

ホームゲートウェイの設計と実装

～デジタル家電の統合制御～

寺本 圭一 門間 信行 斎藤 健

(株) 東芝研究開発センター 情報・通信システム研究所

近年の AV/白物家電分野では、新たに機器間の通信・制御を可能とする新しいデジタル情報家電機器の登場が求められるようになってきた。各種家電機器が接続される家庭内ネットワークのメディア/プロトコルに関しては様々な方式が提案されているが、我々は、これら複数のネットワークを相互に接続/収容し、更に、家庭外のインターネット端末からも通信・制御可能なゲートウェイ機能の実現を目指している。特に、WWW や JAVA における技術を積極的に利用することによって、通常の Web ブラウザから家庭内機器を制御するアーキテクチャについて検討している。

Design and Implementation of Home Gateway

– Integrated Control of Digital Consumer Appliances –

Keiichi TERAMOTO, Nobuyuki MONMA, Takeshi SAITOH
Communication and Information Systems Research Labs., R&D Center,
TOSHIBA CORPORATION

Recently, there appear digital consumer appliances in the market. They enable such new applications as remote control and communication. There are several types of home network media and protocols, such as IEEE1394, Echonet, and the Internet, and we have to consider how to interconnect them. In this paper, we propose home network architecture and its key component "home gateway", which interconnects those several networks. It has a capability of the Internet connectivity, and the user can control home appliances via web browser through the Internet. This technology is based on WWW, JAVA, and the related Internet technology.

1 はじめに

近年の家庭内へのインターネット導入により、パソコンにとどまらず、新たに家電分野に対しても機器間の通信・制御を可能とするネットワーク技術を装備した新しい情報家電の登場が求められるようになってきている。

各種家電機器が接続される家庭内ネットワーク (IEEE1394, 電灯線, 無線, 電話線など) に関する規格化/標準化については、既に様々な方式が提案されているが、それぞれ異なるネットワーク媒体や伝送/制御プロトコルを用いて開発が進められている。我々はこれらを統合的に扱い、ネットワーク経由で機器の操作を行うユーザに対して、ネットワーク媒体やプロトコルの違いを意識させない (隠蔽する) ゲートウェイ・アーキテクチャを提供しようとしている [1]。

さらに、家庭外からインターネット経由あるいは直接公衆網を経由したダイヤルアップ接続 (無線も含む) によって、家庭内の家電機器を容易にかつ安全に制御するための仕組みも提供しようとしている。

ここで、ゲートウェイとは、家庭内の複数の異なるネットワークを束ね、各ネットワークに接続された機器を集中的に制御・管理するコントローラとしての役割を有するが、我々は更に、接続される機器をインターネットや公衆網を介して遠隔制御できるような家庭内と外界との接点的

な役割をも担う物として位置付けている。我々はこれを、「ホームゲートウェイ」と呼称し、異種ネットワーク間接続やインターネットとの接続を強く意識したものとして扱うことにしている。

今回の設計では、ホームゲートウェイを WWW 技術や JAVA 技術を積極的に利用することによって、セキュアで各種モジュールのモデル化の見通しが立て易い設計となるよう考慮している。具体的なホームゲートウェイの実装形態としては、Web サーバを servlet により機能拡張する形式で検討している。

本稿では、まず、我々が想定するホームゲートウェイが装備すべき機能について考察し、現在設計/試作中のシステムの概要について述べる。

2 ホームゲートウェイ機能

ホームゲートウェイは、ネットワーク接続された全家電機器の集中制御装置としての役割のほかに、家庭内における複数の異なるネットワークを束ねる役割、および、家庭外の公衆網やインターネットといった外界との接点としての働きを備える。実体は PC や STB、TV、SOHO ルータの一部の機能として搭載されていくものと想定している。各種ネットワークに接続される家電機器を制御するための機能は、各ネットワーク毎に用意される集中コントローラ機器上に装備されることも考えられるが、本報告ではホームゲートウェイ機能と併せて STB などに集約される形式を考えている。

図 1 に、現在想定しているホームゲートウェイに関わる機器構成例を示す。

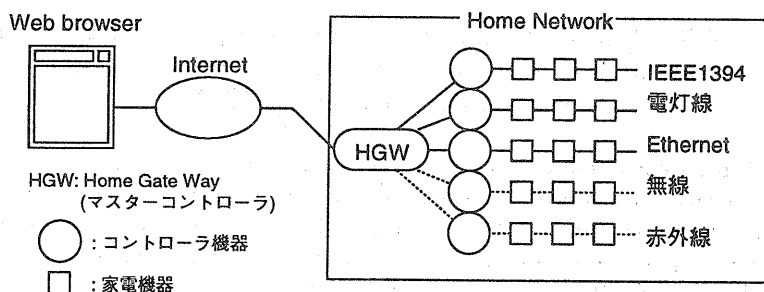


図 1: 想定するホームゲートウェイおよび制御家電機器の構成

このように、ホームゲートウェイをマスタコントローラ、および、異種 (混在) ネットワーク間、インターネット接続の接点としてとらえた場合、以下のような機能が必要であると考えている。

- ネットワーク状態管理機能
- plug and play(hot-plug-in) 管理機能
- 制御モジュール (デバイスドライバ、処理サーバ、GUI アプリケーションなど) の登録/制御/削除管理機能
- 機器エンティティのアドバタイズ (機器プロパティなどを対象としたディレクトリサービス)
- クライアントへの制御メソッド提供機能 (メソッド単位での公開、非公開の指定)
- 機器間協調作業時の連携/調停機能 (複数の機器を同時に操作し複雑な共同作業を行う)
- 各種ネットワーク間での相互接続/制御を可能にするプロトコル変換機能 (proxy 機能など)

- ユーザ認証, アクセス設定/制御機能 (セキュリティ)
- セッション管理機能 (状態保持含む)
- 障害回復機能 (不慮, 不正な操作後の状態回復など)
- データ転送機能 (含 AV データ)
- firewall 機能

今回の設計では, 主にインターネットを介して, 遠隔地から家庭内の家電機器を制御するという側面に注目して基本機能の考察を行った. 上記項目の中では特に, プロトコル変換機能 (IP と家庭内各種ネットワークプロトコルの変換), クライアントへのメソッド提供機能, の実装検討を行った.

以下では, インターネット経由の遠隔制御を行う場合に, 我々が必要と考える要求条件について述べる.

2.1 要求条件

まず, 外界 (インターネット, 公衆網) との接続を考慮した制御を考えると, 様々な場所や環境から家庭内にアクセスされる可能性を考慮しなければならない. 電話で家庭内に直接ダイヤルアップする場合は別として, インターネット経由では, 会社や学校, 公共施設に置かれた個人所有/共有の PC/WS/端末/PDA からアクセスしたい要求が高いと考えられる. こうした多様な端末からのアクセスを容易に実現するには, クライアント側で提供される Web ブラウザを利用することが有効と考えられる. 特に JavaVM をサポートする Web ブラウザであれば, applet を動的にダウンロードすることによって, ブラウザ側にあらかじめ家電機器制御アプリケーションをインストールしておくことなく, 実行することが可能になる. また, plug-in module のような形式で専用アプリケーションをブラウザに組み込んで使用することも可能であるが, 共用端末での利用は制限されるであろう.

次に, アクセス制御をホームゲートウェイ上に設定する場合には, 接続先の対象が家族である個人専用の端末やブラウザから必ず行われることになるのか, それとも, 誰でも使える共有端末から行われるのかを考慮する必要がある. 個人専用のブラウザであると特定できる場合であれば, アクセス認証やセッション状態といったプライベート情報を Cookie ファイルのような形式で端末側に保持させておき, 以降これを利用した認証やセッション管理を行うことが可能であるかもしれないが, 共用ブラウザの場合は問題がある.

また, 会社や学校のインターネットインフラを利用して, 家庭内のホームゲートウェイに接続する場合, 特に会社であれば, firewall 問題を考慮する必要がある. 多くの会社ではセキュリティ上の問題から, 不要なポート番号 (プロトコル) を経由したインターネットへのアクセスを制限しているが, 通常, http 用のポートは利用可能になっていると思われるので, これを利用した接続形態が望ましい.

更に, ホームゲートウェイ機能自体は, PC 以外の STB やホームルータ, 何らかの組み込み機器上に対しても導入されることを想定しているため, 複数のターゲットに対応できる可能性が高いと思われる Java プラットフォームを利用することが望ましい.

以上まとめると, 以下のようなことを考慮してホームゲートウェイ機能を実現する必要があると考えられる.

- クライアント側は Web ブラウザ (JavaVM サポート) を想定.
- クライアントとホームゲートウェイの通信には, firewall 超えを考慮して http を使用.
- ホームゲートウェイ機能の実現はプラットフォーム非依存を考慮し, 現状では Java ベースの実装を考える.

3 Web サーバの機能拡張

通信プロトコルに http を使用するために、ホームゲートウェイには Web サーバ (httpd) を配置する。これに従い、ホームゲートウェイ自身を Web サーバの機能拡張により実現することを検討する。

Web サーバを機能拡張するには、「CGI 方式」、「サーバサイドスクリプト方式」、「Java 方式」などの方法が考えられるが、今回は、要求条件にて述べたように Java 方式を採用する。

Java では、セッション管理やトランザクション管理の仕組みなどクラスライブラリ群も充実しているおり、高度なプログラムが比較的容易に実現できる可能性が高い。しかしながら、アプレットを使った実装では、ダウンロード自体に時間がかかるといった欠点もある。

4 applet, servlet, backend server による実現

今回の設計では、ブラウザ側には applet を提供し、Web サーバ側に servlet を配置して、実際の機器制御処理を担当する backend server へ仲介させる方式を採用する (図 2)。

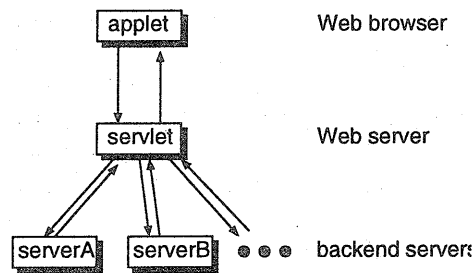


図 2: applet, servlet, backend server 方式

こうした applet, servlet, backend server による 3 層構造を採用する理由について述べる。

applet は、JavaVM をサポートする多くのブラウザで利用できることと、高度なプログラミングが比較的容易に実現できる点があげられる。

次に、servlet を使用して backend server への仲介をはかった理由としては、firewall 超えを考慮した http ベースでの Web サーバとブラウザ (applet) 間通信を前提としており、この通信を servlet が受けて、対応する backend server に振り分けるためである。別の実現方式として、Web サーバ上に backend server を配置し、これと applet が直接 http を使って RMI 通信する形式も実現可能であるが、この場合、Web サーバ側で RMI 用の registry を http と同じポートで立ち上げ、Web サーバ用と RMI 用とで区別するなどの処理が必要となる。また、Web サーバ以外のマシンに backend server を置いた場合、applet との直接的なオブジェクト間通信は、applet のセキュリティ上の制約からできない。

一方、servlet を介する場合には、servlet 自身が各 backend server に向かって RMI などを直接呼び出して通信できるため、applet に代わって家庭内ネットワーク接続機器に対して自由かつ柔軟に Java 通信プログラムなどを適用することが可能になる。もちろん、Web サーバ上に backend server を配置しても良い。

5 ホームゲートウェイの試作

今回の試作は、AV 網 (IEEE1394) と電灯線 (X10[3]) の 2 種類のホームネットワークを例にとり、これらを統合的に制御・管理するための機能、さらに、機器の操作を家庭外からインター

ネット経由で制御するための機構について PC 上で実現するための検討/試作を行った。

前述のように、今回のホームゲートウェイの試作では、Java 技術 (applet, servlet, RMI), WWW 技術 (http) をベースに作成することによって、Java 環境での開発環境の統一、既存ブラウザの流用、firewall 経由での制御を考慮している。

5.1 開発環境

基本的に、試作したプログラムにはすべて Java を使用している。開発用およびターゲット用 PC には、標準的な Pentium 搭載パソコンを使用した。

- PC: ホームゲートウェイ用 (白物機器制御サーバ含む), クライアント用 (Web ブラウザ表示), バックエンドサーバ用 (AV 機器制御サーバ) の 3 台
- OS: Windows95 および FreeBSD 2.2.6-RELEASE (二つの環境で試作/動作確認)
- JDK: 1.1.6 版
- Web サーバ: Windows95 用に Java Web Server v.1.1[2] および FreeBSD 用に Apache v.1.2.6
- X10 コントローラおよび接続白物家電機器: CP-290(X10 コントローラ), デスクスタンド, TV など を電灯線を介して接続
- AV 家電機器: デジタルビデオカメラ (DVC), デジタルビデオレコーダ (DVTR) を IEEE1394 を介して接続

5.2 ソフトウェア構成

ホームゲートウェイを介した家電機器の遠隔制御に必要なソフトウェア構成として, applet, servlet, backend server による 3 層構造を採用した。

ホームゲートウェイの実体はこのうち, servlet により機能拡張される Web サーバ自身を指すが, Web サーバ上にはこの servlet と通信するフロントエンド GUI となる applet も格納される。

今回, servlet から実際の処理を依頼される backend server は, AV 家電機器制御用 backend server に関しては別途コントローラマシン (PC) を用意して走行させている一方, 白物家電機器 (X10) 制御用 backend server に関しては, ホームゲートウェイ上にこれを配置して走行させている。しかしながら, どちらの server とも, servlet 間の通信には Java RMI を利用しているの、それぞれホームゲートウェイ上に配置されても, 別のコントローラ用マシン (機器) に配置されても構わない。

以下, applet, servlet, backend server の実装についてそれぞれ説明する。

applet ~ フロントエンド部

applet は, インターネットを経由して, 家庭内の機器を制御するための操作パネル (GUI) を提供する。

AV 機器や白物機器用の各クラスを構成する際に, 共通の部分 (Label や gif イメージ, servlet への送信/受信用基本メソッド) を抽象クラス化することによって, 再利用を考慮したコンポーネント設計を行っている。実際に, 各 AV 機器や白物機器を表現するオブジェクトは, この共通クラスからの派生クラスとして実現されている。ユーザへの家電機器制御操作画面は, 見かけ上, AV 機器や白物家電機器が接続される各々のネットワーク (IEEE1394, 電灯線) やそのプロトコル (AV/C, X10) の違いを, ユーザに全く意識させないような形で表現した。今回の試作では, 基本制御機能の検証のみを目標としたため, デザイン上の表現は簡素にし, 機器の gif イメージと

AWT によるシンプルな GUI コンポーネントからなる。制御ボタンは、AV 機器に関しては再生、停止などの基本的な操作、X10 対応機器の制御も On/Off、ライトの明暗のみとなっている。

servlet と applet 間で簡単な制御プロトコルを用意した。applet は、servlet へ request 送出後、servlet から response を受信する。具体的には、applet 側から URLConnection を AV 機器用 servlet あるいは白物機器用 servlet に対して確立した後、このコネクションに対する出力ストリームを作成し、マジックコード、ネットワーク種別 (AV 系、白物系など)、制御対象機器 ID、操作の種類に応じたイベントコードを 1 メッセージにして送出する。すると servlet からは、このメッセージを解釈し実行した結果として result code (成功か無効とその要因) が返信される。結果を受け取ると applet は、成功時には、その操作を行った結果として次回の操作で利用不可能となる制御ボタンを disable にし、失敗時には、エラーを表示するなどする。

servlet ~ web サーバ

servlet は、request メッセージを受け取り、メッセージの解釈/正当性チェック、対応する backend server の RMI 呼び出しを行う。更に、backend server の処理が終了した後、servlet は backend server からの結果をもとに result メッセージを作成し、これを applet 側に返信する。

backend server ~ 機器制御用各種サーバ

backend server は、各種制御機器毎に用意される管理サーバ自身である。本サーバは RMI に基づき外部に対して公開可能な API を用意し、servlet からこれ呼び出させる形式で実際の制御処理を行う。backend server は Web サーバと同じ装置上になくても良く、別途ネットワーク固有のコントローラ上に散在する状況でも良い。

6 さいごに

今回の試作では、Java による開発環境の統一、既存ブラウザの流用、firewall 経由での制御を考慮した設計を行った。比較的小さなメッセージの送受信で applet/servlet 間通信を行い、servlet と backend server 間を RMI を使って構築するという形式で、ホームゲートウェイを Web サーバの機能拡張により実現できる可能性について検証した。このように、servlet と backend server 間の通信には RMI を利用しているため、backend server の位置透過性が高まり、柔軟な機器構成が可能になる。

今回の試作では、遠隔操作を行うユーザに対する認証、アクセス制御、セッション管理などについては含まれていない。これらは、Cookie、アクセスコントロールリスト、Session Tracking 機構など、既存の (Java) Web Server が提供する機能を流用する方法が考えられるが、このままでは家電機器を扱うユーザが容易に設定/管理するのは難しい。また、インストールに必要なオブジェクト群を自動的に設定する方式についても現在検討中である。

参考文献

- [1] 斎藤、高島、橋本、岡本: “デジタル情報家電の接続を考慮した家庭ネットワークアーキテクチャ,” 信学技報, IN97-128 (1997-11).
- [2] JavaSoft. Java Web Server, <http://java.sun.com/products/java-server>
- [3] X10.ORG. X-10.ORG - X-10 Technology and Resource Forum, 1997. <http://www.x10.org/>.
- [4] RFC-2068 Hypertext Transfer Protocol - HTTP/1.1