

修士論文

周辺機器をサービス化して
柔軟な連携を可能とする技術の研究

Study of Flexible Cross-Device Interaction

平成 31 年 01 月 31 日提出

指導教員

入江 英嗣 准教授

東京大学大学院 情報理工学系研究科
電子情報学専攻

48-176410 喜多 司

概要

ユーザ周囲の様々な機器がインターネットに接続されるようになった。これら I/O 機器が増加するにつれて、ユーザがそれらを自在に相互接続させられる枠組みがより重要となる。ホームアシスタントなどの登場によって、プライベートな空間では様々な機器を簡単に連携・利用できるようになった。しかしパブリックな空間では、異なる所有者や異なるネットワーク上の機器間における柔軟かつ安全な連携は難しい。また、その空間においてどのようなサービスが存在するのかを適切にユーザに提示することも必要である。

本論文では、周辺機器をサービス化して柔軟な連携を可能とし、これらのサービスをユーザ機器との相対的な位置関係に基づいて適切に提示するシステムを提案する。本研究では、機器とサービスの位置、サービスの I/O、およびアクセス権を管理することで、Augmented Reality (AR) を用いたインタフェースとドラッグ・アンド・ドロップによるインタラクティブな機器連携をユーザに提供できるデモシステムを実装した。

目次

第 1 章	はじめに	1
第 2 章	背景と関連研究	5
2.1	ホームネットワークシステム（HNS）に見る相互接続性と相互運用性	5
2.1.1	サービス指向アーキテクチャ（SOA）	5
2.1.2	動的なサービス情報の取得	6
2.2	周辺機器の定義およびユーザとの位置関係	9
2.3	周辺機器との様々なインタラクション	10
2.3.1	物理的な接触を用いる手法	10
2.3.2	言葉や音を用いる手法	12
2.3.3	カメラや深度センサを用いる手法	14
第 3 章	周辺機器をサービス化して柔軟な連携を可能とする技術の提案	16
3.1	提案の概要	16
3.2	システム構成	21
3.2.1	機器	21
3.2.2	測位システム	21
3.2.3	クラウドデータベース	21
3.2.4	専用アプリケーション	22
3.2.5	ウェブサービス	23
第 4 章	実装	25
4.1	想定したシナリオ	25

4.2	概要	26
4.3	システム構成	27
4.3.1	スマートフォン, ノート PC × 2	27
4.3.2	UWB 測位システム	27
4.3.3	機器の位置とサービスを管理するクラウドデータベース	28
4.3.4	Android アプリケーション	28
4.3.5	ウェブサービス	30
第 5 章	評価と考察	32
5.1	動作	32
5.1.1	測位システム	32
5.1.2	ユーザインタフェース	32
5.1.3	タップおよびドラッグ・アンド・ドロップによる機器連携	33
5.2	理論	34
5.2.1	ユーザ観点	34
5.2.2	開発者観点	35
5.3	今後の課題	35
5.3.1	無線機器同士の連携	35
5.3.2	ユーザビリティの評価	35
5.3.3	機器をウェブサービス化することの代償	36
第 6 章	おわりに	37
	参考文献	39

1.1	IFTTT の概念図	2
2.1	サービス指向アーキテクチャ (SOA) の概念図 ([1] より転載) . . .	6
2.2	SOA を適用したホームネットワークシステム (HNS) の例 ([1] より 転載)	7
2.3	パーソナルスペース	10
2.4	Yang ら [2] による, ユーザの視線方向に応じたサウンドインタラク ション	11
2.5	WatchConnect [3] でのインタラクションの例	12
2.6	Tracko [4] による, データ転送をもたらすユーザインタラクション . .	14
3.1	提案システムの概念図	17
3.2	クラウドデータベース・ストレージを用いるウェブサービスの連携 . .	18
3.3	ユーザと周辺機器の位置関係に応じたインタフェースの提示	19
3.4	データベースに登録されるサービスプロトコル	20
3.5	位置・サービスデータベースの概要	22
3.6	空間ごとの, 機器の座標情報とサービス情報の管理	23
3.7	機器の持つサービス情報の内訳	24
4.1	実現したいインタラクションの例	26
4.2	UWB 測位システムで用いる移動機 (左) と親機 (右) [5]	28
4.3	ノート PC に機器情報が AR 表示される様子	29

4.4	画面共有サービスがデータベースに書き込むサービス情報（左）と サービスプロトコル（右）	31
4.5	投稿画像表示サービスがデータベースに書き込むサービス情報（左） とサービスプロトコル（右）	31
5.1	測位の誤差のヒストグラム	33
5.2	画面共有をドラッグ・アンド・ドロップで実現している様子	34

表目次

5.1	測位の頻度と誤差の最小値，最大値，平均値	33
-----	--------------------------------	----

第 1 章

はじめに

インターネットに接続できる機器は 2020 年には 300 億個に達するという統計が出ている [6]。周囲の機器が増加するにつれて、ユーザがそれらを自在に相互接続させることは機器の利便性を向上させるだけでなく、複数のユーザが共同で作業するような状況も作り出すことができる [7]。関連して、スマートホームという言葉が近年よく耳にする。日本国内では 2000 年代からホームネットワークシステム (HNS) と呼ばれる、宅内のサーバと複数の家電からなるシステムが開発されるようになった [8] [9]。家電の消費電力を削減することを目的としたホームエネルギーマネジメントシステム (HEMS) の開発が主流だった一方で、当時からメーカーの異なる家電同士の相互接続性を向上させる研究が為されてきた [10] [11]。現在は、異なるメーカーの家電を統一的に扱うため、HTTP を用いてネットワーク越しにインタフェースを呼び出すことができる WebAPI を各メーカーが規格に沿って製品に実装するのが通例となっている。WebAPI を連携させる代表的なハブサービスとしては、IFTTT [12], Google Assistant [13], Amazon Alexa [14] などがある。これらのサービスでは、あらかじめトリガとアクションを関連付けて設定しておくことで操作を自動化することができる。図 1.1 には IFTTT の仕組みを示した。たとえば、IFTTT のアプリケーションをインストールしたスマートフォンが自宅の Wi-Fi に接続されることをトリガに、自宅のスマートランプの点灯をアクションに設定することで、ユーザが帰宅すると同時に部屋の明かりを点灯させるというようなインタラクションを実現できる。Google Assistant, Amazon Alexa ではマイク付きのホームアシスタントを用意することを基

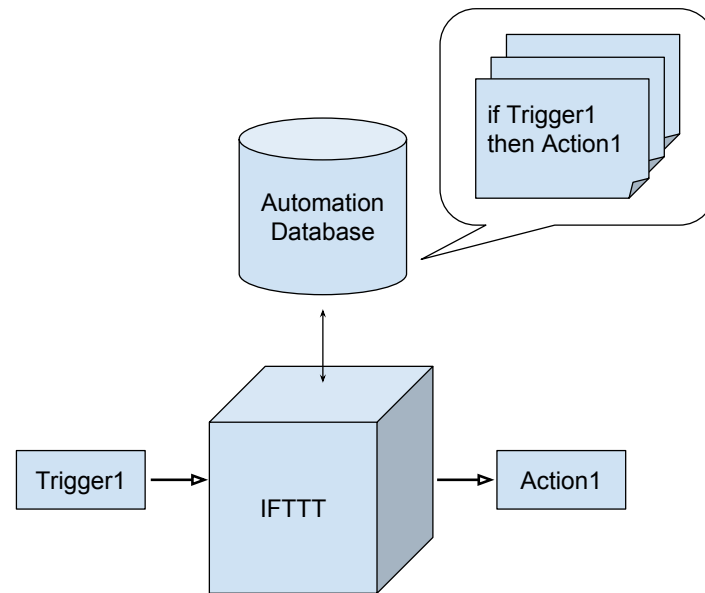


図 1.1 IFTTT の概念図

本としており，ユーザの音声に該当キーワードが含まれることをトリガに設定することができる．このように複数の機器が特定の規格に沿ったインタフェースを持ち通信できることを指して相互接続性と言う．

しかし，ユーザ周囲の機器の相互接続性が向上しているのは自宅やオフィスといったプライベートな空間に限られている．21 世紀のコンピュータの在り方として Weiser [15] が提唱したユビキタスは，単にコンピュータが遍在するという意味ではない．人間が生活する上で，意識することなく当たり前のようにいついかなる場所でもコンピューティング能力を用いることができることをユビキタスコンピューティングと言う．したがって，プライベートな空間だけでなく，パブリックな空間においても，ユーザが自在に周囲のコンピューティング能力を用いることができる環境を整えていく必要がある．しかしここには，安全性とユーザインタフェースに関する課題がある．プライベートな空間における機器連携手法ではあらかじめ決められている機器同士のみが通信できる場合がそのほとんどである．パブリックな空間で機器連携を行うことを考えると，たとえば任意のユーザの持つモバイル機器（ユーザ機器）と他者のユーザ機器やその場にある共用の機器とを相互に接続することが想定される．NFC, BLE,

Wi-Fi といった安全な無線技術が存在するため、現状、これらを用いた接続が多い。たとえば次のような場合である。

1. 交通機関を利用する際、改札を通るときには NFC を利用できる。
2. レンタルオフィスで、スピーカーを利用するときには BLE を利用できる。
3. カラオケでは、カラオケ機器が持つ専用 Wi-Fi に接続することでスマートフォン上のアプリケーションから選曲することができる。

しかしこれらの無線接続は、通信路を形成する際にユーザの行動が制限される他、パスワードの入力を要求されるなど接続の仕方が煩雑な場合がある。ユーザが、どの機器がどんな通信手段を持っているかを意識せずに、周囲の機器を扱えるようなインタフェースが求められる。また、この三つの例は、ユーザ機器とその他の単一の機器とのインタラクションを実現している。同じ無線技術を利用できる機器であれば、その通信路を延長してより複数の機器同士を接続することもできる。しかし、ある二つの機器が共通した無線技術を持ち合わせていなかった場合、それらの間でどうデータのやり取りをするべきかは自明ではない。こういった heterogeneous な機器同士の接続は、家電同士の連携に見ても、インターネット上の HTTP を用いて行うことが一般的と言える。つまり、パブリックな空間においてその場にいる誰もが利用できる機器の機能をインターネット上に公開する、ということになる。パブリックな空間では、どの機器同士が通信しうるかをあらかじめ設定しておくことができないため、その場にいるユーザにのみ周囲の機器にアクセスする権限を持たせるためには、ユーザの位置による認証が有効である。近年、Global Positioning System (GPS) から得られる位置をもとに様々なサービスを提供するスマートフォン用アプリケーションが流行している。こういったアプリケーションでは、その場所に行かないと手に入れられないコンテンツを用意することでユーザをその場所に誘導する。しかし GPS は、衛星が発する信号を受信した機器がその信号に載った情報を元に座標を計算するその性質上、位置を偽ることができる。そこでユーザの位置を偽れないようにする技術が発展してきた。

機器連携におけるユーザビリティのことを、本論文では相互運用性と呼ぶ。ユーザビリティとは、操作のしやすさ、分かりやすさ、便利さ、使い心地の良さの総称である [16]。読んで字のごとく、操作のしやすさとは、ユーザが煩雑な操作なく機器連携を

行えることを指す。プライベートな空間においては、ユーザが連携可能な機器について熟知していることが想定されている。対してパブリックな空間においては、初めてその空間に訪れるがために提供されているサービスに関して事前知識を持たないユーザも考慮し、実現できる機器連携をいかにユーザに提示するかが重要な役割を果たす。これは機器同士が接続可能かどうかとは直交して存在する問題であり、分かりやすさに関連する。便利さとしては次のような例をあげる。仮にスマートランプに、入力として受け取った音声の音程に合わせて明かりの色を変えるような仕組みを持たせたいとする。このときホームアシスタントが音声データそのものをスマートランプに伝達できれば、相互接続された機器がハードウェア的に存在しえなかった I/O を扱うことができたという意味で、サービスの幅が広がっており、便利になったと言える。実際には、スマートランプの開発者が製品にマイクを取り付けることで同様の機能を実現できるが、需要が個人によって異なること、設計時に全ての利用法を想定することは不可能であること、センサを持ちネットワークに接続される機器の設計にはプライバシー上の慎重さが必要とされることから好ましくない。この例で言えば、マイクというセンサは一つの空間において一つあれば十分であるということも示唆される。こういったユーザへの配慮は、使い心地の良さに繋がっていく。

本論文ではパブリックな空間における相互接続性と相互運用性を向上させるため、周辺機器をサービス化して柔軟な連携を可能とし、これらのサービスをユーザ機器との相対的な位置関係に基づいて適切に運用・提示するシステムを提案する。本研究では、機器とサービスの位置、サービスの I/O、およびアクセス権を管理することで、Augmented Reality (AR) を用いたインタフェースとドラッグ・アンド・ドロップによるインタラクティブな機器連携をユーザに提供できるデモシステムを開発した。

第 2 章

背景と関連研究

2.1 ホームネットワークシステム（HNS）に見る相互接続性と相互運用性

2.1.1 サービス指向アーキテクチャ（SOA）

サービス指向アーキテクチャ（SOA）[17] とは 2000 年代に入ってから注目を集めるようになったネットワーク上に分散する複数の異なるドメインの制御下にあるシステムを柔軟に編成するための手法であり，特に HNS の開発において頻繁に提唱された [18].

HNS を活用するうえで問題となるのは，各メーカーが独自のプロトコルで制御可能な家電を生産していたことだった．SOA では図 2.1 のように，各システムの機能をサービスという単位でくくりだし，これらの組み合わせでより高度なサービスを組み上げていく [1] [19].

中村ら [1] は，各家電をデバイス層およびサービス層の 2 層で構成している．デバイス層は，ネットワーク家電のハードウェアおよび制御 API を指す．一方サービス層は，これら制御 API を論理的な機能単位でまとめてローカルネットワークに公開する．サービス層は，各家電のアプリケーションとして実装され，家電のストレージに格納される．ローカルネットワークにおけるサービス公開にあたっては，W3C [20] でまとめられている標準的な仕様にしたがって URI を割り振っている．このような HNS におけるサービス指向フレームワークを用いることで，図 2.2 に示すように様々な家電

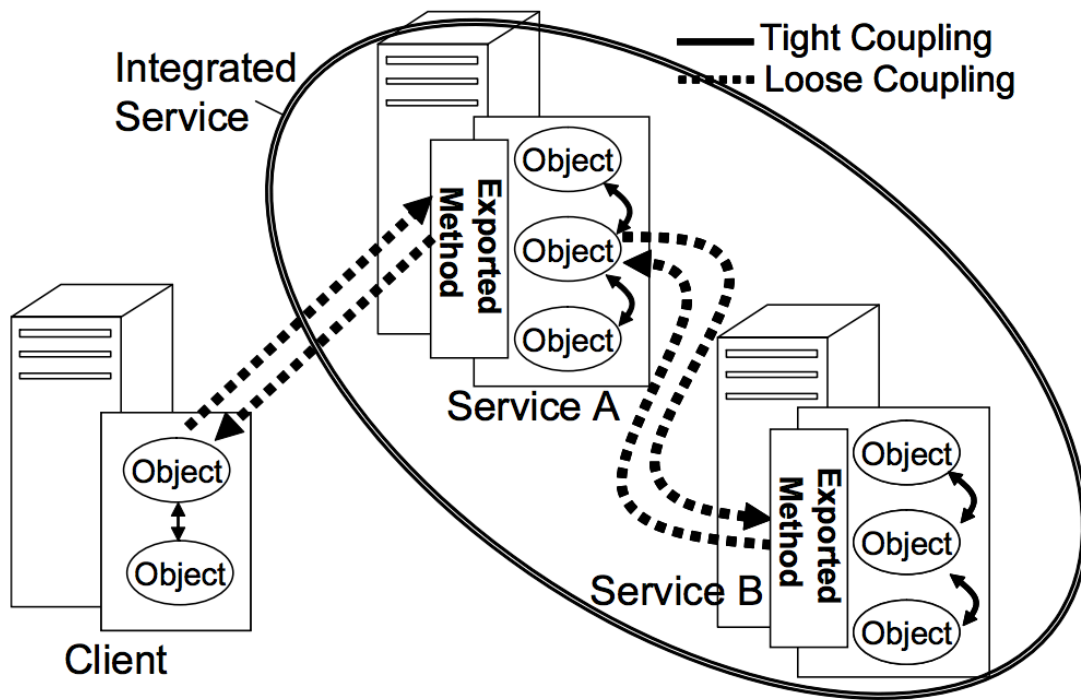


図 2.1 サービス指向アーキテクチャ（SOA）の概念図（[1]より転載）

の相互接続性が向上し、多様な連動を設定することができる。

日本国内では 2012 年に ECHONET Lite [8] が標準規格として扱われるようになり、その他の規格に沿って生産された家電も含めたあらゆる家電を ECHONET Lite をベースにした共通規格へと落とし込むような地道な努力が続いている [21] [11]。田中ら [10] は、制御 API の存在しない従来の家電も、赤外線を利用する遠隔操作機能などがあれば、マルチリモコンなどを利用することで HNS サービス指向フレームワークに当てはめることができることを示した。以上の考え方は IFTTT, Google Assistant, Amazon Alexa などのハブサービスにも反映されており、SOA が様々な機能を連携して自動化することに適している手法だと言える。

2.1.2 動的なサービス情報の取得

HNS にはサービスプロバイダ型と DIY（Do It Yourself）型の二つのモデルがある [22]。サービスプロバイダ型は、ユーザではなく業者がスマート家電を設置・運用する。これに対して、DIY 型ではユーザ自身が利用の他に設置・運用も担うため、サー

2.1. ホームネットワークシステム（HNS）に見る相互接続性と相互運用性

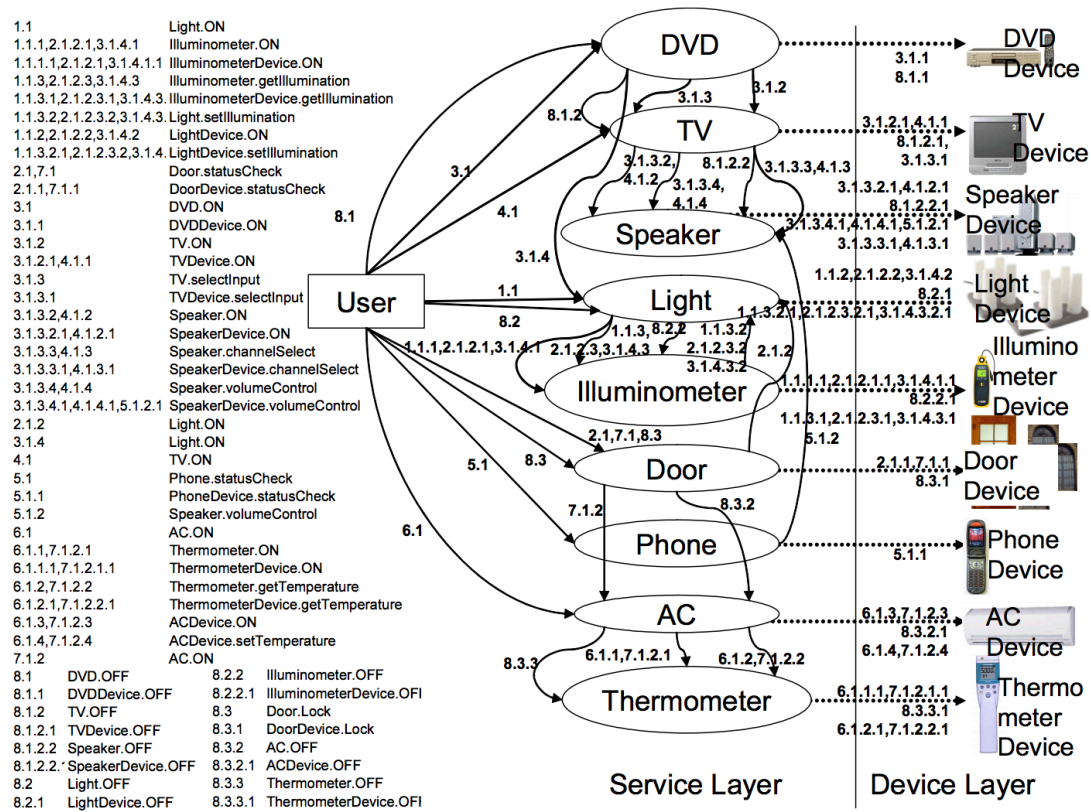


図 2.2 SOA を適用したホームネットワークシステム（HNS）の例（[1] より転載）

ビスプロバイダ型の HNS よりも利便性を追求できる．しかし両者とも静的な機器連携にとどまる．すなわち，相互接続される機器同士はどちらもユーザの所有物であり，システムに他人の機器が参入することは想定されていない．また，他人の機器が参入してこない以上，当然ながらこれらのシステムを利用するシチュエーションも自宅やオフィスなどといったプライベートな空間に固定される．パブリックな空間において，異なる所有者の異なる機器間でいずれかの機器が展開するサービスをユーザに提供するためには，ユーザに対して動的にサービス情報を提供することが要される．

Android アプリケーション（以降，アプリ）の開発において，インストール不要の Google Play Instant というコンセプトが提案されている [23]．アプリ全体をインストールすることなく，そのアプリの持つ単一の機能をユーザが即座に利用するためにダウンロードするのが Google Play Instant である．ユーザが Google Play Instant の機能をリクエストすると，その特定の機能を実行するために必要なコードのみがダウンロードされる．また，ユーザがその機能を実行し終わると，システムにより機能のコー

ドが破棄される．このように単一の機能をユーザの必要に応じて提供するため，アプリケーションは小さなモジュールに分割され，機能モジュールとしてパッケージ化されている．Google Play Instant は，多種多様なアプリが存在する昨今，わざわざインストールすることなく即座にそのサービスを試してみたいというユーザの要求を満たすことが第一に考えられている．

このように，必要に応じて提供されるサービスに関連する機能をダウンロードしてくればよいという考え方を持っている研究として，二村らによるユーザ機器と BLE, Wi-Fi など無線で接続される周辺機器を簡単につなげるウェブドライバ技術 [24] がある．二村らは，HTML や JavaScript で書かれた機器制御モジュールを必要なタイミングでユーザ機器のウェブランタイム上で動的に挿抜でき，ウェブアプリと動的に紐付けて実行することが可能なウェブ型デバイスドライバアーキテクチャを提案している．動作フローを次に示す．

1. 検出：ユーザ機器が周辺機器に近づくと，ユーザ機器がその周辺機器を検出する．
2. 機器情報取得：ユーザ機器は，周辺機器に関する情報を取得するため，周辺機器に接続して，それがどのような機器で，どのような機能を提供しているのか特定する．
3. ウェブ型デバイスドライバ配備：取得した機器情報をもとに，対応するウェブ型デバイスドライバをダウンロードし，これを事前に用意しておいたウェブアプリと結合する．
4. 機器利用：ウェブアプリからウェブ型デバイスドライバにアクセスすることで，周辺機器を利用する．

ウェブ型デバイスドライバを挿抜できるウェブアプリはウェブランタイム上で動く．ウェブランタイムは，OS の違いおよび OS バージョンごとの違いを吸収することができるため，対応ブラウザで HTML が開ける任意の機器がユーザ機器として利用できる．当然だが，ユーザ機器には BLE や Wi-Fi モジュールが備わっていなければ対応する周辺機器を扱うことができないことには注意されたい．ウェブ型デバイスドライバは，以下から構成され，周辺機器 1 つにつき，1 セット開発を行う．

1. ドライバ情報ファイル
2. 機器制御 API を提供する JavaScript プログラム
3. アプリケーション上に表示される機器のアイコン

これらはパッケージ化され、ウェブサーバに置かれる。したがってユーザ機器で周辺機器を利用する際、必要となるのは事前に用意したウェブアプリのみとなり、OS や周辺機器ごとの専用アプリを不要とし、動的に周辺機器を利用できる。

2.2 周辺機器の定義およびユーザとの位置関係

そもそもユーザの周辺機器とは、どのような範囲内の機器のことを言うのか定義したい。これに対して近接学、すなわちユーザのパーソナルスペースを考慮して人と機器との関係を明らかにしようとしている Proxemic Interaction の研究 [25] [26] がある。パーソナルスペースとは、文化人類学者の Hall が唱えた、人を囲む空間のことである [27]。図 2.3 のように、人を中心に内側から「親密な距離」、「個人的な距離」、「社会的な距離」、「公共の距離」とあり、それぞれの内側が対応する空間となる。

人が誰かと接触する際、自身を中心としたいずれの距離内にあるかで振る舞いが異なると言われている。人の振る舞いが人相手だけでなく機器やサービスに対しても変化すると仮定すれば、パーソナルスペース内に存在する機器をユーザの周辺機器と定義づけることに意味がある。すなわち、ユーザが周辺機器を利活用する際、この 4 つに分かれた空間内のどこに機器があるかを判別することができれば、相互運用性を向上させることができる。たとえば、パブリックな空間に共用の機器があったとして、パーソナルスペースのうち「親密な空間」に存在する共用の機器は他のユーザからは利用できないようにする、といった管理が可能となる。Yang ら [2] は、図 2.4 のように、ユーザの向いている方向からの音のみをユーザに聞こえるようにした VR アプリケーションを実装している。

以降、本論文ではユーザのパーソナルスペース内に存在する機器をユーザの周辺機器と呼ぶ。

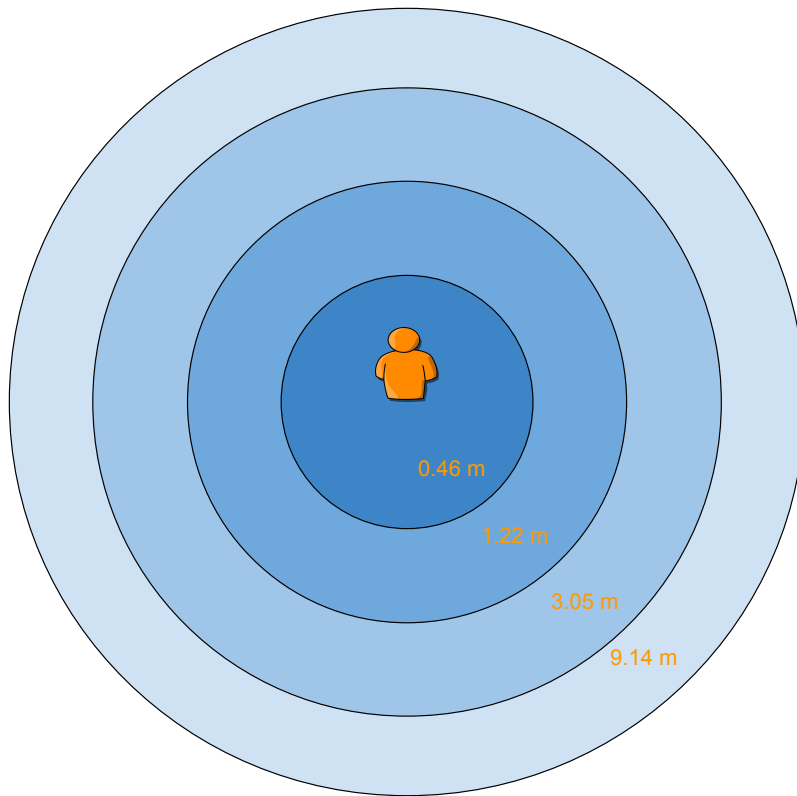


図 2.3 パーソナルスペース

2.3 周辺機器との様々なインタラクション

2.3.1 物理的な接触を用いる手法

2.3 節では、ユーザと周辺機器のインタラクションについて位置関係に着目しながら説明する。

WatchConnect [3] はたとえば図 2.5 のように、ユーザがタッチセンサ付きのディスプレイ機器に触ることでタッチした画面先に対応した情報を手元のスマートウォッチに表示する、といったインタラクションを実現している。実物の機器にユーザが直接触れることになるので、ユーザが機器を選択する操作は直観的である。スマートウォッチをヒューマンインタフェースデバイスとして利用するといった機能も提案されている。スマートウォッチ上に表示される筆アイコンをタッチした状態でディスプレイ機器の画面に触れれば絵や文字を書き込むことができ、消しゴムアイコンをタッ

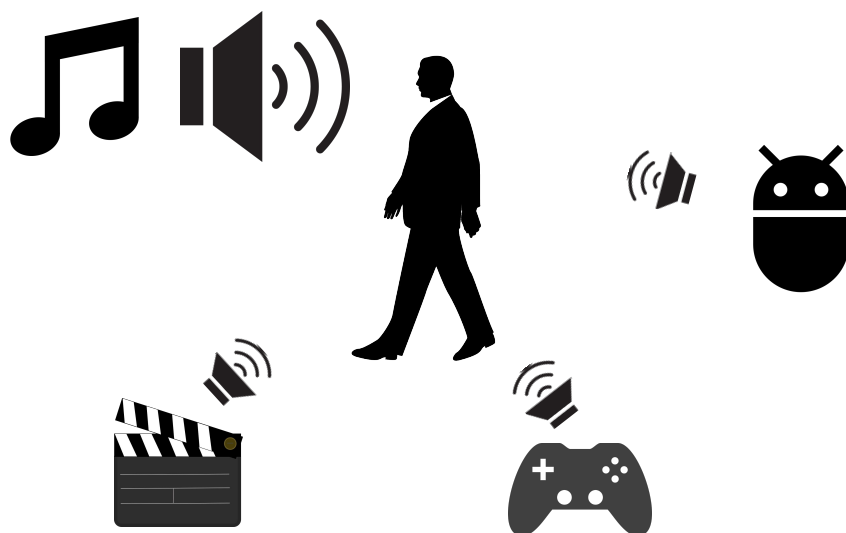


図 2.4 Yang ら [2] による，ユーザの視線方向に応じたサウンドインタラクション

チした状態ではその書き込みを消去できるといったインタラクションを示している．ディスプレイ機器への多様な入力を実現するため，スマートウォッチには測距センサや加速度センサ，磁気センサなどが備わっており，スマートウォッチを取り付けた腕や手の姿勢も検知できるようになっている．たとえば，スマートウォッチを取り付けた左手でディスプレイ機器の画面をタッチしている状態のまま，スマートウォッチの上部をスワイプするように右手を動かすことで，スマートウォッチ上の画面をディスプレイ機器上に画面共有させるといった例も示されている．しかし，このようにインタラクションの幅を広げるために多様な動作を検知できるようにしても，どの動作でどの機能が使えるのかをユーザに対して明示することは難しく，分かりやすさが乏しい．複雑な動きを検知できるようにしても，操作のしやすさが損なわれる．

スマートウォッチと複数のディスプレイ機器とのインタラクションは随時ユーザが操作対象の機器に接近することで実現されるが，ディスプレイ機器同士のインタラクションは想定されていない．実際の実装では，ディスプレイ機器への入力を検出した際に，スマートウォッチが検出した腕や手の姿勢およびタッチしている画面先に応じたフィードバックを有線の通信路でスマートウォッチに出力している．つまり，複数のユーザがいるような状況は想定されていない．しかし，スマートウォッチとディス

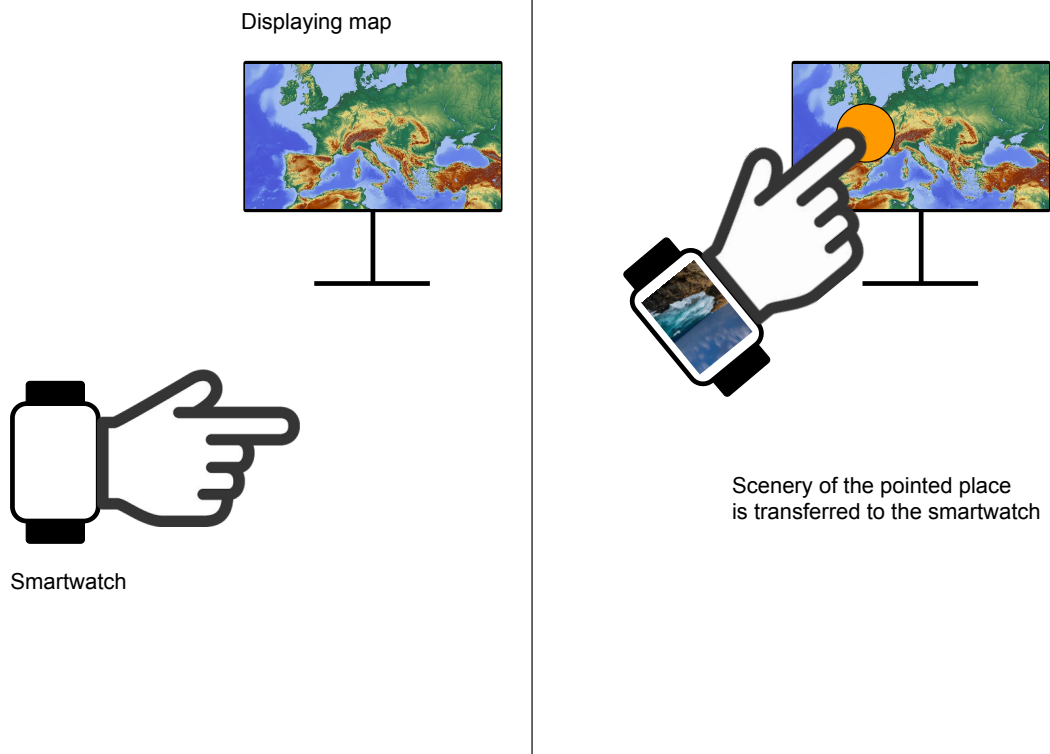


図 2.5 WatchConnect [3] でのインタラクションの例

プレイ機器との間の通信路がたとえば Near Field Communication（NFC）などによる無線で行えれば，このスマートウォッチを使う複数のユーザがいる場合にも，ディスプレイ機器をタッチしているのがどのユーザなのかをシステムが判断できる。

次に，システムが把握できるユーザや機器の位置関係について考える．このような接触によるインタラクションにおいては，システムが把握できるユーザと周辺機器の位置関係はユーザを中心とする「親密な空間」程度の近距離に限られ，その把握は接触が生じる散逸的なタイミングに限られ連続的には成し得ない。

2.3.2 言葉や音を用いる手法

第 1 章でも述べたように，Google Assistant や Amazon Alexa といったハブサービスでは，マイクを搭載した機器を用意し，ユーザは音声によってその他の機器を操作するといったインタラクションが可能である．しかし，「スマートランプを明るくして」などとユーザが言葉で指令を出すとき，複数の似通った機器がある場合，ユーザがど

のスマートランプのことを指しているのかをシステムが理解することは難しい。「ベッドの近くのスマートランプを明るくして」などと事細かくユーザに説明させることでシステムがユーザの意図する機器を正しく汲み取ることはできるが、ユーザへの負担も増えかねない。またこの場合、あらかじめスマートランプがベッドの近くにあることをシステムに登録しておく必要もある。

SONDI [28] では、ユーザはスマートフォンなどのユーザ機器のマイクをインタフェースに周辺機器を操作する。各周辺機器が指向性のスピーカを搭載しており、それぞれが固有の信号の乗った超音波を発している。ユーザ機器のマイクにそれらの信号を受信させ、その強度からユーザがどの機器の近くにいるかを判定する。これにより、ユーザはハンズフリーに音声のみで操作できる。しかし、信号の強度が最も高い機器を操作対象とするため、ユーザは操作したい機器の傍まで寄る必要がある。

Tracko [4] は、SONDI と同じく超音波を使うが、超音波が減衰し切らない程度の距離であれば、スマートフォン同士の姿勢を含む相対的な位置関係が分かる手法である。スマートフォン同士の距離が 1 m 程度であれば測位の誤差は 15 cm 程度と高精度な測位手法である。姿勢に関しては、一般的なスマートフォンには加速度センサやジャイロセンサといった Inertial Measurement Unit (IMU) が備わっているため、簡単に取得できる。二つの機器間で超音波を送受信し合う際の Time of Arrival (ToA) を測定することにより、音速を一定値と仮定すれば機器間の距離が測定できる。スピーカが二か所あるスマートフォンを用意し互いに超音波を送受信し合えば、合計 4 つの直線の測距ができる。スマートフォンの形は一定なので、スマートフォン上のスピーカやマイクとの間の相対的な位置関係は拘束されている。そのため、変数に対する拘束条件の数を考えれば、数学的にスマートフォン同士の相対的な位置関係が求まる。

Tracko はこうして求まったスマートフォン同士の姿勢や位置関係が分かることから、図 2.6 のように画面上の物体を他のスマートフォンにフリックするといった直観的な動作による通信を実現している。

以上の例から、ユーザの声や超音波を含む機器が出す音を用いてシステムがユーザと機器の位置関係を把握しようとする場合、大雑把にユーザが機器の付近にいる、すなわちマイクに音が届く程度の範囲にいることは簡単に知ることができる。そして、ユーザにとって「個人的な空間」程度の範囲内にある機器となれば、詳細な位置関係

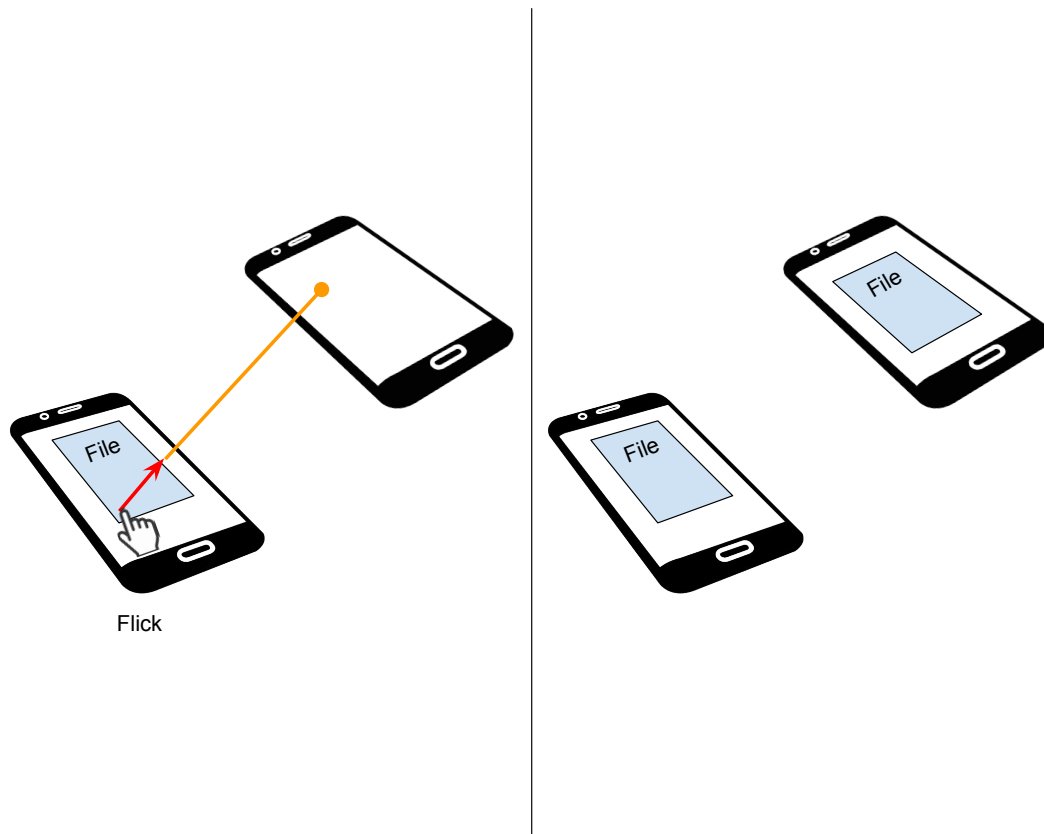


図 2.6 Tracko [4] による，データ転送をもたらすユーザインタラクション

を取得できる．また，これらの位置は連続的に取得できる．

2.3.3 カメラや深度センサを用いる手法

機器にマーカをつけ，ユーザ機器のカメラを用いてそのマーカをとらえることで，ユーザの意図している機器を特定する手法 [29] [30] [31] がある．しかし，マーカを機器に取り付けることはデザイン性を損なうことに加え，光環境の影響を受けやすいという欠点がある．マーカは使わず，アイトラッキング技術を用いてユーザの視線をとらえることで機器を特定する手法 [32] もあるが，アイトラッキングのためにユーザは専用のヘッドマウント機器を装着しなければならない．

Kong ら [33] は家電の見た目を機械学習しクラスタリングすることでスマートグラスをかけたユーザの見ている機器を判定している．クラスタリングするだけでは似通った見た目の機器は区別できないが，対象を家電に限ることで，同じ部屋に同じ見た目をしている家電は存在しないと仮定している．そして，ユーザが宅内のどの部屋

にいるかは、スマートフォンのセンサ情報や受信した Wi-Fi の強度のフィンガープリンティングによって特定している。フィンガープリンティングは、人間の指紋が人によって異なるように、様々なセンサ情報を場所ごとに収集していけばどの場所として全く同じパラメータにはならないだろうという仮定に基づく。センサには個体差があるものが多く、全く環境でも異なった値を持つことがある。そのため、フィンガープリンティング用のデータベースを機器ごとに作らないと正しく判定されないことも多い。

カメラが取得する情報のみを使い、システムがユーザと機器との位置関係を把握する場合、視界に機器が入っている程度には近くにいることを保証できる。最新のスマートフォンではカメラに深度センサが付いている場合があり、こういった機器は周囲の 3 次元データを取得することができる。周囲の 3 次元データを取得して保管することは、地図を作製することと同義で、地図を参照することで機器の位置を「公共の空間」程度の範囲内であれば正確にシステムが把握できる。

第 3 章

周辺機器をサービス化して柔軟な連携を可能とする技術の提案

3.1 提案の概要

本論文の提案は次の 2 つに大別される。

1. 機器をウェブサービス化して展開することによる相互接続性の向上.
2. 周辺機器の持つサービスをユーザに測位に基づき提示することによる相互運用性の向上.

図 3.1 に提案システムの概念図を示す.

機器をウェブサービス化して展開することによる相互接続性の向上

パブリックな空間において、異なる所有者の heterogeneous な機器間でいずれかの機器が展開するサービスをその場にいる任意のユーザに提供し利用させることができる環境を整えるためには、ユーザ周辺で提供されるサービス情報を動的にユーザ機器が取得することが求められる。二村ら [24] の研究では、ユーザ機器がいかに周辺機器のサービスを得るかに主眼を置いており、ユーザ機器以外の複数の周辺機器の持つサービスを協調的に扱うことは考慮されていない。

機器の連携にはデータのやり取りが不可欠である。そこで我々は、異なるサービス間でもうまくデータのやり取りができることを目的とした、クラウドデータベース・

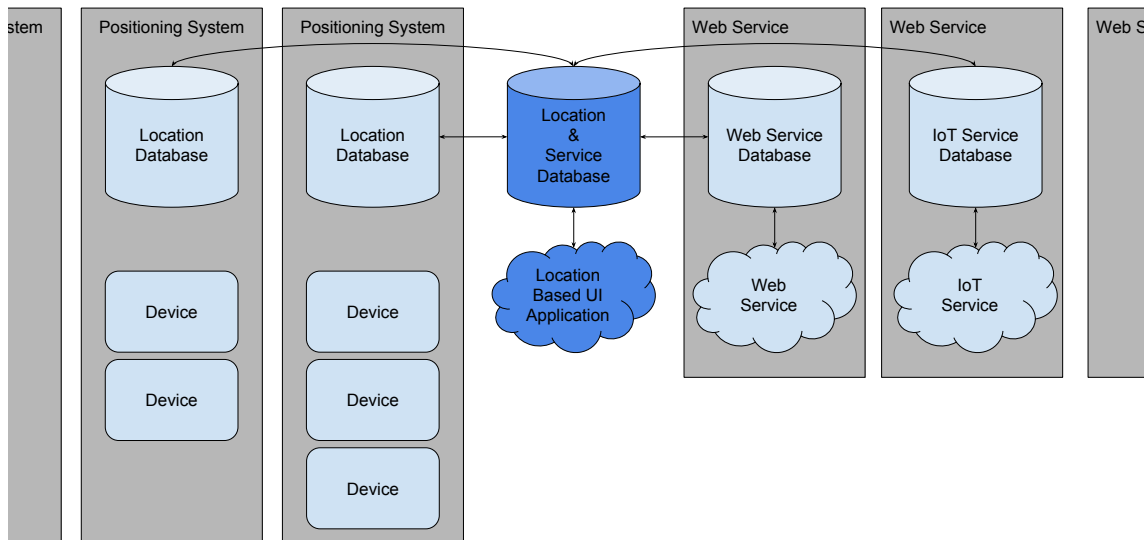


図 3.1 提案システムの概念図

ストレージを用いる I/O に重点をおいたウェブサービスの組み方を提案する。OS などになるべく依存しないクロスプラットフォームなサービスを提供するためには HTTP を利用することが一般的となっているため、各機器はウェブブラウザ（以下、単にブラウザ）からウェブサービスに接続することでサービス化する。たとえば、次のような二つの異なるサービスがあったとする。

1. サービス A：カメラで取得しているリアルタイムの画像を出力するサービス
2. サービス B：入力された画像の雰囲気に合わせた音楽を生成するサービス

このとき、サービス A の出力をサービス B に入力できれば、カメラで取得しているリアルタイムの画像からその雰囲気に合わせた音楽を流すことができる。この例では、やり取りされるデータは画像であり、サービス A がカメラから取得した画像をクラウドストレージにアップロードした上で、そのアドレスをサービス B に伝達することでスムーズな連携が実現する。つまり、サービス化した機器はクラウドデータベース上の特定の領域にある値の変化を監視し、その変化に応じて関数を実行する。図 3.2 に、先の例におけるクラウドデータベース・ストレージの構造、およびサービス間でどうデータをやり取りするかを示した。

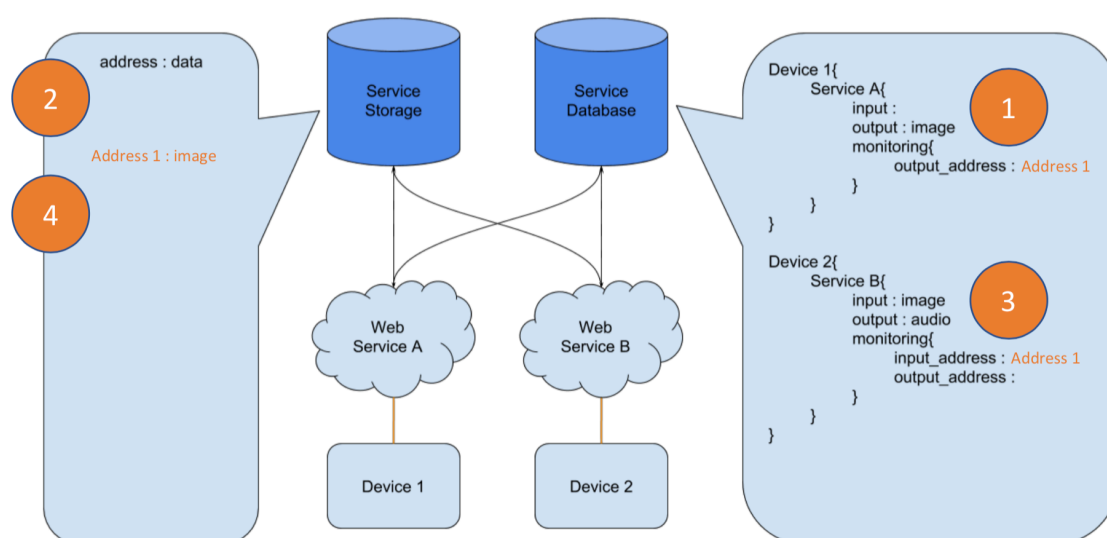


図 3.2 クラウドデータベース・ストレージを用いるウェブサービスの連携

1. サービス A の監視領域にデータをアウトプットするストレージ上のアドレスを書き込む。
2. サービス A がストレージ上にデータを書き込む。
3. サービス B の監視領域にインプットするデータのストレージ上のアドレスを書き込む。
4. サービス B がストレージ上のデータを読み込む。

周辺機器の持つサービスをユーザに測位に基づき提示することによる相互運用性の向上

2.2 節で明示したように、ユーザや周辺機器の位置や向きおよびその連続的な動きを追うことがインタフェースのデザインに重要である。

プライベートな空間では、どのようなインタラクションが存在するのかはユーザにとって自明であった。しかし、パブリックな空間ではその空間に存在するサービスを知らないユーザがいることを考慮しなければならない。したがって、スマートフォンのようにユーザが日常的に持ち歩くユーザ機器を用いて適切なインタフェースを提示したい。ユーザとサービスとの相対的な位置関係に基づいて、たとえば Augmented Reality (AR) で提示する。このようサービスディスカバリを実現するためには、ユー

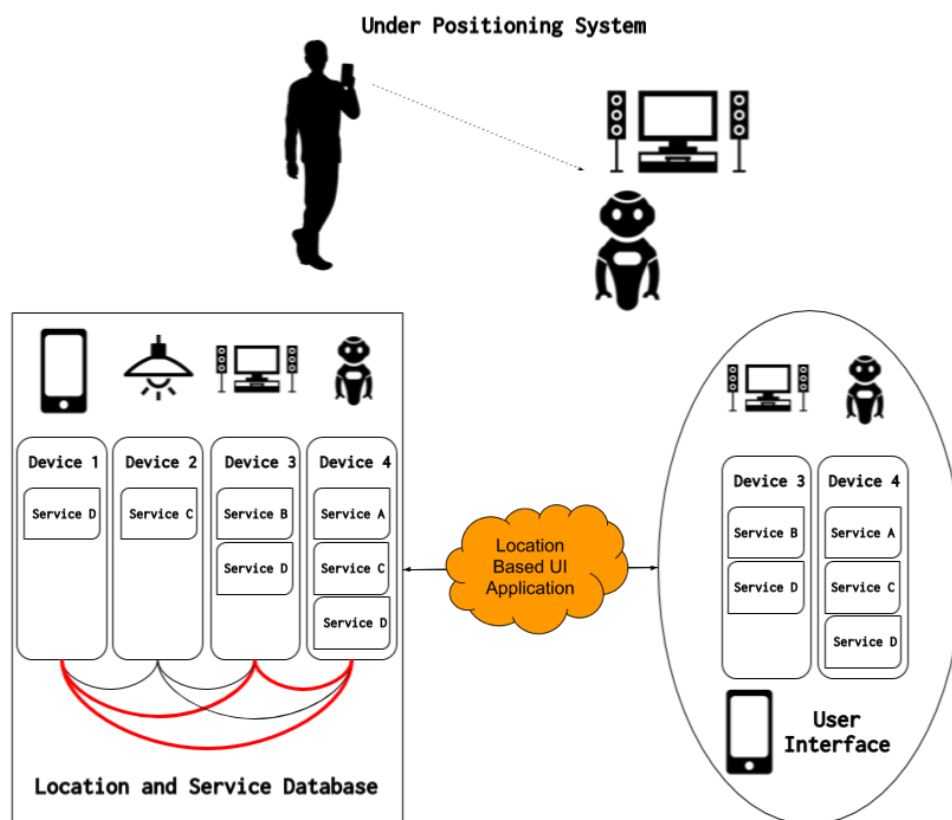


図 3.3 ユーザと周辺機器の位置関係に応じたインタフェースの提示

ザ機器が、周辺機器との相対的な位置関係およびその機器の持つサービス情報を取得しなければならない。したがって、これらを紐づけクラウドデータベースに書き込む。これにより図 3.3 のように、ユーザが向いている方向に存在する周辺機器に関する情報だけをユーザに提示する、などといったフィルタリングが可能となる。ユーザに対して過大な情報を提示することは好ましくない。

また、ユーザが提示されたインタフェースを用いて機器を連携させる際に取りうる動作としては、タップやドラッグ・アンド・ドロップなどといった、既存の GUI で用いられるようなものを考える。ユーザ機器からある単一の周辺機器に接続するときにはタップ、複数の周辺機器同士を連携させるときにはドラッグ・アンド・ドロップが有効である。先述した、サービス A の出力をサービス B に入力するというインタラクションを起こすとき、たとえば、提示されているサービス A のインタフェースをサービス B のインタフェースにドラッグ・アンド・ドロップするといった操作によって実現できれば直観的である。

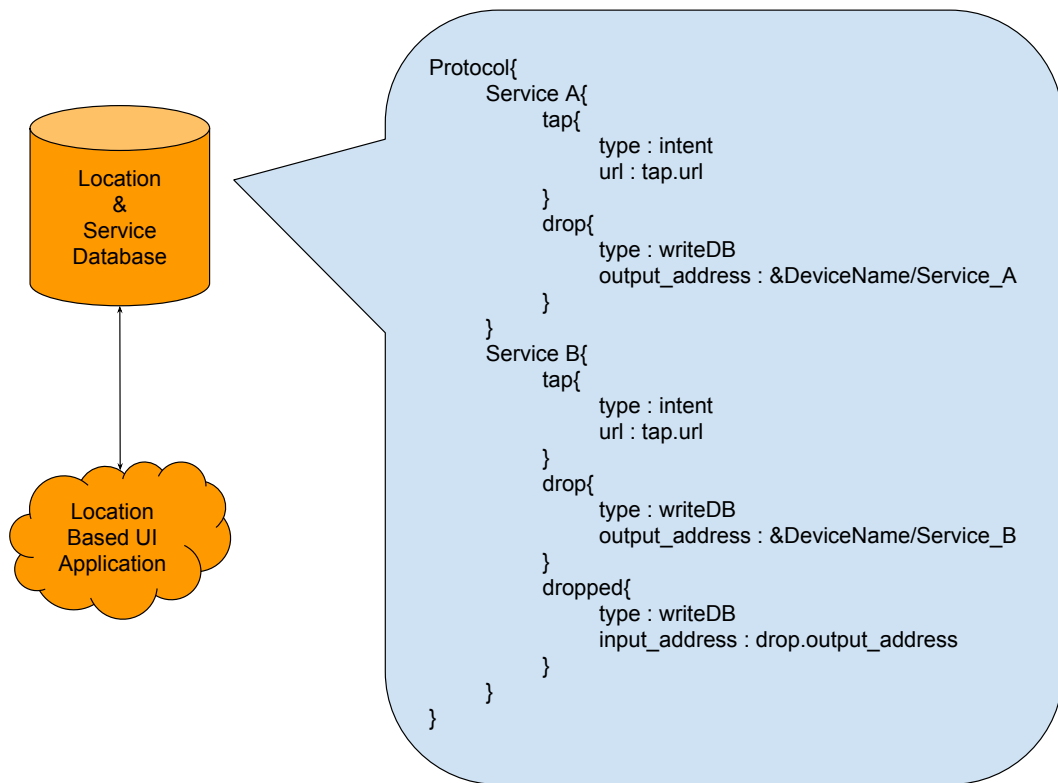


図 3.4 データベースに登録されるサービスプロトコル

以上のように、ユーザのインタフェースに対する動作に応じて何かしらの処理をさせるためには、データベースに各サービスのプロトコルが書き込まれていれば良い。図 3.4 にサービス A および B におけるプロトコルの例を示す。

備考

以上のようなシステムの組み方は、部分的には、既存の Service Discovery Protocol (SDP) をクラウドデータベース上に実装しているだけに見えるかもしれない。しかし、既存の SDP との違いは、サービスを発見するためだけではなく、サービスを利用・連携させるためのプロトコルも用意していることにある。

3.2 システム構成

3.2.1 機器

本提案におけるシステム下の機器は、ユーザ機器とそれ以外に分けられるが、周囲のユーザにサービスを提供する場合はウェブサービスに接続する必要があるためブラウザが動くことが求められる。また、専用アプリケーションを起動できる機器をユーザ機器と呼称しているに過ぎず、どちらもサービスを展開できる。また、測位システム下で機能する測位用モジュールが組み込まれてあると仮定する。

3.2.2 測位システム

次の要件を満たすものならばどんなシステムを利用しても良い。

1. 対応する測位モジュールを持つ全ての機器の測位ができること。
2. ユーザの位置に対して周辺機器がパーソナルスペースのどの空間にあるのかが分かる以上の精度・範囲があること。

測位は、周辺機器に対してサービスを提供したい、あるいは周辺機器のサービスを利用したいという所有者の意志にしたがって同意を得てから開始されるものであると良い。

3.2.3 クラウドデータベース

図 3.5, 3.6, 3.7 に位置・サービスデータベースの階層構造を示す。機器の座標情報および提供しているサービス情報の一覧が紐づけられ、測位システム下ごとに位置・サービスデータベースへと書き込まれ、これが随時更新される。

このデータベースへのアクセスは専用アプリケーションと、後述する各種あらかじめ登録されたウェブサービスに限る。また、ウェブサービスは登録の際に、ウェブサービスごとのプロトコルをデータベースへと書き込む。このプロトコルには、専用アプリケーション上でユーザがインタフェースに対して何らかの操作をするのに対し、ど

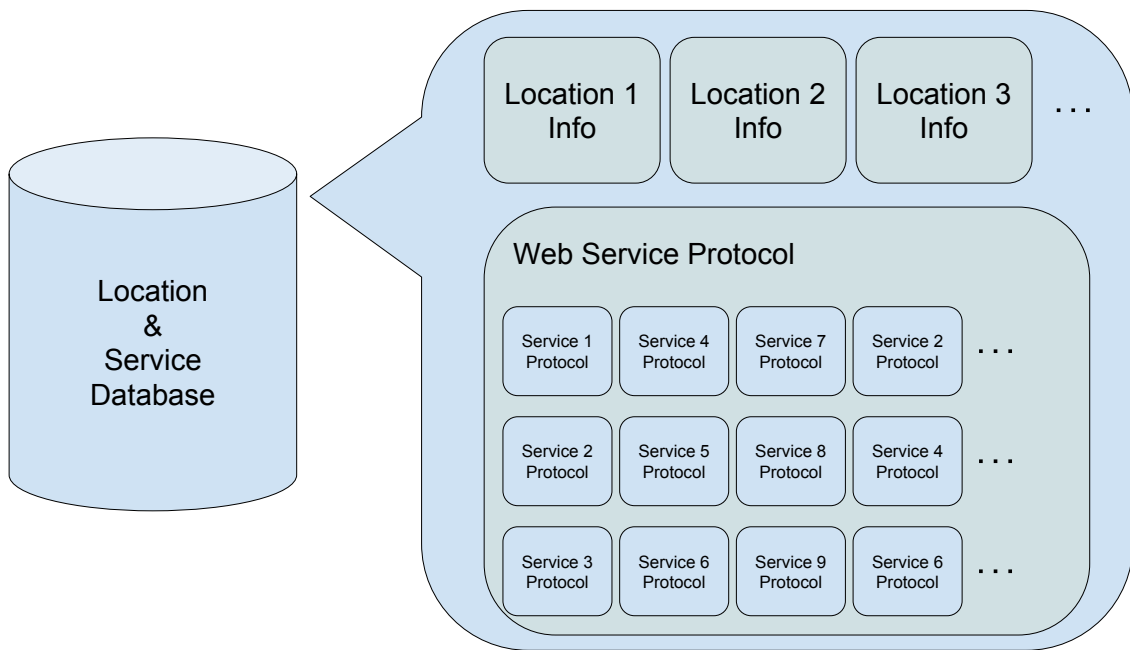


図 3.5 位置・サービスデータベースの概要

ういった処理をすれば良いのかが記述される。

3.2.4 専用アプリケーション

専用アプリケーションは、ユーザ機器で周辺機器の位置およびサービス情報を取得する。ユーザ機器から鍵を受け取り、位置・サービスデータベースへとアクセスする。ユーザ機器を含む周囲の機器の位置をこのデータベースから取得し、ユーザ機器と他の機器の相対的な位置関係を計算し、対応したインタフェースを表示する。また、ユーザに提示する情報が過多にならないようフィルタリングする機能も担っている（図 3.3）。

提示しているインタフェースに対してユーザが何らかの操作をしたときには、位置・サービスデータベースに存在するウェブサービスのプロトコルを確認し、その内容とユーザの操作を照らし合わせ、このデータベース上のウェブサービスが監視する領域をそのプロトコルにしたがって書き換える。

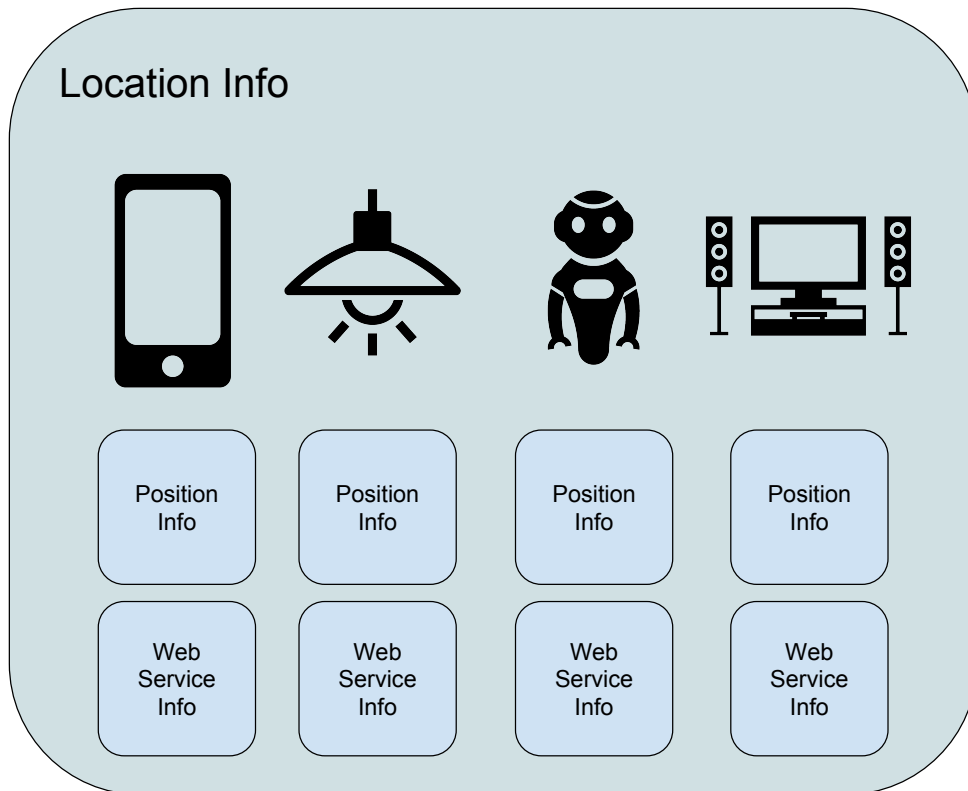


図 3.6 空間ごとの、機器の座標情報とサービス情報の管理

3.2.5 ウェブサービス

ウェブサービスは、接続してきた機器からその機器が存在している空間の情報を取得する。そして、位置・サービスデータベースへとアクセスし、接続してきた機器のサービス情報を更新する。図 3.7 に示したように、サービス情報にはサービスの名前やウェブサイトのアドレスなどの基本的な情報に加え、サービスに設定される関数を呼び起こすために監視していなければならない領域がある。

すべてのサービスはいずれかの機器と紐づいている必要があるが、サービスの位置は必ずしも機器自体の位置と一致する必要はない。たとえば、プロジェクタがスクリーンにプロジェクションするように、機器自体とは離れた位置に機器が出力を行う場合がある。ユーザがスクリーンに対して何らかの画像や映像を映したいと考えたと

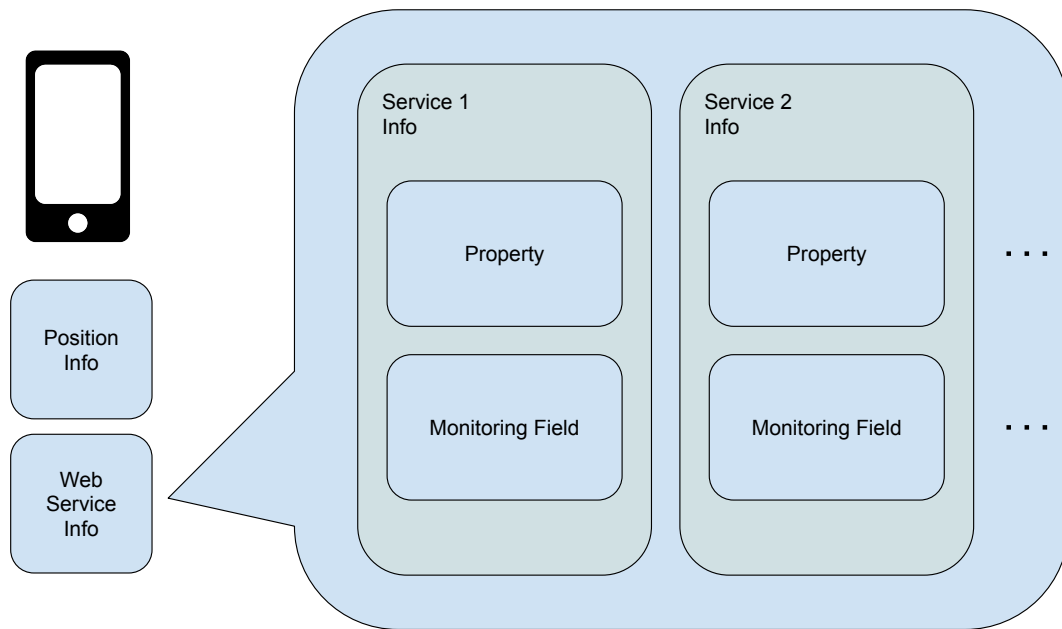


図 3.7 機器の持つサービス情報の内訳

き，ユーザが直観的にインタラクトするのはスクリーンであってプロジェクタではない．つまりプロジェクタが，入力として受けとった画像や映像をプロジェクションするサービスを展開する場合，そのサービスの位置はプロジェクタと同じ位置ではなくスクリーンの位置とするべきである．

第 4 章

実装

4.1 想定したシナリオ

貸し会議室にて

ユーザ 1 は Windows のノート PC を持ち込んでおり，これは持参したポータブル Wi-Fi に接続されている．ユーザ 1 はこのノート PC でプレゼンテーション用のスライドを開いた．画面共有ができるウェブサービスの存在を知っているユーザ 1 は，ブラウザを開きこのウェブサービスに接続し，スライドを画面共有できるよう設定した．

ユーザ 2 は標準的な電話回線につながっているスマートフォンを持っている．ユーザ 2 は専用アプリケーションを開き，ユーザ 1 のノート PC をスマートフォンのカメラで覗くとアイコンが重畳された．ユーザ 1 のノート PC が画面共有サービスを提供し始めたことを受け，ユーザ 2 はそのアイコンをタップした．するとユーザ 2 のスマートフォン上でブラウザが立ち上がりユーザ 1 のスライドが画面共有されているサイトが開いた．

ユーザ 2 は関連する資料を持っていたため，これをユーザ 1 に見てもらいたくなった．ユーザ 2 は持ち込んでいた Mac のノート PC で同様のウェブサイトを開き，共有したい資料を画面共有できるように設定した．専用アプリケーションを使い，新しく自身のノート PC に重畳されて描かれているアイコンをユーザ 1 のノート PC に重畳されているアイコンにドラッグ・アンド・ドロップすると，ユーザ 1 のノート PC 上のブラウザにユーザ 2 の資料が画面共有された．

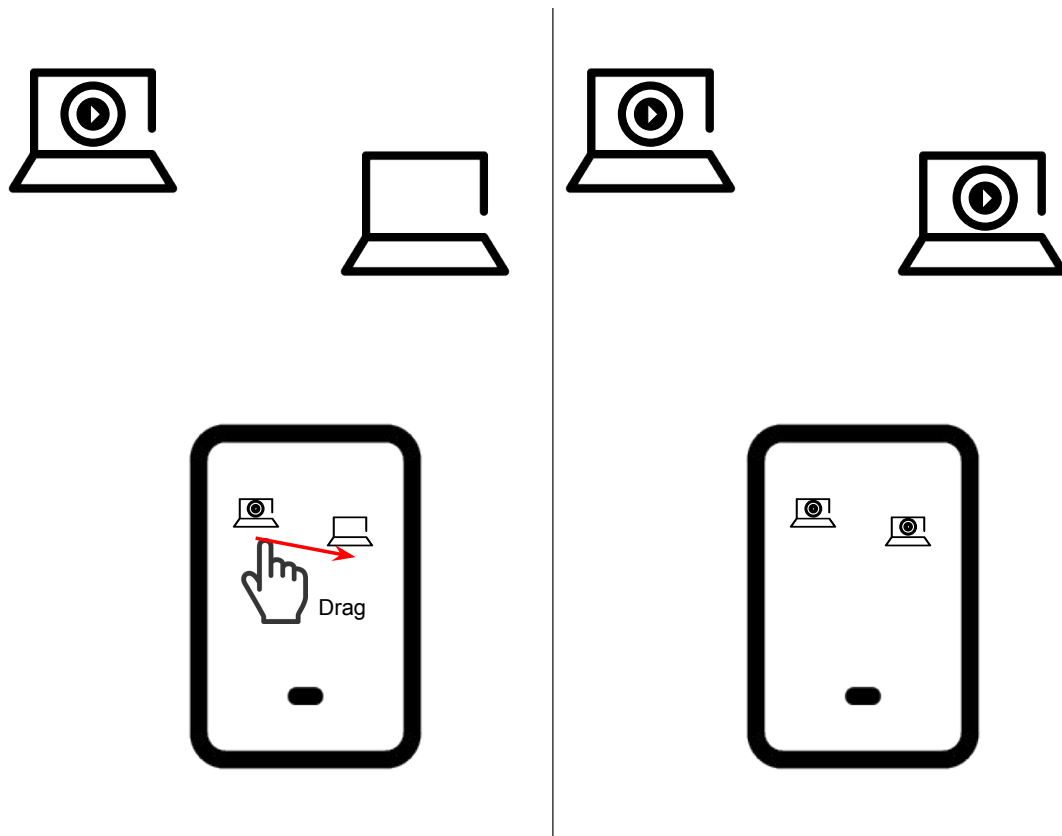


図 4.1 実現したいインタラクションの例

4.2 概要

第 3 章で述べた提案システムの適用シナリオを 4.1 節に記した．図 4.1 は，シナリオの一部のイメージ図である．

実装したデモシステムの構成は次のとおりである．

1. スマートフォン，ノート PC × 2
2. UWB 測位システム
3. 機器の位置とサービスを管理するクラウドデータベース
4. Android アプリケーション
5. 画面共有ウェブサービス

4.3 システム構成

4.3.1 スマートフォン，ノート PC × 2

使用したスマートフォンおよび周辺機器は次の通りである．

1. Android 端末 : Xperia X Dual F5122
2. ノート PC : MacBook Pro (13-inch, 2016, Four Thunderbolt 3 Ports)
3. ノート PC : dynabook R734/M

4.3.2 UWB 測位システム

ユーザ周囲の様々な機器が測位できる環境を構築するため，図 4.2 に示すような測位タグを使う UWB 測位システムを用いた．事前に計測した 4 箇所に親機を設置することで，移動機の位置が mm 単位で算出される．ユーザ機器と共にこの移動機を持ち歩くことで，ユーザ機器の位置が測位されることとした．測位環境下における全ての機器がリアルタイムに測位されていることが望ましかったが，移動機の数に限りがあるため，ユーザ機器として今回利用した Android 端末以外の機器はあらかじめ測位した場所に設置することにした．

この測位システムでは，親機は Wi-Fi に接続し，測位した移動機の位置を Wi-Fi 経由で通知する．そのため，ノート PC の dynabook を測位システムと同じ Wi-Fi に接続し，dynabook をサーバ化した．dynabook は受け取った移動機の位置を，随時クラウドデータベースに Android 端末の位置として書き込む．

この UWB 測位システム [5] では，Ultra Wide Band (UWB) という高周波で直行性のある電波を，親機と移動機との間で行き来させ，その往復にかかった時間と光速を掛け合わせることで 2 点間の距離を算出している．親機はあらかじめ計測した位置に設置するため，各親機と移動機との距離が 3 つ以上揃えば移動機の位置が点として算出できる．



図 4.2 UWB 測位システムで用いる移動機（左）と親機（右） [5]

4.3.3 機器の位置とサービスを管理するクラウドデータベース

実装環境における機器の位置やサービス情報をインターネットを通じて取得できるようにするため，Google の Firebase が提供する Cloud Firestore および Cloud Storage [34] というクラウドデータベース・ストレージを用いた．

4.3.4 Android アプリケーション

周辺機器との相対的な位置関係に応じて機器情報やサービス情報を図 4.3（左）のように AR 表示する Android アプリケーションを開発した．Firebase と連動させ，機器の位置とサービスを管理するクラウドデータベース・ストレージにアクセスできるため，測位環境下にある全ての機器の位置およびサービス情報を把握する．

図 4.3（右）では，機器情報が画面上部に格納されている．これは，一度場所と共に確認した機器情報を保持しておくためのコンテナの役割を果たす．機器に AR 表示されるアイコンをタップしたり，他のアイコンへとドラッグ・アンド・ドロップすることで，その機器のサービスにアクセスしたり，他の機器同士での連携を実現する．図

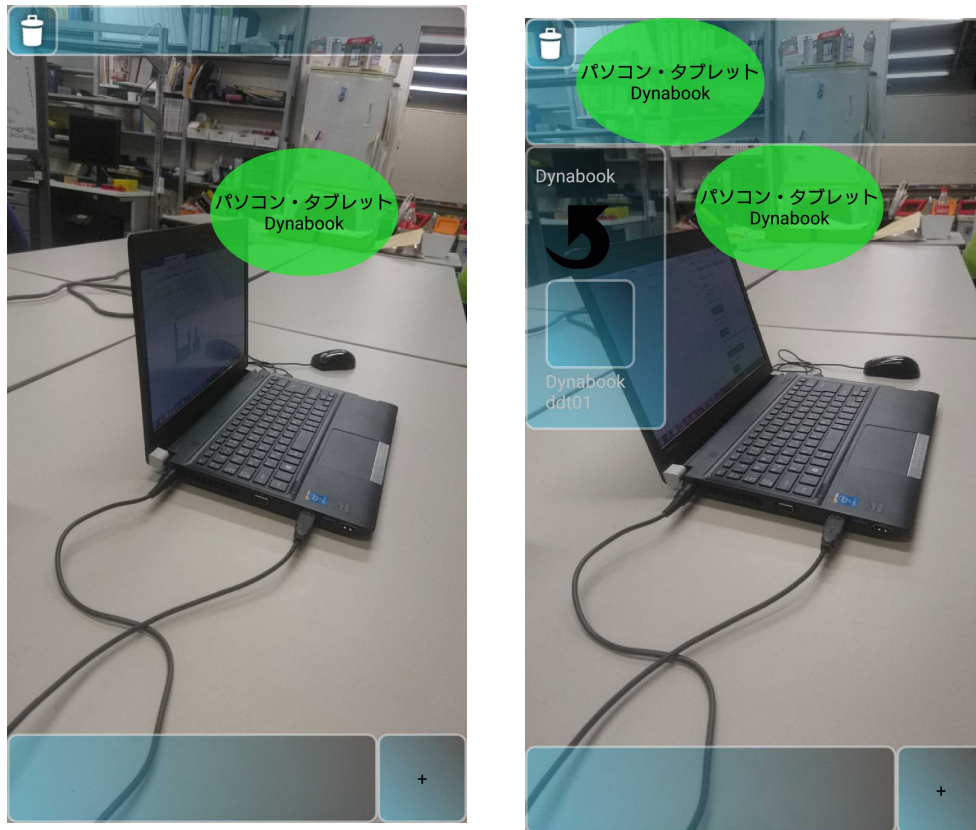


図 4.3 ノート PC に機器情報が AR 表示される様子

4.3（右）の画面左側にはノート PC dynabook の持つサービス情報が表示されている。ここで得られるサービス情報は、後述するウェブサービスにブラウザで接続することでウェブサービスがデータベースに書き込むものである。アイコンに対してタップやドラッグ・アンド・ドロップが発生したときは、どのサービスをユーザが操作したかを確認し、クラウドデータベース上のサービスプロトコルを参照し、そのプロトコルにしたがってデータベース上の所定の箇所を書き換えるまたはブラウザを起動する。

図 4.3 の画面下部には、Android 端末の持つデータをストックしておけるコンテナが用意されている。右下のボタンから内部ストレージ上のデータを選択できる。選択したデータはコンテナ内にアイコン表示され、これもドラッグ・アンド・ドロップに対応している。このアイコンを周辺機器のサービスに対してドラッグ・アンド・ドロップすることで、Android アプリケーションがクラウドストレージにデータをアップロードし、そのアドレスをサービスの監視領域に書き込むことで、周辺機器が Android 端末内のデータを使ってサービスを提供することができる。

4.3.5 ウェブサービス

画面共有サービス

異なる所有者，異なる OS の機器間でも画面共有ができるようなウェブサービスを Firebase のホスティング機能を利用し立ち上げた．ブラウザからウェブサービスに接続することで，その機器がサービス化される．すなわち，クラウドデータベースに該当機器のサービス情報を書き込んだ上で，監視領域への書き込みを待つ状態へと移行する．ウェブサイトにアクセスした機器を識別するため，Firebase のメールアドレスとパスワードによる認証機能を用いた．

このウェブサービスの構築には SkyWay [35] というブラウザ上での WebRTC を簡単に実装できる JavaScript 用の SDK を使っている．SkyWay では，ウェブサービスに接続するごとに無二の文字列が発行される．この文字列を知っている相手とのみ画面共有が行われる．クラウドデータベース上の自身に割り当てられる監視領域を定期的にチェックし，通信相手の文字列が書き込まれれば，その文字列を用いて相手との通信を確立する．

実際にデータベースへと書き込むサービス情報とサービスプロトコルを図 4.4 に示した．

投稿画像表示サービス

画面共有サービスと同様，クラウドデータベースに機器のサービス情報を書き込んだ上で，監視領域への書き込みを待つウェブサービスを開発した．

実際にデータベースへと書き込むサービス情報とサービスプロトコルを図 4.5 に示した．

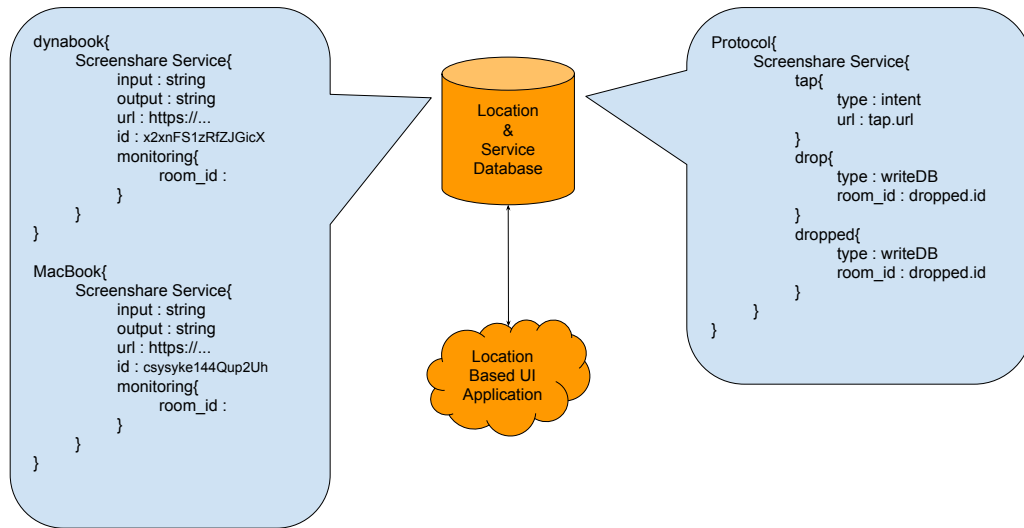


図 4.4 画面共有サービスがデータベースに書き込むサービス情報（左）とサービスプロトコル（右）

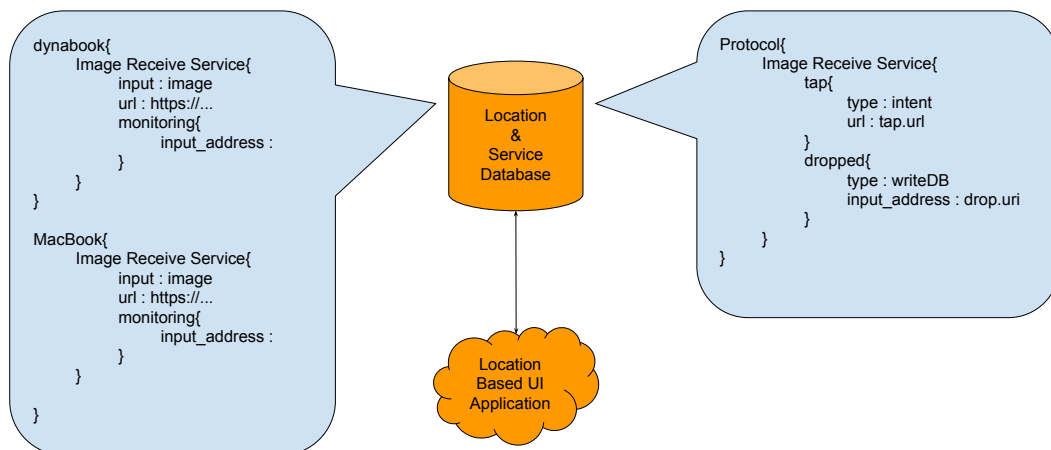


図 4.5 投稿画像表示サービスがデータベースに書き込むサービス情報（左）とサービスプロトコル（右）

第 5 章

評価と考察

5.1 動作

5.1.1 測位システム

実装したシステムで使った UWB 測位システムの精度について評価した。測位システムの移動機が実際の位置と比べてどれだけずれていたか、また、秒間に何度測位されるのかを計測した。実際の位置はレーザー距離計を用いてあらかじめ計測している。このあらかじめ計測した位置に移動機を設置し、測位システムが計測した座標を収集した。

2m 程度離れた二つの座標で計測を試みたが、表 5.1 図 5.1 のような結果になった。UWB による測位手法も、他の電波を用いる手法と同様に、マルチパス問題が発生する。そのため、場所によって測位の精度が異なってくる。平均的に 1m 未満の誤差に収まっているが、これは「親密な空間」にある機器を「社会的な空間」にあるものとシステムが判断する精度である。したがって将来的には、より精度の高い測位システムが要求される。

5.1.2 ユーザインタフェース

機器に重畳される AR は測位システムによる誤差に加え、スマートフォンの IMU の誤差の影響を受ける。つまり、ユーザがなるべく動かずにスマートフォンを固定しているつもりでも、AR は絶えず揺れ動く。この現象によりユーザがあまり不快に感じ

18 points in 10 sec			13 points in 10 sec		
error	min	356 mm	error	min	631 mm
	max	1236 mm		max	1024 mm
	mean	645 mm		mean	807 mm

表 5.1 測位の頻度と誤差の最小値，最大値，平均値

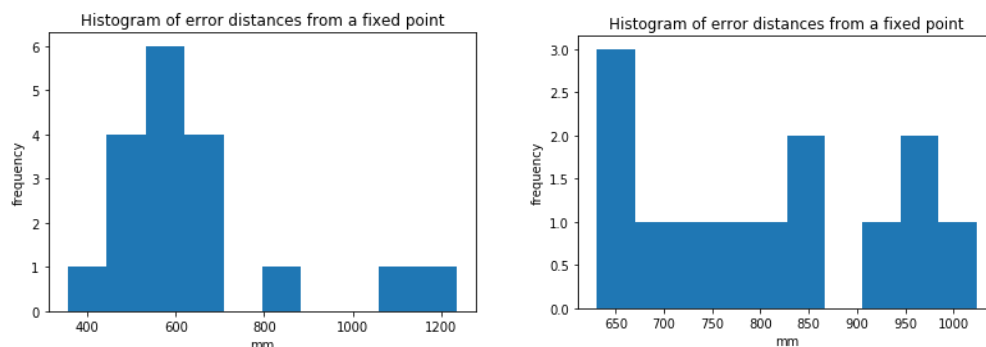


図 5.1 測位の誤差のヒストグラム

ないよう，図 3.3（右）のように，画面上部にインタフェースを固定できるようにした。

5.1.3 タップおよびドラッグ・アンド・ドロップによる機器連携

ノート PC がウェブサービスに接続していれば，データベース上にサービス情報が書き込まれる．サービス情報のインタフェースをタップやドラッグ・アンド・ドロップすることで所望する連携ができた．すなわち，タップしたときにはブラウザを開く intent が呼び出され，ウェブサービスに対応したブラウザを選択することでそのブラウザが開き，ノート PC の画面が共有されるウェブサイトへとアクセスできた．図 5.2 には，ドラッグ・アンド・ドロップによって，MacBook 上の画面を dynabook に共有させた過程を示している．図 5.2 に写っているのが dynabook で，画面共有が開始されることで画面上に表示されている映像（左）が MacBook へと共有され，MacBook の画面が dynabook へと共有されている（右）。

正常な WebRTC 通信が確認できたブラウザは Mozilla Firefox と Google Chrome に限られるが，これは SkyWay の仕様に依る．また，スマートフォンの画面を共有する機能は SkyWay にはまだ実装されていないが，スマートフォンのカメラとマイクで取得したビデオを送信することはできた．

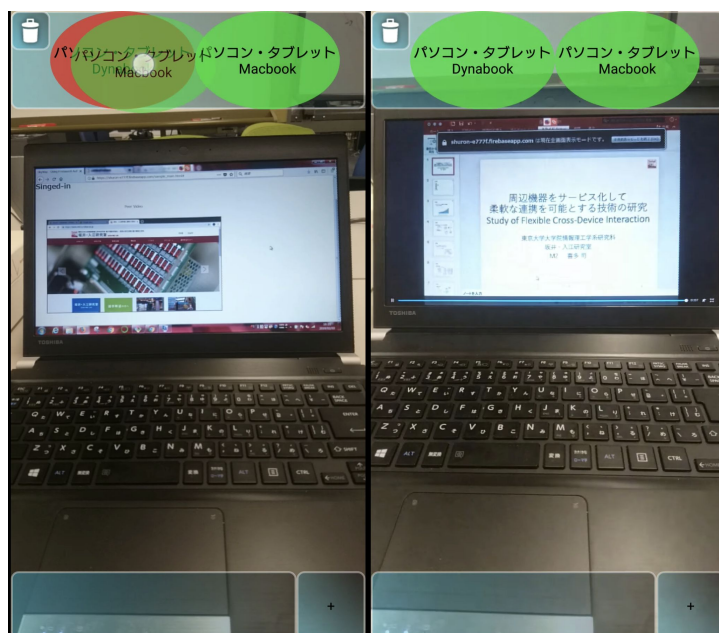


図 5.2 画面共有をドラッグ・アンド・ドロップで実現している様子

ウェブサービスを使用する短所としては、データベースにサービス情報を書き込んだ後、ウェブサービスがその情報を削除しないでブラウザを閉じたときなどに、すでに存在しない過去のサービス情報がデータベースに残り続けてしまうという問題が起きた。複数の測位システムが展開した場合にも、同じ現象が起きうる。しかし、これはデータベースの更新時にタイムスタンプをつけ、データベース側で一定期間更新がなかったデータを削除するように決めれば解決する。

5.2 理論

5.2.1 ユーザ観点

これまで周辺機器を利用する際には、どこに何の機器がありどんな機能を持っているかを事前にユーザは把握しておく必要があった。しかし、実装したシステムのように AR が表示されることで、ユーザは自身が利用できるサービスの存在を新たに知ることができる。

また、ユーザが各機器のネットワークを意識することなく周囲のインターネット機器を利活用できるようになる。従来は専用のアプリケーションなどをインストールし

た機器同士でしか連携できなかったことが、提案システムにより実現できる。

5.2.2 開発者観点

機器同士の連携を実現する多様なサービスが考えられるが、従来であれば OS や機種に依存する専用のアプリケーションを構築していた。我々の提案に則ったウェブサービスであれば、一つの開発で OS に依らない機器を対象としたサービスが提供できる。

5.3 今後の課題

5.3.1 無線機器同士の連携

本研究では周辺機器のうち、ブラウザを開くことができる機器のみを対象とした。しかし、実際には我々の身の周りには BLE 機器や Wi-Fi 機器などの無線通信機器が一定数存在する。二村ら [24] は、ユーザ機器からこれらの情報をユーザ機器のブラウザ上に表示する手法を提案しているが、それらを連携させて使えるような枠組みには言及していない。

我々の提案システムに彼らの手法を取り入れることで、異なる無線機器同士を連携させることもできると考えられる。すなわち、彼らの提案した無線機器固有のウェブサービスに接続したユーザ機器が、接続した無線機器の機能をサービスとして周囲に提供するという流れになる。

こういった実装を進めることで、真に、ユーザは周辺機器を自在に操作・連携させることができるようになる。

5.3.2 ユーザビリティの評価

本論文では、被験者による提案システムのユーザビリティ評価を行っていない。Proxemic Interaction も考慮したより実用的なサービスを実装し、少なくとも 2 箇所以上の空間において、提案システムを体験してもらうことで本研究の意義を強く主張していきたい。

5.3.3 機器をウェブサービス化することの代償

クラウドデータベース上のデータをどこからでも読み書きできるようにしてはならない。クラウドデータベース上の特定の領域を監視し値の変化に応じて関数が実行されるウェブサービスを提案したが、悪意のあるユーザが勝手にクラウドデータベース上の値を書き換えることができてしまってはいけない。あくまでユーザが周辺機器を利活用する際の助けとなるようなシステムの構築が目的であるため、クラウドデータベースへの読み書きの権限も空間ごとに与えられるようなものが良い。そこで、測位システムがクラウドデータベース・ストレージ認証用の鍵を生成することが求められる。一度でも訪れた空間であれば認証用の鍵を知っていていつどこからでもデータベースにアクセスできる、とさせないために定期的に鍵が更新されることが望ましい。

以上の問題を解決できるよう、測位システムの要件を改める必要があると同時に、具体的にどのような鍵を生成すべきかを熟考することが求められる。

第 6 章

おわりに

本論文では、インターネットにつながるユーザ機器を含む多種多様な周辺機器がサービス化する状況を想定し、機器同士あるいはサービス同士がクラウドデータベース上に情報を共有することで柔軟に連携できる技術を提案した。ユーザ機器と周辺機器を高精度に測位することで、ユーザ機器を中心とした相対的な位置関係が計測できる。そして、専用アプリケーション上でこの相対的な位置関係を用いた周辺機器に対する AR による機器インタフェースの表示が行えることを確認した。さらに画面上に表示されるこの機器インタフェースに対してユーザがタップやドラッグ・アンド・ドロップすることで直観的な機器連携ができるデモシステムを実装した。

また、我々の実装ではユーザ機器として汎用的な機器であるスマートフォンを用いた。しかし、ユーザ周囲の空間において提供されているサービスを可視化する技術である以上、複合現実との相性が良いことは明らかである。ヘッドマウントディスプレイが PC から独立してモバイル化した今、こういったユーザ機器においてどのようなインタラクションがユーザにとって操作しやすいか、分かりやすいか、便利か、使い心地が良いかを考えることの重要性が増してきた。

Bluetooth Special Interest Group (SIG) によれば、Bluetooth 5.1 では cm 単位の測位ができるようになる [36]。我々の提案における最も乗り越えがたい課題の一つが汎用的な測位システムの存在であった。Bluetooth のように汎用的に様々な機器に組み込まれた無線通信規格で高精度な測位を行えるようになることで、ユーザや機器の位置関係を利用するサービスの開発にこれまでより多くの人々が携わることになるだろう。

ユーザビリティの高いサービスが数多く登場するに並行して、それらを連携させる技術が重要となる。我々の提案のような、ユーザと機器の位置が分かることで相互運用性の高いシステムを構築できるという主張は現実味を増した。

ユーザは、どのような周辺機器とどういった位置関係にあるときに、どのような欲求を持つのか。これらは、我々の実装したシステムを拡張していくことで、次第に明らかになっていくだろう。位置だけでなく周囲のコンピュータが持つ多様なセンサ情報も扱うことで、ユーザの意図を自然と理解するコンテキスト・アウェアなシステムの実現が近づいている。以上を以って、今後の展望とする。

参考文献

- [1] Masahide Nakamura, Hiroshi Igaki, Haruaki Tamada, and Ken ichi Matsumoto. Implementing integrated services of networked home appliances using service oriented architecture. In *Proc. of 2nd International Conference on Service Oriented Computing (ICSOC2004)*, pp. 269–278, 2004.
- [2] Jing Yang and Gábor Sörös. Augmenting smart object interactions with smart audio. In *Proceedings of the 9th Augmented Human International Conference, AH '18*, pp. 28:1–28:3, New York, NY, USA, 2018. ACM.
- [3] Steven Houben and Nicolai Marquardt. Watchconnect: A toolkit for prototyping smartwatch-centric cross-device applications. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pp. 1247–1256, New York, NY, USA, 2015. ACM.
- [4] Haojian Jin, Christian Holz, and Kasper Hornbæk. Tracko: Ad-hoc mobile 3d tracking using bluetooth low energy and inaudible signals for cross-device interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST '15*, pp. 147–156, New York, NY, USA, 2015. ACM.
- [5] Global Interface Technologies. 株式会社 日本ジー・アイ・ティー. <http://www.git-inc.com/index.html>.
- [6] IHS. Internet of Things (IoT) Connected Devices Installed Base Worldwide from 2015 to 2025 (In Billions). <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>, November 2016.
- [7] Jishuo Yang and Daniel Wigdor. Panelrama: Enabling easy specification of cross-device web applications. In *Proceedings of the SIGCHI Conference on Human Fac-*

- tors in Computing Systems, CHI '14, pp. 2783–2792, New York, NY, USA, 2014. ACM.
- [8] ECHONET CONSORTIUM. ECHONET. <https://echonet.jp/>.
- [9] ホームアプライアンス・オープンイノベーションシンポジウム ECHONET2.0 ビジョン発表 平松 勝彦. <https://www.youtube.com/watch?v=2K8GG5G4LJQ>.
- [10] 田中章弘, 中村匡秀, 井垣宏, 松本健一. Web サービスを用いた従来家電のホームネットワークへの適応. 電子情報通信学会技術研究報告, Vol. 105, No. 628, pp. 067–072, 2006.
- [11] 梅山莉奈, 増田剛志, 鈴木秀和ほか. 通信プロトコルの違いを意識しない直感的家電制御システムの実装. 第 78 回全国大会講演論文集, Vol. 2016, No. 1, pp. 261–262, 2016.
- [12] IFTTT. IFTTT helps your apps and devices work together. <https://ifttt.com/>, 2019.
- [13] Google. Google assistant, your own personal google. <https://assistant.google.com/>, 2019.
- [14] Amazon. Ways to build with amazon alexa. <https://developer.amazon.com/alexa>, 2019.
- [15] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, Vol. 3, No. 3, pp. 3–11, jul 1999.
- [16] 北原義典. イラストで学ぶヒューマンインタフェース, 2011.
- [17] Seng Wai Loke. Service-oriented device ecology workflows. In Maria E. Orlowska, Sanjiva Weerawarana, Michael P. Papazoglou, and Jian Yang, editors, *Service-Oriented Computing - ICSOC 2003*, pp. 559–574, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [18] Kathryn B Laskey and Kenneth Laskey. Service oriented architecture. *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 1, No. 1, pp. 101–105, 2009.
- [19] Akihiro TANAKA, Masahide NAKAMURA, Hiroshi IGAKI, and MATSUMOTO Ken-ichi. Adapting conventional home appliances to home network systems using web services. *IEICE technical report*, Vol. 105, No. 628, pp. 67–72, mar 2006.
- [20] 2019 W3C. World Wide Web Consortium (W3C). <https://www.w3.org/>.

- [21] 松倉隆一. 家電を web で制御するサービスプラットフォーム. Technical Report 48, 富士通 (株), mar 2014.
- [22] 経済産業省. スマートハウス・ビル標準・事業促進検討会 普及促進タスクフォース 中間とりまとめ. http://www.meti.go.jp/shingikai/mono_info_service/smart_house/fukyu_sokushin/201809_report.html, 2018.
- [23] Google. Google Play Instant. <https://developer.android.com/topic/google-play-instant/>.
- [24] 二村和明, 伊藤栄信, 坂本拓也, 中村洋介, 西垣正勝. スマート端末と周辺デバイスを簡単につなげる web ドライバ技術. 情報処理学会論文誌コンシューマ・デバイス&システム (CDS) , Vol. 7, No. 1, pp. 10–20, jan 2017.
- [25] Nicolai Marquardt, Robert Diaz-Marino, Sebastian Boring, and Saul Greenberg. The proximity toolkit: Prototyping proxemic interactions in ubiquitous computing ecologies. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pp. 315–326, New York, NY, USA, 2011. ACM.
- [26] Till Ballendat, Nicolai Marquardt, and Saul Greenberg. Proxemic interaction: Designing for a proximity and orientation-aware environment. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pp. 121–130, New York, NY, USA, 2010. ACM.
- [27] Edward Twitchell Hall. *The Hidden Dimension*. ANCHOR BOOKS EDITIONS, 1966.
- [28] Hannu Kukka and Pauli Marjakangas. Sondi: Audio-based device discovery and pairing for smart environments. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, NordiCHI '14, pp. 1059–1062, New York, NY, USA, 2014. ACM.
- [29] Jun Rekimoto and Yuji Ayatsuka. Cybercode: Designing augmented reality environments with visual tags. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, DARE '00, pp. 1–10, New York, NY, USA, 2000. ACM.
- [30] S. Mihara, A. Sakamoto, H. Shimada, and K. Sato. Augmented reality marker for operating home appliances. In *2011 IFIP 9th International Conference on Embed-*

- ded and Ubiquitous Computing*, pp. 372–377, Oct 2011.
- [31] Ankit Mohan, Grace Woo, Shinsaku Hiura, Quinn Smithwick, and Ramesh Raskar. Bokode: Imperceptible visual tags for camera based interaction from a distance. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pp. 98:1–98:8, New York, NY, USA, 2009. ACM.
- [32] Fulvio Corno, Alastair Gale, Päivi Majaranta, and Kari-Jouko Räihä. *Eye-based Direct Interaction for Environmental Control in Heterogeneous Smart Environments*, pp. 1117–1138. Springer US, Boston, MA, 2010.
- [33] Quan Kong, Takuya Maekawa, Taiki Miyanishi, and Takayuki Suyama. Selecting home appliances with smart glass based on contextual information. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pp. 97–108, New York, NY, USA, 2016. ACM.
- [34] Google. Firebase. <https://firebase.google.com/>.
- [35] NTT Communications. SkyWay - Enterprise Cloud WebRTC Platform. <https://webrtc.ecl.ntt.com/>.
- [36] Bluetooth SIG. Bluetooth enhances support for location services with new direction finding feature. <https://www.bluetooth.com/news/pressreleases/2019/01/bluetooth-enhances-support-for-location-services-with-new-direction-finding-feature>, Jan 2019.

研究業績

■主著：口頭発表（査読なし）

1. 喜多 司，高原 隼太郎，入江 英嗣，坂井 修一：高精度位置測位技術を用いたARによる直観的な機器操作，信学技報, vol. 116, no. 510, CPSY2016-151, pp. 333-338, 2017 年 3 月, CPSY2016-151, ISSN 0913-5685.41

■共著：デモポスター発表（査読あり）

1. 高原 隼太郎，喜多 司，入江 英嗣，坂井 修一：ユーザ周囲の多種多様な情報機器を「サービス化」する LBS インタラクション，25th Workshop on Interactive Systems and Software (WISS 2017)

謝辞

本研究を進めるにあたり，多くの方にお世話になりました．

研究の方向性や実現したいインタラクションのアイデアに関して，ミーティングの時間に限らずいつでも指導してくださった坂井修一教授，入江英嗣准教授に深く感謝いたします．特に指導教員の入江英嗣准教授には，学会の締め切りが近いときはたとえ年末年始でも丁寧に指導していただき，感謝しきれません．

また，研究室生活で様々なサポートをしてくださった秘書の八木原晴水さん，赤羽彩子さんに感謝いたします．

相談に対して真摯に回答してくださった先輩方，些細な不安を取り除くために忙しいときも雑談に付き合ってくれた同期，応援してくださった後輩一同にも感謝いたします．

私が本研究に全力を投じることができるよう支えてくれた家族にも頭があがりません．

本論文は，私の身の回りの多くの方々のおかげで完成させることができました．心より皆様に感謝いたします．

本論文の研究は，一部公益財団法人セコム科学技術振興財団の助成による．