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家電連携サービスの安全性に関する考察

閻　奔† 　中村　匡秀† 　リディ ドゥ ブスケ‡ 　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
‡ヨセフ・フーリエ大学（グルノーブル第 1 大学），フランス

E-mail: †{hon-e, masa-n, matumoto}@is.naist.jp, ‡lydie.du-bousquet@imag.fr

**あらまし**　複数の家電をネットワークを用いて接続・連携させ，付加価値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・開発が盛んである．本稿では，付加価値サービスを開発する際に，家庭内のユーザに対して保障しなければならない安全性について議論を行う．また，安全性の検証を考慮に入れた HNS のモデルを提案し，検証方法についてのアイデアを述べる．

**キーワード**　ホームネットワーク，連携サービス，安全性，モデル，JML，契約による設計

# Considering Safety in Integrated Services in Home Network System

Ben Yan† 　Masahide Nakamura† 　Lydie du Bousquet‡ 　and 　Ken-ichi Matsumoto†

†Nara Institute of Science and Technology 8916-5 Takayama-cho, Ikoma-shi, Nara, 630-0192 Japan
‡LSR Laboratory, IMAG, Joseph Fourier University (Grenoble I), BP72 F-38402 Saint-Martin d'Hères Cedex France

E-mail: †{hon-e, masa-n, matumoto}@is.naist.jp, ‡lydie.du-bousquet@imag.fr

**Abstract**　The HNS (Home Network System) integrated service is to orchestrate multiple networked appliances to add value of user's life. Recently, the research and development of the HNS are getting more and more attention. Assuring the safety of HNS is the crucial problem to protect the users and the home properties. In order to tackle this problem, this paper addresses the verification of safety of the HNS integrated services. Moreover, we propose a model of HNS for the safety validation, and a concrete procedure of the validation.

**Keyword**　Home Network System, Integrated services, Safety, Model, JML, Design by Contract

## 1. はじめに

家電機器やセンサを家庭内ネットワークに接続し，様々なサービスを提供可能にするホームネットワークシステム（HNS）の研究・開発が盛んである．ネットワークに接続された複数の家電（**ネットワーク家電**と呼ぶ）をアプリケーションから連携制御することで，ユーザに便利で快適な付加価値サービス(**HNS 連携サービス**と呼ぶ)を提供することができる [9][10]．例えば，TV，DVD プレーヤ，5.1ch スピーカ，照明，カーテンを連携制御することで，映画館の雰囲気で DVD 視聴を楽しめる **DVD シアターサービス**を実現できる．

HNS 連携サービスの開発・提供にあたっては，そのサービスがユーザや家財に対して**安全**であることを保証しなくてはならない．従来から，各家電の取扱説明書には，安全性に関する注意事項（**安全事項**--safety instructions--と呼ぶ）が記載されている．ユーザは安全事項を遵守することで機器を安全に使用することができる [4]．

しかしながら，HNS 連携サービスの場合 (a) 操作者が人間のユーザではなくソフトウェアであること，(b) 1 つの連携サービスは複数の家電を組み合わせて使用すること，(c) 複数の連携サービスが同時に実行されうること，といった特徴がある．これらの理由から，連携サービスの開発・実装においては，従来家電を単体で使用する場合と比べ，より慎重かつ綿密な安全対策が必要となる．ソフトウェアのバグやロジック誤りが，深刻な事故や被害を引き起こす可能性がある．残念ながら，HNS 連携サービスの安全性に関する研究や方法論は，これまでほとんど報告されていない．

そこで本稿では，HNS 連携サービスの安全性を，厳密に検証するための枠組みを提案する．具体的には，まず HNS 連携サービスの安全性について定義を行い，連携サービスに求められる条件の洗い出しを行う．次に，安全性検証のための HNS および連携サービスのオ

ブジェクト指向モデリングの提案を行う．最後に，契約による設計(Design by Contract) [8] と JML (Java Modeling Language) [11] を用いた安全性検証のアイデアを述べる．

## 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．各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，各機器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．ネットワーク家電間の通信は，専用の家電プロトコルに基づいて行われる．現在，多くの家電プロトコルが標準化されつつあり，代表的なものに情報家電用の DLNA[2]や白物家電用の ECHONET [3]等が知られている．

また典型的な HNS には**ホームサーバ**が設置され，ネットワーク家電の操作や管理，外部ネットワークへのゲートウェイ機能の実行等を行う．

**HNS 連携サービス**は，複数のネットワーク家電の機能を連携しユーザに付加価値を与えるサービスである．通常，ネットワーク家電の制御 API を決められたロジックで実行するソフトウェアアプリケーションとして実装され，ホームサーバにインストールされる．

### 2.2. 連携サービスの例

例として，いくつかの HNS 連携サービスのサービスシナリオを導入する．

**SS1： DVD シアターサービス**

デジタル TV, DVD プレーヤ，サウンドシステム，照明，カーテンを連携して，映画館の雰囲気で DVD を視聴できるサービス．ユーザがサービスを起動すると，各家電の電源がオンになり，カーテンが閉まり，照明が暗くなる．また，プレーヤ，TV の入力設定，サウンドシステムの音量調節等を自動的に行う．

**SS2： リラックスサービス**

DVD プレーヤ，サウンドシステム，照明，エアコン，電気ポットを連携して，ユーザがリラックスできる空間を演出するサービス．ユーザがサービスを起動すると，音楽が自動的に流れ，空調を適温に，照明を適度な明るさに調節する．さらに，電気ポットを沸騰モードにして，コーヒー用のお湯を準備する．

**SS3： お風呂予約サービス**

タイマー，ガス給湯器を連携し，設定した時間・湯量・温度でお風呂を沸かすサービス．ユーザの設定時間に湯の注入が完了するように，設定時間の一定時間前にサービスが起動し湯の注入を開始する．

```
public DVDThaterService {
    DigitalTV tv = new DigitalTV();
    DVDPlayer dvd = new DVDPlayer();
    SoundSystem sound = new SoundSystem();
    Light light = new Light();
    ElectricCurtain curtain = new Curtain();

    tv.on();                        /* Turn on TV */
    tv.visualInput('DVD');
    dvd.on();                       /* Turn on DVD player */
    dvd.setSoundOutput('5.1');
    sound.on();                     /*Turn on Sound System*/
    sound.setInput('DVD');
    sound.setVolumeLevel(25);
    curtain.close();                /*Close curtain*/
    light.setBrightnessLevel(1);    /*Minimize brightness*/
    dvd.play();                     /*Play DVD*/
}
```

図1　DVD シアターサービス

**SS4： シャワーサービス**

ガス給湯器，シャワーバルブ，エアコン，バスルームエアコンを連携し，快適なシャワー設定を提供するサービス．ユーザがサービスを起動すると，シャワーの湯温が最適な温度に設定され，シャワーバルブが開く．また，ユーザが寒くないよう，あらかじめ脱衣場と部屋の温度をエアコンで調節する．

**SS5： 調理準備サービス**

ガスバルブ，換気扇，オーブン，キッチン照明を連携して，炊事準備を行うサービス．ユーザがサービスを起動すると，キッチンの照明が点灯し，ガスバルブが開いて，換気扇が回り，オーブンの予熱が始まる．

図 1に DVD シアターサービスの Java ライクな擬似コードを示す．以降，X.Y() は，家電 X の制御 API Y() を実行することを示す．

## 3. HNS 連携サービスの安全性

### 3.1. 安全性

安全性の定義は様々なものが存在するが，HNS に焦点をあてて広義に以下のように定義する．

**定義 1（広義の安全性）：** HNS 連携サービスが安全であるとは，そのサービスが，住人，設備，環境，家財，近隣住人・環境に対して，死亡，けが，破損，損害を生じうるいかなる条件からも脱却していることである．

一般に，100%の安全性を達成することは難しいため，安全性はリスクという形で表現され，リスクを最小限に抑えるために従うべき条件やガイドライン（**安全性性質 safety property** と呼ぶ）が設定される[4]．以降の節では，リスクを最小限にするため，連携サービスが満たすべき条件について提案する．

### 3.2. 各家電の安全性性質

各家電には，機器を安全に使用するための注意事項（**安全事項, safety instruction**）が定められている．従来こうした安全事項は人間のユーザ向けのものであるが，連携サービスは使用する各機器の安全事項をソフトウェア上で遵守しなければならない．

例えば，以下は電気ポット(electric kettle)の安全事項の一例である．

L1:「お湯が沸騰中に上蓋を開けないでください．やけどのおそれがあります.」

このとき，このポットを使用する連携サービス（例えば2.2のSS2）は，沸騰中に上蓋が開かないように実装されていなければならない.実装の一指針としては，沸騰を開始する前に上蓋が閉まっていることを確認し，サービス開始後は終了するまで蓋を開けることを禁止する．

また全ての家電の安全事項には，その家電の**仕様(Specification)**にあった環境で使用することが記載されている．仕様には，定格電流・電圧，消費電力，動作温度，湿度などの属性が含まれる．

安全事項は家電ごとに定められる**ローカル**な性質（の集合）であり，家電ベンダによって与えられるものとする．

### 3.3. 複数家電をまたがる安全性性質

連携サービスは複数の家電機器を組み合わせて使用するため，複数の機器をまたがった**大域的**な性質も考慮すべきである．例えば，2.2のSS4ではユーザの火傷や心臓発作を防ぐため，

G1:「シャワーバルブを開く時には，給湯器の設定温度が34度から38度までの間でなくてはならない」

といった性質が考えられる．また，SS5では中毒を防ぐため，以下の性質が考えられる．

G2:「ガスを使用する際には，かならず換気扇をつけなければならない」

これらの複数機器にまたがった大域的性質は，サービスの内容に依存するため，個々の家電の安全事項でカバーされないことがある．したがって，連携サービスごとにサービス開発者によって慎重に決定される必要がある．またサービスは，その大域的性質を遵守するように実装されていなければならない．

### 3.4. 環境に対する安全性性質

連携サービスは，HNSが設置される環境（家屋，部屋，建物等）に対して安全でなくてはならない．各家や建物には，その安全に生活するためのガイドラインが定まっていることが多い．例えば，大抵の家には電気容量が決まっていて，

E1:「電気の同時使用量は30Aを超えないこと」

といった安全性性質がある．また，

E2:「火災の時にはドアや窓をロックしないこと」

といった非常時のものや，

E3:「夜10時以降は，大きな音を出さないこと」

というコミュニティに迷惑をかけないようにするための性質も考えられる．

環境に関する安全性性質は，その家に収容される家電やサービスに関係なく独立に定義される.本稿では，建物の使用説明書や非常用マニュアル，自治会や町内会での取り決めなどとして与えられるものとする．

### 3.5. HNS 連携サービスの安全性定義

上記の議論に基づき,HNS連携サービスの3種類の安全性を定義する．

**定義 2（HNS 連携サービスの安全性）：** s を与えられた連携サービスとする．いま，

- App(s)={d1, d2, ..., dn}を s が使用するネットワーク家電の集合，
- LocalProp(di)={lpi1，lpi2，...,lpim}を家電 di∈App(s)の安全事項の集合，
- LocalProp(s) = $\cup_{di \in App(s)}$ LocalProp(di)，
- GlobalProp(s)={gp1，gp2，...，gpk}を s で規定された大域的安全性性質の集合，
- EnvProp(s)={ep1，ep2，...，epl} を s が提供される環境の安全性性質の集合，

とするとき，以下の 3 種類の安全性を定義する．

**[ローカル安全]** s が LocalProp(s)の全ての安全性性質を満たすとき，s は**ローカル安全**であるという．

**[グローバル安全]** s が GlobalProp(s)の全ての安全性性質を満たすとき，s は**グローバル安全**であるという．

**[環境安全]** s が EnvProp(s)の全ての安全性性質を満たすとき，s は**環境安全**であるという．

また，s がローカル安全かつグローバル安全かつ環境安全であるとき，s は**安全**であるという．

上記定義において，LocalProp(di)は家電diの取扱い説明書等の形で家電ベンダによって提供されるものとする. GlobalProp(s)は，サービスsの開発者が決定する．また，EnvProp(s)は，家の説明書や自治体のルール等によって与えられるものとする．

### 3.6. 安全性検証問題

与えられた連携サービスsが安全かどうかを検証する問題は以下のように定式化される．

**定義3（安全性検証問題）：**

**入力：ホームネットワークシステム**hns，連携サービスs，LocalProp(s), GlobalProp(s), EnvProp(s).

**出力：**hns上でsが安全かどうかの判定．

## 4. 安全性検証のための HNS モデリング

安全性検証問題を実施するために，本章では HNS および連携サービスのモデル化手法の提案を行う．

### 4.1. オブジェクト指向モデリング

HNS に収容される各ネットワーク家電は，内部状態（動作モード，電源状態など）と操作インターフェース（制御 API）を持つ．したがって，各家電を**属性**（attributes）と**操作**（methods）とを持つ**オブジェクト**としてモデル化することが自然である．実際に，我々は先行研究[7][10]において，ネットワーク家電をオブジェクト指向的にモデル化する手法を提案している．しかしながら，3.5で述べた 3 種類の安全性を検証するためには，家電のみのモデル化では不十分である．

そこで本稿では，家電モデルに加えて，連携サービスと家(環境)のオブジェクトを新たに導入し，3 種類のオブジェクト間の関係を定義する．

図 2に提案モデルの全体図を示す．記法は UML のクラス図に従っている．モデルは，大きく分けて

- **Appliance**：家電機器オブジェクト
- **Service**：連携サービスオブジェクト
- **Home**：家（環境）オブジェクト

の 3 種類のオブジェクト（クラス）から構成される．これらの間の関係は次のとおりである．

- 各 Home は，複数の Appliance を持つ．
- 各 Home は，複数の Service を持つ．
- 各 Service は，複数の Appliance を使う．

以降で各オブジェクトについて詳細な説明を行う．

### 4.2. 家電機器オブジェクト(Appliance)

家電機器オブジェクトは，実際のネットワーク家電をモデル化する．提案モデルでは，家電に共通した属性や操作を Appliance クラスとして，実際の機器のクラスがそれを継承する．

Appliance クラスは，仕様（Specification）を持ち，各家電の静的な仕様情報（定格電流・電圧，動作保証温度等）を保持できるようになっている．電源のオン・オフ(powerON(), powerOFF())や現在の消費電力の取得(getCurrentConsumption())，仕様情報の参照(getApplianceSpecification())など，全ての家電に共通した操作は，Appliance クラスに集約される．

一方，TV チャンネル設定(TV.selectChannel())，DVD 再生（DVD.play()），冷房(AirConditioner.setMode(“cold”))，ポット蓋オープン(ElectricKettele.openLid())など，家電の機種に依存する操作は，個別の家電クラスで定義されている．操作が実行されると家電オブジェクトの属性が更新され，家電の現在状態が変化する．現在の家電オブジェクトの状態（全ての属性の現在の値の組）を外部からオンデマンドで取得できるように，各家電のクラスは現在状態を取得するメソッド(getState())を実装することが推奨される．

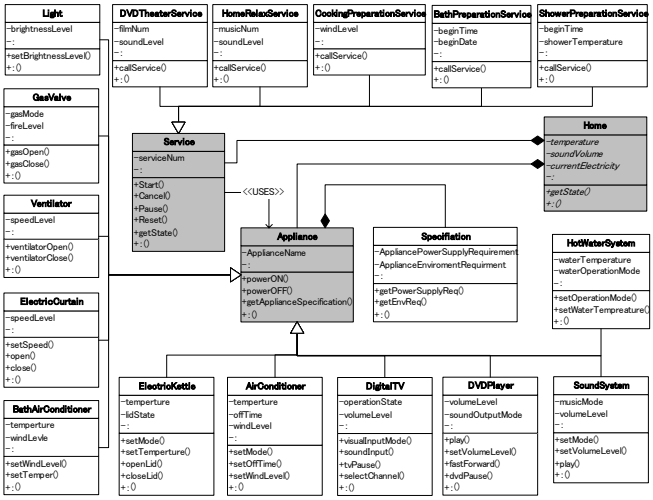

図 2　HNS のオブジェクト指向モデリング

### 4.3. 連携サービスオブジェクト(Service)

連携サービスオブジェクトは，連携サービスをモデル化する．各連携サービスクラスは，サービス内容に応じて，複数の家電オブジェクトを使用する．Appliance クラス同様，サービス開始・終了といった汎用的な操作は Service クラスに集約し，サービス毎に異なる家電や操作は個別のサービスクラスで定義される．例えば，2.2の SS1 で述べた DVDTheaterService クラスはその内部で，DigitalTV, DVDPlayer, SoundSystem, Light, ElectricCurtain クラスを使用する．

### 4.4. 家(環境)オブジェクト(Home)

家（環境）モデルは，すべての家電およびサービスを収容する単一のオブジェクト（Home クラス）としてモデル化される．Home クラスは，現在の家の気温や湿度，消費電力といった家の大域的な環境情報を属性として持つ．こうした環境情報は，各家電やサービスの現在状態から計算され，更新されるものとする．例えば，現在の消費電力は，各家電 d について，d の仕様に記載されている消費電力値を計算し，現在電源がオンになっている全ての家電の消費電力を合算することで計算可能である．

## 5. DbC による安全性検証

以上のモデル化に基づいて，実際に安全性検証を行う方法を提案する．

### 5.1. キーアイデア

安全性検証を実施するために，我々は**契約による設計(Design by Contract**，以降 **DbC**)[5][8]と呼ばれる実装戦略を採る．

DbC では，プログラム中のある操作を実行するために必要な条件や，操作の前後で変わってはならない条件などを「**契約**」として記述し，プログラムの実行時（厳密にはテスト時）にその条件をチェックすることができる．これにより，操作を実行する際に呼び出し側と操作側で責任の所在が明確になり，各オブジェクトが満たすべき様々な性質を検証することができる．

DbC における契約は，以下の 3 種類が存在する．

**事前条件（Pre-Condition）**：ある操作（メソッド）を実行するために，最低限必要な条件を定めたもの．

**事後条件（Post-Condition）**：ある操作（メソッド）を実行した後に，成立していなければならない条件を定めたもの．

**クラス不変表明（Class Invariant）**：あるクラスについて，どのような操作を行っても常に満たされているべき条件を定めたもの．

プログラムの実行中，もし契約が破られた場合には，通常，例外が投げられるかエラーが報告される．

安全性検証問題は，安全性性質 prop と HNS の実装 hns および連携サービスの実装 s が与えられたとき，s が hns 上で prop を遵守するか検証する問題である．いま，hns および s が4章で提案したモデルに基づいて実装されているものとする．我々のキーアイデアは，prop を DbC の契約として図 2のオブジェクト内に埋め込み，定義 2 における 3 種類（ローカル，グローバル，環境）の安全性の検証を行うことである．このとき，prop がどの安全性に関する性質かによって，prop を埋め込むオブジェクトを変える．

### 5.2. ローカル安全性の検証

ローカル安全性は，家電単体で定義される安全性性質を満たすことであるから，図 2の家電クラス（Appliance クラスまたはその子クラス）内に契約を規定する．

例えば，3.2の性質 L1 は，電気ポットオブジェクト ElectricKettle の蓋を開けるメソッド openLid()メソッドに対して，以下の契約を規定することで表現できる．

メソッド：ElectricKettle.openLid()

事前条件：heatingStatus != ‘boiling’

事後条件：lid == open && heatingStatus != ‘boiling’

この契約によって，「連携サービスが openLid()を呼び出す際には，加熱状態が沸騰中であってはならないこと」という安全性性質を表現できる．もし，沸騰中に実行された場合には，契約が侵害され例外が投げられプログラムがストップする．

```
public class ElectricKettle {
  private /*@spec_public@*/ LidState  lid;
  private /*@spec_public@*/ HeatingState hstate;
...
// JML Contract for openLid() [L1]
/*@ requires hstate != “boiling” ;
  @ ensures  lid = “open” && hstate != “boiling”;
  @*/
  public void openLid() {
    ... // Implementation
  }
...
}
```

図 3　JML による契約記述（5.2の L1）

### 5.3. グローバル安全性の検証

グローバル安全性は，連携サービスの内容に依存した複数の家電間にまたがる大域的性質を満たすことである．したがって，図 2の連携サービスクラス内に契約を規定する．

例えば，3.3の性質 G2 は，SS5：調理準備サービスのクラスの不変表明として規定できる．

クラス：CookingPreparationService

クラス不変表明：gasValve.status == ‘open’ ->
ventilator.power == ‘ON’

上記の契約は，CookingPreparationService で使用されるガスバルブ(gasValve)と換気扇(ventilator)について，「もし，ガスバルブが開いているならば，換気扇の電源が必ず入っていなければならない」ことを表す．この条件は，サービス中いかなる操作が実行されても保証されなければならない性質であり，侵害されると例外が投げられる．

### 5.4. 環境安全性の検証

環境安全性は，家電やサービスの内容に依存しない，家に規定される性質を満たすことである．したがって，図 2の Home クラス内に契約を規定する．

例えば，3.4の性質 E1 は，Home クラス内に以下の契約として規定できる．

クラス：Home

クラス不変表明：home.getTotalElectricity() <= 30

上記契約は，Home クラスの現在の消費電流を計算するメソッドを呼び出し，その戻り値が 30(A)以下であることを規定している．

### 5.5. JML を用いた安全性検証手順

HNS と連携サービスの実装が Java で行われた場合，上記の安全性の検証に JML（Java Modeling Language）[11]を用いることができる．JML は，DbC の契約を Java のソースコードの注釈（コメント）として挿入するための言語である．契約が埋め込まれた Java ソースコー

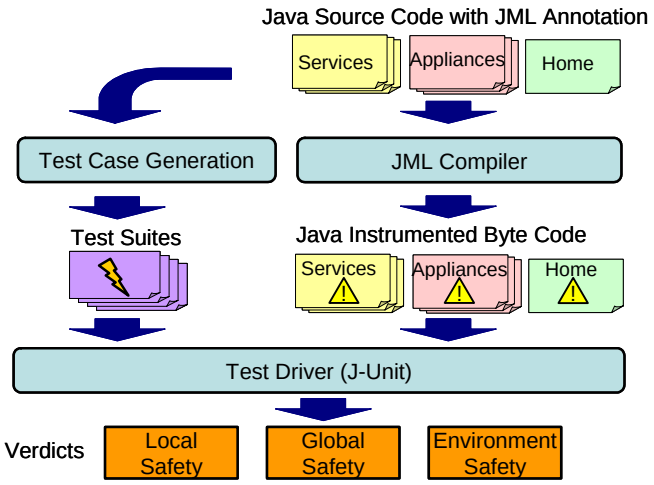


図 4　HNS モデル検証手順図

ドは，JML コンパイラによって，契約のチェックルーチン付バイトコード(Instrumented Code)へとコンパイルされる．

3.2で述べた電気ポットの契約 L1 を JML で記述した例を図3に示す．図中，メソッド openLid()の上部にコメントとして契約が挿入されている．図中，requires は事前条件を，ensures は事後条件をそれぞれ記述するための予約語である．spec_public はその属性値を JML 契約内で参照するという意味を表す．

図 4に JML を用いた安全性検証の手順を示す．JML 注釈付のソースコードが与えられると，検証者は JML コンパイラでコンパイルする．同時に，ソースコードからテストケースの生成を行う．テストケースの生成は,連携サービスのロジック,パラメタ値に基づいて，手動で生成するか，TOBIAS [1][6]等のテスト生成ツールで行う．生成したテストケースをバイトコードに与えてテストを行う．もし，JML の契約（すなわち安全性性質）が侵害されれば，テストが失敗する．これにより，3 種類の安全性の検証できる．テストにおいては，Java のテストのためのフレームワーク JUnit [12]を利用することでテストの効率化が図れる．

## 6. おわりに

本稿では，ホームネットワークシステム（HNS）における連携サービスの安全性検証についての提案を行った．まず，HNS 連携サービスの 3 種類の安全性の定義を行い，問題の定式化を行った．次に，安全性検証のための，オブジェクト指向モデルを提案した．最後に，提案モデルに基づいて，Design by Contract (DbC) および，Java Modeling Language (JML)を用いた安全性検証の手順を提案した．

今後の課題は,実際の HNS を想定した安全性検証の実験を行うこと，また，提案法の評価を行うことが挙げられる．また，HNS におけるサービス競合問題を考慮に入れた安全性検証手法の開発を行っていきたい．

## 謝辞

本研究は，文部科学省科学技術研究費（若手研究(B)-18700062），および，21 世紀 COE プログラム(NAIST-IS:ユビキタスメディアコンピューティング)の助成を受けている．

## 文　献

[1] L. du Bousquet, Y. Ledru, O. Maury, and P. Bontron, “A case study in JML-based software validation, ” Proceedings of 19th Int. IEEE Conf. on Automated Software Engineering (ASE'04), Linz, pages 294-297. IEEE Computer Society Press, Sep. 2004.

[2] Digital Living Network Alliance, http://www.dlna.org

[3] ECHONET Consortium, http://www.echonet.gr. jp/

[4] International Electrotechnical Commission, “Household and similar electrical appliances --- Safety”, IEC 60335-1, Sep. 2006.

[5] G. T. Leavens and Y. Cheon, “Design by Contract with JML,” Java Modeling Language Project, Internet: http://www.jmlspecs.org, 2003.

[6] Y. Ledru, L. du Bousquet, O. Maury, and P. Bontron. “Filtering TOBIAS combinatorial test suites,” In Proceedings of ETAPS/FASE’04 - Fundamental Approaches to Software Engineering , LNCS 2984, Springer-Verlag, Mar. 2004.

[7] P. Leelaprute, M. Nakamura, T. Tsuchiya, K. Matsumoto, and T. Kikuno, “Describing and Verifying Integrated Services of Home Network Systems,” In Proc of 12 th Asia-Pacific Software Engineering Conference (APSEC 2005), pp.549-558, Dec. 2005.

[8] B. Meyer, “Applying Design by Contract”, IEEE Computer, vol. 25, no. 10, pp. 40-51, Oct. 1992.

[9] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, K. Matsumoto, “Adapting Legacy Home Appliances to Home Network Systems Using Web Services,” Proc. of Int'l Conf. on Web Services (ICWS 2006), pp.849-858, Sep. 2006.

[10] M. Nakamura , H. Igaki , and K. Matsumoto , “Feature Interactions in Integrated Services of Networked Home Appliances -An Object-Oriented Approach-,” Proc. of Int'l. Conf. on Feature Interactions in Telecommunication Networks and Distributed Systems (ICFI'05), pp.236-251, Jul. 2005

[11] “The Java Modeling Language – JML,” http://www.cs.iastate.edu/~leavens/JML/

[12] “JUnit, Testing Resources for Extreme Programming”, http://www.junit.org/