

ユーザの移動に伴う機器と環境の状態引き継ぎを実現するホームネットワークシステムの提案

井垣 宏[†] 長江 洋子[†] 山田 松江[†] 中村 匡秀^{††} 松本 健一^{††}
青山 幹雄[†]

[†] 南山大学数理情報学部, 愛知県

^{††} 奈良先端科学技術大学院大学情報科学研究科, 奈良県

E-mail: [†]{igaki,03mt064,03mt110}@nanzan-u.ac.jp, ^{††}{masa-n,matumoto}@is.naist.jp,
^{†††}mikio.aoyama@nifty.com

あらまし 家庭内におけるネットワークとネットワークに接続可能な機器の普及が著しい昨今では, それらの複数機器を連携制御する新しい付加価値サービスの提供が進みつつある. 本稿では, その種の新しい付加価値サービスとして部屋状態引継ぎサービスを提案する. このサービスは, 複数の部屋から構成される家庭内におけるユーザの移動に着目し, 移動時に行われる機器制御を支援するサービスである. 我々の提案するサービスが支援可能なシナリオとしては次の二つのような例が挙げられる. 居間にいたユーザがベッドルームに移動する場合, (1) 居間で見ていた TV 番組を移動先のベッドルームでも続けて視聴する, (2) 室温や照度といったベッドルームの環境の状態を居間と同じにする. このように, 継続的に部屋の状態を等しくするための機器制御を我々のサービスが支援することで, ユーザの利便性の向上を図る. 本稿では, この部屋状態引継ぎサービスの実現方法の詳細について述べ, ドールハウス型プロトタイプシステムによる動作の検証を行う.

キーワード ホームネットワークシステム, 連携サービス, 引継ぎ, ネットワーク家電

Home Network System Taking over the Status of Appliance and Environment along with User's Movement

Hiroshi IGAKI[†], Youko NAGAE[†], Matsue YAMADA[†], Masahide NAKAMURA^{††}, Ken-ichi MATSUMOTO^{††}, and Mikio AOYAMA[†]

[†] Department of Information and Telecommunication Engineering, Nanzan University, 27 Seirei, Seto, Aichi 489-0863 Japan

^{††} Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0192 Japan

E-mail: [†]{igaki,03mt064,03mt110}@nanzan-u.ac.jp, ^{††}{masa-n,matumoto}@is.naist.jp,
^{†††}mikio.aoyama@nifty.com

Abstract Home network system which includes networked home appliances and a home network are becoming popular remarkably. As a result, the several value added services to integrate multiple networked appliances are being provided by the home network system. In this paper, as the kind of a new value added service, we propose a room state migration service. This service supports to integrate the appliance methods when a user moves in the inside of a home which consists of multiple rooms, For example, suppose that a user in a living room moves to a bed room. The following two are examples of the scenario which our service can support. (1) A user continues watching a TV program which was being watched in the living room also in the bed room. (2) A user makes the state of the environment (such as room temperature and brightness) of the bed room the same as the living room. Thus, a user's convenience can be improved with our service supporting the appliance orchestration for making the state of the rooms equal continuously.

Key words Home network system, networked home appliances, integrated services, migration

1. はじめに

多様な家電機器を家庭内のネットワークに接続するための基盤技術の開発・普及が進んでいる [2] [8] . ホームネットワークシステム (以降 HNS) とは, そのようなネットワークに接続する能力を持った家電機器やセンサ類とそれらを統合するホームサーバなどから構成されるシステムである . HNS 上でネットワークを通して制御可能なエアコン・照明・DVD といった機器や各種センサ類などが増えるにつれて, 複数の機器/センサを組み合わせた新しい付加価値サービスの提供が増加しつつある . 例えば [10] では, 携帯電話を利用してエアコンや照明の電源を遠隔操作するサービスや, 複数の機器機能を登録しておくことで, ボタンを押すと登録された機器機能をまとめて実行してくれるサービスなどを提供している .

そこで我々は, 複数機器を制御しユーザの利便性を向上する新たな付加価値サービスとして部屋状態引継ぎサービスを提案する . 一般的に, 家は複数の部屋に分かれており, ユーザはその部屋間を移動する . 移動時には, 移動元の部屋で視聴していた TV を移動先でも引き続き視聴したり, 移動先の部屋の室温や照度といった環境要因を移動元と同じにするために空調や照明を調節したりといった, ユーザの居る部屋の状態を継続的に同じにするための機器制御が行われることがある . 我々の提案する部屋状態引継ぎサービスは, 移動前と移動後の二つの部屋において, 機器の状態と環境要因の状態といった部屋状態を近づけるための機器連携制御を支援するサービスである . 部屋状態の引継ぎを目的とすることで, ユーザが部屋から部屋へ移動するだけで, 移動先の部屋における機器連携制御の内容が自動的に決定される .

本稿では 2. で, 各機器を機器メソッドと機器プロパティを持つオブジェクト, 環境を環境プロパティを持つオブジェクトとして定義した我々の先行研究 [7] に基づき, 機器プロパティの値および環境プロパティの値として表現される機器状態と環境状態の定義について説明する . この機器状態と環境状態をユーザの移動に伴って引き継ぐ際の問題点と, それを解決するための具体的な提案内容について 3. において述べる . 4. では, 提案手法をドールハウス型プロトタイプシステムとして実際に構築し, 複数の部屋間を移動する際の部屋状態引継ぎサービスの動作内容を検証している . また, 5. ではその結果についての考察を既存研究との比較を交えて行っている .

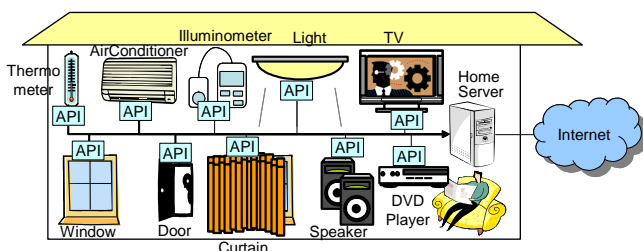


図 1 ホームネットワークシステムの構成例

2. 準備

2.1 ホームネットワークシステム

図 1 にある構成例が示すように, 一般に HNS は複数のネットワークに接続された機器とそれらを制御するホームサーバを持つシステムである . ネットワークに接続された機器はその機能を制御するための API を公開しており, ホームサーバを通じてそれらの API を通じて機器機能呼び出すことができる . 実際に, 複数の機器制御を目的として, DLNA や OSGi, ECHONET 等のさまざまな標準が提案されている [2], [3], [8] .

HNS において利用される機器としてはエアコンや DVD 等の情報家電や設備家電だけでなく, 温度計や湿度計といったセンサ類, RFID のタグとリーダなども増えてきている . 特に RFID を用いた機器やユーザの位置を検知する仕組みが注目されており, [6] ではユーザの携帯電話や財布といった身の回りのものの場所を検知し, ユーザが探しているものの場所を知らせてくれるサービスを RFID を用いて実現している .

2.2 部屋の状態モデル

本稿では, HNS が存在する部屋の状態をその部屋に配備されている各機器の状態とユーザを取り巻く環境状態の集合として表す . ここで, 機器状態とは機器のプロパティの示す値の集合であり, 環境状態は照度や室温といった環境プロパティの示す値の集合である . 例えば, TV の機器プロパティとして電源, チャンネル, 音量があるとすると, TV の状態 $s_{TV} = \langle \text{ON}, 3, 30 \rangle$ はその TV の電源が ON で 3 チャンネルを表示しており, 音量が 30dB であることを示す . また, 環境プロパティとして室温, 照度, 湿度を持つ場合, 環境状態が $s_e = \langle 26, 300, 30 \rangle$ と表されたときは, その部屋の室温が 26°C, 照度が 300lx, 湿度が 30%であることを示すものとする .

機器/環境プロパティは機器機能の実行に影響を受ける . そこで我々は機器を機器メソッドと機器プロパティを持つオブジェクト, 環境を環境プロパティを持つオブジェクトとする定義を行った [7] . 機器プロパティは機器メソッドの実行により直接的にその値が変化する . 一方で環境プロパティは, 機器プロパティと機器メソッドの関係ほど直接的には変化しない場合がある . 例えば, AirConditioner の機器メソッド cooling(22) を呼び出すことで, 22°C の設定で冷房が起動するものとする . ここで, AirConditioner の機器プロパティである TempSetting は 22°C に変化するが, 他の機器の状態や部屋の室温次第では, 環境プロパティの室温の値は必ずしも 22°C にはならない .

このように, 環境プロパティに機器メソッドが与える影響を正確に見積もることは困難である . そこで我々は, 機器メソッドが特定の環境プロパティに与える影響を上昇, 下降, 不明という方向で表現し, その環境プロパティと影響の方向を合わせて環境インパクトと定義する . 例えば, エアコンの cooling() メソッドは実行されると部屋の室温を下降させる . そのため, このメソッドの環境インパクトは $(\text{Temperature}, \downarrow)$ となる . 影響方向が不明であるという環境インパクトは, 対象のメソッドを実行することにより特定の環境インパクトを上昇もしくは下降させる可能性があるが, 現時点では分からないということ

表 1 機器定義の例

ApplianceName	ApplianceClass	ApplianceProperty	ApplianceMethod	IP _e
Living AirConditioner	AirConditioner	Power TempSetting ModeSetting	cooling(tTempSetting temp)	(Temperature, ↓)
			heating(tTempSetting temp)	(Temperature, ↑)
			dehumidify()	(Temperature, ↓) (Humidity, ↓)
			off()	
Kitchen Window	Window	WindowStatus	open()	(Temperature, -) (Humidity, -)
			close()	
Kitchen Curtain	Curtain	CurtainStatus	open()	(Temperature, -) (Brightness, -)
			close()	
LivingTV	TV	Power Channel Volume	setChannel(tChannel channel)	
			setVolume(tVolumeSetting volume)	
			off()	
LivingLight	Light	Power BrightSetting	on()	(Brightness, ↑)
			off()	
Dehumidifier	Dehumidifier	Power	on()	(Humidity, ↓)
			off()	
Humidifier	Humidifier	Power	on()	(Humidity, ↑)
			off()	

を示す．例えば，Window の open() メソッドは実行により室温を変動させる可能性があるが，外気温と室温次第で，上昇/下降のどちらになるかが変わるため，現時点では影響の方向は不明である．表 1 はこれらのモデルを利用した機器と機器メソッドおよび環境プロパティの関係を示した例である．この表において，ApplianceName は機器の名称，ApplianceClass は，各機器の種類を示しており，この ApplianceClass が共通の機器同士は共通の機器プロパティを持つものとする．さらに，ApplianceProperty と ApplianceMethod はそれぞれその機器が持つプロパティと機器メソッドをあらわしており，IP_e は環境インパクトを示している．部屋の状態と機器メソッドが及ぼす各プロパティへの影響をこのようにモデル化することで，機器メソッド実行後の部屋の状態を予測することが可能となる．

2.3 ユーザの移動と部屋の状態引継ぎ

一般的に，家は複数の部屋から構成され，ユーザはその部屋間を移動しつつ生活する．ユーザは部屋を移動するたびに，部屋の状態をユーザの期待する理想的な状態に近づけるために複数の機器を制御する．

そこで我々は，ユーザが部屋を移動したときに，前の部屋の状態を引き継ぐサービスを提案する．このサービスでは，移動した先の部屋の状態を移動する前にいた部屋の状態に近づけることで，ユーザがどの部屋に移動しても同じ部屋状態を維持し続けることを目的としている．例えば，あるユーザがリビングルームで映画を見ていたとする．そのユーザがリビングからベッドルームに移動したときに引継ぎサービスを利用することで，同じ映画をベッドルームにある TV で見続けることができる．また，機器の状態だけでなく，室温，照度，湿度といった環境状態に関しても，引継ぎサービスを適用することでユーザの居る場所どこにおいても，同じ環境状態が維持し続けられる．

このように，ユーザの移動時に部屋状態を引き継ぐサービスは，ユーザの利便性の向上に寄与する新たな付加価値サービスとして有用であると考えられる．以降では，この部屋状態引継ぎサービスを実現するための具体的な方法について説明する．

3. 部屋状態引継ぎサービス

3.1 キーアイデア

我々の提案する部屋状態引継ぎサービスは，ユーザがある部屋 R_S から別の部屋 R_G に移動したときに， R_G の部屋状態すなわち機器状態と環境状態を R_S の部屋状態に可能な限り近づけることを目的としている．機器/環境状態はともに，その部屋に配置されている HNS を通じた機器制御によって変動させることができる．そのため，RFID リーダーとタグの類を利用してユーザの移動を検知し， R_S で動作していた機器と共通の機器を R_G において共通の動作内容で動作させることで，引継ぎサービスを実現することができると考えられる．しかしながら，引継ぎサービス実現のためには以下の二つの問題が解決される必要がある．

(1) R_S と R_G に配置される機器が異なる場合の環境状態の引継ぎ

(2) R_S と R_G の元々の部屋の環境状態が異なる場合の環境状態の引継ぎ

部屋間で配置される機器が異なる場合，共通の機器を動作させることができなくなるため，完全に全ての機器において機器状態を等しくすることは不可能である．しかしながら，存在しない機器の機能呼び出しがユーザに要求されることは考えにくい．そのため，機器状態の引継ぎに関しては配置される機器が異なる場合は問題にはならない．すなわち，TV が無い部屋で TV を再生することが要求されるようなケースは考えない．しかしながら，環境状態の引継ぎにおいては，室温や湿度といった環境プロパティに影響を与える機器は複数存在する場合がある（例えばエアコンの除湿機能と除湿機の除湿機能等）．そのため，配置される機器が異なっていたとしても，代替機器の機能実行による環境状態の引継ぎが可能な限り行われることが望ましい．

二つ目の問題の場合，仮に R_S と R_G の機器の配置が等しかったとしても，単純に同じ機器機能を同じ内容で実行するだけでは，引継ぎサービス後の環境状態が R_S と R_G の間で異なってしまう可能性がある．そのため，環境状態を引き継いでユーザの期待する理想的な状態に R_G の環境状態を遷移させる

ためには、現在の環境状態を考慮して機器機能とその実行内容を調整する必要がある。

これらの問題に対応するためのキーアイデアとして、我々は HNS に存在する全ての機器を環境インパクトを持つメソッドを含むものと含まないものに分類した。機器状態の引継ぎでは、環境プロパティに影響を与えない機器のみを対象とすることで、環境状態への影響を考慮せず、 R_S と R_G に共通して存在する機器状態の引継ぎを実現する。また、環境状態の引継ぎでは、環境プロパティに影響を与える機器を対象とし、 R_G の引継ぎ前の環境状態や配置されている機器に応じて、実行する機器とその実行内容を柔軟に変化させることで、 R_S と R_G の環境状態を近づける。我々は、このように部屋状態引継ぎサービスを機器状態の引継ぎサービスと環境状態の引継ぎサービスの二つの異なる方法を組み合わせることで実現する。以降では、ユーザが R_S から R_G に移動するという前提の下で、二つの引継ぎサービスを実現する方法について説明する。

3.2 機器状態引継ぎサービス

機器状態引継ぎサービスは環境プロパティに影響を与えない機器のみを対象とし、 R_S から R_G へ機器状態を引き継ぐサービスである。このサービスを実行するためには、 R_S の状態を保持し、 R_G に配置されている機器と同種のもが存在するかを比較したのちに、引継ぎを行う必要がある。そこで、Appliance State DB を用意し、引継ぎサービスを行う際に R_S に存在する環境プロパティに影響を与えない機器の ApplianceClass とその機器の状態を組み合わせることで記録しておく。この Appliance State DB では、各 ApplianceClass の機器状態は引継ぎサービスが実行されるたびに更新されるが (R_S にその ApplianceClass の機器が存在すれば)、ApplianceClass の情報そのものが削除されることはない。

以下にユーザが R_S から R_G へ移動する際の機器状態引継ぎサービスの手続きを示す。

機器状態引継ぎサービスの手続き

STEP1: R_S に存在する環境プロパティに影響を与えない機器の ApplianceClass とそのときの機器状態を Appliance State DB に保存する。

STEP2: Appliance State DB に保存された ApplianceClass と一致するものを R_G に配置された機器から抽出する。

STEP3: STEP2 で抽出された機器の状態を Appliance State

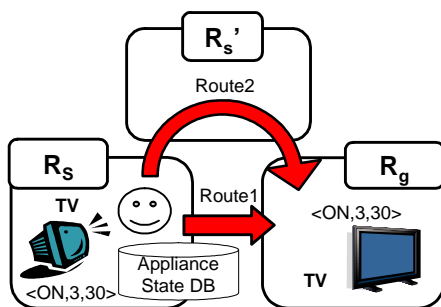


図 2 R_S から R_G へ移動する場合と R_S から R'_S を経由して R_G へ移動する場合の機器状態引継ぎサービス

DB に保存されているものと一致させるための機器機能呼び出しを行う。

図 2 に、 R_S から R_G へ二つの経路で移動する際の機器状態引き継ぎサービスの実行例を示す。STEP3 の機器機能呼び出しの内容は、機器プロパティとそのプロパティの値により決定される。例えば、表 1 の LivingTV の場合、 $s_{TV} = \langle ON, 3, 30 \rangle$ が与えられると、LivingTV の setChannel(3) と setVolume(30) が実行される。

また、Appliance State DB を利用することで、図 2 が示すように R_S から R'_S を経由して R_G に行く場合、 R_S と同じ ApplianceClass の機器が R'_S に無くて R_G にあるようなときでも、 R_S から R_G へと機器状態の引継ぎを行うことができる。

3.3 環境状態引継ぎサービス

ユーザの居る部屋の環境状態を常にユーザにとって理想的な状態にするためには、すべての環境プロパティをその部屋に配置された機器によって制御する必要がある。しかしながら、ユーザの移動するすべての部屋において常に理想的な環境状態を維持できるとは限らない。例えば、窓も照明機器も無い部屋では、その部屋の照度を機器制御によって調節することは難しい。そのため、我々の提案する環境状態引き継ぎサービスでは、ユーザの期待する理想的な環境状態を用意しておき、ユーザの移動する先々の部屋の環境状態をその理想的な環境状態に近づけることを目的とする。本研究では、ユーザごとに期待する理想的な環境状態が異なることを考え、Environment State DB をユーザごとに用意することを提案している。

ユーザの移動先である部屋 R_G の環境状態をユーザの理想に近づけるためには、まず R_G の現在の環境状態とユーザの理想との間の差を計測し、どの環境プロパティをどのように遷移させるべきかを算出する必要がある。すなわち、 R_G の現在の環境状態とユーザの期待する理想的な環境状態を比較することで、環境プロパティを上昇させるべきか、下降させるべきか、といった情報を環境インパクトの形で算出する。

環境インパクトが得られた後に、その環境インパクトを満たすことができる機器メソッドを R_G に配置された機器の中から探索する。ここで獲得される機器メソッドの集合 AVM は以下の定義を満たすものとする。

定義:与えられた環境インパクトを満たす機器メソッド集合 AVM 環境状態引継ぎサービスを行う部屋に配置された機器の集合を D_G 、 D_G が持つ機器メソッドの集合を M_G とし、さらに IP を与えられた環境インパクトの集合とする。ここで、 $m \in M_G$ である m の持つすべての環境インパクトの集合が IP の部分集合となっているとき、そのようなメソッド m の集合を環境状態引き継ぎサービスにおける AVM と呼ぶ。

ただし、機器メソッドの持つ環境インパクトの影響の方向が不明のままでは、 AVM の獲得に支障がでる場合がある。そこで、環境状態引き継ぎサービス実行時に、影響の方向を決定する手段を用意する。機器メソッド実行時の環境プロパティに与える影響の方向を調査する手段は、対象となる機器メソッドごとに異なる可能性があるが、その部屋に配置された複数のセンサ類の示す値を HNS 上で統合することで影響の方向

を知ることが十分可能であると考えられる。例えば、表 1 の KitchenWindow の open() メソッドは、窓の内側と外側の温度および湿度を比較することで、open() の実行による環境プロパティの室温、湿度への影響の方向を知ることができる。このように、環境インパクトを引き継ぎサービス実行時に更新しておくことで、より正確な AVM の獲得が可能になる。

以下に、ユーザが R_S から R_G に移動する際の環境状態引き継ぎサービスの手順を示す。

環境状態引き継ぎサービスの手続き

STEP1: R_G に存在する環境インパクトを持つ機器メソッドの中から、環境インパクトにおける影響の方向が不明(すなわち、'-' になっている)ものを抽出する。

STEP2: STEP1 で抽出されたメソッドに対応した方法でそのメソッドの実行が環境プロパティに与える影響の方向を求め、環境インパクトを更新する。

STEP3: R_G の環境状態と Environment State DB に記録されているユーザの環境状態を比較し、 R_G をユーザにとっての理想的な環境状態に遷移させるための環境インパクトを導出する。ここでは、例えば環境プロパティとして室温があり、 R_G の室温が 10°C で、ユーザの期待する室温が 22°C であったとすると、(Temperature, \uparrow) という環境インパクトが得られる。この処理をすべての環境プロパティに対して行う。

STEP4: STEP3 で得られた環境インパクトの集合にもとづいて、その環境インパクトを満たす機器メソッド集合 AVM を求める。

STEP5: STEP4 で得られた AVM を R_G の HNS が実行する。

図 3 は環境状態引き継ぎサービス実行の例として示した R_S と R_G の例である。この例では、環境プロパティとして室温、照度、湿度が与えられており、図が示す機器が配置されている。ここで、ユーザが移動を開始すると、 R_G の現在の環境状態とユーザの期待する環境状態を比較し、(Temperature, \uparrow), (Brightness, \uparrow), (Humidity, \downarrow) という環境インパクトが上記 STEP3 において得られる。この環境インパクトの集合により、STEP4 において、AVM として Heater の on(), LivingAirConditioner の heating(), LivingLight の on(), Defumidifier の on() が選択される。STEP5 では、これらの機器メソッドの実行が行われる。この例では、Heater と LivingAirConditioner の二つで (Temperature, \uparrow) の環境インパクトに対応しているため、冗長である可能性がある。STEP5 の後の処理で、ユーザの要求に応じて、任意のもの一つのみを実行するという調

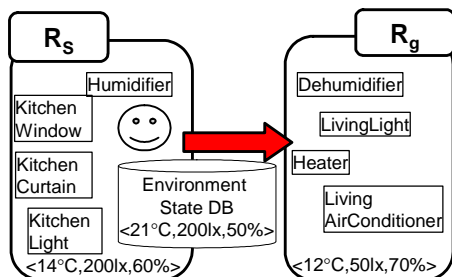


図 3 環境状態引き継ぎサービス時の部屋 R_S , R_G の例

整を行うことも考えられるが、詳細は今後の課題である。

4. プロトタイプ開発

図 4 は部屋状態引き継ぎサービスの動作を検証するために作成したドールハウス型プロトタイプシステムの写真である。図 5 はドールハウスの内部に配置された機器の種類と各部屋の構成と環境状態を示している。また、環境プロパティとしては、室温、照度、湿度を与えた。

開発環境は以下に示すとおり。

- JDK 1.5_09
- Tomcat 5.5.20
- Apache Axis 1.4
- Phidget(phidgetRFIDkit, 4-Motor PhidgetServo Kit, phidgetInterfaceKit8/8) [9]

Window, Curtain, Light は PhidgetServo Kit や LED を組み合わせて実際の機器の振る舞いを作成し、AirConditioner や TV 等のドールハウスにおける振る舞いの開発が困難な機器に関しては、PC 上の GUI アプリケーションで機器状態の詳細確

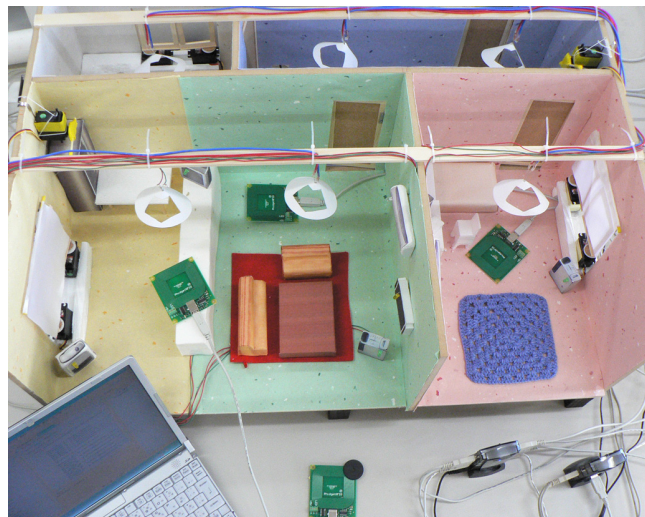


図 4 開発したドールハウスの写真

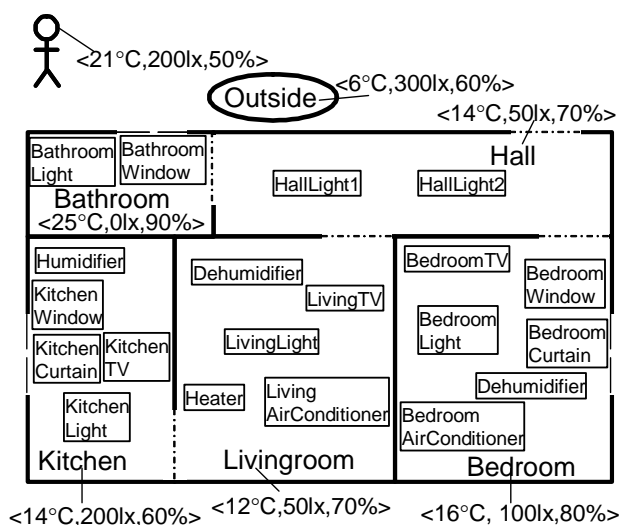


図 5 ドールハウスの構成

認を表示するようにした。これにより、引き継ぎサービス実行時の動作状況や機器状態が確認できる。さらに、phidgetRFIDkitのRFIDリーダをすべての部屋に配置することで、タグを保持したユーザがリーダを通過したときに部屋間を移動したものと検知する仕組みを実現した。また、すべての機器の振る舞いを制御するための機器メソッドをWebサービス[1]を利用して公開した。

この枠組みを利用し、ユーザの移動を検知した後に部屋状態引き継ぎサービスを実行し、対応する機器メソッドを制御するJavaプログラムを作成した。ここでは、ケーススタディとして以下のようなシナリオによってユーザが移動した際の、部屋状態引き継ぎサービスの動作検証を行った。

ケーススタディシナリオ

ユーザは最初Kitchenにあり、KitchenTVを利用して3チャンネルの番組を見ている。その後、Livingroomへ移動し、引き続き同じ番組をLivingTVで視聴する。次に、チャンネルの3を7に変更してそれをしばらく見た後、Hallを経由してBedroomに移動する。

このケーススタディでは、(1)KitchenからLivingroom、(2)LivingroomからHall、(3)HallからBedroomに移動する際の3回、部屋状態引き継ぎサービスが実行される。(1)の際には、KitchenTVで見ている3チャンネルやそのTVの音量の情報が機器状態としてLivingroomのLivingTVに引き継がれる。環境状態はユーザの保持している(21°C, 200lx, 50%)と現在のLivingroomの環境状態である(13°C, 50lx, 70%)にもとづいてDehumidifierのon()、LivingLightのon()、Heaterのon()、LivingAirConditionerのheating()が実行される。(2)では、HallにTVが存在しないため機器状態の引き継ぎは行われない。また、環境状態に関しては、HallLight1/2による照度の制御のみが行われる。(3)のときには、LivingroomのLivingTVで視聴していた7チャンネルの番組がBedroomTVで表示され、BedroomCurtainのopen()、BedroomLightのon()、Dehumidifierのon()、BedroomAirConditionerのheating()がそれぞれ実行される。

5. 考 察

ユーザの移動や現在の部屋の室温等の情報に基づいて、HNSによる機器制御を行う研究は多く行われている。[5]では、ユーザの移動や部屋の明るさ等の情報をコンテキストとして扱い、機器制御と関連付けることで多彩な状況に対応することを目的としたサービスが多数紹介されている。コンテキストウェアの考え方を代表とするこれらの分野では、コンテキストと機器制御の関連付けの作成が困難な場合が多く、ユーザの行動履歴などから適応的に関連付けの情報を更新する研究[4]などが行われている。

我々の提案する部屋状態引き継ぎサービスでは、ユーザがどこに移動しても、その部屋状態を継続的に同じにすることを目的としている。ユーザの期待する理想的な環境状態と移動元の部屋の機器状態を移動先の部屋に引き継ぐという仕組みを実現したことで、コンテキストと機器制御の関連付けといった従来必

要であった準備作業がほとんど必要でなくなり、ユーザ/開発者双方の負担が減少している。

一方で、環境状態引き継ぎサービスの対象ではないような複雑な機器制御(例: DVDシアターサービス)を新たに始めるような場合の支援には我々の提案するサービスは対応していない。ユーザが新規に開始する機器連携サービスの支援にはコンテキストウェアアプリケーションを含む従来の複雑な機器連携制御実現手法を我々の提案内容に組み入れることで、改善が可能であると考えている。

6. まとめと今後の課題

本稿では、HNSの提供する新しいサービスとして部屋状態引き継ぎサービスを提案した。部屋状態を機器状態と環境状態に分け、すべての機器を環境プロパティに影響を与えるものと与えないものに分類をしたうえでそれぞれの引き継ぎサービスを作成した。これにより、機器/環境それぞれの状態に適した形で構成された各引き継ぎサービスが、ユーザの居る部屋状態を継続的にその前に居た部屋状態に近づけることが可能となった。

今後の課題としては、機器メソッドが環境プロパティに与える影響をより詳細に分析し、現状の上昇/下降だけでなく、より正確に環境状態をユーザの期待するものに遷移させるためのモデル構築を行うことを考えている。

謝辞 この研究は、科学技術研究費(若手研究B 18700062, 若手研究(スタートアップ) 18800060)、および、21世紀COEプログラム「NAIST-IS:ユビキタス統合メディアコンピューティング」の助成を受けて行われている。

文 献

- [1] E. Cerami, "Web Services Essentials - First Edition," O'Reilly & Associates Inc. 2002.
- [2] DLNA: "Digital Living Network Alliance", <http://www.dlna.org>, 2007.
- [3] ECHONET Consortium: <http://www.echonet.gr.jp/english/index.htm> (1997-2004).
- [4] M. Huebscher and J. McCann, "Adaptive middleware for context-aware applications in smart-homes," Proc. 2nd International Workshop on Middleware for Pervasive and Ad-Hoc Computing, Middleware 2004, pp.111-116, Oct. 2004.
- [5] S. Meyer and A. Rakotonirainy, "A survey of research on context-aware homes," Proc. the Australasian information security workshop conference on ACSW frontiers 2003, Vol. 21, pp. 159-168, Feb. 2003.
- [6] M. Minoh and T. Yamazaki, "Daily life support experiment at ubiquitous computing home," Proc. the 11th Information Processing and Management of Uncertainty in Knowledge-Based Systems International Conference(IPMU 2006), Jul. 2006.
- [7] M. Nakamura, H. Igaki and K. Matsumoto, "Feature interactions in integrated services of networked home appliances -an object-oriented approach-," Proc. Int'l. Conf. on Feature Interactions in Telecommunication Networks and Distributed Systems (ICFI'05), pp.236-251, Jun. 2005.
- [8] OSGi Alliance: <http://www.osgi.org/>, 2006.
- [9] Phidgets Inc.: "Unique and easy to use usb interfaces," <http://www.phidgets.com/>, 2003.
- [10] TOSHIBA Consumer Marketing Corp.: "Toshiba home network - femininity", http://www3.toshiba.co.jp/femininity/femininity_eng/index.html, 2005.