

Instaraction : 容易な機器導入を実現する 機器連携ミドルウェアの開発

井村 和博^{†1} 橋爪 克也^{†2} 伊藤 友隆^{†2}
河田 恭兵^{†2} 西條 晃平^{†1} 生天目 直哉^{†2}
伊藤 昌毅^{†2} 中澤 仁^{†1}
高汐 一紀^{†1,†2} 徳田 英幸^{†1,†2}

近年、複数の機器を相互接続するためのミドルウェア技術が発達し、多くの機器が場所や時間を問わず接続できるようになった。また、機器やサービスを実世界インタラクションが多数提案されている。これらの進化の一方で、ユーザが機器の提供するサービスを利用するためには依然として各機器にどのような設定が必要であり、どのようなサービスがその機器上で提供されているかをユーザが正しく理解している必要がある。また、実世界指向インタラクションは機器を導入する時点で識別のためのタグや専用の機材が必要になるなど、ユーザに対する負担が大きい。そこで本論文では機器を設置する際に必要な設定を最小限に抑え、かつマルチモーダルなインタラクションに対応する機器連携ミドルウェア Instaraction のプロトタイプを設計・実装し提案する。

Instaraction : Middle-ware for device collaboration enabling multi-modal interaction and easy installation

KAZUHIRO IMURA,^{†1} KATSUYA HASHIZUME,^{†2}
TOMOTAKA ITO,^{†2} KYOHEI KAWADA,^{†2} KOHEI SAIJO,^{†1}
NAOYA NAMATAME,^{†2} MASAKI ITO,^{†2} JIN NAKAZAWA,^{†1}
KAZUNORI TAKASHIO^{†1,†2} and HIDEYUKI TOKUDA^{†1,†2}

As middlewares for connecting devices have evolved, many devices are now able to connect each other any time and anywhere. At the same time, real world oriented interaction for handling devices and services are suggested. Regardless of these evolution, users still need to be conscious of service provided by device, and understand how to set up it. Furthermore, it is necessary to install tag

or exclusive device for device recognition when installing real world oriented interaction. This is exacerbated when user is trying to install device to home network. We present a middleware enabling easy device installation and multi modal interaction for device collaboration, called Instaraction.

1. はじめに

近年、機器を相互接続するために多くのミドルウェアが開発されている。Microsoft 社が提唱している UPnP^{?)} や米 Apple 社が開発した Bonjour^{?)} などのプロトコルを用いれば機器同士が IP ネットワーク上でコンフィグレスに接続することができる。ネットワークを介して機器同士が接続されるようになったため、今後はそれらの機器がどのように連携を行うかを定義する事が重要な課題となる。また、これまでに機器やサービス进行操作するためのインタラクションは多数提案されている。慶應義塾大学の岩本らが提案している u-Photo^{?)} は、ARToolkit^{?)} を使った 2 次元コードを機器に貼り、それをカメラで撮影する事により機器の指定を行う。また同河田らが提案する Snappy^{?)} は、SR コードと呼ばれるジェスチャコードを機器に貼り、コードに示されている矢印通りに携帯端末を振ることにより機器の指定を行う。このように、画面上で機器やサービスの操作を行うのではなく、実世界指向インタラクションを用いて操作を行う手法が提案されている。他にも名古屋大学の岩崎らが提案している Touch-and-Connect^{?)} や、ソニーコンピュータサイエンス研究所の綾塚らが提案している Gaze-Link^{?)} などが挙げられる。これらのインタラクションはどれも特定の環境下において有用であるが、インタラクションは時と場合によって使い分けができる必要がある。例えば自宅でテレビを操作する際にテレビ自体に付いているボタンを押すか、リモコンのボタンで操作するかといった場合、距離的な関係に基づいてユーザがどちらの操作方法が便利であるかを判断する。従って、機器操作はインタラクションが複数あり、それをユーザが選択することを前提に設計されることが望ましい。しかし、複数の機器と複数の実世界指向インタラクションが存在する場合、ユーザが機器を導入する際に今まで以上に多くの負担がかかることが想定される。これに対して、機器導入を容易にするシステムの開発が望まれている。そこで、本論文では容易な機器導入を実現する機器連携ミドルウェア

^{†1} 慶應義塾大学 環境情報学部

Keio University Faculty of Environment and Information Studies

^{†2} 慶應義塾大学 政策・メディア研究科

Keio University Graduate School of Media and Governance

Instaraction のプロトタイプを設計・実装し提案する。Instaraction は実世界指向インタラクションが複数使用されている環境で、機器導入に必要な操作をミドルウェアが自動的に行う事によりユーザの機器導入に対する負担を軽減する。また、複数のインタラクションを解決するモジュールを備え、今までに存在したインタラクションに加え、これから開発されるインタラクションにも対応する。

2. Instaraction：容易な機器導入を実現するミドルウェア

2.1 シナリオ

Instaraction は、様々な機器操作やサービス操作のためのインタラクションが存在する環境で、機器をホームネットワークに容易に追加できるようになることを目指している。以下で Instaraction が想定するシナリオを挙げる。

シナリオ 1：購入した機器の導入

龍太郎は新しくデジタルフォトフレームを購入した。早速自室に導入したいと思い、デジタルフォトフレームを自室の無線 LAN に接続した。するとプリンターからインタラクションに対応するコードが印刷されたので、龍太郎はそれをデジタルフォトフレームに貼った。龍太郎は普段から使用して慣れているインタラクションでデジタルフォトフレームを操作した。

シナリオ 2：インタラクションの追加

龍太郎は誰かが開発した新しいインタラクション手法を使おうと思った。龍太郎はインタラクションに必要なファイルをホームサーバにダウンロードしインストールを行った。その後、新しいインタラクションで自室にある機器の操作を行った。

2.2 問題意識

上述のシナリオを実現するためには、インタラクション層とミドルウェア層の連携が必要である。以下に連携を行う上での問題点を挙げる。

機器依存のサービス記述

現在普及している機器連携のためのミドルウェアとして UPnP と Bonjour が挙げられる。両者とも IP ネットワーク上で機器連携を行う事を想定しているが、互換性がないため UPnP に対応した製品と Bonjour に対応した製品が連携を行う事はできない。このように異なる規格を採用している製品が存在する場合にも、連携を取れるミドルウェアが必要となる。

インタラクションの乱立

現在普及しているミドルウェアは、機器同士が接続できることを保障するに留まっている。例えば UPnP ではアドレッシング、ディスクバリーにより機器同士がネットワークレイヤで接続できる事を保障する。またデスクリプション、コントロール、イベント通知、プレゼンテーションにより機器同士がアプリケーションレイヤで接続できる事を保障している。しかし、実際に機器を操作するインタラクションについては標準化がされていないため、結果として提案されているインタラクション手法が実際に用いられる事はない。また、インタラクションは時と場合により使いやすいものが異なり、さらに今後も新しいインタラクションが開発される可能性があるため、複数のインタラクションが存在することを前提に、それらに柔軟に対応できるミドルウェアが必要となる。

機器導入時のユーザの負担

複数の実世界指向インタラクションが存在する場合、ユーザは各インタラクションを用いるために、機器導入時に多くの負担を強いられる。例えば前述の u-Photo であれば AR タグが必要であり、Snappy では SR コードが必要である。その他に RFID などの接触により機器指定を行うインタラクションがあるとすれば、ユニークな ID を持つ RFID タグを作成する必要がある。現状ではこれらのインタラクションに必要なコード生成を行うための規格が存在しない場合、ユーザは一つ一つの機器ごとに手で ID を振ったりコードを生成する必要がある。機器を導入する際にユーザにとって大きな負担となっている。したがって、機器導入時にユーザの負担を軽減するミドルウェアの開発が望まれる。

2.3 機能要件

以上で述べたシナリオと問題点から Instaraction システムに必要な機能を以下に述べる。機器情報の柔軟な記述

互換性のないミドルウェアを用いている機器同士が連携を行う際に、機器に依存しない形式で柔軟に機器情報やサービス記述し、かつ通信を行ってサービスとリソースの共有を行う必要がある。これは同一のメーカーや機器間でしか協調動作しない規格ではなく、横断的に一つの規格に基づいて運用される事が求められる。

未知なインタラクションへの対応

インタラクションに関しては多くの提案があり、今後も新しいインタラクションが提案される可能性がある。その際、未知なインタラクションに対応できるように柔軟なインタラクション記述方式を検討する必要がある。

機器導入時に必要な操作数の削減

複数の実世界指向インタラクションを想定する際、各機器に対して何らかの準備が必要となる場合が多い。その際、その準備の段階を可能な限り減らす事により、ユーザの負担を軽減する必要がある。

3. 関連研究

機器の協調動作を支援するためのミドルウェアとして UPnP や、それをベースに作成された DLNA²⁾ などが挙げられる。また、Bluetooth 機器と UPnP 機器といった異なるネットワークインタフェースを持つ機器同士をシームレスに連携させる技術として uMiddle³⁾ が提案されている。以下では各提案について考察する。

DLNA

DLNA は UPnP をベースとして、情報家電やコンピュータを相互接続する為に開発された。DLNA を採用した機器同士は通信を行い、リソースやサービスの共有を行う事ができる。プリンターやテレビ、HDD レコーダーといった製品が採用しており、今後も対応製品は増えると考えられる。

uMiddle

uMiddle は慶應大学の中澤らが開発した、異なるミドルウェアを採用している機器同士が協調動作するためのミドルウェアである。uMiddle では、UPnP や Bluetooth といったミドルウェアにネイティブな機器が Mapper と呼ばれるコンポーネントを用いて機器の発見を行う。UPnP ネイティブの機器が Bluetooth ネイティブの機器と通信を行う場合には、Mapper が Translator と呼ばれるコンポーネントを生成し、プロトコルの変換を行う。これにより、複数の異なるミドルウェアが動作する環境においても通信を行う事が可能となる。

このように近年のネットワーク化された情報家電に対して多くの基礎技術が整備されつつあるが、機器の提供するサービスをホームネットワークに登録する部分や、機器導入にかかる負担をいかに減らすかといった点についてはまだ議論が行われていない。そのため、ホームネットワークに対してアプリケーションのレイヤで機器をどのように登録し、機器同士が連携するための準備を行うかについて議論を進める必要がある。

4. 設 計

4.1 想定するインタラクションと妥当性

今回の実装では機器指定の際に3つのインタラクション手法を想定している。

振る

機器に貼られたマルチインタラクションコードの中央に描画されている矢印と同じ方向に携帯端末を振る事により機器の指定を行う。図1の場合は携帯端末を右に振った後、下に振る動作を示している。携帯端末の代表例である携帯電話や携帯オーディオプレイヤーなどの多くは既に加速度センサを搭載しており、振る動作を検知する事が可能である。

撮る

機器に貼られたマルチインタラクションコードをカメラで撮影する事により機器の指定を行う。多くの携帯電話にはカメラが搭載されており、写真の撮影を行う事が可能である。

触れる

機器に貼られたマルチインタラクションコードを携帯端末で触れる事により機器の指定を行う。マルチインタラクションコードにはユニークな ID の付与された RFID タグが含まれており、RFID リーダを搭載した端末であれば、機器がコードに触れた事を検知する事が可能である。

以上のインタラクションはユーザビリティと実現可能性を考慮して選択した。これらのインタラクションは全て現在普及している携帯端末で実現可能なものであり、今後これらのインタラクションが普及するにあたりユーザの機器に対する追加投資が無いといったメリットがある。

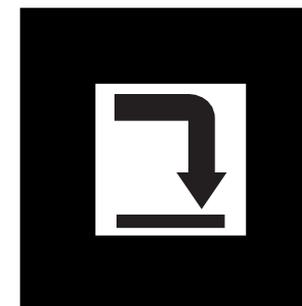


図1 マルチインタラクションコード

4.2 動作概要

Instaraction は図 2 のようにホームサーバと無線 LAN によって構築されたローカルネットワークが存在し、各機器の提供するサービスは携帯端末を用いて操作することを想定している。このような環境下で、Instaraction は以下のような挙動をする。

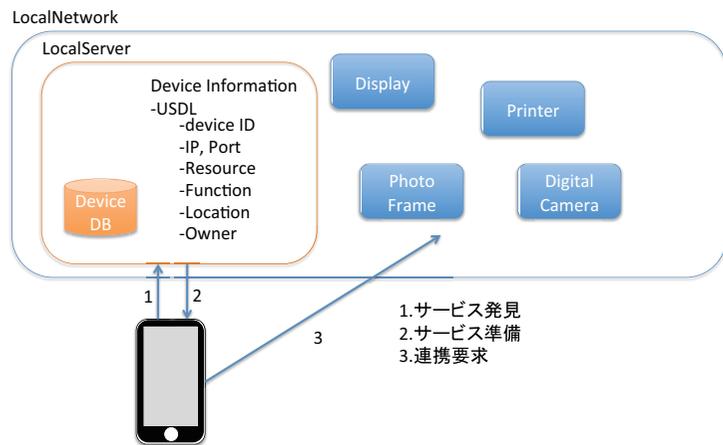


図 2 Instaraction における機器導入時の挙動

(1) サービス発見

携帯端末をローカルネットワークに接続した際、携帯端末上のクライアントアプリケーションはローカルサーバに対して接続を行い、環境内で提供されているサービスの発見を行う。

(2) サービス準備

ローカルサーバ内のデバイスデータベースに記述されている機器情報を携帯端末に返送する。これにより、携帯端末はどのようなサービスが提供されているかを把握する。

(3) 連携要求

サービスを使用する準備が整った上で、ユーザは携帯端末を経由して各サービスに対して連携要求を行う。

携帯端末はサービスをコントロールするこれにより、ユーザの機器導入時の負担は減り、任意の実世界指向インタラクションを選択し機器を指定・操作することが可能となる。

4.3 ソフトウェア構成

ソフトウェア構成を図 3 に示す。以下では機器登録モジュール、インタラクションコー

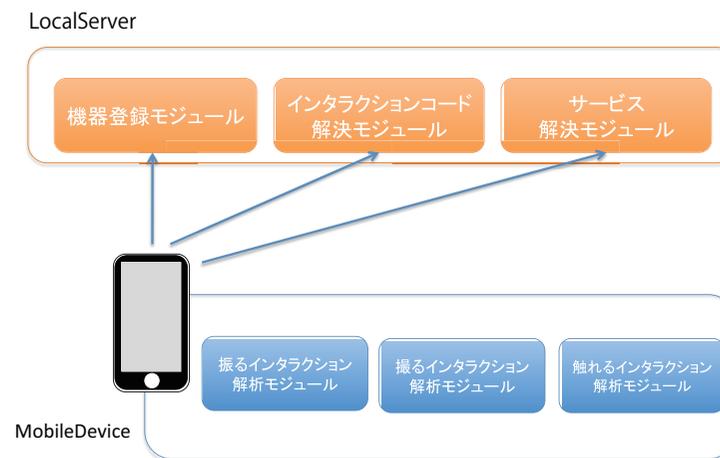


図 3 Instaraction システムのソフトウェア構成

ド解決モジュール、サービス解決モジュールについて詳しく述べる。

4.4 機器登録モジュールの設計

機器導入時にユーザの負担を軽減する為には、機器の登録作業とインタラクションに必要なコード生成の自動化を行う必要がある。図 4 は Instaraction の機器登録時における動作概要を示している。携帯端末を起動した際、端末はローカルサーバに機器情報を自動的に送信し、サーバが登録処理を行う。これに対してデジタルフォトフレームやミュージックプレイヤーが起動された場合、過去に接続された機器であるかどうかを確認を行った上で、もし初めて接続された機器であればサーバが登録処理を行うと同時にコードの生成と発行を行う。

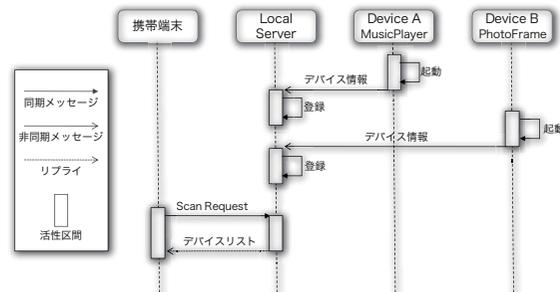


図 4 機器登録の動作概要

コードの生成は各実世界指向インタラクションに依存する．例えば振る動作の場合であれば left-up-right のような文字列の生成を行い，撮る動作の場合であれば AR タグを生成する．機器登録モジュールはこれらをまとめて生成し，ユーザが利用しやすい形式で出力する．

4.5 インタラクションコード解決モジュールの設計

既に存在する多くのインタラクションやこれから開発される未知なインタラクションに対応するためには，インタラクションを柔軟に定義できるスキーマが必要となる．今回想定した3つのインタラクションは，それぞれ異なる形式のデータを記録する必要がある．例えば，振るインタラクションに関しては”up-right-left-up”のように方向を表す4つの文字列を記述しなければならない．撮るインタラクションに関しては AR マークを含む写真データを記録しなければならない．触れるインタラクションに関しては RFID のタグ内に記述されている文字列を記述する必要がある．これらの要件を満たすために，インタラクションコード解決モジュールではインタラクションを XML ファイルに文字列で記述する．写真データに関してはそのままの形式では書き込めないため，文字列にエンコーディングし，必要なタイミングでデコードを行う．

4.6 サービス解決モジュールの設計

柔軟な連携サービスを提供するためには，異なるメーカーの製品を共通のプラットフォーム上で動作させる必要がある．しかし，メーカー各社が現在提案している機器連携用ミドルウェアはメーカー独自の仕様に基づいているため，異なるメーカー製品を連携させることができない．よって，本論文では機器非依存型のサービス記述手法を用いる．この手法として慶應大学の中澤らが研究を行っている USDL(Universal Description Language) を拡張し，機器記

述のための XML として利用する．サービス解決モジュールでは，USDL フォーマットで記述された XML ファイルの生成と解釈を行い，機器をオブジェクトとして捉えるオブジェクト指向による連携を行う．機器の指定には IP アドレスや MAC アドレスなどの情報をカプセル化した機器のオブジェクトを用いる．また，オブジェクト間の通信を Input/Output として捉え，ある機器（オブジェクト）のリソース（コンテンツなど）を Input とし，それらを処理するサービスを Output として，異なる機器のファンクションで動作するモデルを用いる．

5. 実装

本節では設計に基づきどのような実装を行ったか詳しく述べる．

5.1 機器登録モジュールの実装

機器登録モジュールでは，ネットワーク上で新しい機器が検出された際に，システムへの機器の登録を行う．同時に，インタラクションに必要なコードの生成を行う．以下では複数のインタラクションのためのコード生成を行う，マルチインタラクションコード生成モジュールの実装について述べる．

マルチインタラクションコード生成モジュール

マルチインタラクションコード生成モジュールでは図5の RFID プリンターを利用して，振る，撮る，触れるインタラクションに対応したコードを自動的に生成する．生



図 5 RFID プリンタ P-touch RL-700S

成モジュールは生成要求として xml ファイルを受信し，コードを生成する．含まれる

情報として文字列や画像、バイト列があり、画像やバイト列は通信のため Base64 でエンコードされる。以下はコード生成に用いる XML ファイルの例である。

```
1 <codeGenerationRequest >
2   <request type="print">
3     <description>this is optional</description>
4   </request>
5   <device>
6     <name>photoframe</name>
7   </device>
8   <interactions>
9     <element type="photo">
10      <code type="artoolkit">IDIiNSayN...</code>
11      <code type="image">iVBORwOKGgoAAAAAN...</code>
12    </element>
13    <element type="swing">
14      <code type="string">left-right-down</code>
15      <code type="image">iVBORwOKGgoAAAAAN...</code>
16    </element>
17    <element type="touch">
18      <code type="string">photoframe001</code>
19    </element>
20  </interactions>
21 </codeGenerationRequest >
```

5.2 インタラクションコード解決モジュールの実装

インタラクションコード解決モジュールは、各インタラクションの解析結果として出力された機器の ID を機器の詳細情報に変換する。サーバ側では過去に登録を行った機器の情報と固有 ID、そして各インタラクションに必要な ID を保持している。これにより複数のインタラクションが使用されている環境においても、どの機器が指定されたかを確認することができる。

5.3 サービス解決モジュールの実装

サービス解決モジュールでは、ローカルサーバ内で以下のような XML ファイルを保持し、それをもとにどの機器がどのようなサービスを行っているかを解決する。サービス解決を行う際、サービス同士の連携が行えるかどうかを判定するために input タグ内の記述と output タグ内の記述を比較する。

```
1 <device>
2   <property>
3     <id>photoframe</id>
4     <name>Ryutaro'sPhotoframe</name>
```

```
5   <icon>photoframe.gif</icon>
6   <description>Ryutaro'sPhotoframe</description>
7 </property>
8
9 <resources>
10  <resource id="current-picture">
11    <name>写真</name>
12    <output>image/jpeg</output>
13    <mode>packet</mode>
14  </resource>
15 </resources>
16
17 <functions>
18  <function id="display-picture">
19    <name>写真を表示する</name>
20    <input>image/jpeg</input>
21    <mode>packet</mode>
22  </function>
23 </functions>
24 </device>
```

5.4 その他のモジュール

RFID リーダ

今回の実装ではユーザが持つ携帯端末として Apple 社製の iPhone を使用した。その際、iPhone には日本メーカー製携帯電話のように RFID 技術が導入されていないため、iPhone 自体に RFID タグを貼り、環境側に図 6 のような RFID リーダを設置した。RFID リーダ



図 6 SunSpot と Skyetek M1-Mini による RFID リーダ

ダには Skyetek M1-Mini, またリーダに通信機能を持たせる為に Sun microsystems

が開発した SunSPOT⁷⁾ を用いた .

6. 議論と今後の展望

本論文では複数のインタラクションが使用されている環境で、機器の導入時におけるユーザの負担を減らすためのシステム Instaraction を設計・実装し、提案した . 複数の実世界インタラクションのためのコードを統合したマルチインタラクションコードは、ユーザがインタラクションのために新しくコードを作成もしくは設置する負担を減らしたと考えられる . また、インタラクション記述のためのスキーマを定義し、より汎用的に利用できるようなミドルウェアを構築した . これらは、今回の実装で想定した振る、撮る、触れるインタラクションに特化したものではなく、今後登場するインタラクションを受け入れられる可能性がある .

謝辞 本研究は、NICT 委託研究「ダイナミックネットワーク技術の研究開発」の研究成果の一部である .

参 考 文 献

- 1) <http://www.upnp.org/>. Universal plug and play forum.
- 2) <http://www.apple.com/support/bonjour/>. Bonjour.
- 3) Genta Suzuki, Shun Aoki, Takeshi Iwamoto, Daisuke Maruyama, Takuya Koda, Naohiko Kohtake, Kazunori Takashio, and Hideyuki Tokuda. u-photo: Interacting with pervasive services using digital still images. In *Pervasive*, pp.190-207, 2005.
- 4) 伊藤友隆, 河田恭平, 中川直樹, 生天目直哉, 橋爪克弥, 伊藤昌毅, 中澤仁, 高汐一紀, 徳田英幸: Snappy: 振る動作による異種ネットワーク間での機器連携の実現, 情報処理学会研究報告 Vol.2009, No.17, pp.31-36, 2009.
- 5) <http://www.hitl.washington.edu/artoolkit/>. ARToolkit.
- 6) Y.Iwasaki, N.Kawaguchi, and Y.Inagaki. Touch-and-Connect: A connection request framework for ad-hoc networks and the pervasive computing environment. In *First IEEE Annual Conference on Pervasive Computing and Communications(PerCom 2003)*, pp.20-29. Citeseer, 2003.
- 7) Y. Ayatsuka, N. Matsushita, and J. Rekimoto. Gaze-link: A new metaphor of real-world oriented user interface. *IPSI Journal*, Vol.42, No.6, pp.1330-1337, 2001.
- 8) Jin Nakazawa, W. Keith Edwards, Hideyuki Tokuda, and Umakishore Ramachandran. A bridging framework for universal interoperability in pervasive systems. In *Pervasive Systems, ICDCS 2006*. IEEE Computer Society, 2006.
- 9) <http://www.dlna.org/>. Digital living network alliance.
- 10) <http://jp.sun.com/products/software/sunspot/>. Sun SPOT.