

# 法政大学学術機関リポジトリ

HOSEI UNIVERSITY REPOSITORY

PDF issue: 2024-09-03

## ホームネットワークにおけるエージェント・ 端末のソフトウェア構成の検討

吉田, 裕 / YOSHIDA, Yutaka / Kobayashi, Takumi / 小林,  
拓史

---

(出版者 / Publisher)

法政大学工学部

(雑誌名 / Journal or Publication Title)

Bulletin of the Faculty of Engineering, Hosei University / 法政大学工学部  
研究集報

(巻 / Volume)

42

(開始ページ / Start Page)

41

(終了ページ / End Page)

46

(発行年 / Year)

2006-03

(URL)

<https://doi.org/10.15002/00003748>

# ホームネットワークにおけるエージェント・端末のソフトウェア構成の検討

## A STUDY ON SOFTWARE CONFIGURATION IN AGENT AND TERMINALS FOR HOME NETWORK

小林拓史\*, 吉田裕\*\*  
Takumi KOBAYASHI and Yutaka YOSHIDA

In this paper, home network is assumed to consist of servers called home agents(HA's) and clients called home terminals(HT's). The HT's are located within a home network but also may be located outside the network via internet. This paper discusses on interactive data exchange or communications interface between these HA's and HT's and further on communications schemes between HT's especially focusing on whether an HA mediates them or not. At last, software configuration for HA's and HT's is described including class definitions used in them.

**Key Words:** Home Network, Home Agent, Home terminal, Software Configuration, Software Interfaces

### 1. はじめに

一般家庭のインターネット接続が普及し、情報家電も接続可能なホームネットワークが期待されつつある。様々な企業や団体が関連する研究・開発を進めつつある。

ホームネットワークの中に宅内の中心となるサーバ(ホームエージェント)を含むシステム構成を想定し、ホームネットワーク内外の端末とサーバについて、備えるべき機能、並びにそれらの間のデータ交換を考慮したソフトウェア構成のあり方について検討する。

### 2. システム構成

ホームネットワークシステムは、ホームエージェント(HA)、端末(HT)からなるシステムを構成する。ホーム端末としては、ホームネットワーク内のものばかりではなく、外部の端末も対象とする。トポロジは、HAを中心とするスター型とする。

#### ○ホームエージェント(HA)

- ・ HT・ユーザの静的・動的情報、ホームネットワーク内機能及び情報を管理する。
- ・ HTとのデータ交換や通信機能並び、ホームネットワーク内の主要サービス(WEBサービス等)を実現する種々の機能を持つ。
- ・ ルータ、ファイアウォールの機能も持つ。

#### ○ホーム端末(HT)

- ・ ホームネットワーク機能を持つ家庭内機器。
- ・ 識別番号(HT\_id)、通信機能を持つ。
- ・ すべてのHTはHAに登録される。

\* 大学院情報電子工学専攻

\*\* 電子情報学科

また、ホームネットワーク外のインターネット中にあり、ホームネットワーク内の機能を利用する端末も関係する。

- ・ 外部HT(XHT, eXternal HT)  
家族の職場や親戚、知人宅内にある端末、あるいは携帯電話、PDA等の移動端末
- ・ 外部HA(FHA, Foreign HT)  
親戚、知人等の宅内にあるHA

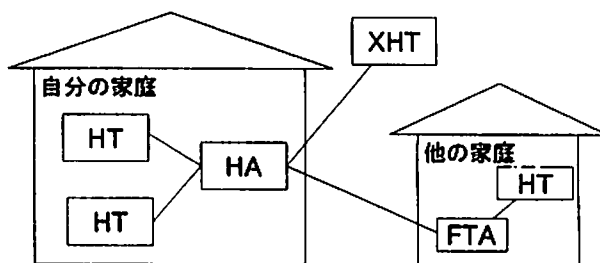


図1. システム構成図

### 3. ソフトウェア構成の検討

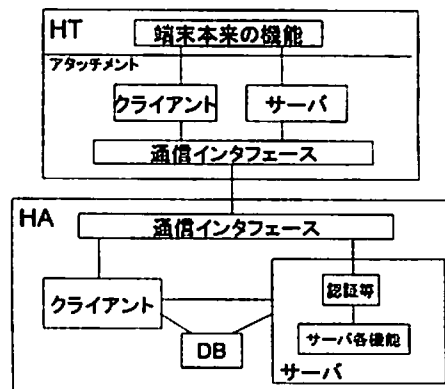


図2. ソフトウェア構成図

HA, HTが備える機能を示す。HAの各機能には実際には様々なものが含まれる。また、HTの本機能は、本来そのHTが持っている部分であり、それにアタッチメントを付加したものがHTとなる。

本章では、通信インタフェース、HT-HT間での通信、HAの各機能のクラス構成、制御・管理部について検討する。

### 3. 1. 通信インタフェース

ホームネットワークにおいて端末同士が通信する際に、ソフトウェア上での通信インタフェースについて、ソケット、通常のRPC、オブジェクト通信であるSOAPのうち、どの方法を用いるかを検討する。

論点としては、ホームネットワーク製品は様々なメーカーによって開発されることになるので、互換性がとりやすいこと、仕様か簡単でわかりやすいことなどが重要となる。

#### (a) ソケット

プログラムからみた通信の対象はストリームである。ソケットを用いたネットワークプログラムは、基本的には、ソケットを通じた入出力である。サーバとクライアントのプログラムの内容は独立だが、プログラム中でこの読み書きのペアが出現する順番は正確に同じでなければならない。コネクションの切断などネットワークの動作がそのままみえてしまっている部分も多く、通信コードの作成、デバッグ、および保守のほとんどの作業をプログラマが行う必要がある。しかし、作成するアプリケーションがオーバーヘッドに対してきわめて敏感な場合に有効である。

通信に使用するバッファに、HT\_idや、実行させたい命令の情報などを入れる。

この方法を用いた場合、クライアント・サーバのプログラム内でネットワーク部分が記述されることになるので、ソフトウェア構成としては、クライアント・サーバは通信インタフェースを含む形となる。

#### (b) RPC (RMI など)

プログラムからみた通信の対象は関数である。プログラム内において、ネットワークは完全に隠蔽されており、サーバとクライアントのプログラムは同期を取る必要がなく、独立に開発が可能である。しかし、下位ではソケットが使われているので、処理、データ量ともに(a)よりオーバーヘッドが大きくなる。

関数を呼び出す際に、引数にHT\_idや、その関数自身の引数を入れる。

ソフトウェア構成としては、ネットワーク部分が隠蔽されているため、クライアント・サーバと通信インタフェースを分離して考えることができる。

#### (c) RPC (SOAP)

通信の対象はオブジェクトである。同じ分散オブジェクト通

信にCORBAがあるが、CORBAはJAVA同士でしか使用できない。ホームネットワークでは様々なメーカーによって製品が開発されるため、異なる言語で作られたプログラム同士でも通信できた方がよい。SOAPも下位ではソケットを用いているので、オーバーヘッドは通常のRPCよりもさらに大きくなる。SOAPでは通常はhttpなど一般的に使用されるポートを使うので、セキュリティを厳しくした状態でも気にすることなく使用することができる。逆に言えば、別の部分でセキュリティを強化しなければならない。こちらも引数にHT\_idなどを入れる。

ソフトウェア構成は、こちらもクライアント・サーバと通信インタフェースを分離して考えることができる。

これらを踏まえた上で、どの方法が最適かを検討する。通信インタフェースとクライアント・サーバ部分は分離されていたほうが、管理面でも共通化でも有効である。RPCであれば、新しくホームネットワーク製品を開発するときに通信部分はほとんど考慮に入れなくて済む。後からプログラムを変更する際など、ソケットでは処理が細かくなっているため、拡張性についてもRPCが有効である。また、SOAPでは通信相手を一つのオブジェクトとしてみることもできる。よって、本稿では通信インタフェースにSOAPを採用する。オーバーヘッドやHTの処理能力については本稿では考慮していないが、センサのような小型で省電力が求められるものに関してはソケットを用いてオーバーヘッドのコントロール等を行ってもいいかもしれない。

### 3. 2. 通信方式

通信インタフェースにSOAPを採用したので、HTからHAへ通信する際、もしくはHAからHTへ通信する際には、通信相手をオブジェクトとして機能を直接呼び出すことができる。ここでは、HTからHTの機能を呼び出す際、もしくはHTからFHA・FHAに登録されているHTの機能を呼び出す際など、直接接続されていない端末間での通信時にどのような形で通信するのがよいか、また、その際に必要な機能やソフトウェア構成の変化について検討する。論点としては、HAでのHT情報の一元管理、呼び出しの簡単さ、オブジェクト指向が重要となる。

HTからHTの機能の呼び出しを例に挙げ検討する。また、呼び出しをする側のHTをアクターHT、呼ばれる側のHTをリアクターHTと呼ぶこととする。

#### (a) HT-HT間で直接呼び出し (認証なし・P2P)

HAはルーティング機能のみを行い、HT同士の通信には直接関与せず、HTとHTが直接通信する方法。もっともシンプルであり、アクターHTから見た場合、リアクターHTをそのままオブジェクトとしてみることもできる。しかし、HT情報はHAで一元管理されているので、認証を行うことができない。

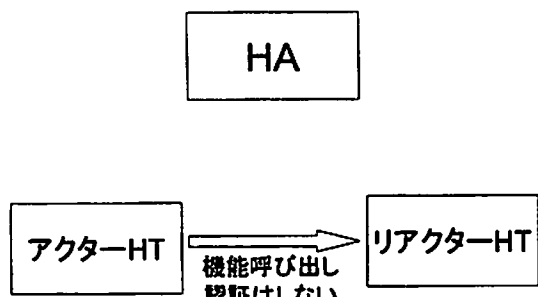


図3. HT-HT間で直接呼び出し(認証なし)

(b) HT-HT間で直接呼び出し(認証を行う・P2P)

(a)と同じようにHAは介さずに直接HT同士で通信を行うが、HT-HT間で認証を行う方法。この方法を実現するにはなんらかの方法でリアクターHTがHAから認証情報を取得しておかなければならない。

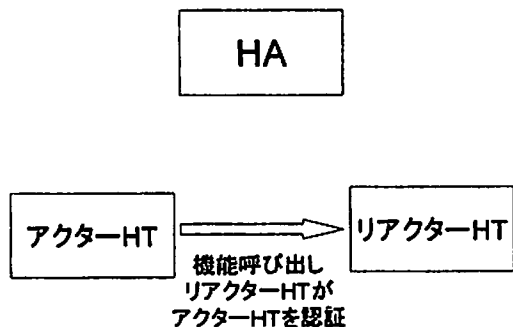


図4. HT-HT間で直接呼び出し(認証を行う)

(c) 認証のみHA(ハイブリッド型P2P)

機能の呼び出し自体はHT-HT間で行うが、認証にHAが介入する方法。通信するまでの手順を以下に示す。

- (1) アクターHTがHAの認証機能呼び出す。
- (2) HAは、リアクターHTが持つアクターHTの認証情報を渡す機能呼び出し、アクターHTにも認証情報を返す。
- (3) アクターHTはHAから受け取った認証情報とともにリアクターHTの機能呼び出し、認証を受ける。

この方法であれば、リアクターHTが情報を持たなくても認証を行うことができ、かつ、HTの機能を直接呼び出すことができる。

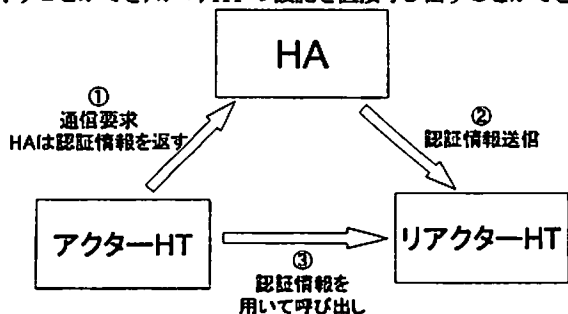


図5. 認証のみHA

(d) HAが呼び出しを中継(クライアント・サーバ型)

HAが呼び出しを中継する方法。HAがリアクターHTの呼び出し中継機能を持っており、以下の手順で通信する。

- (1) アクターHTはHAの中継機能呼び出す。
- (2) 中継機能がアクターHTに代わってリアクターHTの機能呼び出す。
- (3) HAは、リアクターHTから受け取った戻り値をそのままアクターHTに返す。

この方法であれば必然的にHAで認証することが可能になる。

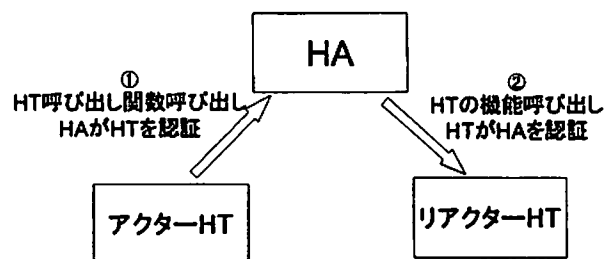


図6. HAが呼び出しを中継

これらのなかでどの方法が最適かを検討する。まず、(a)の方法では認証を行っていない。家庭内のみでHTを使うのであれば、HAのルータ機能(ファイアウォール機能)や無線LANのセキュリティを正しく設定してあれば、認証を行わなくてもそこまで問題ないだろう。しかし、家庭外の端末から直接HTが呼ばれることを考慮するとセキュリティは必要になる。(b)の方法ではリアクターHTが他のHTの情報を取得する機能を持つことが必要になり、その情報を保持していなくてはならない。つまり、HAにはHTに認証情報を渡す機能が、HT側には、HAから認証情報を取得し、それを管理する機能が必要になる。

(c)の方法ではHT情報を一元的に管理しているHAで認証することができるが、呼び出し時の動作が少し複雑である。(d)の方法では、呼び出しに一手間かかるが、必ずソフトウェア面でHAを経由するため、セキュリティ面、管理面ともに効果がある。また、要求を受けた側はアクター側を認証するが、アクター側もリアクター側を認証した方が、安全である。(c)(d)の方式であれば、リアクターHTにとって、アクター側は常にHAなので、HT側に認証情報が少なく済むという利点がある。オブジェクト指向という観点からみると、(a)、(b)、(c)の方法ではアクターHT側からみてリアクターHTをそのままオブジェクトとしてみることもできるが、(d)の方法ではHAの機能呼び出すため、リアクターHTを直接呼び出し相手としかみることが出来ない。

以上の検討から、本稿では、HAでの管理と処理の簡単さを重視し、(d)の方法をとる。リアクターHTのオブジェクト化という点に関しては、一工夫してHAをリアクターHTのようにみせるようにする。

HTからFHAの機能呼び出す際には、リアクターHTをそのままFHAと読み替えばよい。HTからFHAに登録されているHTを呼び出す際には、リアクターHT→HA→FHA→の順に呼び出すことになる。その際、FHAからみたHAは、HTとほぼ同等のものとみなすと処理が共通化できる。

### 3. 3. クラス構成

HAが持つ機能は、以下の3種類に分類できる。

(1) 呼び出し時に必ず実行する機能

- ・ 認証：端末の認証
  - ・ アクセス権チェック：操作するユーザの認証と、その操作を実行できる権限を持っているかのチェック。
- など。

(2) どのような家庭のHAでも必ず持つ機能

- ・ 在宅状況管理：家庭に所属するユーザの在宅状況の管理を行う。ユーザが外出、帰宅したときに呼び出される。
- ・ HT状態変化管理：各HTの電源のON・OFFなど、状態が変化したときに呼び出される。HTの連携操作のために必要になる。
- ・ 他のHTの情報・機能取得：他のHTの機能呼び出すために、情報を取得しておく機能
- ・ 他の端末の機能呼び出し：3. 3章で検討した(d)の方法のために使用する中継機能

(3) 各家庭のHAによって固有の各機能

家庭内に設置されたセンサが何かを検知したときに、その検知を受け取り、状況に応じて最適な通知先に通知する機能や、HA自体をデータサーバとする機能等。これらは、ユーザが必要とすれば適宜HAにインストールされるものとする。

3. 1章で出てきた、ソケットでの通信を仮想的にSOAP呼び出しとする機能は(b)に分類できそうであるが、ソケット用の別プロセスでサーバを動かすことになるので、この議論には含まれない。

クラス構成を検討する目標として、呼び出し時に必ず実行される機能は一つにまとめるなど、なるべく処理を共通化したい。まず、HTからHA内の機能呼び出しと、他の端末の機能呼び出しについてユースケース図で表すと、図7のようになる。

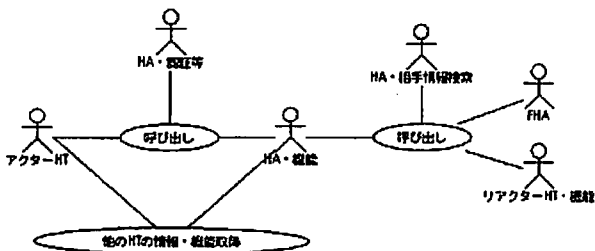


図7. ユースケース図

次に、これをアクティビティ図で表すと、図8のようになる。

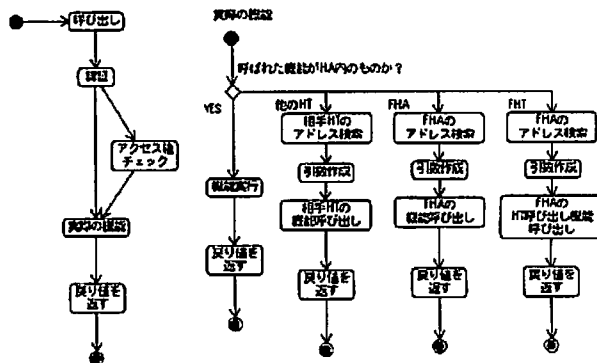


図8. アクティビティ図

認証やアクセス権チェックなど、機能呼び出し時に必ず実行されるものは、一つのクラスにまとめておく。このクラスをHAクラスと定義する。

これをもとに、以下の案についてクラス図を用いてどの案がよいか検討する。管理面、拡張性などが論点となる。

(a) 全ての機能のクラスをHAクラスが持つ。

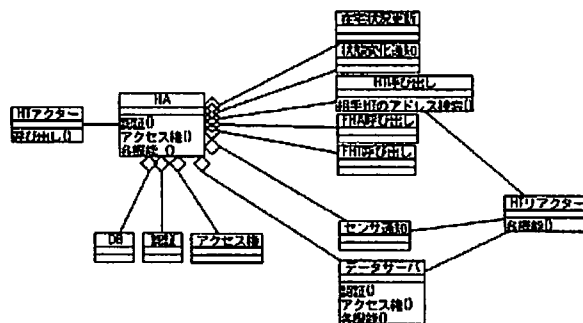


図9. (a)のクラス図

(b)HAが必ず持つ機能のクラスは、HA機能クラスにまとめ、HA機能クラスとそれぞれの個別の機能のクラスはHAクラスを継承する。

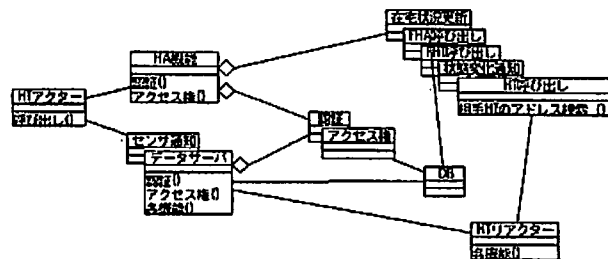


図10. (b)のクラス図

(c) HA が持つすべての機能を個別にし、それぞれのクラスがHAクラスを継承する。

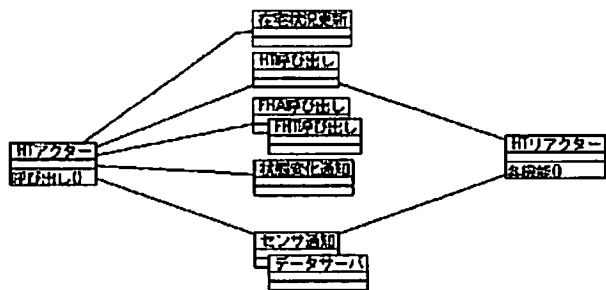


図11. (c) のクラス図

これらの案の中で、どれが最適かを検討する。呼び出す側から見た場合はどの方法でも変わらないので、HA内だけの問題となる。

まず(a)の方法では、初めに開発する際にはプログラムの負担が小さいが、後から拡張や変更が利きづらい。ホームネットワークはこれからまだまだ発展していく分野なので、拡張性に富んだ形にしておくほうがよい。しかし、この方法でないと実装できないプログラム言語もある。(b)の方法では、HAが必ず持つ機能がまとまっているので、仕様を管理する際に見やすくすっきりしている。個別の機能も後から追加するのが簡単である。(c)の方法では、HAが必ず持つ機能も後から追加しやすい。しかし、HAが必ず持つ機能が追加されることはそれほど多くないと思われるので、ここでは管理面を重要視し、(b)の方法を採用する。

実際にアクターHTからリアクターHTの機能呼びたいときには、HT\_id、認証情報、呼び出し関数名、その関数の引数などを引数に入れ、アクターHTが、HAが持つHT呼び出し関数を呼ぶ。HAは、リアクターHTからの戻り値をそのままアクターHTに返す。

### 3. 4. 制御・管理部

図2において、HA・HTともに、実際にはサーバ・クライアントを制御・管理する部分が必要になる。この制御・管理部の機能には以下のようなものが挙げられる。

- ・ アタッチメント全体の管理・制御
- ・ サーバが持つ機能の管理、クライアントとの連携
- ・ 他のHTが持つ機能の管理

HA内であれば、サーバとクライアントの間に置けばよい。しかし、HTにおいては端末本来の機能との兼ね合いや処理の仕方など、考慮する点がある。この機能をどこに置くかを検討する。

#### (a) サーバ・クライアントと端末本来の機能の間

HAの場合と同じ位置。端末本来の機能から、制御・管理部を通してサーバ・クライアント部へアクセスする。

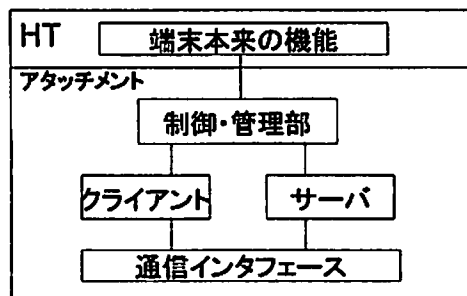


図12. (a) のHTソフトウェア構成図

#### (b) 端末本来の機能の内部

端末本来の機能の内部に制御・管理部を入れてしまう。

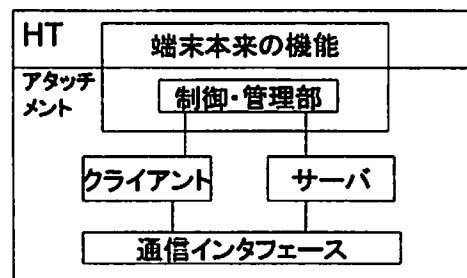


図13. (b) のHTソフトウェア構成図

#### (c) サーバの内部

サーバの内部に制御・管理部を入れてしまう。

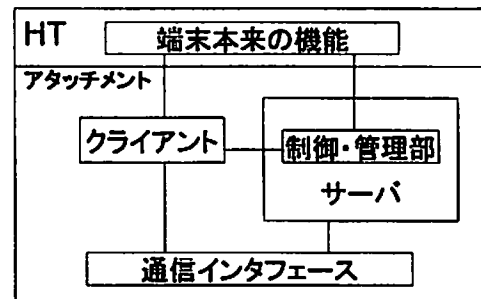


図14. (c) のソフトウェア構成図

これらの案の中で、どれが最適かを検討する。

(a)の形はサーバ側と同じ形であり、もっとも基本的な形である。

(b)の形は、HTからHAや他のHTを操作するときには、実際には端末本来の機能からの操作になり、管理・制御部は端末本来の機能と密接に関係するため、直接つながっているほうが制御しやすいかもしれないという考えから出した。しかし、実際に動作させるときには効率はいまいだろうが、設計面としては、端末本来の機能とアタッチメントとの境界が不明確になる。また、端末本来の機能の開発時にも(a)の形よりもアタッチメント側の部分を考慮しなくてはならない。

(c) は特殊な形であるが、クライアント部と他の部分との関係は、端末本来の機能とサーバから指示を受けるだけであり、クライアント部から HT 内の他の部分へ指示が出されることは普通ないという考え方から出した。また、この機能が使用されるのは、実際にはサーバに関する部分がほとんどである。しかし、ここでは制御・管理部が他の HT や HA の情報を持つことにしているので、端末本来の機能からクライアントに指示を出す際には、一度サーバ側にアクセスしなくてはならず、データの流れが不自然になる。他の HT や HA の情報を端末本来の機能で持つならばこれは問題にはならない。

結果としては、HA と同じ構成であり、特に問題のない (a) の形を選択することにした。

#### 4. 評価

3. 1 章では、オーバーヘッドは考慮にいれず、仕様の共通化・単純化とオブジェクト指向という点を重視した。オーバーヘッドについては、現在の端末の処理能力を考慮すれば問題ないと思われるが、本来であれば実験・考察するべきだっただろう。3. 2 章では、処理の共通化・単純化を重視した。直接呼び出すことができず、オブジェクト指向で扱えなくなってしまった。3. 3 章では、拡張性と単純化を重視した。3. 4 章では、HT 側の制御・管理部を HA と同じように、スタンダードにサーバ・クライアントと端末本来の機能の間に置くことにした。ホームネットワークは様々なメーカーで共通の仕様でなくてはならず、これからさらに発展していくことを考慮し、このような検討結果となった。

#### 5. まとめ

ホームネットワークに用いる宅内に設置するサーバ (ホームエージェント) に関するソフトウェア構成を検討した。通信インターフェースは SOAP を用いる。端末から別の端末の機能呼び出す際には、一度 HA が持つ中継機能呼び出し、HA に実際に呼び出したい端末の機能呼び出してもらう。クラス構成は、HA が必ず持つ機能は一つのクラスにまとめ、個別の機能は別々のクラスとして端末から呼ばれる。本稿では、様々なメーカーや団体が共通に使えること、ホームネットワークはこれからさらに発展していくことを重視した検討結果となった。まだホームネットワークは実際には普及してはいないが、これからさらに普及し、ホームネットワークが一般的なものになってきたときに、どのような仕様になっているか、どのような使われ方をされているかが注目である。

#### 参考文献

- 1) ECHONET CONSORTIUM  
<http://www.echonet.gr.jp/>
- 2) 小林拓史, 吉田裕 “ホームネットワークにおけるホーム端末属性登録方式—ホーム端末機能のオブジェクト化によるホームエージェントへの登録—” 信学ソ大, B-7-37, 2004
- 3) 小林拓史, 吉田裕 “ホームネットワークにおけるセンサの多目的利用” 信学ソ大, B-7-47, 2005
- 4) 齊藤紀明, 吉田裕 “ホームネットワークにおけるアクセス制御方式の検討” 信学ソ大, B-7-44, 2005