

ホームネットワークシステムにおける家電連携サービスのための 競合解消方式の考察

吉村 悠平[†] 池上 弘祐[†] 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: †{yuhey,ikegami,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動作が行われることがある。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、サービス競合の動的な検出と、優先度による解消を行うオンライン競合検出・解消法を提案した。本稿では、より適切なサービス競合の解消を行うために、サービスの有効期間であるアクチベーション、サービス内の重要なメソッドである必須メソッド、サービスの中断再開処理を導入した、サービス競合解消方式の考察を行った。

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、アクチベーション、必須メソッド、中断・再開処理

Feature interaction resolution in integrated services of networked home appliances

Yuhei YOSHIMURA[†], Kosuke IKEGAMI[†], Hiroshi IGAKI[†], and Masahide NAKAMURA[†]

[†] Kobe University rokkoudaityou 1-1, nada-ku, Kobe, Hyogo, 657-8501 Japan

E-mail: †{yuhey,ikegami,igaki,masa-n}@cs.kobe-u.ac.jp

Abstract The integrated services in the home network system (HNS) provides value and convenience 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 feature interaction problem. In our previous research, we proposed a run-time detection and resolution method for service interactions. In this paper, we introduce three new concepts to make the method more reasonable. They are activation which is lifetime of service, mandatory methods which guarantee essential and optional operations in services, and suspend/resume mechanism which allows lower-priority services to sleep temporarily and to wake up later when all conflicting services are terminated.

Key words home network, appliance integrated services, service interactions, runtime detection and resolution, activation, mandatory methods, suspend/resume mechanism

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 レコーダー、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。

DVD シアターサービス：DVD レコーダー、テレビ、スピーカー、カーテン、照明を連携し、映画館の雰囲気ユーザが

DVD 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 DVD レコーダーとテレビが再生モードで ON になり、カーテンが閉まり、照明が暗くなり、スピーカーの 5.1ch が選択され、DVD が再生される．

おかけサービス： ユーザが帰宅した際に、照明を明るくするサービス．

おでかけサービス： 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．

セキュリティ監視サービス： ビデオカメラ、テレビ、DVD レコーダーを連携し、監視を行うサービス．実行すると、ビデオカメラによる撮影が開始され、映像がテレビに映され、DVD レコーダーによって録画される．

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 DVD 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、おかけサービスを実行したとする．このとき、ユーザ A の DVD 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 のおかけサービスにより照明が明るくなってしまふ．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、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動的検出・解消法を提案している．具体的には、連携サービスを各機器のメソッド (以下、機器メソッド) の実行系列からなるシナリオとして定義する．そして、サービス競合を動的に検出するための機器メソッド競合監視機構 (以下、競合監視機構) が、連携サービスを実行した際に実行された機器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、検出された競合を自動的に解消するために、機器メソッド単位の優先度を導入し、競合が発生した際には優先度の高い機器メソッドを優先的に実行することで、競合を解消する．

しかしながら、上記の手法には以下の 3 つの問題点が存在する．

問題 P1: サービスの有効期間が曖昧である．

問題 P2: サービス内の機器メソッドが全て同じ重みである．

問題 P3: 競合解消時に優先度が低い方のサービスが強制終了される．

そこで、本稿では、それらの問題を考慮した、サービス競合解消法を提案する．提案手法では、有効期間を明確化するた

めに、連携サービスに、有効期間としてアクチベーションを明示的に与える．それによって検出の対象となる期間を制御することが可能となる．また、機器メソッドに重みを付与するために、サービス内で必ず実行されなければならない重要な機器メソッドを必須メソッドとして設定する．必須メソッドが実行できない場合は、サービスそのものを実行しない．そして、競合発生時のサービスの強制終了を回避するために、中断・再開処理を導入する．中断・再開処理では、競合発生時に優先度の低いサービスを一時的に中断し、優先された方のサービスが終了した際に再開する．

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 先行研究：HNS サービス競合の動的検出・解消法

我々は先行研究 [6] [7] において、サービス競合の動的検出・解消法を提案している．まず、サービス競合を検出するために、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、機器メソッドの実行系列からなる連携サービス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．図 1 に 4 つの連携サービスシナリオの例を示す．以後、この 4 つのサービスを利用して説明を行う．

そして、検出を動的に行うために、競合監視機構を導入している．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．新規にサービスが実行された際、現在実行中のメソッドと新規に実行予定のメソッドが矛盾するならば競合を検出する．

そこで、競合監視機構は、現在実行中のメソッドを管理するために、実行中メソッドごとに、以下の情報を保持する．

| Priority 4 SS ₁ :DVD-Theater (DVD-T) | Priority 6 SS ₂ :Coming Home (CH) | Priority 10 SS ₃ :Security Monitor (SM) |
|--|--|--|
| <pre>begin() { 1.1. DVD_Recorder.on(); 1.2. DVD_Recorder.changeInput(0); 1.3. TV.on(); 1.4. TV.changeInput(1); 1.5. Curtain.close(); 1.6. Light.setBrightness(1); 1.7. Speaker.changeMode(5.1ch); 1.8. DVD_Recorder.play(); } end () { 1.11. DVD_Recorder.stop(); 1.12. DVD_Recorder.off(); 1.13. TV.off(); 1.14. Light.setBrightness(10); 1.15. Curtain.open(); }</pre> | <pre>begin() { 2.1. Door.notify(OPEN); 2.2. Light.setBrightness(10); }</pre> | <pre>begin() { 4.1. MotionSensor.notify(DETECT); 4.2. VideoCamera.on(); 4.3. TV.on(); 4.4. TV.changeInput(2); 4.5. DVD_Recorder.on(); 4.6. DVD_Recorder.changeInput(1); 4.7. DVD_Recorder.recording(); } end () { 4.11. DVD_Recorder.stop(); 4.12. DVD_Recorder.off(); 4.13. TV.off(); 4.14. VideoCamera.stop(); }</pre> |
| | Priority 8 SS ₃ :Leaving Home (LH) | |
| | <pre>begin() { 3.1. DVD_Recorder.off(); 3.2. TV.off(); 3.3. Curtain.close(); 3.4. Light.off(); 3.5. Speaker.off(); 3.6. VideoCamera.off(); }</pre> | |

図 1 連携サービスシナリオ

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

例えば”DVD_Theater.Light.setBrightness.1.4”という情報からは、「DVD_Theater サービスの setBrightness(1), 優先度 4」というメソッドが実行中であることがわかる。新規に連携サービスを動作させたとき、現在実行中の各メソッドの情報を取得し、それを利用してメソッドのモデルを取得する。そして、新規に実行する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。

また、検出された競合を自動的に解消するために、メソッド単位の優先度を導入している。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。ただし、本稿では同じサービスシナリオ内のメソッドの優先度は等しいものとし、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っている。図 1 に、サービスシナリオに加えて、各サービスの優先度が記してある。優先度が大きいものが優先順位が高くなっている。本稿では、これら 4 つのサービスを利用する際、この優先度に従うものとする。

2.3 先行研究の問題点

先行研究の手法には、1 章で述べた 3 つの問題点がある。各問題点を具体例で説明する。

P1: 連携サービスが競合検出の対象となる有効期間が曖昧である。例えば、ユーザが外出時におでかけサービスを実行したとする。そして、帰宅時におかえりサービスを実行した際、この 2 つのサービスが競合を起こしてしまうという不自然な状況に陥る可能性がある。これは、お出かけサービスの有効期間が、ユーザの帰宅時まで続いていたことが原因である。

P2: 連携サービス内のメソッドが全て同じ重みである。例として、セキュリティ監視サービスが実行中のときに、DVDシアターサービスを実行する状況を考える。このとき、セキュ

リティ監視サービスの、DVDレコーダーを録画モードにするメソッド”DVD_Recorder.changeInput(1)”, 録画するためのメソッド”DVD_Recorder.recording()”が実行中であり、DVDシアターサービスの、DVDレコーダーを再生モードにするメソッド”DVD_Recorder.changeInput(0)”, 再生するためのメソッド”DVD_Recorder.play()”とそれぞれ競合する。そして、DVDシアターサービスの方が優先度が低いので、DVDレコーダーを再生モードにして再生するという制御は行えない。しかし、DVDシアターサービスに含まれる競合を引き起こさない他のメソッドは実行されてしまうので、結果として、DVDシアターサービスにとって重要な、DVDを再生するためのメソッドが実行されず、DVDを視聴できないにもかかわらず、カーテンが閉まり、照明が暗くなるなどの意味のない制御が行われてしまう。これは、サービス内の重要なメソッドとそうでないメソッドの重みが全て同じであることによって発生する。

P3: 競合発生時に、優先度が低い方のサービスが強制終了され、実行できる状態になっても終了したままである。DVDシアターサービスが実行中の際に、セキュリティ監視サービスが実行される場合を考える。このとき、P2の例と同じメソッド間で競合が発生し、優先度の低いDVDシアターサービスの競合したメソッドは強制終了される。そして、後にセキュリティ監視サービスが終了した際、DVDシアターサービスは実行できる状態に戻るが、終了したままである。これは、優先度の低いサービスのメソッドを強制的に終了させてしまったことが問題である。

ユーザの利便性、快適性を損なわないためにも、これらの問題点を考慮した、サービス競合解消が求められる。

3. 提案手法

本稿では、以下の 3 つの要求を満たすことを目的とする。

要求 R1: サービスの有効期間を明確にする

要求 R2: メソッドに重みを付与する

要求 R3: サービスの強制終了を回避する

R1 を満たすため、アクチベーションを、また R2 を満たすため、必須メソッドを、そして R3 を満たすため、中断・再開処

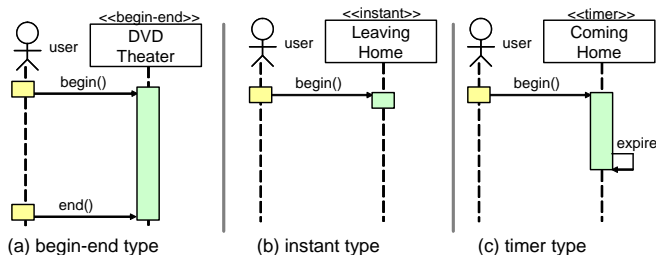


図 2 アクチベーション

理を導入する。

3.1 アクチベーション

サービスが競合検出の対象となる有効期間を明確にするために、各サービスに明示的に有効期間を与える。そこで、有効期間を、アクチベーションとして 3 つのタイプに分類した。

begin-end タイプ: ユーザが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期間が続くタイプ。

instant タイプ: ユーザがサービスを開始すると、メソッドの実行のみが行われ、有効期間が存在しないタイプ。

timer タイプ: ユーザがサービスを開始してから、指定時間後まで有効期間が続くタイプ。

begin-end タイプは、ユーザが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サービスに適用する。instant タイプは、サービスを実行中として管理しておく必要がないサービスに適用する。timer タイプは、利用時間が決まっているサービスなどに適用する。各タイプのシーケンス図を図 2 に示す。サービスごとに、アクチベーションとして begin-end タイプ、instant タイプ、timer タイプのいずれかが適用される。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。全タイプには開始処理である begin() メソッドがあり、終了処理が必要と思われるサービスには end() メソッドを用意した。

本稿では、DVD シアターサービスは、ユーザが DVD を視聴し終わったときに明示的にサービスを終了するものと考え、begin-end タイプを適用する。おでかけサービスは、外出時に家電の電源を OFF にさえできればいいので、instant タイプを適用する。おかえりサービスは、ユーザが帰宅してから一定時間環境を整えるものであり、また、セキュリティ監視サービスは、サービス開始から一定時間の撮影・録画が行われるので、共に timer タイプを適用する。

アクチベーションの導入によって、各サービスに適切な有効期間を設定できるようになり、有効期間が明確になる。したがって、R1 が満たされる。

3.2 必須メソッド

メソッドに重みを付与するために、サービス内において、サービスを成立させるのに不可欠なメソッドを、必須メソッド、不可欠ではないメソッドを通常メソッドとして定義する。そして、全メソッドに、メソッドタイプとして、必須メソッド・通常メソッドのどちらであるかの情報を保持させる。そして、サービスを実行する際は、まずそのサービスが含む全ての必須メソッ

ドが実行可能かどうかをチェックする。実行可能とは、他のメソッドと競合しない、または競合する全メソッドよりも優先度が高いということであり、競合するメソッドより優先度が低い場合は実行不可能である。競合が発生し、1 つでも必須メソッドが実行不可能である場合は、サービス全体を実行しない。また、あるサービスが実行された際、競合した実行中サービスの必須メソッドが 1 つでも実行不可能になった場合は、その実行中サービス全体を終了させる。

例えば、DVD シアターサービスにおいて、DVD レコーダー、TV、スピーカーを制御するメソッドは、サービスの目的である DVD の視聴に不可欠であるので必須メソッドとして、カーテンや照明を制御するメソッドは、不可欠でないので通常メソッドとして設定される。

サービス内の重要なメソッドを必須メソッドとして設定することで、メソッドに重みを付与することができ、R2 が満たされる。

3.3 中断・再開処理

サービスの強制終了を回避するために、中断・再開処理を導入する。サービス競合が発生した際、優先度の低い方のサービスを強制終了させずに、一旦中断する。そして、優先度の高い方のサービスが終了した際、中断していたサービスを再開させる。

ただし、サービスによっては中断・再開を行うべきものとそうでないものがある。そこで、各サービスに再開フラグを用意する。競合により実行中メソッドが実行不可能になった場合、そのサービスの再開フラグが 1 ならば中断・再開処理を行い、0 ならば終了させる。

中断・再開処理を行うためには、競合監視機構において、メソッドが、現在実行中なのか、中断中なのかを把握しておく必要がある。そこで、各メソッドに状態プロパティを用意し、”実行中”、”中断中”のどちらであるかという情報を保持させる。これにより、実行中メソッドと中断中メソッドの両方を管理することができる。競合が発生し、実行中メソッドの優先度の方が低く、その再開フラグが 1 であった場合、メソッドの状態が”実行中”から”中断中”に変更され、中断中として管理される。それが必須メソッドであった場合は、そのメソッドを含むサービスの全メソッドの状態が”中断中”に変更される。サービスを実行する際、中断中のメソッドも競合検出の対象となるが、優先度に関係なく実行予定のメソッドが優先されるものとする。

また、あるメソッドの終了・中断時に、再開可能性があるメソッドを選出するためには、実行中の各メソッドが中断させているメソッドを把握しなければならない。そのために、各メソッドに、中断リストを用意し、メソッドを中断させる際には、中断させるメソッドの情報を格納する。ただし、必須メソッドを中断させることによりサービス全体を中断させる場合は、その必須メソッドを含むサービスの全メソッドの情報を格納する。

あるメソッドを終了・中断させる際には、まず、そのメソッドの中断リストを参照する。再開できる可能性があるのは、中断リストにあるメソッドの内、これから終了させるメソッドより優先度が低い、かつ、他のメソッドと競合していない、また

| | | |
|---|---|--|
| <pre> <<begin-end>> Priority 4 Resumable SS₁:DVD-Theater (DVD-T) begin() { 1.1. * DVD_Recorder.on(); 1.2. * DVD_Recorder.changeInput(0); 1.3. * TV.on(); 1.4. * TV.changeInput(1); 1.5. - Curtain.close(); 1.6. - Light.setBrightness(1); 1.7. * Speaker.changeMode(5.1ch); 1.8. * DVD_Recorder.play(); } end () { 1.11. - DVD_Recorder.stop(); 1.12. - DVD_Recorder.off(); 1.13. - TV.off(); 1.14. - Light.setBrightness(10); 1.15. - Curtain.open(); } </pre> | <pre> <<timer>> Priority 6 SS₂:Coming Home (CH) begin() { 2.1. * Door.notify(OPEN); 2.2. * Light.setBrightness(10); } </pre> | <pre> <<timer>> Priority 10 Resumable SS₃:Security Monitor (SM) begin() { 4.1. * MotionSensor.notify(DETECT); 4.2. * VideoCamera.on(); 4.3. - TV.on(); 4.4. - TV.changeInput(2); 4.5. * DVD_Recorder.on(); 4.6. * DVD_Recorder.changeInput(1); 4.7. * DVD_Recorder.recording(); } end () { 4.11. - DVD_Recorder.stop(); 4.12. - DVD_Recorder.off(); 4.13. - TV.off(); 4.14. - VideoCamera.stop(); } </pre> |
| | <pre> <<instant>> Priority 8 SS₃:Leaving Home (LH) begin() { 3.1. * DVD_Recorder.off(); 3.2. * TV.off(); 3.3. * Curtain.close(); 3.4. * Light.off(); 3.5. * Speaker.off(); 3.6. * VideoCamera.off(); } </pre> | |

図 3 連携サービスシナリオ (提案手法適用後)

は競合する全メソッドよりも優先度が高いメソッドである。再開処理では、再開可能性のある全メソッドに対し、優先度の高いものから順に実行可能かどうかをチェックし、実行可能であれば実行する。再開に伴う競合によって、さらに再開可能性のあるメソッドが生じた場合、そのメソッドについても同様に再開処理を行っていく。

4. 提案手法の適用

4.1 サービス競合検出・解消の流れ

3章で述べた3つの提案手法を適用した競合監視機構による、サービス競合検出・解消の流れを記す。

開始条件：新規に連携サービス S_{new} が実行される。

Step 1: S_{new} と実行中サービスとの競合を検出する。

Step 2: S_{new} が実行可能であるか検証する。

Step 3: 実行中・中断中メソッドを更新する。

Step 4: S_{new} を実行する。

Step 5: 再開可能性のあるメソッドの再開処理を行う。

Step 6: 終了・中断するサービスを実行する。

Step1では、2.2で紹介した機器のモデル化を利用して、 S_{new} の全メソッドと実行中サービスの全メソッドとのサービス競合を検出している。Step2では、検出結果を用いて、 S_{new} の全メソッドが実行可能かどうかを調べる。実行不可能な場合、 S_{new} は実行されない。実行可能な場合はStep3に進む。Step3では、優先度やメソッドタイプを考慮した解消結果に応じて、中断リストへ中断されるメソッドを格納し、メソッドが実行中や中断中になる際の状態プロパティの変更を行う。Step4では、 S_{new} の実行可能なメソッドを実行し、各機器の制御を行うと共に、 S_{new} をアクチベーションに応じて、実行中として管理する。Step5では、競合解消によるメソッドの終了に伴って生じた再開可能性のあるメソッドを、3.3に記述した再開処理にかける。Step6では、終了処理が必要なサービスの機器制御を行う。

4.2 解消例

2.3で述べた以下の3つのケースにおいて、提案手法を用いることで問題が解決されることを確認する。

Case1: おでかけサービスの実行後に、おかえりサービスを

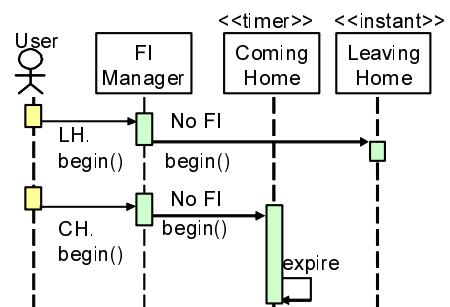


図 4 シーケンス (Case1)

実行する。

Case2: セキュリティ監視サービスの実行中に、DVDシアターサービスを実行する。

Case3: DVDシアターサービスの実行中に、セキュリティ監視サービスを実行する。

まず、図1に記した各サービスに対し、提案手法を適用したものを図3に示す。

各サービスには、アクチベーションが与えられており、3.1で記したように、DVDシアターサービスにはbegin-endタイプ、おでかけサービスにはinstantタイプ、おかえりサービスとセキュリティ監視サービスにはtimerタイプを適用している。また、サービスが含む全メソッドは、必須メソッドが通常メソッドのどちらかに設定してあり、必須メソッドには“*”が記してある。そして、DVDシアターサービスとセキュリティ監視サービスは、再開フラグを1としているので、中断・再開を行うサービスを意味する“Resumable”が表記してある。

Case1では、おでかけサービスとおかえりサービスが、競合を引き起こしてしまうことが問題であった。しかし、アクチベーションの適用により、おでかけサービスにinstantタイプが適用されたので、サービスを実行すると機器の制御は行われるが、有効期間は存在しない。したがって、おでかけサービスの実行後におかえりサービスを実行しても、おでかけサービスは競合の対象とはならないので、競合は発生しない。シーケンスを図4に示す。

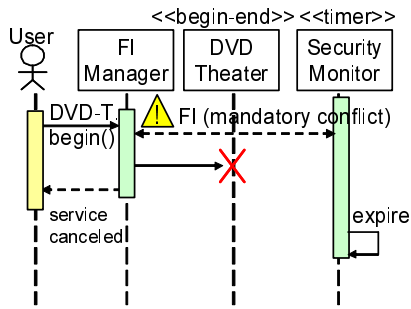


図 5 シーケンス (Case2)

Case2 では、DVD シアターサービスの実行時に、競合によって DVD を視聴できないにもかかわらず、競合していない機器の無意味な制御を行ってしまうことが問題であった。セキュリティ監視サービスの実行中に DVD シアターサービスを実行すると、図 3 における DVD シアターサービスのメソッド (1.2), (1.4), (1.8) と、セキュリティ監視サービスのメソッド (4.6), (4.4), (4.7) がそれぞれ競合する。また、優先度は DVD シアターサービスの方が低いので、メソッド (1.2), (1.4), (1.8) は実行することができない。ここで、DVD シアターサービスにおいて、DVD レコーダーを視聴するために必要なメソッド (1.1), (1.2), (1.3), (1.4), (1.7), (1.8) は、必須メソッドとなっており、今回競合するメソッドは全て必須メソッドである。必須メソッドが実行不可能な場合は、サービス全体を実行しないので、DVD シアターサービスのメソッドは全て実行されない。よって、無意味な機器の制御はなくなった。シーケンスを図 5 に示す。

Case3 では、DVD シアターサービスの実行中にセキュリティ監視サービスが実行され、競合によって DVD を視聴できなくなった後、セキュリティ監視サービスが終了しても視聴できないままであるということが問題であった。DVD シアターサービスの実行中にセキュリティ監視サービスを実行すると、Case2 と同じメソッド間で競合が発生する。そして、実行中の DVD シアターサービスは、必須メソッドが競合の発生により実行不可能になるので、サービス全体を止める必要がある。ここで、DVD シアターサービスは再開フラグが 1 なので、中断・再開を行うため、サービスに含まれる全てのメソッドが“中断中”として管理される。また、セキュリティ監視サービスの競合しているメソッドの中断リストに、DVD シアターサービスのメソッドの情報が格納される。セキュリティ監視サービスが終了すると、中断リストにある DVD シアターサービスのメソッドが再開可能か調べられ、再開可能であれば再開される。よって、ユーザは再び DVD を視聴できる。シーケンスを図 6 に示す。

3 つのケースにおいて問題が解決できたことで、提案手法が実際のサービスの問題を解決できることが確認された。

5. 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とした、従来のサービス競合解消法の問題点を挙げ、それを考慮した解消方式の提案を行った。そして、提案手法を

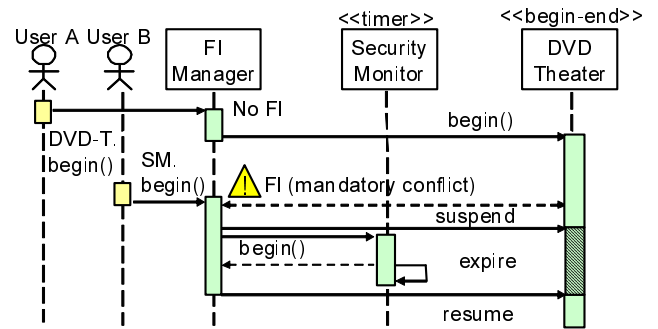


図 6 シーケンス (Case3)

適用することで問題点が解決できているか考察し、解決できていることを確認した。

今後は、我々が提案手法を用いて検出・解消システムを実装し、他に提案されているサービス競合検出・解消法 [10] [11] と比較し、評価していく必要がある。

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] パナソニック電工株式会社, “ライフフィニティ”, <http://denko.panasonic.biz/Ebox/kahts/>
- [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, ディベ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] 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.
- [7] 吉村悠平, 井垣宏, 中村匡秀, “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”, 電子情報通信学会技術研究報告, Vol.108, No.136, pp.35-40, July 2008.
- [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>
- [10] M. Wilson, M. Kolberg, and E. H. Magill, “Considering side effects in service interactions in home automation - an online approach,” in Feature Interactions in Software and Communication Systems IX (L. du Bousquet and J.-L. Richier, eds.), pp. 172-187, IOS Press, Amsterdam, 2007.
- [11] M. Kolberg, E. H. Magill, and M. Wilson. Compatibility issues between services supporting networked appliances. IEEE Communications Magazine, 41(11):136-147, November 2003.