

## 家電機器連携サービスにおけるサービス競合の検出

井垣 宏<sup>†</sup> 串戸 洋平<sup>†</sup> 石井 健一<sup>†</sup> 中村 匡秀<sup>†</sup> 松本 健一<sup>†</sup>

<sup>†</sup> 奈良先端科学技術大学院大学 情報科学研究科 〒630-0192 奈良県生駒市高山町 8916-5

E-mail: †{hiro-iga,youhei-k,keni-i,masa-n,matumoto}@is.naist.jp

**あらまし** ホームネットワークシステムの一アプリケーションとして、複数の家電機器を連携・制御し、ユーザの日常生活における快適性・利便性を高める家電機器連携サービスの研究・開発が進んでいる。このような連携サービスは単独では正常に動作するが、複数同時に実行されると互いに干渉・衝突を起こし、結果としてユーザの意図したとおりに動作しなくなる可能性がある。複数のサービス間に発生するこの種の不具合のことをサービス競合と呼ぶ。本稿では、家電機器連携サービスにおける2種類のサービス競合(機器競合、環境競合)を定義し、検出する方法を提案する。機器競合とは、同じ機器において複数のサービスが呼び出す機能が衝突する直接的な競合である。一方、環境競合は異なる機器の機能が、環境の条件に対して間接的に衝突する競合である。ケーススタディでは、具体的な家電機器における連携サービスに対して提案手法を適用することで、実際にサービス競合の検出を行っている。

**キーワード** ホームネットワーク、家電連携サービス、サービス競合、検出

## Detecting Feature Interactions in Integrated Services of Networked Home Appliances

Hiroshi IGAKI<sup>†</sup>, Youhei KUSHIDO<sup>†</sup>, Ken-ichi ISHII<sup>†</sup>, Masahide NAKAMURA<sup>†</sup>, and Ken-ichi MATSUMOTO<sup>†</sup>

<sup>†</sup> Graduate School of Information Science, Nara Institute of Science and Technology 8916-5, Takayama, Ikoma, Nara, 630-0192 Japan

E-mail: †{hiro-iga,youhei-k,keni-i,masa-n,matumoto}@is.naist.jp

**Abstract** The integrated services are one of the major applications of the home network system. Features of multiple networked appliances are combined to provide more comfortable living for home users. When multiple integrated services are executed simultaneously, the features executed in the services may conflict with each other. This conflict is generally known as the feature interaction problem, which decreases the total quality of services. This paper formulates the feature interaction problem in the integrated services of the home network system. Specifically, we define two types of interactions: device interactions and environment interactions. The device interactions are direct conflicts of features on the same device, whereas, the environment interactions are indirect conflicts on certain environment properties. We conduct a case study of interaction detection to demonstrate the effectiveness of the proposed method.

**Key words** home network, integrated services of home appliances, feature interactions, detection

### 1. はじめに

ユーザの日常生活の利便性を向上させるため、エアコン、照明機器、AV機器等の家電製品を家庭内のネットワークに接続し、複数機器の連携や宅外からの操作を実現するシステムの普及が進んでいる。このようなシステムを一般にホームネットワークシステム(HNS)と呼び、近年いくつかの製品が商品化されている[6][7][8][10]。

複数の家電機器を連携動作させる家電機器連携サービス(以降、連携サービスと呼ぶ)はHNSのアプリケーションの一つとして研究・開発が進められている[12][14]。以下に連携サービスの例を示す:

**DVD Theater サービス:** DVDの電源を入れると、TVがDVDモードでONになり、スピーカが5.1チャンネルで起動する。また、部屋のブラインドが閉まり、照明が暗くなる。

**帰宅サービス:** ユーザがドアを開けて帰宅すると、部屋の照

明が点き、エアコンにより室温が調整される。

HNS の付加価値を高めるため、様々な連携サービスが開発されている。また、ユーザ自身が自由に連携サービスを構築できる環境 [9] の開発も期待される。

しかしながら、複数の連携サービスが各家庭の HNS 上に導入されると、単独では正常に動作する連携サービスが互いに干渉・衝突を起こし、ユーザの意図した通りに動作しなくなる可能性がある。例えば、上記の二つの連携サービスの場合、あるユーザ A が DVD Theater サービスを実行しているときに別のユーザ B が帰宅したとする。この時、ユーザ A は DVD Theater サービスを実行中で、照明を暗くしておいて欲しいと思っているが、ユーザ B が帰宅することによって帰宅サービスが実行され、照明が明るくなってしまふ。結果としてユーザ A の意図とユーザ B の意図が照明機器において衝突してしまうこととなる。

本稿では、このような連携サービス間の衝突を HNS におけるサービス競合 (Feature Interactions) と呼ぶ。サービス競合は、従来主に電話通信システムの分野で行われてきた研究分野 [2] であり、個々のサービスは正常に動作するが、複数組み合わせると発生する不具合を指してきた。

上で述べたように HNS においてもサービス競合は発生する。すなわち、個々の連携サービスは正常に動作するが、複数の連携サービスを同時に利用すると、ユーザの意図や機器の機能が競合を起こす場合がある。この時、サービス競合はユーザの快適性・利便性を損ない、HNS の品質を低下させる要因となる。連携サービスが多様化し、随時動的に変更・追加されるような状況では、こうしたサービス競合を ad hoc に扱うことは不可能である。従って、HNS の連携サービス間のサービス競合を体系的に検出・解消する体系的な枠組みが必要になる。

そこで、本稿では HNS におけるサービス競合問題を検出するための枠組みを提案する。具体的には、HNS における家電機器、および、連携サービスをモデル化し、2 種類のサービス競合 (機能競合、環境競合) の定式化を行う。また、いくつかの実連携サービスに対して適用し、競合検出のケーススタディを行う。以降では、2 章で HNS 連携サービスのモデル化を行う。3 章では 2 種類のサービス競合とその競合を検出する方法について述べる。その後、4 章でケーススタディを行い、5 章で考察と今後の課題について説明する。

## 2. ネットワーク家電連携サービスの形式化

連携サービス間のサービス競合問題を定式化するために、本章では、HNS に収容される家電機器のモデル化、および、連携サービスの形式的な定義を行う。

### 2.1 連携サービスシナリオ

以降の議論における理解を深めるため、連携サービスのシナリオ例を導入する。HNS を構成する要素として 10 の家電機器 (DVD プレーヤ、TV、スピーカ、照明、照度計、ドア、電話、エアコン、温度計、ブラインド) を想定している。以下に、連携サービスとして実現する 7 つのサービスシナリオ例 (以降では  $SS_i$  ( $1 \leq i \leq 7$ ) と書く) を示す。

$SS_1$ : TV の電源を入れると、TV の画面が表示され、スピーカ

表 1 機器プロパティ

DeviceName	DevicePropertyName	PropertyType
AirConditioner	動作状態	ON/OFF
	温度設定値	数値 (単位°C)
Thermometer	動作状態	ON/OFF
	温度計測値	数値 (単位°C)
Speaker	動作状態	ON/OFF
	スピーカ入力	1(TV),2(DVD)
	スピーカチャンネル	2,5,1
	機器音量	数値 (単位dB)
Light	動作状態	ON/OFF
	照度レベル	数値 (単位lx)
Illuminometer	動作状態	ON/OFF
	照度計測値	数値 (単位lx)
Door	開度検知状態	1(Open),2(Close)
	動作状態	ON/OFF
Phone	通話状態	1(着信), 2(呼び出し中), 3(接続), 4(待機)
DVD player	動作状態	ON/OFF
TV	動作状態	ON/OFF
	TV入力	1(TV),2(DVD)
Blind	動作状態	ON/OFF
	開閉設定	開/閉

が 2ch モードで、音量が TV 用に調節されて起動する。

$SS_2$ : DVD の電源を入れると、TV が DVD の出力を表示し、スピーカが 5.1ch で、音量が DVD 用に調節されて起動し、部屋の照明が暗くなりブラインドが閉められる。

$SS_3$ : ユーザが帰宅して Door を開けると、Light の明るさが Illuminometer の示す部屋の明るさをもとに調整される。

$SS_4$ : ユーザが帰宅して Door を開けると、AirConditioner が起動し、Thermometer の示す室温をもとに調整される。

$SS_5$ : 電話を着信すると、スピーカの音量を落とす。

$SS_6$ : 日中は Blind を開ける。

$SS_7$ : 就寝時、あるいは外出時に全ての機器の電源を落とす。

### 2.2 機器のモデル化

HNS に収容される家電機器は、その機器をネットワークから制御するためのインターフェースを有する。より厳密には、各家電機器は、自身の状態を保持する属性 (プロパティ) と、属性を外部から参照または更新するためのいくつかのメソッドから成るオブジェクトとみなすことが出来る [7]<sup>(注1)</sup>。例えばエアコンは、動作状態、温度設定値などのプロパティを持つ。各プロパティには取りうる値を規定する型が存在し、プロパティ  $Prop$  の型を  $tProp$  と書く。例えば、 $t$  動作状態 = {ON, OFF},  $t$  温度設定値 = 数値 (単位°C) 等が与えられる。表 1 に前節で述べた機器のプロパティの例を示す。

メソッドは、機器のプロパティの値を参照・更新するための公開されたインターフェースである。例えば、上記エアコンの例であれば、 $setPower(t$  動作状態 onoff),  $setTemperature(t$  温度設定値 temp) 等のメソッドを持ち、実際には  $setPower('ON')$ ,  $setTemperature(26)$  のようにネットワークから機器を制御する。

各メソッドのモデル化には、様々な抽象度レベルが考えられるが、本稿ではモデルの汎用性を考え、メソッド実行に必要な前条件、および、実行後に成立する後条件でモデル化することにする。前条件および後条件は、それぞれメソッドが依存する

(注1): 機器オブジェクトの標準化は年々進んでおり、例えば、ECHONET [7] では機器のプロパティまで厳密に定義されている。

表 2 家電機器モデル

DeviceName	DeviceMethod	Pre <sub>d</sub>	Post <sub>d</sub>	R <sub>e</sub>	W <sub>e</sub>
AirConditioner	setPower(t動作状態 onoff)		動作状態=onoff		
	setTemperature(t温度設定値 temp)	動作状態='ON'	温度設定値=temp		室温
Thermometer	setPower(t動作状態 onoff)		動作状態=onoff		
	getTemperature()	動作状態='ON'		室温	
Speaker	setPower(t動作状態 onoff)		動作状態=onoff		
	setInput(tスピーカ入力 splInput)	動作状態='ON'	スピーカ入力=splInput		
	setChannel(tスピーカチャンネル spChannel)	動作状態='ON'	スピーカチャンネル=spChannel		
	setVolume(t機器音量 spVolume)	動作状態='ON'	機器音量=spVolume		音量
TV	setPower(t動作状態 onoff)		動作状態=onoff		
	setInput(tTV入力 tvInput)	動作状態='ON'	TV入力=tvInput		
DVD	setPower(t動作状態 onoff)		動作状態=onoff		
	setPower(t動作状態 onoff)		動作状態=onoff		
Light	setBrightness(t照度レベル lx)	動作状態='ON'	照度レベル=lx		明るさ
	setPower(t動作状態 onoff)		動作状態=onoff		
Illuminometer	getBrightness()	動作状態='ON',照度計測値=*		明るさ	
Door	getDoorStatus()	動作状態='ON',開度検知状態=*			
Phone	ringing()	通話状態='着信'	通話状態='呼び出し中'		音量
	conncted()	通話状態='呼び出し中'	通話状態='接続'		音量
Blind	setPower(t動作状態 onoff)		動作状態=onoff		
	setGate(t開閉設定 gateStatus)	動作状態='ON'	開閉設定=gateStatus		明るさ,室温

プロパティの条件で指定する。例えば、上記の setTemperature(t 温度設定値 temp) の実行には、電源が'ON' である必要があり、実行後は温度設定値が temp で指定された値になるという実装であれば、前条件：動作状態='ON'、後条件：温度設定値=temp とモデル化する。

より一般的に、本稿では前条件、後条件を各プロパティの論理式の積で与えるものとする。

**[定義 1] (プロパティ条件)**  $P = \{p_1, p_2, \dots, p_n\}$  を与えられたプロパティの集合とする。この時、 $c = f_{p_1} \wedge f_{p_2} \wedge \dots \wedge f_{p_n}$  ( $f_{p_i}$  は  $p_i$  に閉じた任意の論理式) をプロパティ条件と呼ぶ。P の全てのプロパティ条件の集合を  $Cond_P$  と書く。また、c に対し、 $\prod_{p_i} (c) = f_{p_i}$  を条件 c のプロパティ  $p_i$  に関する射影と呼ぶ。

射影は任意のプロパティ条件から、単一のプロパティに関する論理式を取り出す演算である。以上を踏まえ、HNS に収容する各家電機器を以下のようにモデル化する。

**[定義 2] (家電機器)** 家電機器  $d$  は、 $d = (P_d, M_d, Pre_d, Post_d)$  で定義される。ここで、

- $P_d$  は  $d$  の全ての機器プロパティの集合、
- $M_d$  は  $d$  の全ての機器メソッドの集合、
- $Pre_d : M_d \rightarrow Cond_{P_d}$  は各メソッドの実行に必要な機器に関する前条件。
- $Post_d : M_d \rightarrow Cond_{P_d}$  は各メソッド実行後、成立する機器に関する後条件。

異なる機器に同じメソッド名が重複する場合を考慮し、機器  $d$  のメソッド  $m (\in M_d)$  を、 $d.m$  と書くことにする。

表 2 に家電機器のモデル例を挙げる。表において、\*は任意の値 (don't care) を表す。

### 2.3 環境のモデル化

HNS に収容された家電機器は、HNS が存在する環境と密接な関わりを持つ。環境は HNS ユーザの快適性に直結するため、競合を考察するに当たり重要な概念である。例えば、温度計は環境から現在の室温を参照し、エアコンは運転を通して快適な室温に保つ (即ち、室温を更新する)。

本稿では、環境を全ての機器から参照・更新される大域的な

オブジェクトとしてモデル化する。具体的には、環境オブジェクトは、室温、照度、音量といった (大域的な) プロパティを持ち、各機器のメソッドの実行によって、間接的に参照・更新される。本稿では、各機器メソッドがどの環境プロパティを参照 (Read) または更新 (Write) し得るのみを考慮した緩いモデル化を行う。これは、環境プロパティに対するメソッドのアクセスは、機器プロパティに対するアクセスほど明示的ではないため、機器における前条件・後条件ほど厳密性を追求できないからである。

**[定義 3] (環境)** HNS に収容される全ての機器の集合を  $D = \{d_1, \dots, d_k\}$ 、全ての機器のメソッドの集合を  $M = \cup_{d_i \in D} M_{d_i}$  とする。この時、環境は  $e = (P_e, R_e, W_e)$  で定義される。ここで：

- $P_e$  は全ての環境プロパティの集合、
- $R_e : M \rightarrow 2^{P_e}$  は、各機器メソッドが参照する環境プロパティ (の集合) を指定する。
- $W_e : M \rightarrow 2^{P_e}$  は、各機器メソッドが更新する環境プロパティ (の集合) を指定する。

環境は、想定する部屋の間取りやユーザの嗜好等を考慮の上、与えられるものと仮定する。

本稿における例では、環境プロパティとして室温、照度、音量の 3 つを想定する。また、表 2 において、各メソッドがこれらの環境プロパティにどのように作用するかを  $R_e, W_e$  として与えている。例えば AirConditioner の場合、setTemperature(t 温度設定値 temp) の  $W_e$  として室温が定義されている。すなわち、AirConditioner の setTemperature を実行すると、環境プロパティである室温が更新され得ることを意味している。

### 2.4 HNS と連携サービスシナリオ

ホームネットワークシステムは、収容される家電機器とその環境によって定義される。

**[定義 4] (ホームネットワークシステム)** ホームネットワークシステムは、 $HNS = (D, e)$  で定義される。ここで、 $D = \{d_1, d_2, \dots, d_n\}$  は HNS に収容される全ての機器、 $e = (P_e, Pre_e, Post_e)$  は HNS が存在する環境である。

$HNS = (D, e)$  が与えられた時、機器が公開するメソッド群

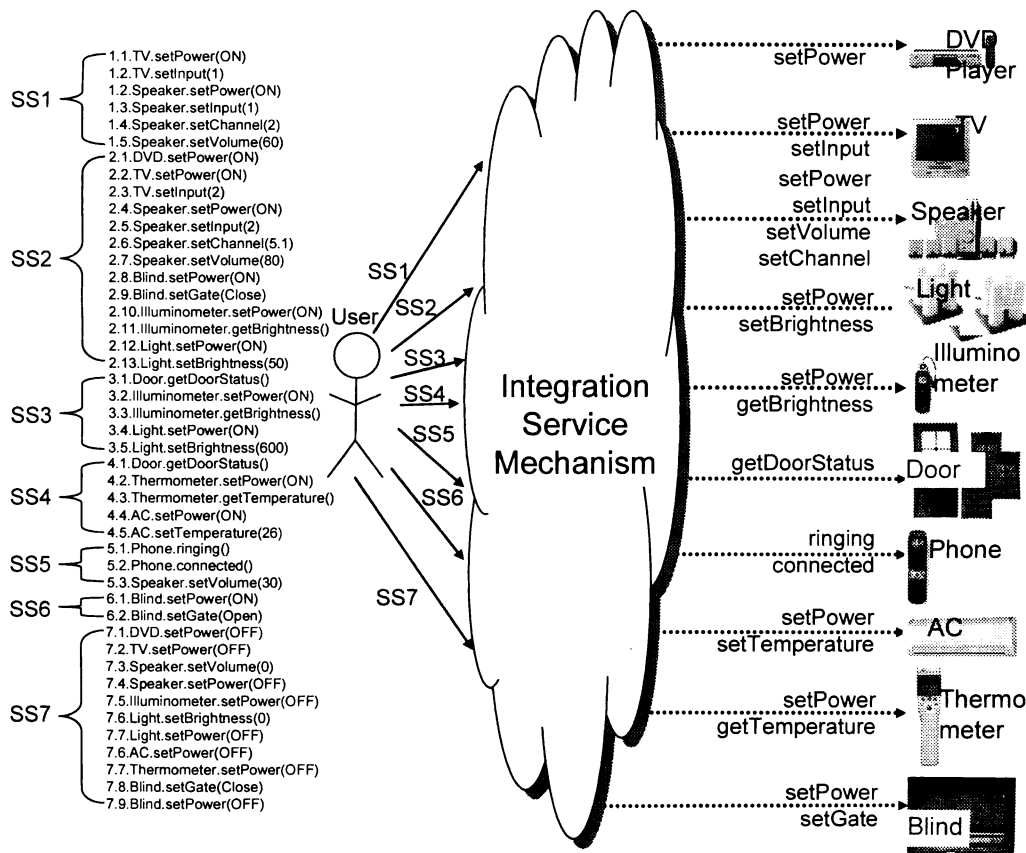


図1 HNS 連携サービス

を適切に選択し、順に実行することで、複数機器の機能を連携した連携サービスを実現できる。このメソッド制御を行う機構(連携機構と呼ぶ)として、ホームサーバ(コントローラ)を用いて中央集権的に行う方式 [8][11][12] や、各機器をインテリジェント化して自律分散的に行う方法 [9] 等が考えられるが、本研究では連携機構に依存しない競合検出法の提案を目指す。

図1にHNSの例を示す。この例では、2.1節で示した7つの連携サービスの一実現例を示している。ユーザが連携機構に連携サービスを依頼(実線矢印)すると、連携機構は当該家電機器のメソッドを順に実行する(破線矢印)。図中、各メソッドはインデクス付けされており、番号  $m.n$  は2.1節における  $SS_m$  において、 $n$  番目に実行されるメソッドであることを意味する。

例えば、 $SS_2$  は以下のように実現されている。

- (1) DVD.setPower(ON)によりDVDの電源がONになる。
- (2) TV.setPower(ON)によりTVの電源がONになる。
- (3) TV.setInput(2)によりTVの入力がDVDに切り替わる。
- (4) Speaker.setPower(ON)により、スピーカの電源がONになる。
- (5) Speaker.setInput(2)により、スピーカの入力がDVDに切り替わる。
- (6) Speaker.setChannel(5.1)により、スピーカが5.1chにセットされる。
- (7) Speaker.setVolume(80)により、スピーカのボリュームが80dbにセットされる。
- (8) Blind.setPower(ON)により、ブラインドの電源がONになる。
- (9) Blind.setGate(Close)により、ブラインドが閉められる。
- (10) Illuminometer.setPower(ON)により、照度計の電源がONになる。
- (11) Illuminometer.getBrightness()により、照度計を用いて部屋の現在の明るさが取得される。
- (12) Light.setPower(ON)により、照明の電源がONになる。
- (13) Light.setBrightness(50)により、照度計が取得した値をもとにして明る

さが50ルクスに調整される。

本稿では、連携サービスが条件分岐の無い一本のシナリオとして与えられるものと仮定し、機器メソッドの呼び出し系列として連携サービスシナリオを定義する。

[定義5] (連携サービスシナリオ)  $HNS = (D, e)$  を与えられたHNSとする。この時、任意の機器メソッドの系列  $ss_i = d_{i1}.m_{i1}, d_{i2}.m_{i2}, \dots, d_{ik}.m_{ik}$  ( $d_{ij} \in D, m_{ij} \in M_{d_{ij}}$ ) を連携サービスシナリオと呼ぶ。

例えば、上で述べた  $SS_2$  は、以下の通りとなる。

$SS_2 = DVD.setPower(ON), TV.setPower(ON), TV.setInput(2), Speaker.setPower(ON), Speaker.setInput(2), Speaker.setChannel(5.1), Speaker.setVolume(80), Blind.setPower(ON), Blind.setGate(Close), Illuminometer.setPower(ON), Illuminometer.getBrightness(), Light.setPower(ON), Light.setBrightness(50)$

### 3. 連携サービス競合

HNSにおいて複数の連携サービスシナリオが同時に実行された場合、連携サービス間で予期しない競合が発生する可能性がある。本稿では、機器競合と環境競合の2種類を提案および形式化する。

#### 3.1 機器競合

機器競合とは、複数のサービスシナリオが、同一機器における相容れないメソッドを同時実行する際に生じる競合を指す。

例として、2.1節の  $SS_1$  と  $SS_2$  を考える。この二つの連携サービスでは、TVとSpeakerにおいて機器のメソッドが共通

して呼び出される。さらに、各連携サービスシナリオによると、例えば Speaker の setChannel メソッドでは  $SS_1$  で setChannel(2) が実行され、 $SS_2$  では setChannel(5.1) が実行される。この時、Speaker.setChannel を通じて、機器プロパティであるスピーカチャンネルに対して同時に '2', '5.1' という異なる値への変更要求が発生することとなり、両方を同時に満たすことが不可能となる。即ち、setChannel メソッドの後条件を同時に満たすことが出来なくなり、競合が発生する。

また、 $SS_1$  と  $SS_7$  の二つの連携サービスが実行される時、 $SS_1$  の TV.setInput(1) は前条件で動作状態が 'ON' であることを要求している。そのため、 $SS_7$  において TV.setPower(OFF) によって動作状態が 'OFF' になると、TV.setInput(1) が正常に動作できず、競合が発生する。この競合は、TV.setPower(OFF) の後条件により、TV.setInput(1) の前条件が成立しなくなるため発生する。

以上の観測から、機器競合は以下のように定義できる。

**[定義 6] (機器競合)**  $HNS = (D, e)$  を与えられた HNS とし、 $ss_i$  および  $ss_j$  を  $HNS$  上で定義された連携サービスシナリオとする。ある機器  $d \in D$  について、 $ss_i$  がメソッド  $d.m_i$  を含み、 $ss_j$  が  $d.m_j$  を含むとする。以下の条件のいずれかが満たされた時、 $ss_i$  と  $ss_j$  は (機器  $d$  において) **機器競合** を生じるといふ。

**条件 D1:** ある機器プロパティ  $p \in P_d$  が存在して、 $\prod_p Post_d(m_i) \wedge \prod_p Post_d(m_j) = \perp$

**条件 D2:** ある機器プロパティ  $p \in P_d$  が存在して、 $\prod_p Post_d(m_i) \wedge \prod_p Pre_d(m_j) = \perp$

### 3.2 環境競合

環境競合は、複数の機器メソッドが共通の環境プロパティに同時にアクセスしようとしたときに発生する。このとき、競合を起こす機器メソッドは、必ずしも同一機器上にあるとは限らないことに注意されたい。

**[定義 7] (環境競合)**  $HNS = (D, e)$  を与えられた HNS とし、 $ss_i$  および  $ss_j$  を  $HNS$  上で定義された連携サービスシナリオとする。また、 $ss_i$  がメソッド  $d.m_i$  を含み、 $ss_j$  が  $d'.m_j$  を含むとする。以下の条件のいずれかが満たされた時、 $ss_i$  と  $ss_j$  は**環境競合**を生じるといふ。

**条件 E1:**  $W_e(m_i) \cap W_e(m_j) \neq \emptyset$

**条件 E2:**  $R_e(m_i) \cap W_e(m_j) \neq \emptyset$

例として、 $SS_3$  と  $SS_6$  の二つの連携サービスが同時に実行されるケースを考えてみる。表 2 から分かるように、 $SS_3$  と  $SS_6$  では  $W_e(Light.setBrightness)$  と  $W_e(Blind.setGate)$  が共通の環境プロパティ: 明るさを更新している。また、 $SS_3$  では  $R_e(Illuminometer.getBrightness)$  が明るさを参照しているため、以上の 3 つのメソッドが環境プロパティ: 明るさにおいて環境競合を起こしているということが検出できる。

### 3.3 サービス競合検出問題

HNS の連携サービスにおけるサービス競合検出問題を以下のように定式化する。

#### サービス競合検出問題

入力: ホームネットワークシステム  $HNS = (D, e)$ , 連携サー

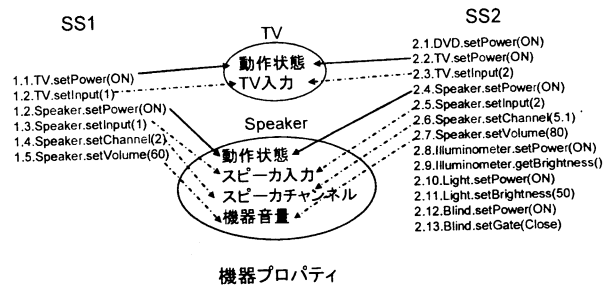


図2  $SS_1, SS_2$  間の機器競合

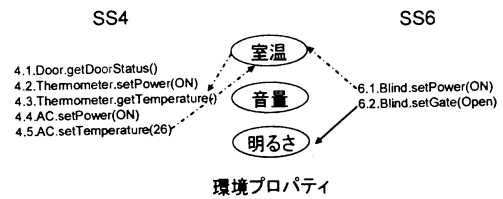


図3  $SS_4, SS_6$  間の環境競合

ビスシナリオ  $ss_1, \dots, ss_n$

出力: 機器競合または環境競合を起こす全てのシナリオのペア。

## 4. ケーススタディ

2.1 節で述べた  $SS_1$  から  $SS_7$  の任意のペアに対して、サービス競合の検出を試みた。検出した機器競合を図 3 に、環境競合を図 4 に示す (表の各エントリには、それぞれの連携サービスの組み合わせにおいて、どの機器メソッドで競合が発生したのかを記述している。空白部分では競合が起らなかったことを示している。また、 $SS_7$  は全ての連携サービスと競合したが、紙面の制限により表からは割愛した)。全体で機器競合が 35 件、環境競合は 33 件を検出した。その中から一例として、図 2 に  $SS_1, SS_2$  間の機器競合検出結果を、図 3 には  $SS_4, SS_6$  間の環境競合検出結果を示した (破線で表示されているプロパティへの通信が競合を表している)。この図 2 では、TV.setInput(), Speaker.setInput(), Speaker.setChannel(), Speaker.setVolume() の各メソッドにおいて機器競合が発生していることを示している。また図 3 では、環境プロパティ: 室温、において  $SS_4$  の Thermometer.getTemperature(), AC.setTemperature() と  $SS_6$  の Blind.setGate() の間で環境競合が生じていることを示している。

このように両方の競合検出において、共通に更新や参照を行う機器・環境プロパティを各メソッドの  $Pre_d, Post_d, R_e, W_e$  に基づいて抽出し、その中から 3.1, 3.2 の各節で定義した競合条件に基づいて、競合検出を行っている。

本稿で述べた競合検出法では、機器プロパティ、環境プロパティに各メソッドが与える影響をもとにして競合を検出した。そのため、このケーススタディの結果からも分かるように、機器モデル、環境モデルの定義さえ行われていれば、メソッド間の関係や機器間関係に新たなモデルを適用することなく競合検出が可能となった。

表3 機器競合検出結果

	SS2	SS3	SS4	SS5	SS6
SS1	TV.setInput Speaker.setInput Speaker.setChannel Speaker.setVolume			Speaker.setVolume	
SS2		Light.setBrightness		Speaker.setVolume	Blind.setGate
SS3					
SS4					
SS5					

表4 環境競合検出結果

	SS2	SS3	SS4	SS5	SS6
SS1				Phone.ringing Phone.connecterd Speaker.setVolume	
SS2		Blind.setGate Illuminometer.getBrightness Light.setBrightness	Thermometer.getTemperature AC.setTemperature Blind.setGate	Phone.ringing Phone.connecterd Speaker.setVolume	Light.setBrightness Blind.setGate Illuminometer.getBrightness
SS3					Illuminometer.getBrightness Light.setBrightness Blind.setGate
SS4					Thermometer.getTemperature AC.setTemperature Blind.setGate
SS5					

## 5. 考察と今後の課題

家電機器等の HNS アプリケーションの多様化、高性能化に伴い、本稿で述べるような複数の連携サービス間の競合検出が重要となってきた。

[1][5] では、サービス競合を検出するために Use Case Maps [3], LOTOS [4] といった形式的記述を利用することで、複数の機能間の関係を記述している。これらの従来手法では、サービス競合を検出するために、複数のサービス間の関係を形式的に記述する必要があり、競合検出の際のコストが増大する。また [13] では、このコストを削減する手法について述べているが、実装レベルに近い関係記述を必要としている。そのため、HNS のように機器やシナリオの動的な追加・変更には対処が困難だと思われる。

また、[14] では、各家電機器の実行が人や環境の持つリソースを消費するという概念を利用することで、環境競合の検出を提案しているが、機器競合の検出にそのまま適用することは困難である。

本稿では、機器および環境プロパティを定義し、HNS をモデル化することで、機器、環境レベルでのサービス競合を検出する方法を提案した。この手法では、サービス間の関係を新たに形式化するというコストをかけずに競合検出を行うことが可能となった。また、一般的な HNS 環境では、機器の種類、実現される連携サービスは非常に多様なものとなることが考えられる。そのような状況を想定した時に、本稿の手法では機器や連携サービスの追加・変更柔軟に対応することが可能である。

一方で、本稿では、全ての機器が同じ部屋にあり、かつ1種類の機器は1つだけしか存在しないという前提でモデル化と競合検出を行った。今後は、同種類の複数の家電機器を含む、より複雑な環境へも対応できるようにモデルを拡張することと、

環境、機器モデルを与えることで自動的に抽出を行えるようなツールの開発を考えている。

## 文 献

- [1] D.Amyot, L.Charfi, N.Gorse et al. "Feature Description and Feature Interaction Analysis with Use Case Maps and LOTOS" *Sixth International Workshop on Feature Interactions in Telecommunications and Software Systems FIW '00. Glasgow, Scotland, 2000.*
- [2] D.Amyot, L.Logrippo (Eds.), "Feature Interactions in Telecommunications and Software Systems VII", *IOS Press, Amsterdam, 2003.*
- [3] R.J.A.Buhr, "Use Case Maps as Architectural Entities for Complex Systems", *IEEE Transactions on Software Engineering, Special Issue on Scenario Management, Volume 24, Issue 12, 1998, pp.1131-1155.*
- [4] T.Bolognesi, E.Brinksma. "Introduction to the ISO Specification Language LOTOS" *Computer Networks and ISDN Systems*, 14, 1986, pp.25-29.
- [5] L. du Bousquet, "Feature Interaction Detection using Testing and Model-checking - Experience Report" *World Congress in Formal Methods, Toulouse, France, 1999.*
- [6] Digital Living Network Alliance - <http://www.dlna.org>
- [7] ECHONET Consortium - <http://www.echonet.gr.jp/>
- [8] Hitachi Home & Life Solutions inc., "horaso network" - <http://www.horaso.com/>
- [9] H. Igaki, M. Nakamura, K. Matsumoto, "Design and evaluation of the Home network systems using the service oriented architecture," *International Conference on E-Business and Telecommunication Networks (ICETE2004)*, August 2004 (To Appear).
- [10] iReady - <http://www.sharp.co.jp/corporate/news/031217-2.html>
- [11] S. W. Loke, "Service-Oriented Device Echology Workflows", *Proc. of 1st Int'l Conf. on Service-Oriented Computing (ICSOC2003)*, LNCS2910, pp.559-574, Dec. 2003.
- [12] Matsushita Electric Industrial Co., Ltd., Kurashi net, <http://national.jp/appliance/product/kurashi-net/>
- [13] A.Metzger, "Feature Interactions in Embedded Control Systems", *Computer Networks, Volume 45, Issue 5, Special Issue on Directions in Feature Interaction Reseach, Elsevier Science, 2004, pp.625-644.*
- [14] Nippon Telegraph and Telephone Corporation, "Home Service Harmony" - <http://www.ntt.co.jp/news/news04/0403/040308.html>