

UPnP エミュレータとテストの実装

吉原 貴仁 堀内 浩規

(株) KDDI 研究所

1. はじめに

あらゆる機器をネットワークに接続することで、いつでも、どこでも機器を操作、あるいは機器同士を協調動作させることで生活を支援するユビキタスネットワークへの期待が高まっている。ユビキタスネットワークの円滑な導入、普及のためには、機器の接続や利用の際の煩雑な設定から利用者を解放し、機器や機器が提供するサービスを簡便に利用可能とすることが重要な課題の一つとなる。このための取り組みの一つに業界標準の UPnP (Universal Plug and Play) [1] がある。著者らは UPnP に準拠する機器やサービスの開発および相互接続試験に必要となる UPnP エミュレータとテストを実装した。本稿ではその概要を述べる。

2. UPnP の概要

UPnP は、IP ネットワークの構成情報の自動設定手順に加え、サービスの自動発見手順やサービスを利用するため手順など、複数の手順からなる。以下、PC がプリンタの印刷機能をサービスとして利用する場合を例に UPnP の概要を図 1 とともに示す。

PC とプリンタは IP ネットワークに接続すると DHCP [2] などを使って IP ネットワークの構成情報をそれぞれ自身に自動設定する (図 1(1))。

サービスを利用する機器 (ここでは PC。以下、UPnP コントロールポイントと呼ぶ) は SSDP [3] M-SEARCH 要求を広報して所望のサービス (ここでは印刷サービス) を自動発見し、サービスを提供する機器 (ここではプリンタ。以下、UPnP デバイスと呼ぶ) からサービスを利用するためのインタフェース (以下、サービスインタフェースと呼ぶ) を入手する (図 1(2))。UPnP デバイスが広報する通知により自動発見する場合もある。

サービスインタフェースは状態変数を引数とし、XML [4] で記述され、印刷実行などの要求とその応答は SOAP [5] を使って送受信される (図 1(3))。

用紙切れなど UPnP デバイスやサービスの状態変化は GENA [6] を使って UPnP コントロールポイントに通知される (図 1(4))。通知を要求する場合、サービスの自動発見以降任意の時点で、UPnP コントロールポイントは subscription 要求を UPnP デバイスに送信する。

3. UPnP エミュレータとテストの必要性

2. で述べたように、UPnP により煩雑な設定をすることなく機器やサービスを簡便に利用できる。一方、次に示すように、UPnP に準拠する機器やサービスの開発および相互接続試験の際には、任意の UPnP デバイスを模擬する UPnP エミュレータと任意の UPnP コントロールポイントを模擬する UPnP テスタが必要である。

3.1 UPnP エミュレータの必要性

UPnP 標準のサービスインタフェースは定義が汎用的であり、開発の際には、サービスインタフェースを新

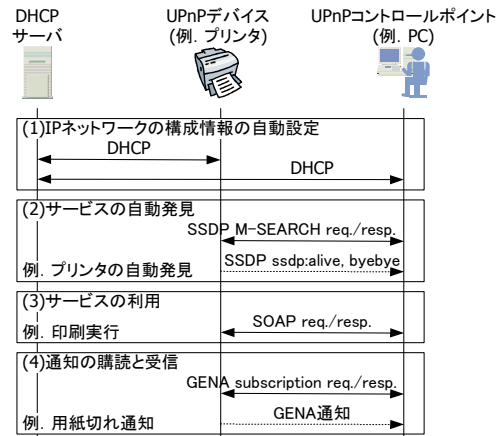


図 1 UPnP の概要。

たに追加拡張しなければならない。例えば、モデムやブロードバンドルータの設定や制御を目的とする UPnP 標準のサービスインタフェース [7] だけでは、動作に必要な IP ネットワークの構成情報さえ十分に設定できない。

このため、UPnP エミュレータを使い、追加拡張したサービスインタフェースを実装する UPnP デバイスを模擬させ、サービスインタフェースが仕様準拠することを試験しながら開発を進める必要がある。

3.2 UPnP テスタの必要性

UPnP 準拠を公称する UPnP デバイスであっても、実際にはサービスインタフェースや通知の形式が仕様にしたがっていない、または、したがっていても仕様通りのタイミングで応答や通知しないものがある。

このため、UPnP テスタを使い、サービスインタフェースや引数を任意に編集して UPnP デバイスに送信し、応答や通知の形式、また、これらのタイミングを試験しながら相互接続性を確かにする必要がある。

そこで以下 4. と 5. で、このための UPnP エミュレータと UPnP テスタをそれぞれ実装する。

4. UPnP エミュレータの実装

4.1 実現機能

任意の UPnP デバイスを模擬するため、次の機能を実現する。

(1) 任意のサービスインタフェースと状態変数の集合を定義する 1 つ以上の XML Service Description をあらかじめ読み込ませる。

(2) 状態変数に自動または手動で初期値を与える。

(3) UPnP コントロールポイントからの要求、具体的には、M-SEARCH 要求、SOAP 要求、subscription 要求に対し、パラメータを任意に編集した応答を任意のタイミングで UPnP コントロールポイントに送信する。

(4) subscription 要求の有無に関係なく、パラメータを任意に編集した GENA 通知を任意のタイミングで UPnP コントロールポイントに送信する。

(5) UPnP コントロールポイントからの要求を契機に、あ

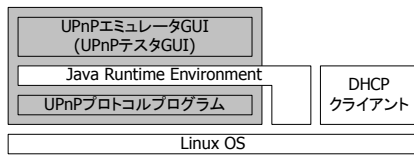


図2 UPnP エミュレータ (テスト) のソフトウェア構成。

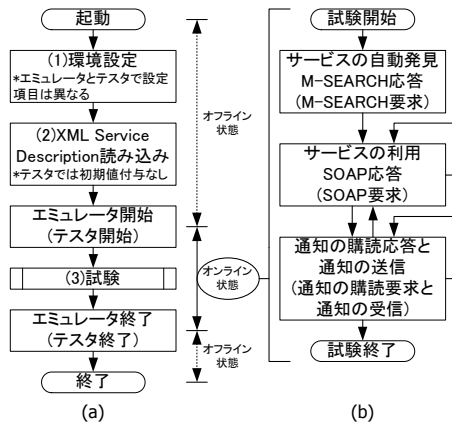


図3 UPnP エミュレータ (テスト) を使った試験手順。

るいは周期的に状態変数の値を更新する。

(6) 受信した要求, 送信した応答や通知のパラメータを実時間表示 (以下, 通信トレースと呼ぶ) する。

(7) 上記機能を試験者が利用するための GUI(Graphical User Interface) を設ける。

4.2 ソフトウェア構成

4.1 の機能実現を目的とする UPnP エミュレータのソフトウェア構成を図2に示す。

OSはDebian GNU/Linux3.0(kernel-2.4.19), DHCPクライアントはOSに付属のものを用いる。UPnPコントロールポイントからの要求を復号, また, UPnPエミュレータからの応答や通知を符号化するUPnPプロトコルプログラムはLinux SDK for UPnP Devices Version 1.2.1を基本に実装する。試験者が利用するUPnPエミュレータGUIはJRE(Java Runtime Environment 1.4.1)上に実装する。また, XML Service Descriptionを読み込むためのXML構文解析ソフトウェアはJREに付属のものを用いる。

4.3 試験手順

図3(a)に示すように, UPnPエミュレータを用いた試験では, 試験設定を行うオフライン状態と, 試験を実施するオンライン状態がある。

(1) 環境設定 (オフライン状態)

試験に用いるHTTPポート番号やSSDP広報用IPアドレスとポート番号, および通信トレースの有効/無効などを設定する。

(2) XML Service Description 読み込み (オフライン状態)

ファイル入力されるXML Service Descriptionを構文解析する。構文が正しい場合, 結果をメモリに展開して閲覧可能とする。また, 状態変数に初期値を与える。

(3) 試験 (オンライン状態)

環境設定やXML Service Descriptionに応じてUPnPデバイスを模擬させ, 図3(b)に示す手順で試験を行う。

5. UPnP テスタの実装

5.1 実現機能

任意のUPnPコントロールポイントを模擬するため, UPnPエミュレータと共通な機能4.1(1)と(7)とともに, 次の機能を実現する。

(8) パラメータを任意に編集した要求, 具体的には, M-SEARCH要求, SOAP要求, subscription要求を任意のタイミングでUPnPデバイスに送信する。

(9) 複数の要求からなるシナリオを編集, 保存, 実行する。

(10) 送信した要求, 受信した応答や通知のパラメータを実時間表示する。

5.2 ソフトウェア構成と試験手順

UPnPエミュレータに準じる。図2と図3で, 括弧内の記述がUPnPテストに該当し, 括弧がない場合にはUPnPエミュレータと共通である。

6. 適用例

一体型モデム(モデム, ブロードバンドルータ, 無線LAN, VoIP機能付き)の自動設定をUPnPで行う自動設定ツールの開発の際に, UPnPエミュレータとテストを試験に適用した。

3.1で述べたように, ここでは無線LANやVoIPの構成情報の値の設定を目的とする77のサービスインタフェースを追加拡張した。モデムへ実装する前に, これらサービスインタフェースが仕様に準拠することや, UPnP標準のサービスインタフェースと競合がないことをUPnPエミュレータを用いて試験した。

上記自動設定ツールは途中で一体型モデムを再起動し, 再起動終了後, 自動設定を継続する。この際, 再起動の終了をGENA通知を使ってモデムがツールに通知する。しかしながら, 期待通りのタイミングで通知しないモデムがあった。UPnPテストを用いて通知の待機時間を調整することにより, ツールが通知を受信し, 受信を契機に自動設定を継続できるように改善できた。

7. おわりに

本稿ではUPnPエミュレータとテストの実装概要について述べた。本エミュレータとテストにより, UPnPに準拠する機器やサービスを実装する前から試験が可能になる。最後に日頃御指導頂く(株)KDDI研究所浅見所長ならびに長谷川執行役員に感謝する。

参考文献

- [1] UPnP Forum: *Universal Plug and Play™ Device Architecture* (2000). <http://www.upnp.org/>.
- [2] Droms, R.: *Dynamic Host Configuration Protocol*, IETF, RFC 2131 (1997).
- [3] Goland, Y. Y., Cai, T., Leach, P., Gu, Y. and Albright, S.: *Simple Service Discovery Protocol/1.0 Operating without an Arbiter*, IETF INTERNET-DRAFT draft-cai-ssdp-v1-03.txt (1999).
- [4] World Wide Web Consortium: *Extensible Markup Language (XML) 1.0 (Third Edition)* (2004). <http://www.w3.org/TR/2004/REC-xml-20040204/>.
- [5] World Wide Web Consortium: *SOAP Version 1.2* (2003). <http://www.w3.org/TR/soap12>.
- [6] Cohen, J., Aggarwal, S. and Goland, Y. Y.: *General Event Notification Architecture Base: Client to Arbiter*, IETF INTERNET-DRAFT draft-cohen-gena-p-base-01.txt (2000).
- [7] UPnP Forum: *Internet Gateway Device (IGD) Standardized Device Control Protocol V1.0* (2001). <http://www.upnp.org/standardizeddcps/igd.asp>.