2005－ＤＳＭ－37（17）
2005／5／13

# リソース間相互作用による状態遷移管理に基づく動的リソース管理方法

土川　公雄　片山　和典　中村　喜宏　伊藤　文彦

NTTアクセスサービスシステム研究所　〒305-0805 茨城県つくば市花畑 1-7-1

E-mail: {kimio, katayama, n.yoshi, f.ito}@ansl.ntt.co.jp

**あらまし**　ホームセキュリティ，エネルギー管理など，多種多様なサービスが共存することが予測されるホームネットワークでは，各サービスは，家電機器や環境などの共通のリソースをサービス間で共有して各々のサービスを提供することが一般に考えられる．このような状況において，各サービスが機能を果たすには，サービス間で生じる相互作用を回避する必要がある．サービス間の相互作用には，２つ以上のサービスが同一の家電機器を同時に使用する際に生じる比較的単純な競合と，環境などの抽象的なオブジェクトを媒介として生じる複雑な相互作用が考えられる．本発表は，後者の相互作用も検出，回避可能な仕組みを実現することを目指し，家電機器だけではなく，環境などの抽象的なオブジェクトもリソースとして取り扱い，さらにリソース操作時に生じる他リソースへの影響も定義したリソース管理のためのフレームワークを提案する．

**キーワード**　ホームネットワーク，抽象サービス，情報家電，サービス競合

# Dynamic Resource Management Method Based on State Transition of Resources Caused by Interaction between Resources

Kimio TSUCHIKAWA　Kazunori KATAYAMA　Yoshihiro NAKAMURA　and　Fumihiko ITO

NTT Access Network Service Systems Laboratories　1-7-1 Hanabatake, Tsukuba-shi, Ibaraki, 305-0805 Japan

E-mail: {kimio, katayama, n.yoshi, f.ito}@ansl.ntt.co.jp

**Abstract**

It is general that services such as Home Security Service and Energy Management Service provide each service with sharing common resources with services in home network. In such a situation, it is required that bad interaction between services is detected and avoided properly if services can achieve their purpose. There are two kinds of the interaction. One is simple interaction which occurs when more than two services are about to operate one appliance at the same time, the other is complicated interaction which occurs by competition for abstract objects such as environment. We propose a framework which can treat not only appliances but also abstract object as resources which services can operate, and can also define interaction between resources, to detect and avoid both of the interactions properly.

## 1. はじめに

### 1.1. ホームネットワークの現状及び問題点

近年，エアコン，冷蔵庫，洗濯機，照明，人感センサなど，多種多様な情報家電が市場に投入されるようになってきた．また，昨今の凶悪事件の増加により，人々の安全に対する関心は高まっていること，資源枯渇問題，環境破壊などにより，エネルギー管理に対する人々の関心も高まりつつあることなどの背景を背にして，今後，情報家電を用いたサービスが一般家庭に普及することが期待されている．実際，ホームセキュリティ，エネルギー管理，エンターテインメントなどのサービスが考えられている．

これらのサービスの普及が進むと，各家庭に複数のサービスが共存することが想定される．このように複数のサービスが共存する環境においては，サービス間での家電機器のリソースに対する競合などのサービス間での相互作用が問題となる．サービス間の相互作用には，複数のサービスが同一の家電機器に対して同時に作用を及ぼす直接的な相互作用と，目に見えない環境などを媒体として生じる相互作用がある．前者は，家電機器の使用状況のみ管理すれば検知でき比較的検知することは容易である．しかし，環境などの抽象的なオブジェクトを媒介として生じる相互作用はサービスレベルでの検知は一般に困難であると言われている．

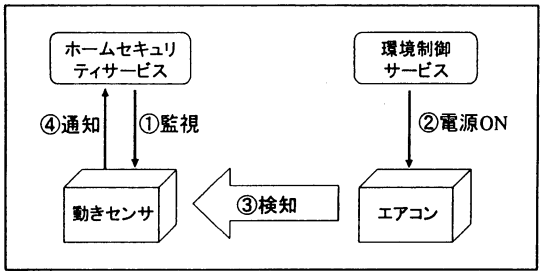


図１．サービス間相互作用の例の例

後者のサービス間での相互作用の例を図１を用いて紹介する．ある家庭において，ホームセキュリティサービスと，環境制御サービスの２つのサービスが同時に動作しているとする．ホームセキュリティサービスは，家の住人が外出している時は，不審者の検知を行っており，不審者が検知された際には，警備会社と住人に通報するよう設計されている．不審者の検知には，動きセンサを使用し，動きがあるかないかで不審者の検知を行う．一方，環境制御サービスは，住人の帰宅時間に合わせて，エアコンの動作を開始するように設計されている．このような状況において，ホームセキュリティサービスが侵入者の検知をしている時に，環境制御サービスがエアコンの動作を開始するとどうなるであろうか．この場合，ホームセキュリティサービスは，エアコンの動作によって生じる動きを侵入者の動きと検知し，警備会社と住人に侵入者ありと通知してしまうことが起こり得る．もちろん，ホームセキュリティサービスのこのような動作は，サービス提供者の意図しない動作であり，望ましくない結果である．

真に情報家電により，人々の暮らしを快適かつ安全なものにするのは，このようなサービス間での相互作用を検知，回避する仕組みが求められる．

### 1.2. 関連研究

先の例を見ても分かるが，家電機器の使用状態を監視するだけでは，サービス間での相互作用を検知するのは難しい．その理由は，家電機器を媒介として相互作用が生じるのではなく，家電機器の動作によって引き起こされた環境の変化を媒介として生じるからである．その点に注目して，環境というキーワードを用いてサービス間の相互作用を取り扱った研究があるので紹介する．

“Compatibility Issues between Services Supporting Networked Appliances”

環境を媒介として生じる相互作用を検出するために，サービス，家電機器，環境の３つの階層を定義し，家電機器の動作により生じる環境への影響も定義されたモデルを用いて，家電機器や環境のリソースにおける競合を検出する仕組みである[1]．このモデルでは，リソースに対する作用として占有のみ定義されており，サービスが家電機器を占有すると，予め組み込まれたルールに従い間接的に環境も占有する．この時のルールは，業界内で合意の取れたルールで行われる．例えば，ブラインドを動作させれば，「動き」という環境と，「温度」という環境に影響を生じる事が予めルールとして決められている．従って，サービスが家電機器を利用すると自動的に環境に対する占有が行われる．つまり，環境に対する作用はサービスの意思に無関係に行われる．また，リソースに対する占有は５種類定義されており，その種類により，その後，同一のリソースに対する占有が発生した場合に，その占有を受け付けるかを判断する．受け付け不可能な操作を受け付けると競合と判断する訳だが，競合が生じた場合には，優先順位により，問題の解決を図る仕組みが提案されている．具体的には，各サービスには予めサービス間で合意が取れた優先順位が付与されており，競合が生じた際には，優先順位の低いサービスから問題のリソースの使用権を剥奪し，高優先のサービスに利用させることで，一応の問題の解決を図っている．

しかし，以下の点が問題点として挙げられる．

◆サービスの環境に対するアクセスは，家電機器の操作を通した間接的な方法に限定される．従って，サービスが自身の希望をダイレクトに表現し，所望のサービスを提供するのは難しいと考えられる．

◆競合が生じた際の対処法は，優先順位に基づく解決のみで，根本的な解決，つまり，なるべく多くのサービスにサービスを提供させることは実現できない．

## 2. 目指すサービスイメージ

本検討では，既存研究の課題を解決し，サービス間の相互作用を検知することはもちろん，適切な手段で相互作用を回避することでなるべく多くのサービスが共存できる環境を構築することを目的とする．

我々の想定するサービスイメージの一例を以下に示す．ある家庭に以下の２つのサービスが同時に動作しているとする．

◆セキュリティサービス

防犯のため動きを監視し，動きを検知すると警報を出す．

◆生活支援サービス

朝８時になると，ユーザに快適な目覚めを提供するために，ブラインドを上げる．

これら２つのサービスを同時に動かすと，朝８時に

セキュリティサービスが警報を出す．この動作は，セキュリティサービスがブラインドの動きを侵入者と誤って検知したために生じたものである．このようなセキュリティサービスの動作は，サービス提供者の意図しない動作である．これを解決するには，以下の３つの手法が考えられる．

①サービス間で予めお互いの動作を把握しておくことで，不具合が生じないように予めサービスを設計しておく．

②既存研究のようにプライオリティによりサービス管理を行い，競合が生じた場合には，優先度の高いサービスを優先する

③ブラインドの上昇動作が完了するまでの時間は，人間が宅内に侵入するのに比べて極短い時間であることを認識し，ブラインドが上昇する間だけセキュリティサービスを止め，ブラインドの上昇動作が完了した後に，セキュリティサービスの動作を再開させる．

①の解決策は，サービス提供者は様々であることを考えると，これらの間で予め合意を取るのは現実的ではない．②については，先にも記述したが根本的な解決に至っていない．そこで，本検討では，ブラインドの動作による動きへの影響を適切に取り扱うことで，③の解決手段を実行できるような仕組みの検討を行う．

## 3. 基本的なオブジェクトの整理

上記を達するために，家電機器や環境などのオブジェクトに対して，以下のようなモデル化を検討した．まず，家電機器や環境などを抽象化して，これらオブジェクトについて基本的な属性や振る舞い(メソッド)を定義し，さらに，オブジェクト間の相互作用を定義する．また，オブジェクトにおける競合を検知することを目的とし，全てのオブジェクトは各オブジェクトに固有の状態により管理される．オブジェクトの状態は，受理可能なメソッドと受理不可能，つまり競合が生じるメソッドを区別する情報と関連付けられ，オブジェクトの状態を見れば，メソッドの受理の可否を判断可能である．また，オブジェクトの状態は，オブジェクトのメソッドを実行すると各オブジェクトについて固有のルールに従い遷移する．

次に，オブジェクト間の相互作用の取り扱いについて説明する．各オブジェクトは，自らが具備するメソッドに対して，メソッドを実行することにより生じる他のオブジェクトへの影響を，影響を生じるオブジェクトのメソッドを呼び出すことで実現する．そのため，各オブジェクトは影響を生じるメソッドごとに，影響を与えるオブジェクトと影響に呼応するメソッドを管理する必要がある．

以上を踏まえると，本提案手法においてリソースが管理すべき情報は，以下の５個である．

◆属性（オブジェクトの基本的な性質を表すパラメータ）
◆メソッド(オブジェクトとしての基本的な振る舞い)
◆状態ごとのメソッドの実行の可否（オブジェクトにおける競合の検知）
◆状態遷移ルール
◆各メソッドを実行した際に，影響を与えるオブジェクトとそのメソッド（オブジェクト間の相互作用）

本検討では，モデル化の対象として，家電機器を抽象化した機器オブジェクトと，環境などの抽象的なオブジェクトを抽象化した抽象オブジェクトの２つを取り扱う．以下，これらオブジェクトの設計例を示す．

### 3.1. 機器オブジェクト

機器オブジェクトは，ホームネットワーク上に存在する家電機器を抽象化したモデルである．機器オブジェクトの設計例として，ブラインドと動きセンサについて検討を行った．図２にそれぞれの設計例を示す．ブラインドオブジェクトは，state 属性を持ち，ブラインドの昇降状態を表す．また，メソッドとして，up() メソッドと down()メソッドを持ちそれぞれ，ブラインドの上昇と，下降の操作を表す．そして，各々のメソッドの処理を行った際に生じる他のオブジェクトのへの影響として，動きオブジェクトの createMotion(5)メソッドを呼び出すことが定義されている．後で説明するが，これは動きを５秒間生じる事を意味する．

| | | | | | | |
|---|---|---|---|---|---|---|
| ブラインド | 属性 | 属性名 | 属性の意味 | | | |
| | | state | 昇降状態(up/down) | | | |
| | メソッド | メソッド名 | メソッドの意味 | オブジェクト内での動作 | 影響を与えるオブジェクト | メソッド |
| | | up() | ブラインドを上げる | state = up | 動き | createMotion(5) |
| | | down() | ブラインドを下げる | state = down | 動き | createMotion(5) |
| 動きセンサ | 属性 | 属性名 | 属性の意味 | | | |
| | | motionFlag | 動きの有無 | | | |
| | メソッド | メソッド名 | メソッドの意味 | オブジェクト内での動作 | 影響を与えるオブジェクト | メソッド |
| | | getMotion() | 動きの有無を通知する | motionFlagを通知する | なし | なし |

図２．機器オブジェクトの設計例

また，ブラインドオブジェクトの状態管理については，あるオブジェクトから状態を占有されているか，いないかのみの情報によって，オブジェクトにおけるメソッドの受け付けの可否は判断可能である．

### 3.2. 抽象オブジェクト

抽象オブジェクトとは，具体的な家電機器を表す機器オブジェクトに対して，サービスシナリオを記述する際に必要な概念や事象を表し，例えば，家電機器により作用される環境（室温，動き，明るさなど）や，サービスの意図を表現する概念（侵入者検知，電力消費状態，音質・音量・騒音など）などを意味する．これらの抽象オブジェクトは，おおよそ多くの人がある程度共通的に合意できる形で共通モデル化することができると考えられる．本検討では一例として，“動き”を表す動きオブジェクトの設計について検討した．

動きオブジェクトは，セキュリティサービスなどから室内の動きの有無を検知する目的で使用されることを想定したオブジェクトである．動きオブジェクトの性質を図３と図４を用いて説明する．図４は，動きオブジェクトが備える属性とメソッドをその意味と共に示している．また，図５は，メソッドの実行により生じる動きオブジェクトの状態の遷移と，各状態における実行可能なメソッド示している．動きオブジェクトは，主に動きの生成・消滅，固定に関するメソッドを持ち，属性として動きが生成された回数を表すパラメータを持つ．動きを生成するメソッドは，引数を持つものと持たないものがあり，持つものは，決まった時間だけ動きを生じる作用を表し，例えば，ブラインドの上昇・下降動作や，プリンタの印刷処理などの場合に用いられる．また，持たない場合は，継続的な動きを生じる作用を表し，ユーザが明示的に終了を宣言するまで動きに作用し続ける動作を表す．例えば，扇風機の動作などが該当する．

また，動きを固定するメソッドは，引数を持ち，引数は時間を表し，引数の時間内で終わる動作なら動きに対する固定を放棄しても良いことを表す．0 が指定された場合は，他からの動きへの影響は受け付けないことを意味する．lockMotion(time)メソッドの実際の内部動作は，呼び出し元のオブジェクトとその時の引数の値の記録と状態遷移である．lockMotion(time)メソッドを受け付けた後に，createMotion(time)メソッドを受け付けると lockMotion(time)で指定された時間と，createMotion(time)で指定された時間を比較して，createMotion(time)で指定された時間の方が短い場合には，lockMotion(time)メソッドの呼び出し元のオブジェクトを止め，その間に createMotion(time)メソッドの呼び出し元のオブジェクトの処理を実行させる．機器オブジェクトにおいても，これらと同種のメソッドや状態を定義することで占有状態を定義することができると考えられる．

動きオブジェクトの getMotion()メソッドは，室内の動きの有無を取得するメソッドであるが，実現するためには実際にこの機能を実現する機器や機能を何らかの方法で検索し，実行する必要がある．このような抽象的な機器操作に関する検討は既に複数存在する[2][3]．今回は簡単のため，既存研究[3]で示されている以下のような制御を想定する．動きオブジェクトには，予め getMotion()メソッドを実現する手段として，動きセンサの getMotion()メソッドを呼び出すことが実装されているとする．

以上が，動きオブジェクトの設計例であるが，図５の状態遷移図を見ても分かるようにかなり複雑な状態遷移を取ることが分かる．さらに各状態において単純にメソッドの受理の可否を判断できず，時間という情報を考慮に入れて判断を行う必要があることが分かる．

## 4. 動作例

これまでは，システムの静的な側面を説明してきた．ここではオブジェクトが連携して動作する動的な側面を図５を用いて説明する．例として先のサービスシナ

| | | | | |
|---|---|---|---|---|
| 動き | 属性 | 属性名 | 属性の意味 | |
| | | createMotionTimes | createMotionメソッドが呼ばれた回数 | |
| | メソッド | メソッド名 | メソッドの意味 | オブジェクト内での動作 |
| | | createMotion() | 継続的な動きを増加させる | createMotionTimes = createMotionTimes + 1 |
| | | deleateMotion() | 継続的な動きを減少させる | createMotionTimes = createMotionTimes − 1 |
| | | createMotion(time) | time秒の間、動きを生じる | createMotionTimes = createMotionTimes + 1をして、time秒後、createMotionTimes = createMotionTimes − 1 |
| | | getMotion() | 動きの有無を取得する | 動きセンサのgetMotion()メソッドを呼び出し、結果を呼び出し元へ通知する |
| | | getCreateMotionTimes() | createMotionが呼ばれた回数を取得する | createMotionTimesを呼び出し元へ通知する |
| | | lockMotion(time) | 動きを固定する。timeが指定されている場合は、その時間だけ固定の解除を許 | 呼び出し元オブジェクトとtimeを記録 |
| | | unlockMotion() | 動きの固定を解除する | 呼び出し元オブジェクトとtimeの破棄 |

図３．動きオブジェクトの設計例

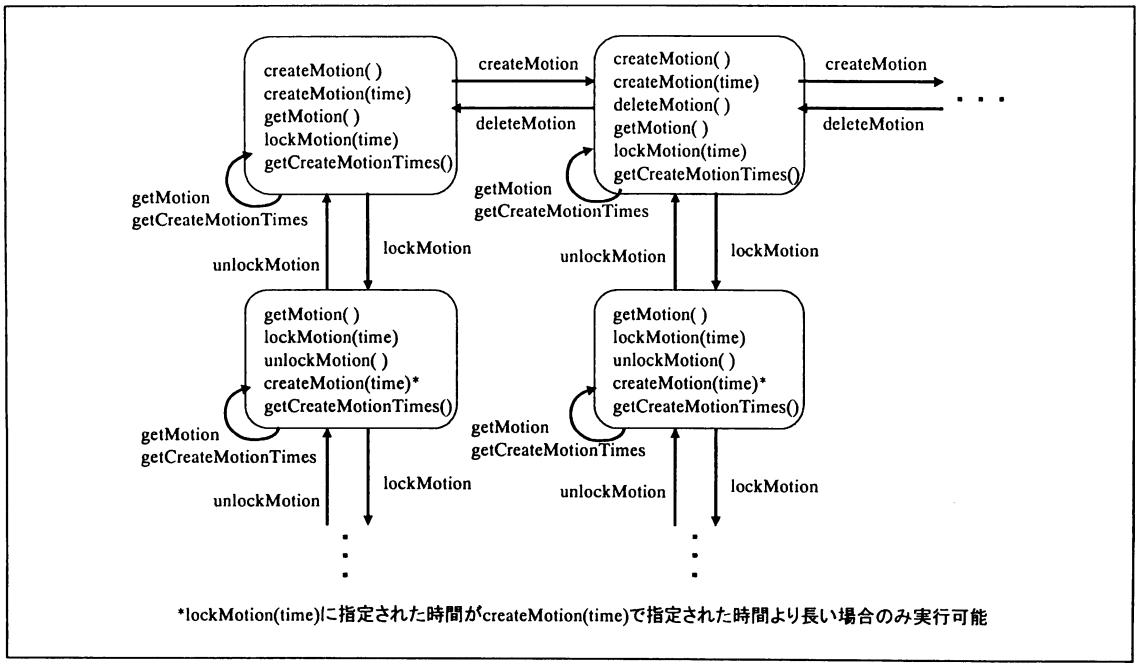


図４．動きオブジェクトの状態管理

リオを取り上げる．動作シナリオは，以下の通りである．

まず，セキュリティサービスが動作を開始するところから説明を始める．

①セキュリティサービスは，侵入者検知を目的として，動きオブジェクトの lockMotion(10)メソッドと getMotion()メソッドを呼び出す．

②動きオブジェクトは，lockMotion(10)メソッドに対する処理として，lockMotion(10)メソッドの呼び出し元としてセキュリティサービスの記録と時間 10 秒を記録する．また，getMotion()メソッドに対しては，動きセンサの getMotion()メソッドを実行する処理を行う．

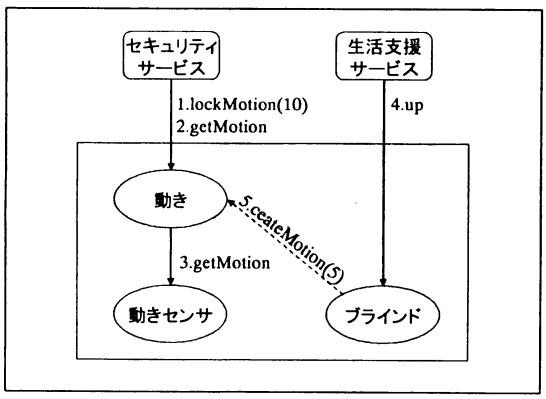


図５．動作例

続いて，生活支援サービスがブラインドの up()メソッドを呼び出した際の動作を説明する．

③生活支援サービスがブラインドの up()メソッドを呼び出す．

④ブラインドは，up()メソッドの実際の処理に先立って，動きオブジェクトの createMotion(5)メソッドを呼び出す．

⑤動きオブジェクトは，lockMotion(time)メソッドで指定された時間と，createMotion(time)メソッドで指定された時間を比較し，createMotion(time)メソッドの受理の可否を判断するが，本ケースでは受理可能と判断する．そして，lockMotion(time)メソッドの呼び出し元であるセキュリティサービスに対して，一時的に動作を止めるように要求する．

⑥セキュリティサービスは，動作を停止したことを動きオブジェクトに通知する．

⑦動きオブジェクトは，ブラインドに動作の許可を出す．

⑧ブラインドは動作を開始する．

⑨５秒後，動きオブジェクトは，セキュリティサービスに再開要求を通知する．

以上が動作例である．

## 5. 考察

### 5.1. 既存研究との関係

本提案手法は，リソースの操作や状態遷移に関して一般的なモデルを構築することで既存研究において残された課題を解決することを目的としていた．以下，本提案手法で既存研究を網羅できることを示し，本提案手法の有用性を示す．

既存研究では，リソースに対する作用は占有のみ取り扱い，リソースに対する占有状態として5種類の占有状態を定義し，各状態に受理可能な占有と不可能な占有を定義していた．従って，本提案手法において，各オブジェクトが具備するメソッドを占有状態を操作するだけの作用と考え，さらに各オブジェクトにおける状態は，占有に関してのみ関連付ければ対応可能であると考えられる．

### 5.2. 提案手法の拡張性

次に一般化したことによる適応範囲の拡張可能性に触れ，提案手法の優位性を示す．本提案手法では，リソースの状態とメソッドを関連付けて管理するモデルを提案した．この仕組みを用いれば，家電機器の操作に関して，知識のないユーザに対する操作のアシスト，意味のない操作の排除，生命や財産に危険を生じるような操作の排除，家電機器を故障させるような操作の排除などを，家電機器の状態を鑑みてメソッドを限定することで実現可能であると考えられる．本論文では家電機器の状態管理については深く議論しなかったが，これを実現するには，電源状態や，機器の特性などを考慮して，家電機器の状態遷移を考える必要があり，今後実現を目指して検討を行いたい．

### 5.3. オブジェクトの構築方法に関する考察

本提案手法では，誰かが何らかの手段でオブジェクトのモデル化を行う必要がある．抽象オブジェクトについては，前章でも触れたが，ある程度合意の取れるものであるので，共通的なモデルを構築することは可能であると考えられる．しかし，家電機器のモデル化に関しては，家電機器は複数のベンダからベンダ独自の様々な機能を実装して提供されるものであり，共通的なモデルを構築してしまうことはベンダ，ユーザ双方に取って不利益を生じることが考えられ，現実的ではない．ベンダ独自の機能を生かすためには，機器ごとにモデル化を行う必要があるが，コストの面で課題が残る．この点に関して，各機器の基本的な機能を実装した基本モデルを共通的に構築し，この基本モデルを継承する形でベンダ独自のモデルを構築する仕組みが考えられる．基本モデルとは，エアコン，ブラインド，照明などの家電機器の種別ごとに準備されるもので，その機器種別について，基本的，共通的な属性とメソッドを持ち，さらに他のオブジェクトへの影響も予め組み込んだものである．ベンダは，独自の機能を新たに実装する場合には，基本モデルを継承することで，その機器の基本的な機能についてはベンダが定義する必要がなく，新機能に該当するメソッドと属性と，さらにそのメソッドの実行によって生じる周囲への影響を表すオブジェクトとメソッドのみを定義することで，新機能を実装したベンダ独自の機器オブジェクトの構築が可能である．また，基本モデルとは異なったリソース間相互作用が生じる場合には，そのメソッドをオーバーライドし，機器特有の処理を記述することで対処可能である．このようにすることで，各ベンダは独自機能を実装した機器オブジェクトをあまりコストをかけることなく構築することが可能であると考えられる．今後は，このような基本モデルの構築を進め，この考えの有用性，実現性を検証したい．

## 6. まとめ及び今後の展望

本検討では，サービス間の相互作用を検知し，適切な手段で相互作用を回避することで，なるべく多くのサービスが共存できる環境を構築するための家電機器，抽象的なオブジェクトのモデル化の検討を行った．そして，サービスシナリオを通して既存研究との差異を明確にして提案手法の有効性を示した．その中で，抽象オブジェクトの一例として動きに関するモデル化を検討したが，その中で既存研究にはない，相互作用を判断する要素を取り扱うことで，なるべく多くのサービスを共存させることができる仕組みを提案した．今後は，さらにモデルケースを増やし，動き以外の抽象オブジェクトについても設計を行い，その中から，家電制御のリソース管理における本質を見出したい．

## 文　献


[1] Mario Kolberg, Evan H. Magill, and Michael Wilson, University of Stirling, Compatibility Issues between Services Supporting Networked Appliances, IEEE Communications Magazine, pp.136-147, November 2003.

[2] Kyeong-Deok Moon, Young-Hee Lee, and Young-Sung Son, Chae-Kyu Kim, Universal Home Network Middleware Guaranteeing Seamless Interoperability among the Heterogeneous Home Network Middleware, IEEE Transactions on Consumer Electronics, vol.49, no3, pp.546-553, August 2003.

[3] 土川公雄，片山和典，伊藤文彦，“抽象サービスによる情報家電制御方法に関する検討，”2004 ソサイエティ大会，B-7-38”.