

ホームネットワークシステムにおける サービス開発を容易化するネット家電標準データモデル

江上 公一[†] 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: †{egami,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) を利用した多様なサービスの研究・開発が進みつつある。しかしながら、HNS サービスと HNS 家電はサービス毎に異なるデータ構造が適用され、密に結合しているのが現状である。そのため、サービス実装の複雑度が増え、開発・運用コストも増加しつつある。本論文では、HNS 上の家電情報を汎用的に利用するための階層的な家電データモデルを提案する。我々のデータモデルは (1)HNS 環境に依存しない標準的な家電クラス情報、(2) サービスに依存しない HNS 環境に含まれる各家電情報、(3) サービス固有の家電情報、という 3 つの階層構造を持つ。実際に複数の HNS サービスを対象に我々のデータモデルを適用し、有効性を検証した。

キーワード ホームネットワークシステム、再利用性、標準化

A Standardized Data Model for Networked appliances in Home Network System

Kouichi EGAMI[†], Hiroshi IGAKI[†], and Masahide NAKAMURA[†]

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

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

Abstract Various Home Network System(HNS) services are being widely developed.HNS appliance information and its data structure in such conventional services differ.As a result, the services and HNS appliances are tightly-coupled.This causes the implementation of the services becomes more complicated.In this paper, we propose a hierarchical HNS appliance data model for improving reusability.Our data model has three layered structures (1) Standardized appliance class information independent of several HNS environment (2) Appliance information included in a HNS environment independent of services (3) Service-specific appliance information.We applied the data model for multiple services, and verified effectiveness.

Key words Home Network System, recycling, standardization

1. はじめに

宅内の各種センサを含む多様な家電機器をホームネットワークに接続し、多種多様なサービスを提供するホームネットワークシステム (HNS) の研究・開発が進みつつある。ネットワークに接続されたテレビやエアコン、扇風機、照明機器、温度計、照度計等の機器 (ネット家電と呼ぶ) を使用することで、これらの遠隔制御 [1] や宅内監視 [2]、複数機器の連携制御 [3] といった付加価値の高いサービスが実際に開発されている。

これらのサービスは一般に、ネット家電が持つ API を組み合わせることで構築されるが、宅内に配置されるネット家電はユーザや家庭ごとに異なる可能性が高い。さらに、現状商品化されているサービスは、単一ベンダ製のネット家電のみに対

応しているものが殆どである。その要因として、ネット家電の API の仕様がそれぞれのベンダによって異なることが挙げられ、このことはサービスとネット家電とを強く結び付けてしまう (密結合化) 原因となる。そのため、サービスのアルゴリズムとそこで利用されるネット家電データの分離 (疎結合化) は、別のユーザ環境におけるサービスアルゴリズムの再利用を容易化する。また、サービスに利用されるネット家電データを特定のモデルにもとづいて標準化し、サービス間で共有することにより、サービス開発時のデータ処理ロジックも再利用可能となり、開発コストのより一層の削減が見込まれる。

そこで本研究では、HNS 上で実現されるサービスと連携可能なネット家電標準データモデルの構築を目指す。我々の提案するネット家電標準データモデルは、以下の 3 つから構成される。

- D1: ネット家電クラス定義
- D2: ネット家電インスタンス定義
- D3: ネット家電サービス定義

ユーザの宅内に配置されるネット家電は非常に多様である。同じ種類の家電であっても提供する機能に差がある場合が非常に多い。そこで本稿では、[4] で述べられているベンダに依存しない標準家電サービスを利用した、この標準家電サービスの持つ標準的なネット家電定義をネット家電クラスとして定義することで、ユーザの宅内環境に依存しないネット家電データを開発者に提供することができる。

ネット家電インスタンス定義では、ネット家電の URI や API 種類等の宅内に配置された各ネット家電への詳細なアクセス方法に関する定義を保持する。ここでいうネット家電インスタンスはユーザの宅内環境に配備された個々のネット家電そのものを指し、すべてのインスタンスは特定のネット家電クラスに属する。ネット家電へのアクセス手法をインスタンス定義として記述することで、提供するサービスに依存しないネット家電の情報を集約することができる。

ネット家電サービス定義はサービスに依存するネット家電データを保持する。サービス開発者はネット家電を用いて特定の処理をユーザに提供するために、ネット家電サービス定義を作成する。ネット家電サービス定義では、サービス実現のためのアルゴリズムとネット家電オブジェクトが記述されている。ネット家電オブジェクトは対象となるネット家電クラスに基づく抽象表現で、個別のネット家電インスタンスに依存しない。定義内のネット家電オブジェクトとネット家電インスタンスの関連づけをサービス定義に記述することで、サービスのアルゴリズムとネット家電インスタンス間の疎結合化が可能となる。結果として、一つのネット家電サービス定義を多様なユーザ環境に対応させることが容易化される。

このように、ネット家電が持つ各種の情報を異なる粒度ごとに分離することで、再利用性の高い標準データモデルを構築することが可能となった。以降では、この提案モデルを我々の開発している HNS 環境において異なるサービスを対象として適用することで、その有用性も確認した。今回利用したサービスは、携帯電話で家電の状態を監視、又は家電を制御する携帯電話用統合インターフェースと、複数の家電を連携制御して一つのサービスを提供する連携サービスの作成支援を目的とした、連携サービス作成支援 GUI(BAMBEE) の 2 つである。

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) とは、宅内のネットワークに複数のネット家電を接続して構成されている。各ネット家電には制御用の API が備えられており、外部のサービスは API を通じてネットワーク越しに機器の状態の監視を行ったり、機器の制御を行うことができる。例えば、携帯電話を通じて自宅の家電の状態を監視・制御するサービス [1] [2]、複数の家電を連携して付加価値の高いサービスを提供する家電連携サービス [3] などがある。

我々の研究室では、田中らの提案する手法 [5] に基づき、実際の HNS (CS27-HNS と呼ぶ) を構築している。各家電はサービス化されており、Web サービス [7] のインタフェースとして家電 API が公開されている。全ての家電 API はそのプラットフォームや開発言語に関係なく、SOAP 又は REST 形式の Web サービスとして呼び出すことが可能である。

2.2 HNS におけるサービス開発

HNS 上で開発/提供される既存サービスの多くは複数の家電 API を利用して構成される。機器 API 利用時のネットワークプロトコルは UPnP [8] 或いは DLNA [9] や Web サービス等で標準化が行われている。既存の多くの HNS サービスはそれらのネットワーク標準を利用し、家電 API を組み合わせることで構成される。我々は先行研究 [4] において、HNS サービスが単一ベンダ製のネット家電にしか対応していないことを問題点として指摘した。そのため、標準家電サービスの作成によりベンダ間の差異を吸収し、マルチベンダに対応した HNS サービスの実現が可能であることを示した。

しかしながら、サービスが利用する HNS 環境やネット家電に関するデータモデルの標準化は未成熟であり、異なるサービス間で HNS 環境や家電情報データの再利用は非常に困難である。

2.3 データモデル標準化の必要性

HNS サービスが実行される HNS 環境はユーザごとに異なる。また HNS サービスが利用する家電構成もサービスごとに異なる。そのため、従来の HNS サービスは固定的な HNS 環境を前提としたものがほとんどである。結果として HNS サービスと家電情報が密に結合してしまい、サービス実装の複雑度が増すこととなった。さらに、特定の家電構成と密に結合した HNS サービスは再利用性も低いいため、サービス開発・運用のコストも増大し、HNS サービス普及の妨げとなっている。

従って、標準化されたデータモデルを構築して各サービス間で共有させることが、HNS の今後の課題と言える。HNS サービスおよびデータモデルの再利用性改善を目的としてデータモデルの標準化を行うためには、以下の 3 つの多様性を考慮する必要がある。

V1: 家電ベンダの多様性

V2: ユーザの HNS 環境における家電情報の多様性

V3: 特定の HNS 環境下において利用される HNS サービスの多様性

2.2 節で述べたように、同じ種類のネット家電であってもベンダが異なればその家電が提供する機能が異なる可能性がある。ベンダ毎にデータモデルが異なれば、HNS サービスの再利用性は低下する。

同じ種類のネット家電であったとしても、異なる HNS 環境に配置されていれば、そのネット家電へのアクセス方法等の詳細情報は異なる。そのため、特定の HNS 環境に配置されたネット家電情報の多様性を考慮したデータモデルの構築が必要となる。

HNS サービスに含まれる家電情報と HNS サービスのアルゴリズムが密に結合しては、サービスの再利用は困難である。そのため、HNS サービスアルゴリズムの再利用性を高めるた

めには、サービスのアルゴリズムと利用するネット家電情報を分離し、かつ、HNS サービスの多様性に対応できるデータモデルが必要となる。

このように、HNS サービスの開発容易性や再利用性による開発・運用コストの低減を図るには、異なるレベルでの多様性に対応したデータモデルが必要である。

3. ネット家電標準データモデル

3.1 HNS とサービス

本稿では、ネット家電を取り巻く情報について、「HNS 機器本体の情報 (D1)」「家庭内の HNS に接続されたネット家電の配置情報 (D2)」「サービスに固有の情報 (D3)」の 3 つに分けて管理する方式を提案する (図 1)。そして、それらを以下のように名付ける。

D1: ネット家電クラス定義。ユーザの宅内情報、及び HNS サービスに非依存な情報を持つ。

D2: ネット家電インスタンス定義。ユーザの宅内情報に依存するが、HNS サービスには非依存な情報を持つ。

D3: ネット家電サービス定義。ユーザの宅内情報、及び HNS サービスに依存する情報を持つ

このように、各情報を異なる粒度ごとに分離することで、それぞれの情報の再利用性を高める。ユーザの構築した HNS 環境は D1 及び D2 によって定義され、各 HNS サービスはその 2 つの情報を利用することで、HNS 環境に関する情報を個別に設定する必要がなくなる。

各情報の様式について、次節から詳細を記す。

3.2 ネット家電クラス定義

家電全般に関する情報を考えると、その家電が提供可能な「機能」と、その機能を使用した際に変化する「状態」とが挙げられる。機能の例としては、TV なら電源 ON やチャンネル変更、エアコンなら冷房運転や暖房運転等があり、状態の例としては、TV なら現在のチャンネルや音量、エアコンなら現在の運転状況や設定温度等がある。これらは家電の種類を問わず存在する。

ここでは、個々の家電をオブジェクト指向における「クラス」として捉え、家電の機能をクラス内の「メソッド」、家電の状態をクラスが持つ「属性」と考える。そして、その種の家電が持つ機能と状態とを抽出した情報を「ネット家電クラス定義」

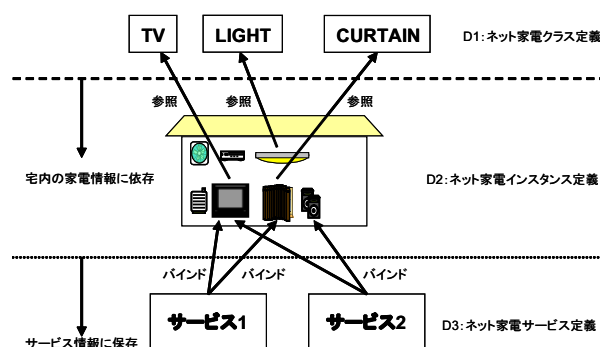


図 1 データの分割

表 1 企業ごとの機能の差異

	電源	音声切替	入力切替	音量	チャンネル	消音	オフタイマー	時刻設定	チャンネル自動切替	無操作自動切替	電力消費調整	字幕	番組内音	BS切替	CS切替	デジタル
A社	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
B社	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
C社	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D社	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
E社	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
F社	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
G社	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
H社	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

として記述する。

そこで問題となるのは、同種の家電であってもベンダや型番によって提供される機能に差異が生じることである。今回は [4] で述べられている標準家電サービスを利用して、各家電に標準的に備わっている機能を調査・抽出し、ネット家電クラス定義に記述することとした。

例として、表 1 は A から H までの各社が提供している TV についての機能を纏めたものとする。表 1 によると、各社製の TV が共通して備えている機能として、電源・音声切替・入力切替・音量・チャンネル・消音・オフタイマーがある。このことから、これら 7 つの機能は TV にとってほぼ必須の機能であると言え、TV における標準的な機能と捉えられる。一方、オンタイマーは 8 社のうち A 社と F 社の TV しか備えられてなく、他社製の TV では利用できない。同様に、字幕機能を搭載している TV も A 社・B 社・H 社の 3 社しかない。よって、このような一部のベンダ製の製品しかサポートしていないような機能は必須機能とは言えず、従って TV における標準的な機能とは言い難い。

このようにして個々の家電の持つ機能を抽出し、ネット家電クラスのメソッドとする。同様にして家電が取り得る状態についても抽出し、そのクラスにおける属性として記述していく。

記述形式は XML を利用する。XML とは、文書及びデータの意味や構造を記述するためのマークアップ言語の一つである。マークアップ言語とは「タグ」と呼ばれる特定の文字列で地の文に構造を埋め込んでいく言語の事を言う。ネット家電クラス定義を XML で書く際の構造を記したスキーマの一部は、例えば図 2 のようになる。斜体字になっている文字列が「タグ」であり、各々の意味は次のようになっている。

```
<xsd:element name="appliance">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="id" type="xsd:string" />
      <xsd:element name="id" type="xsd:positiveInteger" />
      <xsd:element ref="attributes" />
      <xsd:element ref="methods" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="attributes">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="attribute" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="attribute">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="val" type="xsd:string" />
      <xsd:element ref="values" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

図 2 ネット家電クラス定義の XML スキーマ (一部)

appliance: ルート直下に置かれる．子要素に name(家電クラス名), alt(説明用代替テキスト), id(識別番号), attributes(機器の属性の集合), methods(機器のメソッドの集合) をそれぞれ持っている．

attributes: 機器の各属性を集合として纏めている．子要素に attribute(機器の状態) を持ち、その数に上限を指定しない．

methods: 機器の各メソッドを集合として纏めている．子要素に method(機器の機能) を持ち、その数に上限を指定しない．

attribute 機器の属性を記述する．子要素に name(属性名), alt, values(値の集合) を持つ．

method: 機器の機能を記述する．子要素に name(機能名), alt, args(引数の集合), invoke(SOAP 又は REST の指定), return(返り値) を持つ．

invoke: Web サービスへアクセスする際の形式を記述する．今回の場合は属性値として SOAP 又は REST のみ認められている．子要素に name(Web サービス呼び出し時の名前) を持つ．

また、name や id 等に記述できるデータ型は string や positiveInteger 等といったように、事前に決定することができる．これにより HNS サービス開発者は、各クラス定義内の詳細な内容を意識することなく、プログラムに組み込むことができる．

このように、ベンダに依存しない標準的な機能及び状態を対象とすることで、ネット家電クラス定義で記述される内容はベンダを問わず利用可能となる．従って、V1 に対応できる．

なお、スペースの都合上一部のタグについての詳細が割愛されていることに注意されたい．

3.3 ネット家電インスタンス定義

家電本体をクラスとして捉えたと、各家庭に配置されている家電群は、家電クラスを実装した「インスタンス」の集合と捉えられる．これをネット家電クラス定義と同様に XML で記述する際のスキーマは図 3 の通り．必要な構成要素は、HNS に接続された各ネット家電が属するネット家電クラスの情報と、そのネット家電への詳細なアクセス情報である．詳細は次の通り．

appliances: ルート直下に置かれる．子要素に appliance(家電機器) を持ち、appliance の数に上限を指定しない．

```
<xsd:element name="appliances">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="appliance" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="appliance">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="class" type="xsd:string" />
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="wsdl" type="anyHTTP" />
      <xsd:element name="namespace" type="anyHTTP" />
      <xsd:element name="baseURI" type="anyHTTP" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:simpleType name="anyHTTP">
  <xsd:restriction>
    <xsd:simpleType>
      <xsd:restriction base="xsd:anyURI" />
    </xsd:simpleType>
    <xsd:pattern value="http:..." />
  </xsd:restriction>
</xsd:simpleType>
```

図 3 ネット家電インスタンス定義の XML スキーマ

appliance: 各家電インスタンスの情報を記述する．子要素に class(実装する家電クラス名), name(各インスタンスを一意に決定する名前), wsdl, namespace, baseURI を持つ．

wsdl とは、Web サービスを記述するために XML をベースとして書かれた技術仕様の事である．namespace とは XML 名前空間を指定するものである．baseURI は Web サービスにアクセスするための基底となる URI を指す．サービスはこれらの情報を基にして、各ネット家電インスタンスの Web サービスにアクセスする．

このように、宅内に配置されたネット家電の情報をインスタンス定義として集約することで、家庭毎に配置されるネット家電が異なる場合でも、インスタンス定義を変更するだけで対応できる．また、インスタンス定義がサービスアルゴリズムに関する情報を一切持たないことで、インスタンス定義を変更した影響がサービスアルゴリズムに及ぶことはない．よって、V2 に対応できる．

3.4 ネット家電サービス定義

これまでの 2 つの定義が HNS 機器に閉じた情報なのに対し、ネット家電サービス定義はサービス側で必要な固有の設定を記述するものである．特に、サービス内で利用されるネット家電オブジェクトとネット家電インスタンスの関連づけを、このサービス定義内で行う．この関連付けを HNS 機器の情報を集約した定義とは別に定義することで、HNS 機器情報に変更が及んだ際にサービス側で受ける影響を極力抑えることが可能となる．また、必要ならばサービス開発者の裁量で他の項目を追加することもできるので、データ構造におけるサービスに対する制約は少ない．スキーマは図 4 で、各詳細は次の通り．

appliances: ルート直下に置かれる．子要素に appliance(家電機器) を持ち、appliance の数に上限を指定しない．

appliance: サービスが利用するネット家電に関する情報を記述する．子要素に instance(対応するインスタンス名), name(プログラム中の名前), anyname(拡張用) を持つ．

anyname: サービス開発者が必要な内容を自由に記述できる．タグ名・データ型は開発者の裁量に任せられており、anyname の数に上限を指定しない．

ネット家電サービス定義に求められることは、前述の通りネット家電オブジェクトとネット家電インスタンスとの対応情報である．その他、anyname タグがサービス開発者の拡張用の

```
<xsd:element name="appliances">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="appliance" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="appliance">
  <xsd:annotation>
    <xsd:documentation>anynameタグをサービス開発者の裁量で自由に修正可能</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="instance" type="xsd:string" />
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="anyname" type="xsd:anyType" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

図 4 ネット家電サービス定義の XML スキーマ

タグで、この部分を修正することによりサービスの多様性を実現できる。

さらに、データモデル記述に XML を利用することによって、サービスの実装技術に XML を利用できるツールがある場合に、このデータモデルのサービスアルゴリズムへの導入が比較的容易になると考えられる。そして、XML を利用できるツールは現在幅広く存在する [10] [11] [12]。従って、V3 に対応できる。

以上のようにデータを 3 つに分割することで、サービス開発者はネット家電クラス定義とネット家電インスタンス定義の詳細な内容を把握していなくても、提案したスキーマに沿った形で利用することでデータを扱うことができる。また、そのようにデータモデルを統一化することで、サービス毎に必要な定義のみ作成するだけで、既存の HNS 上への導入が可能となる。

4. ケーススタディ

本稿で提案したモデルを、実際に我々のグループで構築している CS27-HNS 上で、幾つかのサービスに適用してみた。作成したクラス定義、及びインスタンス定義のサンプルは図 5、図 6 の通り。

4.1 携帯電話用家電統合インターフェース

これは携帯電話で家電の状態の監視・制御を可能にするインターフェースを提供するサービスである。実装技術は Perl プログラムである。サービスは HTTP リクエストにより家電名を受け取ると、その家電の提供できる機能と状態とを HTML 形式に加工して Web ブラウザ上に表示する。ユーザはそのペー

ジを閲覧することで家電の状態を知り、表示された Web サービスへのリンクをクリックすることで家電を制御できる (図 7)。

一方、図 8 がこのサービスで利用しているサービス定義の一例である。簡単な詳細は以下の通り。

appliance: ルート直下に置かれる。

class: 利用するクラス名を記述する。

name: プログラム内で使われる変数名。

description: 画面に表示される説明文。

imgs: 画像アイコン等の情報を纏めている。この例では、子要素に topimg(トップ画像) タグを持っている。

topimg: 画面トップに表示される画像アイコンの情報を記述する。子要素に src(ソース場所), size(画像サイズ) タグを持っている。

size: 画像アイコンのサイズを記述する。子要素に width, height を持つ。

図 8 は画面トップの画像アイコンの設定部分についての詳細である。これと同じように、Web サービスへのリンクの方法についても、タグで設定することで容易に切り替えることができる。

4.2 連携サービス作成支援 GUI

照明の ON/OFF やカーテンの開閉などは、それ単体でもユーザの要求を満たすことができる。しかし、それら家電の一つ一つの操作を纏めてシステムに行わせることで、連携サービスと呼ばれるより付加価値の高いサービスを提供することができる。

連携サービスは、家庭内の HNS に接続された家電の状況や、ユーザ自身のニーズに合わせて提供されなければならない。

```
<appliance>
  <name>Fan</name>
  <id>221</id>
  <attributes>
    <attribute>
      <name>power</name>
      <alt>電源</alt>
      <values>
        <value><val>true</val><alt>稼働中</alt></value>
        <value><val>false</val><alt>オフ</alt></value>
      </values>
    </attribute>
    :
  </attributes>
  <methods>
    <method>
      <name>on</name>
      <alt>つける</alt>
      <args></args>
      <invoke type="REST"><name>on</name></invoke>
      <return></return>
    </method>
    :
  </methods>
</appliance>
```

図 5 クラス定義 (Fan クラス) のサンプル

```
<appliances>
  <appliance>
    <class>TV</class>
    <name>TV</name>
    <wsdl>http://[URL]</wsdl>
    <namespace>http://[URL]</namespace>
    <baseURI>http://[URL]</baseURI>
  </appliance>
  <appliance>
    <class>AirConditioner</class>
    <name>AirCon</name>
    <wsdl>http://[URL]</wsdl>
    <namespace>http://[URL]</namespace>
    <baseURI>http://[URL]</baseURI>
  </appliance>
  :
</appliances>
```

図 6 インスタンス定義ファイルのサンプル



図 7 UI 画面のサンプル図

```
<appliance>
  <instance>Fan</instance>
  <name>fan</name>
  <alt>扇風機</alt>
  <description>扇風機のオン、オフ、スイングができます。</description>
  <imgs>
    <topimg>
      <src>img/fan-black.GIF</src>
      <size><width>100</width><height>100</height></size>
    </topimg>
  </imgs>
  <attributes>
    <attribute><name>power</name></attribute>
    :
  </attributes>
  <methods>
    <method><name>on</name></method>
    :
  </methods>
</appliance>
```

図 8 サービス定義のサンプル (扇風機用・一部)

その要求に応えるためのサービスが、連携サービス作成支援 GUI(BAMBEE) である (図 9)。実装技術は Adobe Flash CS3 Professional と Perl である。インターフェース部分を Flash で、Web サービスへのアクセス部分を Perl で構築されている。

BAMBEE における固有の設定は、選択したアイコンと HNS 機器の機能との対応関係である。Flash 側で選択されたアイコンから、Perl で Web サービスにアクセスするための記述に変換する (或いはその逆の変換をする) ために利用される (図 10)。簡単な詳細は次の通り。

appliances: 利用する appliance(家電) を纏める。
 appliance: 各家電について設定を記述する。
 class: 実装する家電クラス名を記述する。
 name: プログラム内で使われる変数名。
 methods: 利用する家電のメソッドを纏める。
 method: 各メソッドについて記述する。子要素に name(メソッド名), args(引数), icon(Flash で利用するアイコン名) を持つ。

図 10 にあるように、各 method タグは子要素に Flash 側で利用されるアイコン名を持っているので、プログラム内で各メソッド名とそれに対応するアイコン名との置換が可能になる。

4.3 実装結果

上記 2 つのサービスは元々独自に開発されていたため、図 5 図 6 で定義されているような情報は、個別に用意する必要が



図 9 連携サービス作成支援 GUI

```
<appliance>
<instance>Fan</instance>
<name>cir</name>
<methods>
<method>
<name>on</name>
<args></args>
<icon>on_icon</icon>
</method>
<method>
<name>air_volume</name>
<args><arg>1</arg></args>
<icon>airflow0_icon</icon>
</method>
:
:
</appliance>
```

図 10 BAMBEE における設定ファイル (一部)

あった。しかし、今回提案したデータモデルに基づいてデータ構造及びプログラムを再構築したところ、その場合でも問題なく動作した。

さらに、CS27-HNS に家電機器が新たに追加された際に、共有しているネット家電クラス定義・ネット家電インスタンス定義を変更するだけで、2 つのサービスに適用することができた。このことは、今後サービスの数が増加した際に更なる恩恵を生み出すと考えられる。

5. ま と め

本稿では、ネットワークレベルでの標準化が既に進められている HNS に対し、扱うデータモデルの標準化を今後の課題と捉え、ネット家電標準データモデルについての一手法を提案した。さらに、そのデータモデルを実際のサービスに適用して運用することで、その有用性も確認した。

これからの研究課題としては、提案したデータモデルが本当に妥当なものかどうか、適用するサービスの数を更に増やして評価検証を行うことが挙げられる。また、インスタンス定義等はユーザ毎に設定を行う必要性も視野に入れて、エンドユーザでも簡単に設定が行えるようなインターフェースの提供も考えていきたい。

謝辞

この研究の一部は、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), パナソニック電工株式会社の助成を受けて行われている。

文 献

- [1] 株式会社 東芝, “東芝ネットワーク家電 Feminity”, <http://www3.toshiba.co.jp/feminity/>
- [2] パナソニック電工株式会社, “ライフィニティ”, <http://denko.panasonic.biz/Ebox/kahts/>
- [3] M.Nakamura, H.Igaki, H.Tamada, and K.Matsumoto “Implementing Services of Networked HomeApplication Using Service Oriented Architecture, ”In *Proc. 2nd International Conference on Seervice Oriented Computing(ICSOC2004)*, pp.269-278, NY, USA, Nov.2004
- [4] 福岡 佑介, 西岡 隆司, 中村 匡秀, 井垣 宏, 松本 健一, “動的サービスバインディング機構を用いたマルチベンダホームネットワークシステムの一実現手法”, 電子情報通信学会技術研究報告, Vol.107, No.525, pp.295-300, March 2008.
- [5] 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.
- [6] SUGIYAMA ELECTRON CO.,LTD “学習機能付マルチリモコン「クロッサム 2+USB」”, <http://www.sugi-ele.co.jp/top.htm>
- [7] web サービス, <http://www.w3.org/2002/ws>
- [8] UPnP Forum, <http://www.upnp.org/>
- [9] Digital Living Network Alliance, <http://www.dlna.org/>
- [10] XML Parser For Java, <http://www.alphaworks.ibm.com/tech/xml4j>
- [11] Flex ActionScript および MXML API リファレンスガイド, http://www.adobe.com/support/documentation/jp/flex/1_5/asdocs_jp/index.html
- [12] Flash CS3 ドキュメンテーション http://livedocs.adobe.com/flash/9.0_jp/main/wwhelp/wwhimpl/js/html/wwhelp.htm