M-047

# エンドユーザが容易に動作規定可能なユビキタスサービス連携フレームワーク

# A Ubiquitous Service Coordination Framework whose Coordination Activity is Easily Definable for End-users

岩田　哲弥　　大石　哲矢　　武本　充治　　島本　憲夫†
Tetsuya Iwata　Tetsuya Oishi　Michiharu Takemoto　Norio Shimamoto

## 1. はじめに

近い将来，身の回りの様々なものにコンピュータが組み込まれ，ネットワークでつながれ，人間が無意識のうちにコンピュータやネットワークが生活をサポートするユビキタス社会が到来するといわれている．本稿では，ユビキタス社会におけるネットワークサービスの実現に不可欠な「サービス連携[1]」という手法のうち特にデバイス連携に着目し，それを実現するためには，UPnP[2], ECHONET[3]等の既存情報家電仕様では問題があることを示し，それらの既存仕様インタフェースを包み込む「ユーザ指向インタフェース」と，その間の変換機構により解決するフレームワークについて提案する．

## 2. サービス連携

ユビキタス社会でのネットワークサービスは，いつでもどこでも継続的に，常にその場の環境や状態に最も適した形態で提供され，各ユーザの嗜好性にあわせて適応化されたものになるであろう．このようなネットワークサービスを作る方法としては，時，場所，人により多様に変化する環境の全てを事前にサービス提供者が考慮して作る方法では不可能で，その場その時にユーザ側環境にある「サービス」（Web Service の他にデバイス，データベース，Grid リソース等も含む）を発見し，それらを連携させて作る「サービス連携」のアプローチが不可欠である．筆者らは，特にデバイスに着目した．その理由は，実世界とのインタラクションが大きく，最もユビキタス的なサービス連携が可能なこと，近年 UPnP, ECHONET, HAVi, OSGi[4], LonWorks, Bluetooth 等，次々と仕様が策定され，製品が次々と登場しているためである．

## 3. デバイス連携サービスの例

＜デバイス連携による自動化＞
- 目覚ましの時刻と連動し，テレビの電源が入る．テレビのチャンネルは好きな番組にセットされる．
- 一定以上の温度で自動的にエアコンの電源が入る．
- 家を映画モードにすると，大画面テレビと PC の接続等を行い，部屋の電気が消され，映画が始まる．

＜デバイス間の設定・データの引き継ぎ＞
- PC 内の好みの音楽を携帯電話の着信音に，また，近くの洗濯機の洗濯完了音に設定する．
- 近くのテレビの受信映像を近くのビデオで録画する．また，その音声を近くのヘッドフォンで聞く．
- 家のテレビで得たお店の URL や位置情報を，カーナビに引き継ぐ．

＜多種多様な環境に適応したデバイス連携＞
- いつでもどこでも映画を見たい．家なら大画面 TV と高性能スピーカーで，電車なら PDA で，飛行機ではシートディスプレイとヘッドフォンで
- いつでもどこでも英語の勉強をしたい．家なら TV 画面に出題され，答えを音声で選ぶ．電車なら，携帯電話に出題され，答えを携帯のボタンで選ぶ．
- いつでもどこでも涼しくしたい．エアコンがないが扇風機ならある部屋ならば，扇風機をまわす．

†日本電信電話株式会社 NTT ネットワークサービスシステム研究所
NTT Network Service Systems Laboratories, NTT Corporation

## 4. 要求条件

### 4.1 連携仕様記述の容易性

連携仕様の記述者としては，オフィスアプリケーションを使いこなせる程度の一般ユーザを想定する．このため，できる限り簡単に，ユーザの直感に沿い，ユーザが指定すべき事項が必要最小限ですむ形式で，連携仕様の記述やカスタマイズが行えることが必要である．その仕様記述形式は GUI ツールでの簡易な操作（関連デバイスを線で結ぶ，ドラッグアンドドロップ，コンテキストメニューからの選択等）で，生成できることが望ましい．

### 4.2 連携仕様記述の汎用性

連携仕様記述が，特定の場所，デバイス，ユーザだけではなく，そのまま，または，少しの修正で，多種多様な環境やユーザに適応できることが必要である．

### 4.3 連携仕様記述のパーソナライズ性

いつどんな環境でも個々のユーザの環境や嗜好性に，より適合するようカスタマイズできることが必要である．

### 4.4 携帯端末からのデバイス連携制御

いつどんな環境でもデバイス連携を可能にしたい．デバイス連携は，PDA,携帯電話等の上で動作するデバイス連携エンジンが，デバイス連携仕様を読み込み，それに従って行われると考える．するとデバイス連携エンジンは小リソースの携帯端末で動作できることが必要である．

## 5. 既存デバイス制御技術での課題

UPnP デバイスや ECHONET デバイスを UPnP や ECHONET の API で直接制御する場合，下記の問題点がある．（注：UPnP はメッセージインタフェースを規定するのみだが対応するベンダ製 API について同様である．）

### 5.1 デバイスタイプ別の API 規定

エアコン，扇風機などデバイスタイプ別に API が規定され，きめ細やかな制御が可能だが，環境の移り変わりとともに，その場のデバイスを柔軟に利用することができない．また，洗濯機の洗濯完了音のように，そのデバイスが持つ機能でも，そのデバイスタイプのメイン機能ではない場合，標準化されていないため利用できない．

### 5.2 連携仕様記述が困難

以下の多くの事項の把握・考慮・記述が必要で，連携仕様の記述は一般ユーザには極めて困難である．
- 利用可能デバイスの検索やユーザの選択・確認処理．
- 各フレームワークで決められた記述作法（ECHONET オブジェクト作成等）
- デバイスごとに定められた手順にのっとった制御．
- 制御前後の状態の確認．状態遷移．
- オペレーションの発行可能状態．所望の処理を行う前に必要となる前手順．
- オペレーション種別（イベント，メソッド，プロパティ，周期的なプロパティ取得等），その名称
- パラメータ・戻り値の名前，順序，データ型，単位，範囲，離散値／連続値，離散間隔，定数値名
- エラー・例外処理．例外名．エラーコード値．

・ コンテンツの場合，フォーマット，品質，ファイル転送プロトコル，ストリーミングプロトコル，サイズ制限，それらごとにインタフェースが異なる等．

## 6. 提案するフレームワーク

### 6.1 提案するフレームワークのコンセプト

デバイスタイプ 別ではなく，ユーザに対してのサービス内容にもとづく ユーザ指向インタフェー スを規定し，それで連携仕様 を記述する．きめ細やかな制御はできないが，ユーザ が必要最低限 の事項を指定するだけでよく，簡潔に記述できる．一方，デバイスは，そのネイティブインタフェース（UPnP, ECHONET 等の標準仕様 or 独自仕様）で，きめ細やかな制御 が可能だが，複雑な手順やインタフェースの考慮が必要である．

この 2 つのインタフェース をつなぐのが コンテクスト情報 で，具体的 には，その環境にあるデバイス の種別や状態，パラメータ 設定値のそれまでの 平均値，ユーザ嗜好性等である．

ユーザ 指向インタフェース で記述 された連携仕様 は，デバイス 連携を制御する端末上 で動作する「リゾルバ」が，その場その時の環境ごとに，デバイス のネイティブインタフェース による制御コードへ解決する．いわば，ユーザ 記述の実デバイス 環境へのダイナミック なコンパイラ・リンカである．この際，リゾルバ は，ネットワーク中の，ユーザ 指向インタフェース 定義やその解決に必要なライブラリ，コンテクスト 情報を参照する． 図 1 にそのイメージを示す．

小リソース の携帯端末 を考慮し，ユーザインタフェース定義情報 やその解決に必要なライブラリ は，利用シーンや利用デバイスごとにコンポーネント 化され，OSGi のバンドル として必要に応じてロード／アンロード される．このコンポーネント は，学習前の初期状態のものをデバイスメーカーが提供することを考えている．

このフレームワーク は，多数の制御仕様 が乱立し，詳細で厳密な取り扱いを必要とするデバイスを，OS のようにユーザフレンドリー に容易に統一的に取り扱うことを可能にし，かつ，環境が移り変わっても 適応しつつ，デバイス連携を行うことを可能にする．

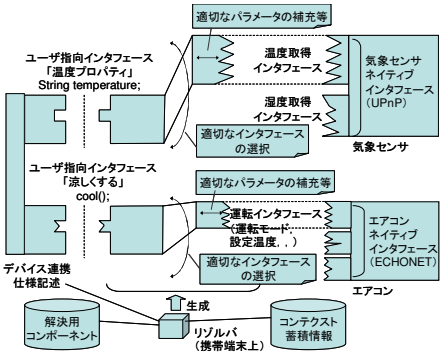


図 1 インタフェース解決イメージ

### 6.2 ユーザ指向インタフェース

3 節の例から考えると，ユーザの直感に沿い，記述の容易なインタフェース とするためには，オペレーション 型，ストリーム 型の双方が必要と考える．前者は，イベントーアクション 型連携の場合等に，後者は，デバイス からデバイスへ設定や情報を渡す場合等に適している．

＜オペレーション型＞

・ デバイスタイプ別ではなく，ユーザにとってのサービス内容で，簡潔なインタフェースを規定する．名称は，動詞 or 名詞 1 語＋修飾語 or 目的語 1 語程度までにする． （例： 冷房(); ，画像表示();）
・ パラメータはユーザにとって重要な意味を持つもの数個までにする． （例：設定温度，風力，風向）サービスに無関係なプログラミング特有概念（オブジェクト，データ型等）が隠蔽される．
・ パラメータはすべて String 型で，省略可能である．

＜ストリーム指向＞

・ プロパティとして，そのコンテンツ種別に着目し，Audio, Image, Video, Message, URL 等を規定する．

ユーザ指向インタフェースの例を図 2 に示す．

例1 その部屋の温度が28℃をこえていたら，涼しくする．
（ユーザ指向インタフェース Temperatureプロパティ,cool()メソッド）

```
import context.uson.org;
import air.context.uson.org;
Context context=new AirContext();
if (context.temperature>28){
   context.cool();
}
```

例2 携帯の着信メロディの設定を，洗濯機の完了メロディの設定に
（ユーザ指向インタフェース Audioプロパティ）

```
import context.uson.org;
import audio.context.uson.org;
Context context1=new AudioContext();
Context context2=new AudioContext();
MobilePhone mobilePhone=new MobilePhone(context1);
WashingMachine washingMachine=new WashingMachine(context2);
washingMachine.context.audio=mobilePhone.context.audio;
```

図 2 ユーザ指向インタフェースの例

### 6.3 ユーザ指向インタフェースの解決過程例

＜オペレーション指向＞

・ 連携仕様に記述されたユーザ指向インタフェースを提供可能なデバイスがデバイス能力記述文書によりその場で検索され，解決される．
・ 対応する最適なネイティブインタフェース （名前，種別）が，コンテクスト情報を参照し，選択される．
・ 事前に必要な状態確認や状態遷移処理が補完される．
・ 詳細なインタフェース （パラメータのデータ型，設定値等），例外処理コードが補完される．

＜ストリーム指向＞

・ デバイス能力記述文書とコンテクスト情報を参照し，代入側と被代入側間で整合可能で最適なネイティブインタフェースを選択し，名前，種別（プロパティ，Get/Set ハンドラ），データ形式（ファイル/ストリーム，フォーマット，URL/ファイル名，制御プロトコル）等が解決される．

### 6.4 提案するフレームワークの効果

本フレームワーク では，ユーザ指向インタフェース がユーザ へのサービス 内容に沿って規定される ため，デバイスタイプ にとらわれず に，その環境に適応化 しつつデバイス 連携を行える．また，デバイス のメイン機能 でなく，既存標準で規定 されない カスタマイズ も可能になる．また，ユーザ 指向インタフェース は個別のデバイス 仕様に依存せず，高度なプログラミングスキル が不要で，連携仕様を容易に記述でき，GUI ツールでの記述生成も可能となる．また，解決に必要な情報を利用シーン ごとにコンポーネント 化し，コンテクスト 情報はサーバで蓄積することで小リソース端末からの連携制御を可能にする．

## 7 今後の課題・まとめ

本稿では，ユビキタス 時代に求められる，ユーザ指向のデバイス 連携フレームワーク のコンセプト，基本アーキテクチャ を提案した．今後は，テスト実装を行い，リゾルバ 動作，コンテクスト 情報蓄積手法，適切なコンポーネント 粒度，携帯端末 実装手法，補完のバリエーションの発散防止 手法 等残された 多くの課題について，具現化・検討し，その性能や有効性等を評価していきたい．

## 8 . 参考文献

[1] M.Takemoto et al.,”The Ubiquitous Service-Oriented Network(USON)-An Approach for a Ubiquitous World based on P2P Technology”, IEEE P2P2002, Sep. 2002
[2] UPnP Forum, http://www.upnp.org
[3] ECHONET Consortium, http://www.echonet.gr.jp
[4] OSGi Alliance, http://www.osgi.org/
[5] 岩田,「環境適応型 サービス連携フレームワークにおけるストリーム指向アプローチの提案」, 2004 年情処全国大会