

### 3.5 プレゼンスセンサ・サービス (PSS)

PSS は、PSA の機能を様々なアプリケーションから使いやすい形にして提供するサービス窓口の役割を果たす。図 2 に示すとおり、PSS は内部に PSA をラップし、センサ状態を取得 (`getSensorState()`) したり、プレゼンスを推定 (`getHumanPresence()`) するメソッドを外部に公開する。また、全てのセンサ (PSD) の状態やプレゼンスを一括に取得するメソッドも備えている。

様々なプラットフォームの HNS サービス、アプリケーションから利用可能とするため、我々は PSS を Web サービスとして公開する。アプリケーションは、REST または SOAP 形式で PSS のメソッドを実行し、結果を XML 形式で受け取ることができる。

## 4. 実 装

### 4.1 PSS フレームワークの実装

提案する PSS フレームワークのプロトタイプを構築し、研究室のスマートスペースに設置した。

図 4 に開発した PST と PSD のプロトタイプを示す。図中、4 本の足が伸びているのが PSD である。このプロトタイプでは、PSD に Phidgets 社の赤外線モーションセンサ (1111.0 - MotionSensor) を利用した。このセンサをインターフェースキット (1018.2 - PhidgetInterfaceKit 8/8/8) に有線で接続し、インターフェースキットからワンボード PC (Raspberry Pi) に USB 接続している。ワンボード PC は研究室の LAN にネットワーク接続されている。3.3 で提案した PST のソフトウェアは Java で実装した。センサへのアクセスは、Phidget 社が提供するライブラリ Phidget21 を利用した。このモジュールを研

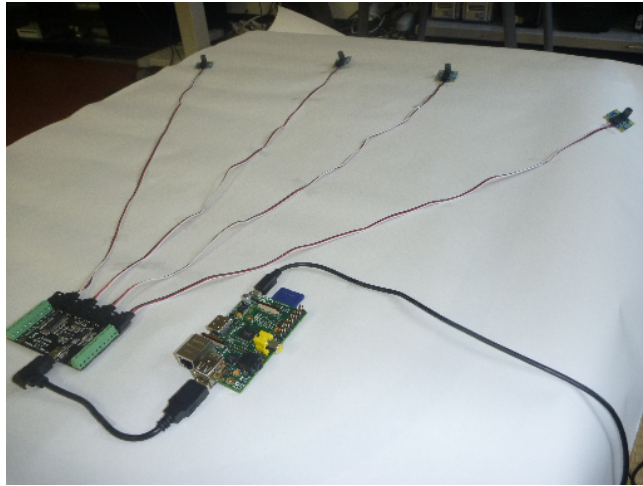


図 4 開発した PST および PSD x 4

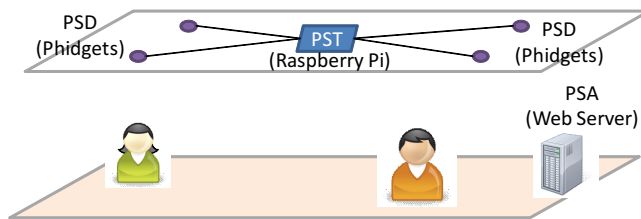


図 5 天井への設置イメージ

表 1 実装に使用した技術

Framework Layer	Technologies Used
Presence Sensor Device (PSD)	Phidgets 1111.0 – Motion Sensor, Phidgets 1018.1 – InterfaceKit
Presence Sensor Terminal (PST)	Raspberry Pi Model B, Phidget21.jar, Java JDK7
Presence Sensor Aggregator (PSA)	Vine Linux 4.2, Apache Tomcat 6.0, Java JDK6, Apache Axis2 1.4

研究室の天井に設置した。図 5 に設置イメージを示す。

PSA および PSS のソフトウェアについても Java で実装した。実装したプログラムを研究室の Web サーバ上に Web サービスとしてデプロイした。Web サービスのミドルウェアは Apache Axis2, Web サーバは Tomcat 6.0 を使用した。

表 1 に実装に使用した技術をまとめる。開発した Java のソースコード総行数は 447 行、開発期間は約 60 日であった。

図 6 に PSS のメソッド `getHumanPresence()` を Web-API で実行した結果を示す。この実行例では、研究室 S101 号室の西側のドアのプレゼンス尤度が 94 であることを示している。

#### 4.2 プレゼンス可視化アプリケーション

開発した PSS のクライアントアプリケーションとして、計測したプレゼンスを Web ブラウザ上のグラフに可視化するアプリケーションを開発した。このアプリケーションは、JavaScript および Google Chart Tools を用いて開発した。まず、PSS の `getHumanPresence()` にアクセスして、研究室の天井の 3 箇所のモーションセンサ付近のプレゼンス尤度を取得する。次にその 3 つのプレゼンス尤度を Google Chart の棒グラフ、または、

```
<ns:getHumanPresenceResponse>
  <return type="jp.kobe_u.cs27.pss.HumanPresence">
    <locationInfo>West Door of S101</locationInfo>
    <presence>94</presence>
    <psdID>01</psdID>
    <pstID>149180</pstID>
  </return>
</ns:getHumanPresenceResponse>
```

図 6 PSS の `getHumanPresence()` の実行結果

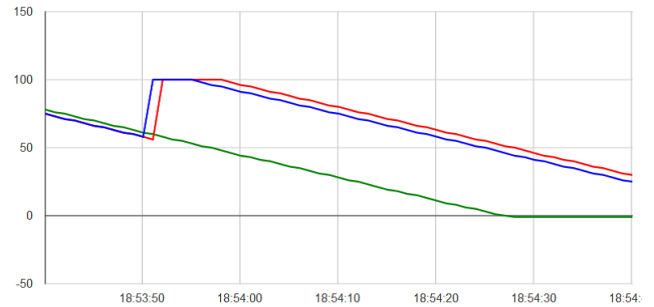


図 7 プレゼンス可視化アプリケーションのスクリーンショット

折れ線グラフで表示する。一定周期で再描画を繰り返すことで、プレゼンス尤度の動的な推移を観測することができる。

図 7 に可視化アプリケーションのスクリーンショットを示す。図は折れ線グラフによる可視化を示している。横軸が時刻、縦軸がプレゼンス尤度を表している。計測日の 18 時 53 分 52 秒ごろに 2 件の PSD がプレゼンスを検知し、同分 55 秒ごろから尤度が減衰していることが見て取れる。このことから、3.4 で提案したプレゼンス尤度のモデルが正しく動作していることが確認できる。

## 5. 考 察

提案フレームワークが 2.2 で述べた要件 R1~R4 をどのように満足するかを考察する。

まず要件 R1 であるが、これは PSD のセンサおよび PST のハードウェアの値段によるところが大きい。今回のプロトタイプで使用した Phidgets センサは 1 つ 4 千円ほどしたため十分安価とはいえない。しかしながら、民生用の安価な赤外線センサは数百円からのものもあるため、実用化・量産において安価なセンサデバイスを利用すべきである。

要件 R2 は提案アーキテクチャによって解決可能である。センシングする場所を増やしたい場合には、その場所に PSD を配置して PST に接続することで、容易に拡張できる。またその際、PSA や PSS を改修したり再設定したりする必要はない。

要件 R3 については、PSS によって満たされる。すなわち、プレゼンス情報が PSS の Web-API を通して様々なアプリケーションで共有可能になる。プラットフォームに依存しない Web サービスを利用することで、HNS サービスだけでなく、ほかの様々な Web サービス (E-mail, SNS サービスなど) とも連携可能である。

要件 R4 に関しては, PSA および PSS を外部のクラウドサーバに配置することで, 技術的には容易に実現できる. PSA の通知先 URL は家庭内 LAN に閉じている必要はないため, PSA を外部クラウドサービスとすることで, 複数の家の HNS からプレゼンス情報を集約することが可能となる. 一方で, プレゼンス情報を外部クラウドに出すことにおいては, プライバシーやセキュリティの問題を対策する必要がある. これについては今後の課題としたい.

## 6. おわりに

本稿では, ホームネットワークシステム (HNS) における効率的なプレゼンスセンシングを行うためのフレームワーク, プレゼンスセンサ・サービスフレームワークを提案した. 提案フレームワークは, 物理的なプレゼンスセンサである PSD とそれを束ねて監視する終端装置 PST, 宅内の全てのセンサ情報を集約する PSA から構成される. PSA に集まったセンサの状態に基づいて, 各 PSD 近傍のプレゼンスが推定され, プレゼンスセンサ・サービス (PSS) として公開される. 各 HNS アプリケーションは PSS を利用して, 宅内の様々な場所のプレゼンスにアクセスできる. 提案手法を, Phidgets 赤外線モーションセンサ, Raspberry Pi, Java を用いて実装した. PSS を利用してプレゼンスを可視化するアプリケーションを構築し, 提案手法が正常に動作していること確認した.

今後の課題として, 提案システムの性能評価 (精度, 応答速度) やプレゼンスセンサの最適配置手法, 過去のプレゼンス情報の履歴を利用できる機能の追加等を考えていきたい.

**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), および, 積水ハウスの研究助成を受けて行われている.

## 文 献

- [1] 液晶テレビ BRAVIA (ブラビア)  
[http://www.sony.jp/bravia/products/KDL-32EX420/feature\\_3.html](http://www.sony.jp/bravia/products/KDL-32EX420/feature_3.html)
- [2] 日立アプライアンス株式会社 省エネ達人プレミアム  
<http://www.hitachi-ap.co.jp/products/business/ac/office/premium/>
- [3] 灯りの広場  
<http://www.akarinohiroba.com/common/guide/exp/sensor-560.php>
- [4] スマートハウス構築のためのホームネットワークとは  
<http://i.impressrd.jp/e/2011/05/27/1137>
- [5] 赤外線位置検出センサ ナノマシンシステム技術研究センター 著  
<http://www.ritsumeai.ac.jp/research/common/file/seeds/robo.env/infrared.light-kimata.pdf>
- [6] 装着した 3 次元磁気センサと姿勢センサを用いたユーザの自己位置推定  
濱口 明宏 著  
<http://yokoya.naist.jp/paper/datas/897/mthesis%280451094%29.pdf>
- [7] センシング技術圧力センサ  
<http://eetimes.jp/ee/articles/1107/14/news104.html>
- [8] 屋内測位・光神経センサを基盤としたセンサネットワークとサービスシステムの実現  
近 哲也 渡辺 一弘 著  
[http://www.t.soka.ac.jp/assets/pdf/grad/cou\\_dc/sys/kon.t.pdf](http://www.t.soka.ac.jp/assets/pdf/grad/cou_dc/sys/kon.t.pdf)
- [9] NEC ネットエスアイ 無線 LAN 位置情報システム  
<http://www.nesic.co.jp/solution/wifi/wps.html>
- [10] 株式会社テクノ大西  
<http://oonishi.info/productinfo/closeup04.html>
- [11] Masahide Nakamura, Akihiro Tanaka, Hiroshi Igaki, Haruaki Tamada, and Ken-ichi 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.
- [12] 井垣 宏, 瀬戸 英晴, 福田将之, まつ本真佑, 中村匡秀, "家庭における省エネ促進のための電力消費振り返りサービスの実装と評価," *電子情報通信学会論文誌*, vol.J95-D, no.4, pp.778-789, April 2012.
- [13] 徳田 啓介, まつ本 真佑, 中村 匡秀, "スマートフォンを利用したピークカット促進 アプリケーションの提案と実装," *情報処理学会研究報告*, vol.2012-MBL-63, no.4, pp.1-7, August 2012.
- [14] 中村 匡秀, 関本 純一, 井垣 宏, 松本 健一, "家庭のエンドユーザを対象としたホームネットワーク機器連携サービス作成支援システム," *ヒューマンインターフェース学会論文誌 「ユニバーサルデザイン」 特集号*, vol.11, no.4, pp.369-379, November 2009.
- [15] Shimpei Soda, Masahide Nakamura, Shinsuke Matsumoto, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto, "Implementing Virtual Agent as an Interface for Smart Home Voice Control," In *Asia-Pacific Software Engineering Conference (APSEC2012)*, pp.342-345, December 2012.
- [16] 徳田 啓介, 稲田 卓也, まつ本 真佑, 中村 匡秀, "ホームネットワークシステムのためのパーソナルリモコン開発フレームワーク," *電子情報通信学会技術研究報告 SS2010-54*, vol.110, no.458, pp.7-12, March 2011.
- [17] Masahide Nakamura, Shuhei Matsuo, Shinsuke Matsumoto, Hiroyuki Sakamoto, and Hiroshi Igaki, "Application Framework for Efficient Development of Sensor as a Service for Home Network System," In the 8th IEEE 2011 International Conference on Services Computing (SCC 2011), pp.576-583, July 2011. (Washington D.C.)
- [18] Masahide Nakamura, Shuhei Matsuo, and Shinsuke Matsumoto, Supporting End-User Development of Context-Aware Services in Home Network System, In Roger Lee, editor, *Studies in Computational Intelligence*, pp.159-170, Springer, 2012.
- [19] センシング技術 IMES 測位  
<http://eetimes.jp/ee/articles/1106/13/news014.html>
- [20] 屋内ロケーション管理技術 奥山敏 森信一郎 小川晃弘 著  
<http://img.jp.fujitsu.com/downloads/jp/jmag/vol64-1/paper15.pdf>
- [21] phidgets 1111 User Guide  
[http://www.phidgets.com/docs/1111\\_User\\_Guide](http://www.phidgets.com/docs/1111_User_Guide)