



## 情報家電プロトコルスタックの開発 成果概要

今野 賢 (Satoshi Konno)

skonno@cybergarage.org

<http://www.cybergarage.org>

## ■オブジェクト指向

UPnP機能を抽象化/隠蔽。開発者はデバイスXML定義およびアクション関数を作成するだけであり、既存アプリケーションのUPnP対応も容易。

### UPnPデバイス定義(CLOCK\_DEVICE\_DESC)

```
<?xml version="1.0" ?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
    .....
    <device>
        <deviceType>urn:schemas-upnp-org:device:clock:1 </deviceType>
        .....
    </device>
</root>
```

### UPnPサービス定義(CLOCK\_SERVICE\_DESC)

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0" >
    ....
    <actionList>
        <action>
            <name>GetTime</name>
            .....
        </action>
    </actionList>
</scpd>
```

```
/*... デバイス初期化 ...*/
CgUpnpDevice *clockDev = cg_upnp_device_new();
/*... デバイス定義情報設定 ...*/
cg_upnp_device_parsedescription(clockDev,
    CLOCK_DEVICE_DESC, strlen(CLOCK_DEVICE_DESC));
/*... サービス取得/定義情報設定 ...*/
CgUpnpService *timeService =
cg_upnp_device_getservicebyname(clockDev, "urn:schemas-upnp-org:service:timer:1");
cg_upnp_service_parsedescription(timeService, CLOCK_SERVICE_DESC, strlen(CLOCK_SERVICE_DESC));
/*... アクション処理関数設定 ...*/
CgUpnpAction *getAction = cg_upnp_service_getactionbyname(timeService, "GetTime");
cg_upnp_device_setactionlistener(clockDev, upnp_clock_actionreceived);
/*... デバイス開始 ...*/
cg_upnp_device_start(clockDev);
.....
/*... デバイス停止 ...*/
cg_upnp_device_stop(clockDev);
/*... デバイス解放 ...*/
cg_upnp_device_delete(clockDev);
```

### アクション関数(upnp\_clock\_actionreceived)

```
BOOL upnp_clock_actionreceived(CgUpnpAction *action)
{
    char *currTime = .....;
    char *actionName = cg_upnp_action_getname(action);
    CgUpnpArgument *currTimeArg =
        cg_upnp_action_getargumentbyname(action, "CurrentTime");
    cg_upnp_argument_setvalue(currTimeArg, sysTimeStr);
    return TRUE;
}
```

## ■ 軽量性

組み込み系であるT-Engine/μITRONを対象プラットフォームとし、高機能版を含めて軽量性を重視した。高機能版は従来より軽量化し、軽量版のオブジェクトサイズは220KB。

## ■ 多言語対応

本成果物を含め、開発者はC/C++/Javaの多様プログラム言語の選択が可能である。また各プログラミング言語版の基本設計は共通であり、その移行は容易。

C++	Java	C
<pre>using namespace CyberLink; Device *dev = new Device(); dev-&gt;loadDescriptoin(...); dev-&gt;setActionListener(dev); dev-&gt;setQueryListener(dev); dev-&gt;start(); ..... dev-&gt;stop();</pre>	<pre>import org.cybergarage.upnp; Device dev = new Device(); dev.loadDescripton(.....); dev.setActionListener(dev); dev.setQueryListener(dev); dev.start(); ..... dev.stop();</pre>	<pre>CgUpnpDevice *dev = cg_upnp_device_new(); cg_upnp_device_parsedescription(dev, ...); cg_upnp_device_setactionlistener(dev, ...); cg_upnp_device_setquerylistener(dev, ...); cg_upnp_device_start(dev); ..... cg_upnp_device_stop(dev);</pre>

## ■ 移植性

一般的なWindows(WIN32)やLinuxプラットフォームだけではなく、Unix全般および組み込み系のT-Engine/μITRONを含む、幅広いプラットフォームを対象としている。機種依存部は基本機能部のみで、他プラットフォームへの移植対応も容易。

## ■ オープンソース

オープンソースの一般的な特徴については割愛するが、公開ライセンスとしてBSDを選択しているため、商用/非商用を問わず同ライセンス下で自由に活動できる。また、試験公開時にもユーザーからの貢献があり、今後も完成度向上に向けて、優秀な技術者の協力が得られる点も魅力。