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

信学技報
TECHNICAL REPORT OF IEICE
DE2004-108, DC2004-23(2004-10)

家電機器連携サービスにおけるサービス競合検出システム

井垣 宏† 中村 匡秀† 松本 健一†

† 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,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実装したプログラムを適用することで，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†, Masahide NAKAMURA†, and Ken-ichi MATSUMOTO†

† Graduate School of Information Science, Nara Institute of Science and Technology 8916-5, Takayama, Ikoma, Nara, 630-0192 Japan
E-mail: †{hiro-iga,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う．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 $(1 \leqq i \leqq 7)$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．

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℃)
Thermometer	動作状態	ON/OFF
	温度計測値	数値(単位℃)
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

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](注1)．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℃）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')，setTemprature(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，

(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_d	$Post_d$	R_e	W_e
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',温度計測値=*		室温	
Speaker	setPower(t動作状態 onoff)		動作状態=onoff		
	setInput(tスピーカ入力 spInput)	動作状態='ON'	スピーカ入力=spInput		
	setChannel(tスピーカチャンネル spChanne	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		
Light	setPower(t動作状態 onoff)		動作状態=onoff		
	setBrightness(t照度レベル lx)	動作状態='ON'	照度レベル=lx		明るさ
Illuminometer	setPower(t動作状態 onoff)		動作状態=onoff		
	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
	connceted()	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

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） $P = \{p_1, p_2, ..., p_n\}$ を与えられたプロパティの集合とする．この時，$c = f_{p_1} \wedge f_{p_2} \wedge ... \wedge f_{p_n}$ (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) し得るか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 $D = \{d_1, ..., d_k\}$，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，$D = \{d_1, d_2, ..., d_n\}$ は HNS に収容される全ての機器，$e = (P_e, R_e, W_e)$ 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

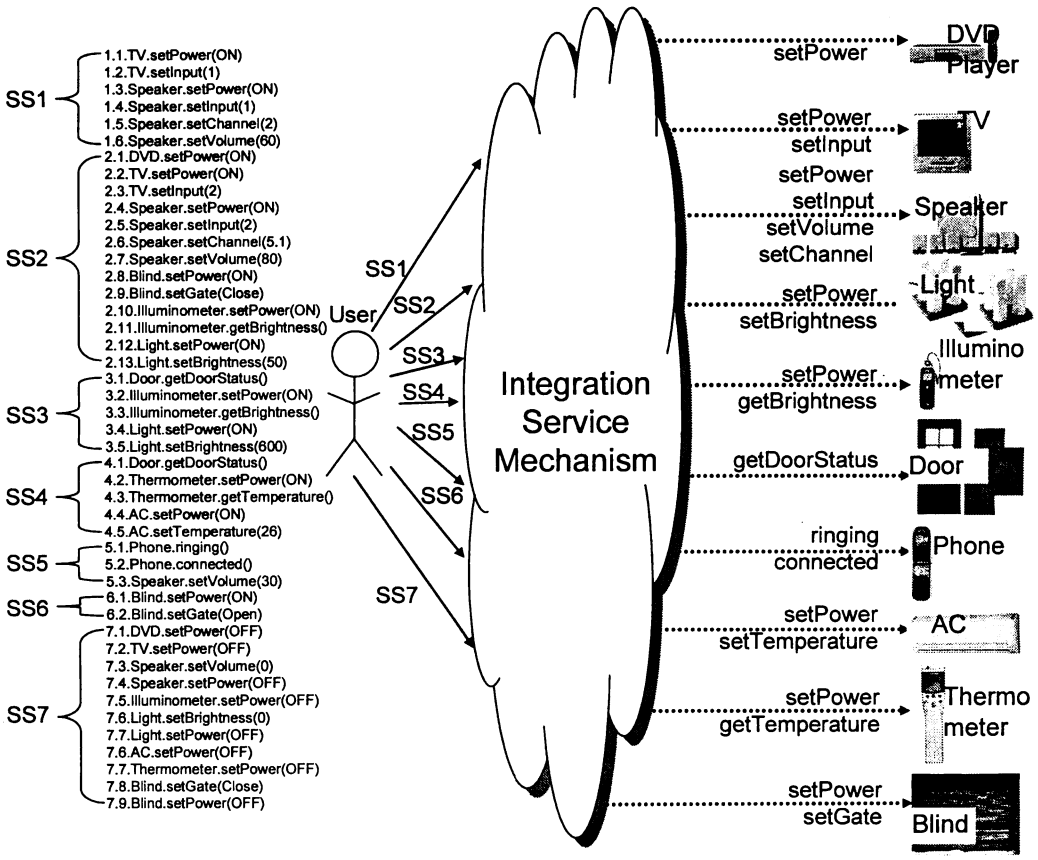

図 1　HNS 連携サービス

(**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.setPpower(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 に切り枠わる．
(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 $ss_i = d_{i1}.m_{i1}, d_{i2}.m_{i2}, ..., d_{ik}.m_{ik}$ $(d_{ij} \in D, m_{ij} \in M_{d_{ij}})$ 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.setInpu(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 種類を提案および定式化する．

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とき，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**という．

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**という．

条件 E1: $W_e(m_i) \cap W_e(m_j) \neq \phi$.

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

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 $W_e(Light.seBrightness)$ 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 $HNS = (D, e)$，連携サービスシナリオ $ss_1, ..., ss_n$

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．

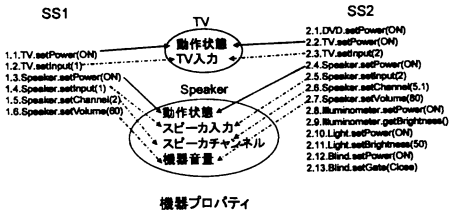

図 2 SS_1, SS_2 の間の機器競合

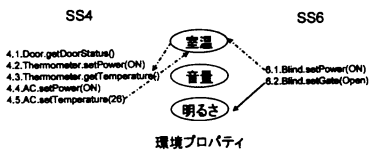

図 3 SS_4, SS_6 の間の環境競合

4. ケーススタディ

本稿では，本稿で例として挙げた HNS と連携サービスシナリオから定義ファイルを構成し，全ての連携サービス間（2.1 節で述べた SS_1 から SS_7）で発生する競合を検出するプログラムを作成した．このプログラムによって検出した機器競合を表 3 に，環境競合を表 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おり，空白部分では競合が起こ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）．機器競合は 35 件，環境競合は 33 件が検出された．

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）．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，HNS および連携サービスシナリオが与えられると，3.1，3.2 によって定義した各競合条件を実装したプログラムによって競合検出が可能であることを確認した．

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，開発者等が設計したサービスの連携機構や各機器の定義ファイルだけでなく，サービス競合を検出するための複数のサービス間の関

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					

係記述が必要となり，競合検出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，実装レベルに近いサービスシナリオ間の関係記述を必要としている．そのため，従来のような手法では，HNS において要求されるような機器やシナリオの動的な追加・変更への対処が困難になる可能性があると考えられる．

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．4. では，提案した競合検出法を実装したプログラムを HNS および連携サービスシナリオに適用することで，提案手法によって以下のような点が実現できることを示した．

(1)　連携サービスシナリオにおけるメソッド間の関係や，機器間の関係を新規にモデル化することなく，機器競合および環境競合を検出することが可能である

(2)　上記のような関係記述が不要なため，機器・連携サービスシナリオの変更にも柔軟に対応することができる

一般的な HNS 環境としては，機器の種類や実現される連携サービスは，非常に多様なものであり，日常的に変更されることが考えられる．そのような状況を想定した時に，本稿で提案した手法のような，変更に柔軟に対応できるという競合検出手法の特徴が重要なものとなる．

本稿のケーススタディでは，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例を利用して競合検出を実際に行った．しかし，提案手法では HNS を部屋単位で定義する等によって，同種類の複数の家電機器を複数の部屋に配置している，といった複雑な HNS 環境においても競合の検出が可能である．

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)*, vol.1, pp.62-69, Setubal, Portugal, Aug. 2004.

[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`