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.

ホームネットワークシステムにおけるサービス競合の動的検出・解消システムの設計と実装

吉村　悠平† 　井垣　宏† 　中村　匡秀†

† 神戸大学　〒 657–8501 神戸市灘区六甲台町 1–1
E-mail: †{yuheiy,igaki,masa-n}@cs.kobe-u.ac.jp

あらまし　ホームネットワークシステム（HNS) のアプリケーションの一つとして，複数の家電を連携制御する家電連携サービスの研究が進んでいる．単体では正常に動作する連携サービスでも，複数を同時に実行すると「サービス競合」と呼ばれる機能の干渉・衝突により，ユーザの意図しない不具合に陥ることがある．HNS のサービス品質を損なわないためにも，サービス競合を検出し解消することが求められる．我々は先行研究において，与えられた連携サービスシナリオから，潜在的な全ての競合をオフラインで検出する静的検出法を提案している．しかし，連携サービスの数が増えると潜在的なサービス競合の数が組合せ的に増加するため，競合対策が困難になる．そこで本稿では，サービス競合を動的に検出するための，オンライン競合検出法を提案する．具体的には，機器メソッド競合監視機構と，競合解消のためのメソッド優先度の導入により，サービス競合を動的に検出・解消するシステムの実装を行った．いくつかの実サービスを用いて実験を行い，想定したすべての競合をオンラインで検出・解消できることを確認した．
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，動的検出・解消

Implementing Run-time Detection and Resolution of Service Interactions in Home Network System

Yuhei YOSHIMURA†, Hiroshi IGAKI†, and Masahide NAKAMURA†

† Kobe University　rokkoudaityou 1–1, nada-ku, Kobe, Hyogo, 657–8501 Japan
E-mail: †{yuheiy,igaki,masa-n}@cs.kobe-u.ac.jp

Abstract　The integrated services in the home network system (HNS) provides value and convinience for home users, by integrating features of multiple home appliances. Even if individual integrated service is implemented correctly, using multiple services together can yield unexpected and undesirable behaviors due to functional conflicts among services. The conflict is generally known as the service interaction problem. In our previous research, we proposed a static method of detecting all potential interactions off-line using given service scenarios. However, the number of all potential interactions is exponential for the number of services, which have made it difficult to manage all interactions. In this paper, we address the run-time detection and resolution of service interactions, which detect and solve the interactions on-line. We have implemented an interaction detection and resolution system, by introducing appliance method run-time monitoring and priority for each appliance method. We have conducted experiments with several practical services, and confirmed that all anticipated conflicts were detected and solved.
Key words　home network, appliance integrated services, service interactions, runtime detection and resolution

1. はじめに

ネットワーク技術の発展と共に，一般家庭にある家電機器を家庭内のネットワークに接続して，宅外からの制御や複数機器の連携を実現する**ホームネットワークシステム (HNS)** の研究開発が進んでおり，近年いくつかの製品が商品化されている [1] [2] [3].

HNS では，テレビや DVD，エアコン，照明，扇風機といった複数の家電を連携制御することで，家電単体で利用する場合に比べてより付加価値の高い**家電連携サービス**（以下，連携サービス）を実現することができる．連携サービスは，ユーザの日常生活における快適性・利便性を高める主要な HNS アプ

リケーションの一つとして研究されている [4] [5]．以下に連携サービスの例を示す．

シアターサービス： テレビ，カーテン，照明を連携して，映画館の雰囲気でユーザがテレビを視聴できるサービス．ユーザがサービスを要求すると，自動的にテレビが ON になり，カーテンが閉まり，照明が暗くなる．

帰宅サービス： カーテンと照明を連携して，ユーザが帰宅した際に照明を明るくし，カーテンを閉めるサービス．

お出かけサービス： HNS 内のすべての機器を連携し，外出時にすべての機器の電源を一括で OFF にするサービス．

HNS に多くの連携サービスが提供されると，利便性が向上する反面，**サービス競合**という新たな問題が発生する [4]．単独で正常に動作する連携サービスを，複数を同時に実行すると互いに干渉・衝突を起こし，ユーザの意図した通りに動作しなくなる現象である．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ため，それらを**検出**し**解消**することが求められる．

上記の連携サービスを使ってサービス競合の例を説明する．ユーザ A がシアターサービスを実行しているときに別のユーザ B が帰宅し，帰宅サービスを実行したとする．このとき，ユーザ A のシアターサービスにより照明が暗くなっているにもかかわらず，ユーザ B の帰宅サービスにより照明が明るくなってしまう．これは照明機器において 2 つの連携サービスの要求が衝突してしまったことによるサービス競合である．この例においては，ユーザ B が帰宅サービスを実行したときに，システムは照明機器に対する競合を検出し，理想的にはうまく解消する必要がある．

我々は先行研究 [7] において，HNS におけるサービス競合の**静的検出手法**を提案している．具体的には，オブジェクト指向モデルを用いて各機器をモデル化し，連携サービスを機器のメソッドの実行系列からなるシナリオとして定義する．各機器メソッドは，その実行前に必要な事前条件，実行後に成立すべき事後条件で性質付けられる．複数の連携サービスを組み合わせた際に，これらの事前・事後条件間に矛盾が生じた場合に，競合であると定義している．

従来の静的検出手法では，与えられた連携サービスシナリオから，シナリオ間に存在する**潜在的な全ての競合**を検出することが可能である．静的検出では，サービスの実行時の振る舞いは考慮しないため，連携サービス提供開始前のオフライン解析においてのみ適用可能である．しかしながら，連携サービスの数が増えると潜在的なサービス競合の数は組み合わせ的に増加するため，オフラインですべての競合対策を行うことが困難になってくる．そうした場合には，競合検出をサービス実行時にオンラインで行い，実際に発生した競合のみを検出・解消する，競合の**動的検出・解消手法**が望ましい．

そこで本稿では，HNS 連携サービスのサービス競合を，動的に検出・解消する新たな手法を提案し，提案手法に基づくシステムを実際の HNS に実装することを目的とする．提案手法では，サービス競合を動的に検出するために，**機器メソッド競合監視機構**（以下，競合監視機構）を導入している．HNS 連携サービスは，複数の機器を制御するために，各機器が持つ機器メソッドを実行する．競合監視機構は，連携サービスが機器メソッドを実行した際に，実行されたメソッドの詳細をリスト化して保持しておく．競合監視機構は，別の連携サービスが起動される毎に，現在実行中の機器メソッドと新たに実行される機器メソッドとが矛盾しないかをチェックし，矛盾するならば競合を検出する．

また，提案手法では，検出された競合を自動的に解消するために，各機器の**メソッド単位の優先度**を導入する．競合監視機構はメソッドの優先度も管理し，競合が発生した際には優先度の高いメソッドを優先的に実行することで，競合を解消する．これにより，サービス競合が発生しても競合が自動解消される．

また本論文では，我々の研究グループで開発している実際の HNS に対して，**サービス競合動的検出・解消システム**の実装を行った．また，いくつかのサービス競合検出・解消実験を行った．その結果，想定したすべての競合をオンラインで検出・解消できることが確認できた．

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，宅内のネットワークに接続された複数のネットワーク家電から構成される．ネットワーク家電は，ユーザや外部エージェントがネットワーク越しに制御できるように，制御 API を備えている．この API 呼び出しを実行するため，ネットワーク家電はプロセッサおよびストレージを持つことが一般的である．HNS の代表的な機能としては，先に記した連携サービスが挙げられ，ユーザの目的を最小限の操作で実現することができる．

我々の研究室では，田中らの提案する手法 [8] をもとに，実際の HNS(CS27-HNS と呼ぶ) を構築している．CS27-HNS では各家電の機能が Web サービス [9] として公開されている．その手法では，赤外線を利用した従来家電を対象とし，PC に接続可能な家電リモコンを用いて家電を制御する．機器ベンダに依存する赤外線レベルのプロトコルをカプセル化し，各機器が持つ機能をベンダ非依存の「サービス」という単位でまとめて，ネットワークに公開している．CS27-HNS は，Apache Axis2 Web サービスを用いて実装され，各機器の操作 (メソッド) は，SOAP または REST 形式で呼び出すことができる．

2.2 HNS におけるサービス競合問題

連携サービスは，複数の家電の機能を用いて付加価値のある機能を実現する．通常，各サービスはそれ単体で矛盾が無いように設計・実装され，正常に動作することが保証される．しかしながら，複数のサービスを同時に実行した場合，サービス間で機能の衝突が起こり，ユーザ意図しない不具合が発生することがある．この問題がサービス競合である．

HNS においてサービス競合が発生する原因は，主に 2 つ考えられる．まず第一には，**機器機能の競合**である．異なるサービスが，同一の家電を互いに相容れない方法（例えば，一方が電源のオン，もう一方がオフ）で使う場合，サービス競合が発生する．第二の原因としては，**環境への影響による競合**であ

表 1　機器プロパティ例

ApplianceName	AppliancePropertyName	PropertyType
TV	power	boolean
	channel	int
	volume	int
Light	power	boolean
	brightness	int
Curtain	openLevel	int

る．一般に，家電の機能は，環境に作用するものが多い．例えば，照明は照度，エアコンは温度に作用する．このとき，異なるサービスが，同一の環境属性に対して相容れない方法でアクセスする際に，サービス競合が発生する．例えば，エアコンの冷房とファンヒータの暖房は，機器機能は競合しないが，温度という環境要因を相容れない方法で更新しようとする．

HNS におけるサービス競合は，サービス数の組み合わせ的に増加するため，場当たり的な方法では対処することが困難とされている．

2.3　先行研究：HNS サービス競合の静的競合検出法

我々は先行研究 [7] において，前節で述べた 2 種類の競合を**機器競合**と**環境競合**として定式化している．さらに，これら 2 種類の競合を形式的に検出するために，HNS の**モデル化手法**を提案している．

このモデル化手法では，各家電を状態 (プロパティ) と操作 (メソッド) を持つオブジェクトとして捉える．まず，各家電は自身の状態を保持する**機器プロパティ**を持つ．表 1 に機器プロパティ例を示す．例えば TV の場合，機器プロパティとして power, channnel, volume を持ち，power は true(電源 ON) か false(電源 OFF) である boolean 型，channel と volume は整数の int 型で与えられる．また，周囲の環境状態を保持する**環境プロパティ**を定義する．これには室温 (Temperature)，照度 (Brightness)，音量 (Volume) 等を想定する．

各家電の操作，すなわち，**機器メソッド**は，これらのプロパティを参照・更新するものとしてモデル化される．具体的には，各メソッドは，

- メソッドの実行に必要な機器に関する事前条件
- メソッド実行後に成立する機器に関する事後条件
- 参照する環境プロパティ
- 更新する環境プロパティ

でモデル化される．表 2 に機器メソッドのモデルの例を示す．例えば TV のメソッド vol(int vol) は，実行後に power が true, volume が引数 vol になっている必要があり，環境プロパティ Volume を更新する．

連携サービスは任意の機器メソッドを組み合わせたものであり，機器メソッドの系列を**連携サービスシナリオ**と呼ぶ．図 1 に連携サービスシナリオの例を示す．

文献 [7] では，このモデル化手法を用いて 2 種類のサービス競合を以下のように定義している：

機器競合： あるサービスに含まれるメソッド m と別のサービスに含まれるメソッド m' について，m の事後条件と m' の事後条件 (または，m' の事後条件) が，同じ機器プロパティにおいて矛盾するとき，機器競合が発生する．

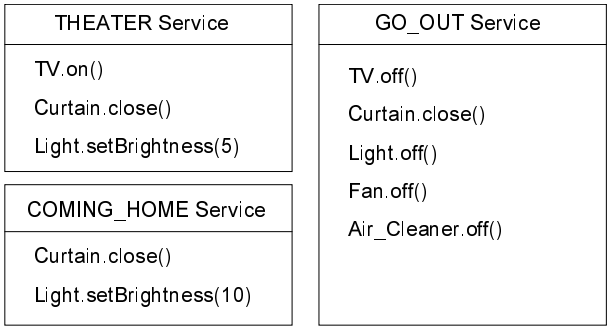

図 1　連携サービスシナリオ例

環境競合： あるサービスに含まれるメソッド m と別のサービスに含まれるメソッド m' について，m と m' が同じ環境プロパティを異なる値で書き換えようとした時，環境競合が発生する．

与えられた連携サービスが含む全ての機器メソッドのペアに対して，上記の競合条件を評価することで，潜在的な全てのサービス競合を静的に検出することができる．例えば，図 1 の潜在的なサービス競合は，以下のようになる：

シアター vs 帰宅： Light の機器競合．照度が異なる競合．

シアター vs お出かけ： TV, Light の機器競合．使用中なのに OFF されてしまう．

お出かけ vs 帰宅： Light の機器競合．ON と OFF の競合．

検出された競合は，サービスシナリオを改訂したり，衝突するシナリオの一方を HNS からアンインストールする等して解消される．

2.4　動的競合検出・解消の必要性

先行研究において，HNS におけるサービス競合を，与えられた連携サービスシナリオから静的に検出してシナリオの編集によって解消したが，この方法では柔軟性に乏しく，機器やシナリオの数・種類が増えるにつれ非常に困難になると予想される．そこで，任意の連携サービスシナリオが実行中の時に，新しいシナリオが実行することで発生する競合の動的な検出・解消方法について記し，それについての実装・評価を行う必要がある．ただし，本稿では**機器競合のみ**に焦点を当てて検証していく．

3.　提 案 手 法

3.1　要　　求

本稿では以下の 2 つを要求を満たすことを目的とする．

要求 R1:　連携サービス競合を実行時に検出する

要求 R2:　検出された競合を自動解消する

基本的に，実行時に発生した競合のみ対象とするため，機器やシナリオの数・種類が増えても，一度に扱うサービス競合の数は比較的少なくすむことが期待できる．

R1 を満たすため，本稿では**機器メソッド競合監視機構**を，また R2 を満たすため，**機器メソッド単位の優先度**を導入する．

3.2　機器メソッド競合監視機構

サービス競合を実行時に検出するには，どのサービスが現在どの機器メソッドを実行中なのかを，絶えずシステムが把握しておく必要がある．新規にサービスが起動された際，現在実行

表 2　機器メソッド例

ApplianceName	ApplianceMethod	PreCondition	PostCondition	ReadEnvironmentProperty	WriteEnvironmentProperty
TV	on()		power=true		
	off()		power=false		
	ch(int c)		power=true,channel=c		
	vol(int vol)		power=true,volume=vol		Volume
Light	on()		power=true,brightness=10		Brightness
	off()		power=false,brightness=0		Brightness
	setBrightness(int level)		power=true,brightness=level		Brightness
Curtain	open()		openLevel=100		Brightness
	close()		openLevel=0		Brightness

中のメソッドと新規に実行予定のメソッドの競合条件を比べ，矛盾があれば競合を検出できる．

そこで，競合監視機構は，現在実行中の機器メソッドを格納しておくための**実行中リスト**を用意する．実行中リストには，現在実行中の機器メソッドごとに，

- サービスシナリオ名
- 機器名
- 機器メソッド名
- 引数
- 優先度

といった詳細を格納する．優先度については後で記す．例えば実行中リストに”THEATER.Light.setBrightness.5.3”という詳細が格納されている時，THEATER サービスの setBrightness(5) が実行中で，その優先度は 3 であるという情報が取得できる．新規に連携サービスを動作させたとき，実行中リストから現在実行中の各機器メソッドの詳細を取得し，それを利用して機器メソッドのモデルを取得する．そして，新規に実行する機器メソッドのモデルと比較することによって競合を検出することが可能となる．

3.3　連携サービスシナリオタイプ

競合監視機構は，サービスによって起動される機器メソッドが「いつまで有効でなければならないか」を明示的に管理する必要がある．そこで，連携サービスシナリオを以下の 3 つのタイプに分類しておく．

Type A:　ユーザが連携サービスを開始してから，明示的にサービスを終了するまで有効なサービス．機器メソッドは，サービス開始時に実行中リストに追加され，サービス終了時に実行中リストから削除される．

Type B:　ユーザが連携サービスを一度開始できれば，後はいつ終了しても構わないサービス．機器メソッドは実行中リストに追加されない．

Type C:　ユーザが連携サービスを開始後，一定時間内は有効なサービス．機器メソッドは，サービス開始時に実行中リストに追加され，ユーザの指定時間後に実行中リストから自動で削除される．

Type A は，ユーザが連携サービスの終了を実行するまで他のサービスからの影響を受けたくない連携サービスに適用する．Type B は機器メソッドを実行中にしておく必要がないサービスに適用する．Type C はその他で，利用時間が決まっているサービスなどに適用する．こうして分類しておくことで，ユーザの要求に合ったサービスを提供することができる．例えば，図 1 のうち，シアターは Type A(映画が終わるまで邪魔されたくない)，帰宅，お出かけは Type B(一度実行すればそれでよい) のように分類可能である．それぞれの動作を図 2 に示す．

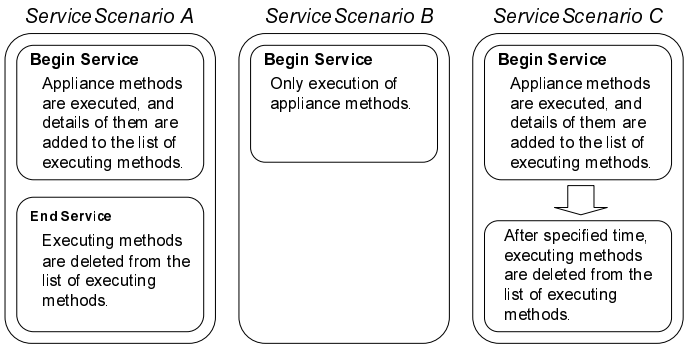

図 2　サービスシナリオ種類別の実行中リストの扱い

3.4　メソッド単位の優先度

R2 を満たすためには，競合を解消するための基準となるものが必要となる．解消方法はいくつか考えられるが，本稿では**機器メソッド単位の優先度**を設定する方法を採用した．これによって，2 つの機器メソッド間で競合が発生した時，優先度が高い方が優先されることになる．ただし今回は，同じサービスシナリオ内の機器メソッドの優先度は等しくし，サービス間に優先順位を与える形で競合の自動解消を行う．

3.5　動的競合検出・解消の流れ

2.3 で述べたとおり，異なるサービス内の機器メソッド m,m' において機器競合が発生する条件は，

- m と m' の事後条件を同時に満たすことができない，
- m の事前条件と m' の事後条件が矛盾する

の 2 つである．競合監視機構は，次の流れにしたがって動的な競合検出・解消を行う．

開始条件：　新規に連携サービスが実行される．

Step 1:　実行中リストから現在実行中の機器メソッドの詳細 (サービスシナリオ名，機器名，機器メソッド名，引数，優先度) を取得する．

Step 2:　現在実行中の機器メソッドと，新規に実行する機器メソッドそれぞれの詳細からメソッドのモデル（事前条件，事後条件）を取得する．

Step 3:　得られた事前条件・事後条件を基づいて，競合条件を評価して競合を検出する．

Step 4: 競合が検出された場合，それぞれの優先度を参照し，それに応じて競合を解消する．

STEP1 のために，実行中リストにはサービスシナリオにおいて機器メソッドが実行される際に，そのメソッドの詳細を格納する．そして新規メソッドとの競合を検出する時，実行中リストに格納されている機器メソッドの詳細を取得する．この詳細によって，どのサービスシナリオでどの機器メソッドが実行されているかという情報が得られ，また優先度も得られるので競合の解消に利用できる．

STEP2 で，それぞれの機器メソッドの詳細からメソッドのモデルを取得することで，競合検出に必要な情報が得られる．

STEP3 のために競合条件を評価するためのモジュールを用意する．このモジュールによって競合が検出されれば，競合結果に応じた競合解消プロセスへの移行が行われる．競合が検出されなければサービスシナリオをそのまま実行する．

STEP4 で新規と実行中それぞれのメソッドに指定された優先度を参照し，優先度が高いメソッドを実行する．新規メソッドの優先度の方が高い場合，新規メソッドが実行中リストに追加されてそのまま実行され，優先度の低い実行中メソッドは実行中リストから削除される．新規メソッドの優先度の方が低い場合，実行中リストはそのままで新規メソッドを実行しない．

4. 実　　装

4.1 サービス競合動的検出・解消システム

2.1 で述べた CS27-HNS 上に，提案手法に基づいた競合検出・解消システムを実装した．今回は，シアターサービス，帰宅サービス，お出かけサービスの 3 つの連携サービスを扱い，3.3 で示したサービスシナリオのタイプの内，シアターサービスには Type A と一定時間でサービス有効期間を終了させる Type C の 2 種類を用意した．これらを以降シアターサービス (A)，シアターサービス (C) と表記する．帰宅サービスとお出かけサービスにはそれぞれ Type B のみの 1 種類を用意し，以降それぞれを帰宅サービス (B)，お出かけサービス (B) と表記する．各シナリオは図 1 で示したシナリオ例のものである．

開発したシステムの主な機能として，次の 3 つが挙げられる．

- 実行したサービスが，サービスシナリオ通りに複数の家電機器を制御する．
- 実行したサービス間で起きた機器競合を検出し，優先度により解消する．
- 競合結果を表示する．競合結果には，競合を起こしたシナリオ名，機器メソッド名と，競合解消により優先されたメソッドが新規に実行したものか実行中のものかという情報が与えられる．

4.2 使用した家電

- テレビ：Panasonic TH-25EA1
- 扇風機：Pieria メタルリビング扇風機
- ライト：Kishima オーブス
- 電動カーテン：Navio Resite
- 空気清浄機：PA-QC13-WX エアブリーズ

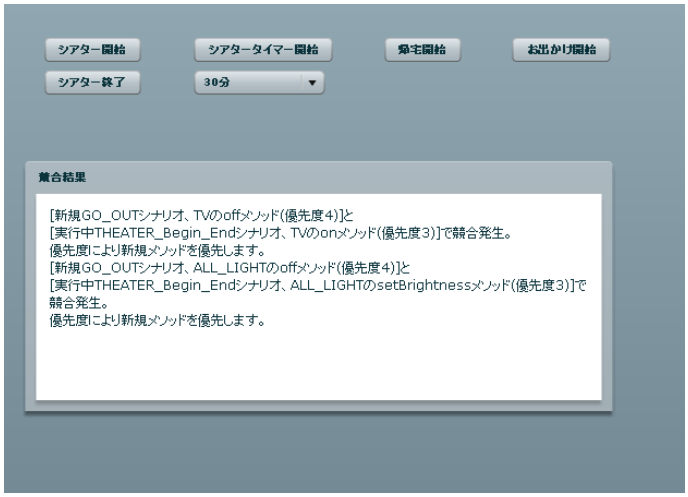

図 3　操 作 画 面

4.3 実 装 環 境

- PC :DIMENSION C521, Athlon64 X2 4600+, 2GB, 250GB
- IrRC I/F：スギヤマエレクトロン - クロッサム 2+USB
- Apache AXIS 2
- Flex Builder3

今回，ユーザインタフェースの設計においては，リッチインターネットアプリケーションのためのフレームワークである Flex 3 Builder を採用した．Flex 3 を利用することで，あらゆるプラットフォームのユーザに対応できる，見ばえの良い操作性に優れたアプリケーションの作成が可能となる．

4.4 操 作 画 面

操作する画面には，各連携サービスを実行するためのボタン，指定時間後に実行解除するサービス用の時間選択ボックス，競合を表示するためのパネルを用意した．

ユーザが任意の連携サービスのボタンを押すことでサービスが実行され，各家電機器がサービスの種類に応じた振る舞いをする．ただし，実行解除時間を指定できるサービスは，時間指定ボックスからあらかじめ実行解除までの時間を選択しておく．

そして，実行中にサービス競合が検出された場合は，設定された優先度により解消される．最後に，実行したサービス内での競合結果が競合表示パネルに表示される．

例として，シアターサービスと帰宅サービスの競合を行った結果の画面とその様子を図 3 に示す．

5. 評 価 実 験

実装した連携サービスを動作させて，機器競合の検出・解消や機器の実行が正しく行われるか，不具合が発生しないかを，実験を通して確認する．本研究では，以下の 3 つの実験を行った．

実験 1　各サービスを単独で実行し，動作を確認する．

実験 2　あるサービス実行中に他のサービスを実行し，競合が検出・解消され，結果が正しく表示・読み上げられるか確認する．

実験 3　実験 2 を，様々なサービスの組み合わせに対して，優

表 3　動作結果例

ServiceScenario	Conflict Method	Prior Method
THEATER(L) vs GO_OUT(H)	<THEATER>TV.on() vs <GO_OUT>TV.off()	<GO_OUT>TV.off()
	<THEATER>Light.setBrightness(5) vs <GO_OUT>Light.off()	<GO_OUT>Light.off()
THEATER(H) vs COMING_HOME(L)	<THEATER>Light.setBrightness(5) vs <COMING_HOME>Light.setBrightness(10)	<THEATER> Light.setBrightness(5)
GO_OUT(H) vs COMING_HOME(L)	No Method	No Method

先度の変更も行った上で確認する．

5.1　実験 1 結果

実験 1 において，各サービスは正確に動作した．シアターサービス (A) では，サービス開始時に各機器メソッドが実行中リストに追加され，テレビが ON になり，カーテンが閉まり，照明が暗くなり，サービス終了時に実行中リストから削除された．シアターサービス (C) では，シアターサービス (A) と同様にサービス開始時に各機器メソッドが実行中リストに追加され，実行された後，指定時間後に実行中リストから削除された．帰宅サービス (B) では，サービス開始時にカーテンが閉まり，照明が点灯し，実行中リストに変化はなかった．お出かけサービス (B) では，サービス開始時に全ての家電機器が OFF になり，実行中リストに変化はなかった．これら全てのサービスの単独動作は全て意図したとおりのものであった．

5.2　実験 2 結果

実験 2 においても各サービスは正しい動作をし，競合の検出・解消を行った．

各サービスの組み合わせに対し，サービス競合が検出された場合は優先度が高い方の機器メソッドが実行中リストに追加，実行されることで解消され，競合結果がパネルに正しく表示された．

5.3　実験 3 結果

手順 3 においても，各サービスの優先度にかかわらず意図したとおりに競合が検出がされ，変更した優先度による競合の解消が行われた．

動作結果例を表 3 に示す．図 1 で示した 3 つのサービスシナリオを実行したときの，シナリオの組み合わせ，その組み合わせで競合が検出された機器メソッド，競合が解決されて優先された機器メソッドを表す．ただし，この例は優先度を帰宅サービスを最も低く，続いてシアターサービス，お出かけサービスの順に高くなるよう設定した場合である．各組み合わせにおいてのシナリオの優先度の高 (H)，低 (L) をシナリオ名の横に表示し，各機器メソッド横に，そのメソッドを実行するシナリオ名を表示してある．

結果的に，実装した連携サービスは問題なく動作し，発生した機器競合は全てサービス実行時に問題なく検出・解消された．提案した動的なサービス競合の検出・解消法の理論が，実際のサービスにも適用できることがわかった．本稿で扱った手法を利用すれば，サービスシナリオの種類や数の増加にも柔軟に対応できると思われる．

6. おわりに

本稿では，HNS アプリケーションの一つである連携サービスを対象に，サービス競合の動的な検出・解消法の提案を行った．さらに，提案手法の実装・評価を行っい，提案した検出・解消法が実際の HNS 連携サービスに十分適用可能であることを確認した．

本稿では，各機器メソッドに優先度を与える形で競合の解消を行ったが，その他にもユーザ間に優先度を与える方法や，ユーザが競合検出時に選択することでの方法などが考えられる．他の解消法に対しても，コストや効率等を考慮しつつ，今後の研究課題として対応していく予定である．また，今回はサービス競合の内の機器競合のみを扱った．今後提案手法を拡張しながら，環境競合についても対応していきたい．

謝　　辞

この研究は，科学技術研究費 (若手研究 B 18700062, 20700027)，および，日本学術振興会日仏交流促進事業（SAKURA プログラム）の助成を受けて行われている．

文　　献

[1] 日立ホーム&ライフソリューション株式会社,“ ホラソネットワーク ”, http://www.horaso.com/

[2] 松下電器産業株式会社,“ くらしネット ”, http://national.jp/appliance/product/kurashi-net/

[3] 東芝,“ 東芝ネットワーク家電 Feminity ”, http://www3.toshiba.co.jp/feminity/about/index.html

[4] 井垣 宏, 中村 匡秀, 松本 健一, ”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,” 信学技報, ディペンダブルコンピューティング研究会, Vol.DC2004-23, pp.11–16, October 2004.

[5] Pattara Leelaprute,Masahide Nakamura,Tatsuhiro Tsuchiya,Ken-ichi Matsumoto,Tohru Kikuno: Describing and Verifying Integrated Services of Home Network Systems, Proc. 12th Asia-Pacific Software Engineering Conference (APSEC ’05),2005

[6] 関本純一, 中村匡秀, 井垣　宏, 松本健一, ホームネットワークにおける家電連携サービス作成支援システムの開発, To Appear

[7] Masahide Nakamura, Hiroshi Igaki, and Ken-ichi Matsumoto, ”Feature Interactions in Integrated Services of Networked Home Appliances -An Object-Oriented Approach-,” In Proc. of Int’l. Conf. on Feature Interactions in Telecommunication Networks and Distributed Systems (ICFI’05), pp.236-251,2005.

[8] 田中章弘, 中村匡秀, 井垣宏, 松本健一, “ Web サービスを用いた従来家電のホームネットワークへの適応 ”, 電子情報通信学会技術研究報告, Vol.105, No.628, pp.067-072, March 2006.

[9] web サービス, http://www.w3.org/2002/ws