修士論文 2009 年度 (平成 21 年度)

機器連携におけるマルチモーダルなインタラクション支援システムの構築

慶應義塾大学大学院 政策・メディア研究科
橋爪 克弥

katsuya@ht.sfc.keio.ac.jp

修士論文要旨 2009 年度 (平成 21 年度)

機器連携におけるマルチモーダルなインタラクション支援システムの構築

論文要旨

情報技術が進化し，ネットワーク接続性や計算能力を持つ情報家電や携帯端末が普及し，単一の機器からではなく，複数の機器が連携し利用者にサービスを提供する機器連携が注目されている．近年では，テレビを中心とした，家電連携が実用化され，利用者が意識することなくサービスを享受出来るようになった．また，機器連携を支援するためのネットワークやミドルウェア技術の発達により，携帯端末とセンサなど異なる機器間の通信が可能になった．

これらの進化の一方で，利用者がサービスを実行するには複雑な設定や操作が必要となるため，実現可能な機器連携サービスの自由度が限定されている．また，今後の機器連携によるサービスは，様々な場所で，状況に応じて機器が必要に応じて動的に組み代わりながら生成されるものであると考えられる．利用者がサービスを利用するためには，何らかの手法で機器に命令を送る必要があるが，既存の実世界インタフェースでは，近い距離だけや，異なるネットワークの機器を指定できないなどの制約がある．また，機器操作に比べて，利用者のメンタルモデルが欠如しているため，機器の接続や利用を容易にすると，サービスの柔軟性が失われるなどの問題が存在する．

本論文では，機器連携を実行しサービスを利用するための指示手法として，マルチモーダルなインタラクションを用いたサービス発見による指示手法を提案する．本手法では，触る，振る，写真を撮る，スキャンするの４つのインタラクションを利用者の距離や好みに応じて利用し，容易で，直感的な機器の特定を支援する．また，環境内に存在する複数のサービスを発見するために，機器に紐付けられたサービスを利用者に表示し，利用者のサービス発見での負担を軽減する．これにくわえ，利用者は携帯端末で動作する統一のインタフェースにより，サービスの柔軟性を確保しつつ，サービスの制御が可能である．連携サービスの利用手法の評価として，本手法のプロトタイプである Chameleon を実装し，ユーザビリティ評価を行い，利用者の主観満足度や認知的負荷を評価した．

キーワード

ヒューマンインタラクション，ユビキタスコンピューティング，機器連携，実世界指向インタフェース，マルチモーダルインタラクション

慶應義塾大学大学院 政策・メディア研究科

橋爪 克弥

Abstract of Master' s Thesis Academic Year 2009

Designing a System to Support Multimodal Interaction for Device Collaboration

Summary

Innovation in Information Technology has brought the wide spread of mobile devices and home electronics capable of computing and networking among people. The end users can enjoy the services provided not only by one medium but also by the corresponding network from various media. In addition, developments of the network and middleware technology to support device collaborations enabled communication between the different devices such as mobile devices and sensors.

However, in order to implement these services in which multiple devices are working together, the user must do complicated setting and operation. The flexibility of the available services is limited. In the future ubiquitous computing environment, providing appropriate services based on the users' situations and preferences by collaborating multiple smart devices will be necessary and significant.

In order to accomplish device collaboration with such diverse and fluid devices, it is necessary to identify devices for collaboration intuitively independently of the relationship between a user and the device. Flexibility of the services is lost when we facilitate connection and the use of devices because users don't have enough mental model of device cooperation.

In this paper, we propose the interaction technique "Chameleon" that uses the multimodal interactions and specify method of service discovery to realize device collaborations. In this technique, the system switches four interaction schemes; Touch, Swing, Snap and Scan corresponding to the user interaction, and enables flexible switching and collaboration of multiple devices. This system also eases the complexity of service discovery, without professional knowledge skills. The control of the services is possible while keep flexibility of the service by interface of the unification that works with a smart phone. We have evaluated the usability of Chameleon by collaborating and registering various computers and mobile devices.

Keywords

Human Computer Interaction, Ubiquitous Computing, Device Collaboration, Real World Oriented Interface, Multi-modal Interaction

Keio University Graduate School of Media and Governance

Katsuya Hashizume

目次

第 1 章　序論 1
1.1　本論文の背景 2
1.2　本論文の目的 2
1.3　本論文の構成 3

第 2 章　機器連携サービス利用のためのインタラクション 4
2.1　機器連携サービスを支援する基盤技術の進化 5
2.1.1　ネットワーク層における通信支援 5
2.1.2　ミドルウェア層における機器連携支援 5
2.1.3　インタフェース層におけるサービス支援 5
2.2　シナリオ 6
2.3　各層における連携のモデル化 7
2.3.1　ネットワーク層における要件 7
2.3.2　ミドルウェア層における要件 7
2.3.3　インタフェース層における要件 7
2.4　機器連携におけるユーザインタフェース 8
2.5　問題点 9
2.5.1　動的な環境下での対象物指定の困難性 10
2.5.2　機器同士の接続操作における利用者のメンタルモデルの欠如 10
2.5.3　統一的サービス利用インタフェースの欠落 11
2.5.4　機器連携サービスの柔軟性と認知的負荷のトレードオフ 11
2.6　機能要件 11
2.6.1　様々な空間に存在する機器の直接指定と操作 11
2.6.2　利用者の認知的負荷が少ない接続指示 12
2.6.3　統一インタフェースによる柔軟性の高いサービス制御 12
2.6.4　柔軟なサービス作成のためのミドルウェア 12
2.7　まとめ 12

第 3 章　マルチモーダルなインタラクションによる機器連携手法　Chameleon 13

3.1 システム概要 . . . 14
3.2 機器との距離に応じた機器指定手法 . . . 14
3.2.1 Distance Based Interaction Method . . . 15
3.2.2 シームレスなインタラクションの自動的切り替え . . . 17
3.3 サービス発見型機器接続手法 . . . 17
3.3.1 連携状態閲覧型操作インタフェース . . . 18
3.3.2 サービス発見型機器選択手法 . . . 19
3.4 統一インタフェースによるサービス制御手法 . . . 21
3.5 機器非依存型サービス記述手法 . . . 21
3.6 関連研究 . . . 23
3.6.1 連携ミドルウェア . . . 24
3.6.2 機器指定手法 . . . 24
3.6.3 機器連携支援インタラクション . . . 24
3.7 まとめ . . . 27

第 4 章 Chameleon の設計 28
4.1 Chameleon の設計概要 . . . 29
4.1.1 ソフトウェア構成 . . . 29
4.1.2 ハードウェア構成 . . . 30
4.2 動作概要 . . . 30
4.2.1 連携サービス実行の動作概要 . . . 31
4.2.2 デバイス発見の動作概要 . . . 31
4.3 まとめ . . . 33

第 5 章 Chameleon の実装 34
5.1 ハードウェアの実装 . . . 35
5.1.1 機器管理サーバ . . . 35
5.1.2 コントローラ . . . 36
5.1.3 機器群 . . . 36
5.1.4 RFID リーダ . . . 36
5.2 ソフトウェアの実装 . . . 37
5.2.1 コード解決モジュール . . . 37
5.2.2 サービス解決モジュール . . . 39
5.2.3 コマンド送信モジュール . . . 39
5.3 まとめ . . . 39

第 6 章 評価実験 41
6.1 機器指定におけるマルチインタラクションの必要性 . . . 42

6.1.1 実験手順 42
6.1.2 実験結果 42
6.1.3 考察 43
6.2 インタラクション切り替え手法 44
6.2.1 実験手順 44
6.2.2 実験結果 44
6.2.3 考察 46
6.3 機器選択型とサービス発見型の機器接続手法の評価実験 46
6.3.1 実験手順 46
6.3.2 実験結果 47
6.3.3 考察 47
6.4 Distance Based Interaction Method を用いた機器連携の評価実験 48
6.4.1 実験手順 48
6.4.2 実験結果 49
6.4.3 考察 50
6.5 まとめ 51

第 7 章 結論 52
7.1 結論 53
7.2 今後の展望 53
7.2.1 アフォーダンスの向上 53
7.2.2 インターオペラブルな機器連携サービスの実現 54
7.2.3 連携時のネットワーク状態の可視化 54

参考文献 56
付録 A 評価用紙 60

図目次

2.1 ミドルウェア層での連携モデル 8
2.2 Out-of-Box Experience For Easy Device Registration 9
2.3 利用者と機器，サービス間のインタラクション 10

3.1 Chameleon の画面遷移 14
3.2 Distance-based Interaction Method 15
3.3 インタラクションの利用者の動きによる動的切り替え 18
3.4 マルチインタラクションコード 19
3.5 操作インタフェース 19
3.6 サービス利用コントローラの雛形例 22
3.7 デバイス記述 XML 23
3.8 Pick-and-Drop 25
3.9 Touch-and-Connect 26
3.10 tranStick 26
3.11 Gaze-Link 26

4.1 ソフトウェア構成 29
4.2 ハードウェア構成 31
4.3 連携サービス実行の動作概要図 32
4.4 デバイス発見の動作概要 32

5.1 プロトタイプ実装 35
5.2 SunSpot と Skyetek M1-Mini による RFID リーダ 37
5.3 RFID プリンタ P-Touch RL-700S 37
5.4 連携実行 XML 40

6.1 a) 実験環境と b) 利用者と機器との位置関係 42
6.2 機器選択時のインタラクション手法の使い分けの割合 43
6.3 自動インタラクション切替と手動インタラクション切替手法のスクリーンショット 45
6.4 タスクを終了するまでの所要時間 45

6.5 NASA TLX による作業負荷の平均値 47
6.6 評価シナリオ 49

表目次

2.1 ネットワーク，ミドルウェア，インタフェース層における連携 7

3.1 デバイス記述例 23

5.1 プロトタイプ実装に用いた計算機の仕様 35
5.2 プロトタイプ実装に用いたコントローラの仕様 36
5.3 プロトタイプ実装に用いた機器群 36
5.4 マルチインタラクション ID 38

6.1 機器選択時のインタラクション手法の使い分け 43
6.2 シームレスインタラクション切替手法の誤認識 45
6.3 質問結果：シームレスインタラクション切替手法 45
6.4 機器接続手法における評価実験時の連携機器 47
6.5 評価シナリオの詳細 48
6.6 Distance Based Interaction を用いた機器連携手法における距離別の評価結果 . . 49
6.7 Distance Based Interaction を用いた機器連携手法における全体の評価結果 . . . 49

第 1 章

序論

本章では，はじめに本研究の背景，目的について述べ，ついで，本論文の構成について述べる．

1.1 本論文の背景

情報技術の発達により，ネットワーク接続性や計算能力を有した携帯端末や情報家電が，家庭やオフィスなど様々な場所へと普及している．例えば，Linux を搭載したフォトフレームや無線 LAN に対応した携帯電話やデジタルカメラなど，プリンタなどオフィス内機器だけでなく，家庭用の機器までもが進化を遂げている．また，空間内に複数のセンサや機器を埋め込み，利用者と機器間の相互接続を実現する知的空間である Smart Space が研究されてきた [36, 22, 25]．これにより，利用者は複数の機器を協調動作させ，単一の機器に比べ，音楽などの出力先を柔軟に切り替えたり，センサからのアラートを照明で知らせたり，などの高度なサービスを利用可能となった．

機器連携実現のための，機器が整備される一方で，これらを現実世界に適応させ，利用者が容易にサービスを構築可能にする必要がある．利用者がサービスを構築するためには，１）機器を連携可能な状態にする，２）連携が動作している状態にする，２つの手順がある．連携が動作している状態にするためには，利用者は何らかの手法で機器に命令を送る必要がある．しかし，一般的に用いられる手法は，IP アドレスやホスト名などの識別子を用いており，複雑な設定や操作が必要である．これに対して，手や視線などの動作により実世界の機器を直接指定し，命令を送る Real World Pointing を実現する研究がある [21, 17, 24]．複数の機器が協調動作する環境下では，利用者と機器との関係性が人の移動や，機器の追加や削除に伴い動的に変化する．そのため，独立した単体のインタラクション手法では，対応することができない．また，これらの手法は，人と機器との関係性が一対一である機器の操作に主眼が置かれている．一対多の関係性を持つ機器連携では，利用者と機器間のインタラクションだけではなく，機器と機器の間のインタラクションを利用者が命令する必要がある．

1.2 本論文の目的

本論文では，ネットワークやコンピュータに対する知識が無い利用者でも，容易にサービスを構築するための，マルチモーダルな機器特定手法と，サービス発見型の機器間命令手法を提案する．本手法では，Touch，Swing，Snap，Scan の複数のインタラクションから，利用者が状況に合わせて適切なインタラクションを選択し機器を特定し命令を送る．これに加え，複数のインタラクションを可能にするためのマルチインタラクションコードを提案する．また，複数の機器を選択する負担やサービスの発見を助けるために，一つの機器を選択すると，この機器と連携可能な機器を検索し，サービスの提案をする．以下に，本手法が実現する機能要件とアプローチをまとめる．

様々な空間にある機器の直接指定と操作

機器に命令を送るために，アドレスなどの識別子ではなく，人の動作などのインタラクションで機器自体を特定する研究が行われている．しかし，触れることで機器を指定する手法など，インタラクションが効果的である空間が限定的である．離れた場所や異なるネットワークにある機器も，

適切なインタラクションにより，少ない負荷で機器を特定することが求められる．本論文では，複数のインタラクションを一つのコントローラで動作させ，利用者がシームレスに手法を切り替えて，機器を選択する．

利用者の認知的負荷が少ない接続指示

利用者は，機器の操作には慣れているが，連携は複雑な設定や操作が必要であり，メンタルモデルが無い為に，利用の負荷が大きい．よって，認知的負荷を軽減させる，機器の接続手法が求められる．本論文では，利用者がサービスが実行される機器を一つ選択するだけで，機器に紐付けられたサービスを利用者に表示し，サービスの発見を援助する．

統一インタフェースによる柔軟性の高いサービス制御

ネットワーク接続性や計算能力を有した機器が家庭，公共空間，オフィスへと普及すると，利用可能なサービスの種類が増えてくる．サービスが多数存在しても，リモコンなど機器に固有のインタフェースでは，利用手法が機器毎に違うため，混乱を与えてしまう．よって，統一したインタフェースによりサービスを制御することが求められる．本論文では，携帯電話上で動作するコントローラにより，複数のサービスを制御する．

柔軟なサービス構築のためのミドルウェア

現在普及している機器連携は，テレビを中心とした家電連携など特定の環境下で，特定の機器同士が相互接続するため，サービスの数が限られてしまう．今後は，家電だけではなく，センサや携帯端末など異なる属性の機器が横断的に連携し，サービスを提供して行くと考えられる．これらのサービス構築を利用者が容易にできる一方で，柔軟なサービスを繋ぎ変えて行くためのバックエンドシステムが必要である．本論文では，機器非依存型のデバイス記述言語を用いて，機器が利用できるサービスを記述することで，異なる機器やメーカ間の連携も支援する．

1.3 本論文の構成

本論文は，本章を含め全７章から成る．第２章では，機器連携に関し考察を行い，本論文における機器連携を定義し，連携を容易に実現する手法における機能要件を整理する．第３章では，機能要件を達成するためのアプローチを延べ，利用者の負荷が少なく，かつ柔軟な連携サービスを実現する手法を議論する．そして，複数のインタラクションによる連携手法 Chameleon を提案する．第４章では，このアプローチを達成するための設計に関して説明し，ついで第５章では，プロトタイプ実装の概要を示し，実装の要点を述べる．第６章では，ユーザ評価実験を行い，システムの評価を行う．第７章では，本論文の課題と結論を示す．

第 2 章

機器連携サービス利用のためのインタラクション

本章では，はじめに機器連携を支援するための技術の発展をネットワーク，ミドルウェア，インタフェースの各視点に分けて述べる．ついで，目指すべき機器連携サービスのシナリオを述べる．次に，各層における要件を整理し，連携のためのインタフェースにおける問題点を定義する．

2.1 機器連携サービスを支援する基盤技術の進化

情報技術の進化に伴い，ネットワーク，ミドルウェアの機器連携支援技術が普及する一方で，利用者にとっては機器の連携が複雑で，設定や操作に時間や手間が掛かるなどユーザビリティの問題が未だに残っている．連携可能な機器が多様化し，サービスの種類や数も増え高度化し，サービスもあらゆる場所で発生する，今後のユビキタスコンピューティング環境では，容易に，かつ直感的にサービスを司るユーザインタフェースが求められる．本節では，機器連携支援技術をネットワーク，ミドルウェア，インタフェース層に分け，述べる．

2.1.1 ネットワーク層における通信支援

ネットワーク層の技術開発が，高い信頼性や拡張性などを備えた機器間の接続を実現している．Wi-Fi や Bluetooth，ZigBee に代表される無線通信規格やハードウェアの進化に相まって，計算機だけではなく，情報家電やオフィス機器，センサ，携帯電話などの機器間での通信が行われている．光ファイバー接続やワイヤレスブロードバンドの普及により通信インフラも整備されつつある．また，セキュアな通信を担保し，異なるネットワークセグメント間でデータのやりとりを実現する VPN などの技術により，世界中のあらゆる機器が，高速に，かつ安全に通信をする社会の到来を迎えている．

2.1.2 ミドルウェア層における機器連携支援

ミドルウェア層が進化により，各機器間の相互接続が可能になるだけではなく，ある機器が別の機器の機能を実行するなどの協調動作が可能になった．例えば，家電機器や PC では UPnP[16] や DLNA[13] などのガイドラインを定めて，機器間の連携を支援している．他にも，利用者が機器をネットワークに指すだけで，簡単に利用ができるような，Bonjour[11] や Jini[14] などもある．また，携帯電話や機器の追加に対応し，協調動作を支援するためのミドルウェアも研究が行われている [20, 45]．このようなミドルウェアを実際の環境に組み込んだものとして，Smart Space がある [36, 22]．各ミドルウェア間の相互連携やセキュリティなどの問題はあるが，人や状況，場所に合わせたサービスを利用者が享受できる社会も近づいている．

2.1.3 インタフェース層におけるサービス支援

上述した結果，異なる機器間，ネットワーク間での連携が可能であり，柔軟なサービスが利用できるが，私たちの身の回りには利用者が機器の連携による，高度なサービスを利用できないままである．これは，ネットワーク，ミドルウェア層で実現出来ている技術を実世界に導入出来ておらず，そのためサービスの自由度が阻まれ，限定的になっている．例えば，機器を操作するインタフェースは，いまだにリモコンが主流であり，利用者は機器毎に異なる操作方法を学習する必要がある．

実世界指向インタフェース [44, 46, 7] により機器を直感的に用いる研究が行われているが，操作が主体であり，連携したサービスをコントロールするものは少ない．共通のインタフェースで，家電連携によるサービスを利用するために，ミドルウェアにおける HDMI-CEC[28, 10, 15] を使い，テレビを頂点としたサービス利用モデルが提唱されているが，屋外やテレビ以外の機器同士の接続によるサービス利用にはまだ適応するできない．このように，インタフェースの進化が留まっており，利用者は静的で，空間的にもサービスの利用が制約されている．

2.2 シナリオ

本論文で目指す，機器連携によるサービスは，従来の単一の機器が単一の機能を有し，単一の目的のために動作するものではなく，様々な場所で，状況に応じて機器が必要に応じて動的に組み代わりながら生成されるものである．本節では，本論文が想定とするサービスをシナリオとして説明する．

シナリオ１：プリントサービスの利用

社会人の美樹は，恋人の恭兵と京都へ旅行に行った写真を寝室のデジタルフォトフレームに表示して楽しんでいる．美樹は，東福寺で２人で撮った写真がお気に入りで，手帳に入れて毎日持ち運ぶことにした．美樹は，デジタルフォトフレームの写真を携帯電話から選択し，印刷サービスを実行し，プリンタから印刷された写真を手帳に挟んだ．

シナリオ２：公共端末でのレコメンドサービスの利用

土曜日，美樹は久々の休日に鎌倉で紅葉を見に行くことにした．２人は，まず紅葉で有名な建長寺に行ったが，次に訪れる場所に悩んでいた．２人は，休憩のために入ったカフェにある公共端末を利用し，レコメンドサービスを受けることにした．恭平は，携帯電話から移動履歴を公共端末に転送し，好みや所要時間を入力すると，公共端末にお勧めとして浄妙寺が表示された．２人は，公共端末から目的地までのルートをダウンロードし，浄妙寺へ向かった．

シナリオ３：遠隔地からのファイル転送

月曜日，美樹は博多港への海上輸送コスト削減のためのプレゼンを１４時から行う予定である．朝のメールチェックを終え，最終確認するために資料を開いたが，最新版ではなかった．美樹は，プレゼンに備えて，前日家で作業をしていたため，ファイルを持ってくるのを忘れてしまった．美樹は携帯電話から，家にあるパソコンに保存されているファイルを，オフィスのパソコンへと転送し，配布資料を印刷した．

2.3 各層における連携のモデル化

上述したような，サービスを実現するためには，インタラクション層に求められる要件が存在する．本節では，ネットワーク層，ミドルウェア層における要件を整理することでインタラクション層の要件を定義する．本論文では，各層で指定，接続，利用の 3 つでモデル化した．本モデルによる分類を表 2.1 に示す．

表 2.1　ネットワーク，ミドルウェア，インタフェース層における連携

	指定・特定	接続	利用・制御
インタフェース	サービス，機器（直感的，容易な手法で）	可視化 or 不可視化	CUI，GUI，ボタン
ミドルウェア	機器の id，オブジェクト	Remote Function Call，input/output	メソッドの実行
ネットワーク	IP アドレス，Mac アドレス	パケット通信，Stream 通信によるデータ通信	

2.3.1 ネットワーク層における要件

ネットワーク層では，機器の接続や連携は，今日の TCP/IP のような信頼できる通信プロトコルに基づいたネットワーク指向で行われる．機器間の通信は，ビット列の伝達により実現される．機器の特定や指定は，IP アドレスや Mac アドレスなどの識別子により行われ，また，機器間の接続や利用は TCP/IP によるパケット通信やストリーミングにより動作する．ネットワーク層では，機器をドメインという単位で管理するために，異なるネットワーク間を繋ぐためには，ネットワークをブリッジするための VPN（Virtual Private Network）が利用される．

2.3.2 ミドルウェア層における要件

ミドルウェア層では，機器をオブジェクトと捉えるオブジェクト指向による連携が実施される．つまり，機器をオブジェクトとモデル化し，連携サービスを機器メソッドの実行形式から成る連携サービスシナリオと定義される．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 ID やオブジェクトとして捉えることが多い．また，UPnP などのミドルウェアでは，ネットワーク上の機器やコンテンツを検索する SSDP（Simple Service Discovery Protocol）や CDS（Content Directory Service）などの機能を備えている．機器間の接続は RFC（Remote Function Call）などが利用される．加えて，オブジェクト間の接続を Input/Output として捉えることが多い．つまり，ある機器のリソース（コンテンツなど）を Input とし，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である．本モデルを図 2.1 に示す．

2.3.3 インタフェース層における要件

ネットワーク，ミドルウェア層では，サービス利用をネットワークやオブジェクト指向でモデル化されると述べてきた．インタフェース層では，これらと異なり，サービス指向であるべきである．

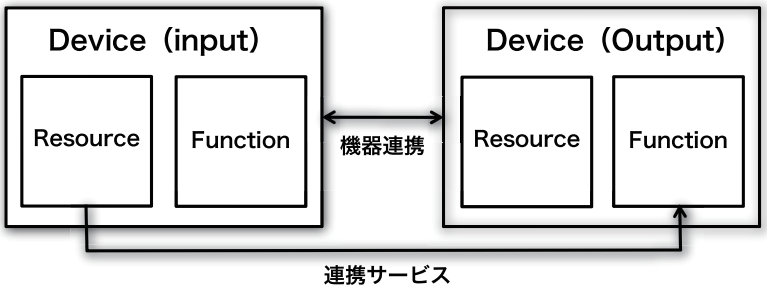


図 2.1　ミドルウェア層での連携モデル

つまり，利用者がサービスを利用したいと考えたときに，ネットワークや機器などのオブジェクトを意識することなく，利用できることが望ましい．そのためには，ネットワーク，ミドルウェアと同じく，指定，接続，利用を容易に，直感的に行えることが必要である．よって，IP アドレスやオブジェクトを指定するのではなく，直感的な手法で写真などのコンテンツや機器を指定し，接続を可視化もしくは非可視化により管理し，統一された操作インタフェースで利用するべきである．

2.4　機器連携におけるユーザインタフェース

本節では，前節の考察を受けて，機器連携におけるユーザインタフェースを考察する．機器が協調動作し，サービスを提供するには利用者にとって，導入と利用の２つが必要である．導入とは，ミドルウェア層で実現している技術をインタラクション層でも実現可能にすることであり，表 2.1 における縦の遷移である．利用とは，インタラクション層の中で，サービスが実行されるまでの動きであり，表 2.1 では横の遷移と示される．以下に，詳述する．

導入におけるインタフェース

利用者が連携したサービスを利用するためには，まず機器を直感的かつ容易な手法で，サービスを利用できる状態にすることが求められる．ネットワーク層で実現している技術をミドルウェア層でも実現するためには，ドライバやソフトウェアのインストールにより，機器でミドルウェアを動作させることが必要である．

導入におけるインタフェースとして，家電へのセンサノード取り付けを支援する Sensor Installation Kit[4] や，日常生活に使うモノとセンサをペアリングするための Spot & Snap がある [43]．しかし，これらはモノや家電のモニタリングが主な目的であり，連携のためのミドルウェアを支援するものではない．私たちは，今までに連携のためのミドルウェアと実世界のオブジェクトである機器をマッピングするために，Easy Device Registration of Out-Of-Box Experience[9] を提案し，この問題の解決に取り組んできた．Easy Device Registration of Out-Of-Box Experience を図 2.2 に示す．利用者は，新しい機器をネットワークに接続すると，アドレスを割り振るのと同時に機器のマシン名や情報をもとに，各機器で行える機能と情報をローカルサーバに登録する．そし

て，インタラクションを行うためのコードを自動的に印刷する．利用者は，このコードを機器に貼ることで，ミドルウェアと機器のマッピングを特別な知識なく行うことができる．

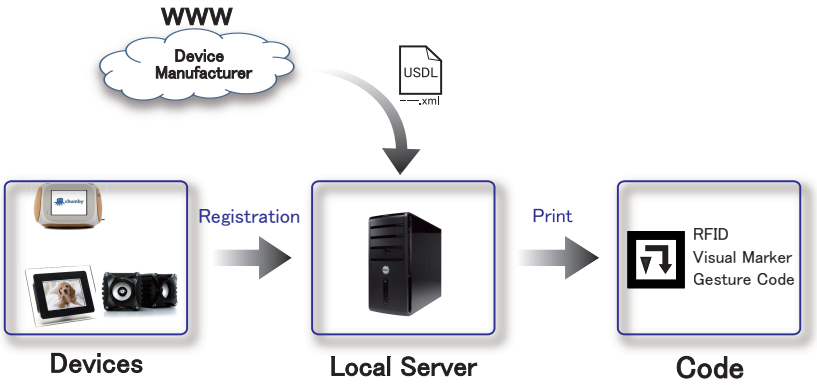


図 2.2　Out-of-Box Experience For Easy Device Registration

利用におけるインタフェース

導入により，機器とのインタラクションが可能，かつミドルウェアが動作する状態になった後，利用を行う．利用は，表 2.1 で述べた，指定，接続，利用・制御までのフェーズを遷移することである．サービスの利用には，サービスの明示的利用と暗示的利用がある [27]．前者は，機器や環境に対して，何らかの主体的操作を行い，サービスを取得する．後者は，コンテクストアウェアに代表される，GPS や RFID により利用者の位置を特定したり，温度や加速度などのセンサで利用者の活動状態を検知し，サービスを提供する手法である．

両者が利用者の好みや状況に応じて，相互に動作することが望ましいが，きめ細かなサービスを利用するには明示的手法が必要である．暗示的手法は，機器が増加し，至る所に存在する空間では，利用者の意図を全て汲み取ることはできない．よって，意識的に行う負荷が少なく，明示的にサービスを利用するシステムの構築が求められる．

ここで問題となるのが，利用者と機器間のインタラクション，連携する複数の機器間のインタラクション，利用者とサービスのインタラクションの方法と負荷である．この関係性を図 2.3 に示す．第 2.3 節で述べた，各層における指定，接続，利用が，Identify，Connect，Control に当たる．

2.5　問題点

上述してきたように，機器連携サービスを，柔軟に組み換え利用するためには，インタフェース層での指定，接続，利用を容易に，直感的に出来なければならない．しかし，人の移動や新しい機器の追加・削除により，人と機器との関係性が今後動的に変化して行くことが予見される．また，これにより動的にサービスが生成され利用されるようになる．現在までに，多くのインタフェース層での連携をハンドリングする研究が存在するが，この問題により，十分なものではない．以下

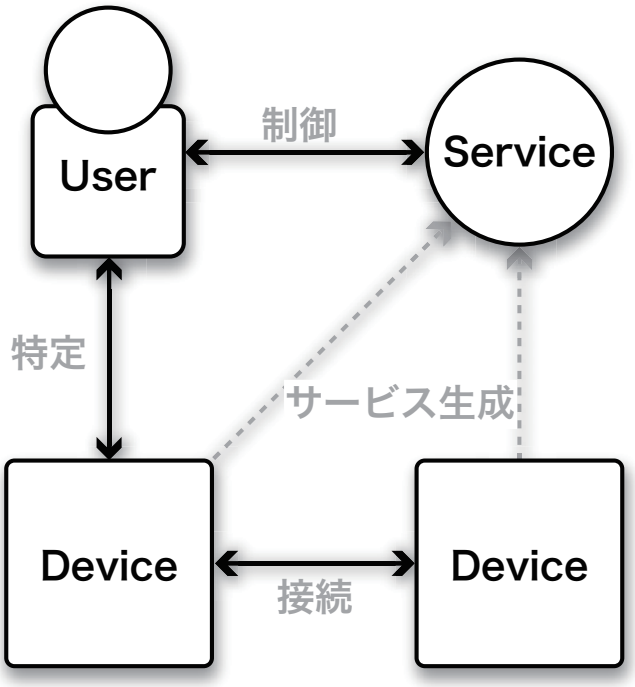


図 2.3　利用者と機器，サービス間のインタラクション

に，指定，接続，制御の各フェーズにおいて，どのような問題点があるかを述べる．

2.5.1　動的な環境下での対象物指定の困難性

サービスを利用するためには，利用者と機器がインタラクションを行い，機器を特定することが求められる．指定には，人と機器間での関係性が変化するために，対象物を容易で，直感的な特定が困難という問題点がある．CUI や GUI により，IP アドレスや Mac アドレスなどの識別子を特定する手法は，識別子を記憶したり，初めて見る機器の識別子を知る必要があるため，利用者の負担が大きい．これに対して，携帯端末や環境側のセンサを用いて，人の動作により機器を直感的に特定する手法がある [41, 42, 23]．しかし，対象物が近い距離にあるなどの限定された環境下では有効であるが，動的環境下では異なるネットワークにある機器や見えない場所にある機器などが対象物となる．

2.5.2　機器同士の接続操作における利用者のメンタルモデルの欠如

サービス享受のためには，機器間でのインタラクションである接続操作が必要である．この操作では，利用者の連携実行のためのメンタルモデルの欠如が問題である．メンタルモデルとは，ある操作を実行する際に，利用者の心の中で決定される操作のモデルである [34]．利用者は，機器を連携させたサービスの利用を，あまり経験していないため，連携のためのメンタルモデルを持ってい

ない．機器の操作は，1チャンネルのテレビを見るというサービスを利用するためには，リモコンで電源ボタンを押し，1チャンネルのボタンを押すという動作のモデルが有るため，容易である．さらに，動的環境下では，家にある機器同士だけではなく，他者の所有物や異なるネットワークにある機器なども繋ぎサービスを利用する可能性がある．

2.5.3 統一的サービス利用インタフェースの欠落

容易にサービスが動作する機器を指定し，接続ができても，利用するためのインタフェースがサービス毎に異なると，利用者に混乱を与える．サービスを利用するためのインタフェースとして，リモコンが一般的に利用されているが，利用できる機器が固定されており，また録画や再生などの機能は同じだが，メーカによるインタフェースが異なるなど，統一的なインタフェースが存在していない．よって，連携を実現するためには操作に関しても，統一的なインタフェースで，容易に直感的な利用が行えることが求められる．

2.5.4 機器連携サービスの柔軟性と認知的負荷のトレードオフ

様々な機器が様々な機能を持つと，利用できるサービスも多様化して行く．しかし，このサービスの多様性は利用者の認知的負荷に影響を与えてしまう．HDMI-CEC による家電メーカが提供する連携サービスは，どのサービスを利用するにしても，テレビに対して操作命令を送ればいいため，利用者の認知的負荷は低い．しかし，固定された機器間同士の接続しか許容していないために，サービスの柔軟性に欠けている．

2.6 機能要件

本節では，前章までに述べた問題点から，利用者が誰でも，どこでも，機器連携サービスを容易に実現するためのインタフェースの機能要件を述べる．本論文における機能要件として，様々な空間に存在する機器の直接指定と操作，利用者の認知的負荷が少ない接続指示，統一インタフェースによる柔軟性の高いサービス制御，柔軟なサービス作成のためのミドルウェアを挙げる．以下に詳述する．

2.6.1 様々な空間に存在する機器の直接指定と操作

機器の追加や人の移動が頻繁に起こる環境下でも，家や公共空間，オフィスなど様々な空間に存在する機器をアドレスなどの識別子を意識することなく指定や操作ができることが求められる．GUI におけるインタフェースデザインとして，See-and-Point や Direct Manipulation の概念がよく用いられる [31, 40]．これは，コマンドや利用方法を記憶するのではなく，認識することで利用するものである．この概念を実世界に拡張し，機器の指定を容易にする必要がある．

2.6.2 利用者の認知的負荷が少ない接続指示

接続指示は，利用者から明示的に指示する手法と，システムが自動的に接続を確立する方法がある．前者は，意識的に指示する必要があるため，利用者の負担が増えるが，利用者が機器を組み換えサービスを構築したり，アクセス制限などが行いやすい．後者は，指示の必要がないが，静的に機器が連携するため，機器の組み換えが困難であり，利用者の意図に沿わない接続が行われる可能性がある．よって，本論文では前者を採用し，認知的負荷の少ない接続指示手法が必要である．

2.6.3 統一インタフェースによる柔軟性の高いサービス制御

複数利用可能なサービスが存在していても，サービスを制御するためのコントローラが複数あったり，サービス利用手法が異なると利用者に混乱を与える．よって，統一されたインタフェースで，複数のサービスが利用可能であることが求められる．加えて，インタフェースを統一することで，利用可能なサービスを限定するのではなく，拡張性を確保する必要がある．

2.6.4 柔軟なサービス作成のためのミドルウェア

柔軟なサービスを実現するためには，各機器の機能やコンテンツを記述し，記述した内容をもとにサービスを生成するためのミドルウェアによる支援が必要である．このミドルウェアは，同一のメーカ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，運用されることが求められる．

2.7 まとめ

本章では，はじめに機器連携の考察をネットワーク，ミドルウェア，インタフェースに分けて行い，目指すべきサービスをシナリオとして述べた．その後，機器連携のサービス利用での問題点として，1）動的な環境下での対象物指定の困難性，2）機器同士の接続操作における利用者のメンタルモデルの欠如，3）統一的サービス利用インタフェースの欠落，4）機器連携サービスの柔軟性と認知的負荷のトレードオフを定義した．最後に，問題点をもとに，機能要件として，1）様々な空間に存在する機器の直接指定と操作，2）利用者の認知的負荷が少ない接続指示，3）統一インタフェースによる柔軟性の高いサービス制御，4）柔軟なサービス作成のためのミドルウェアを導いた．

第 3 章

マルチモーダルなインタラクションによる機器連携手法　Chameleon

本章では、はじめに本システムの概要を述べ，ついで機能要件を達成するためのアプローチとして，複数のインタラクションによる連携サービス利用システム Chameleon を提案する．最後に，本論文で挙げたアプローチをもとに，先行研究を整理し，本研究の意義を示す．

3.1　システム概要

本論文では，連携サービスをネットワークやコンピュータに特別な知識が無い人でも，容易に利用するためのマルチモーダルな連携対象指定手法と，利用者の理解が容易な連携モデルに基づいて設計されたサービス利用手法 Chameleon を提案する．Chameleon は，利用者と対象機器の関係性に基づき，Touch, Swing, Snap, Scan という複数の機器選択手法を提案する．利用者は，対象物との距離や自分のアクティビティの状態によって，その場にあった適切なインタラクションを選ぶことで容易な機器選択を実現する．また，1 つの機器を選択すると，システム側で利用可能なサービスを提示することで，空間内で行えるサービスの発見を手助けし，またサービス利用までのステップ数の減少につなげ認知的負荷を下げる．また，複数のインタラクション手法を可能にするための，機器に選択可能なインタラクションコードを開発した．図 3.1 に Chameleon の画面遷移を示す．Chameleon は，メイン画面，機器選択画面，機器制御画面から構成される．

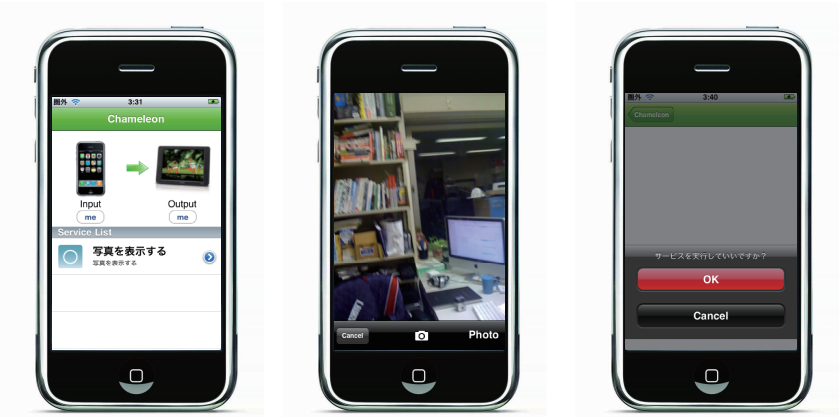


図 3.1　Chameleon の画面遷移

3.2　機器との距離に応じた機器指定手法

機器に連携の命令を送るための指定を容易に行う手法として，本論文では対象物との距離に応じてシームレスに複数の最適なインタラクションを切り替える Distance Based Interaction を提案する．本節では，はじめに本手法の概要と各インタラクションの説明を行い，次いでシームレスに切り替える手法について述べる．

3.2.1 Distance Based Interaction Method

動的に人と機器との関係性が変化するユビキタスコンピューティング環境では，単一のインタラクションだけであらゆる状況に対応することは不可能である．第 2.5.1 項で述べたように，各インタラクションには不得意な状況が存在する．よって，利用者の好みやコンテクストに合わせ，複数のインタラクションから適した手法を選択することが利用者の快適性や利便性につながると考えた．

コンテクストには，手が塞がっているや走っているなどの身体的，人混みや電車の中などの社会的など複数ある．この中で，人々が普遍的に影響を受ける距離に着目をし，インタラクションを整理した．また，対象物との距離が離れると視覚的に確認出来無いため，直感性が失われてしまう．

よって，物理的距離と認知的距離（知っている，知らない）といった関係性をもとに Touchable, Visible, Recallable, Scannable の 4 つに距離を整理し，Distance-based Interaction Method として提案する．そして，携帯端末のセンサや画面などの機能と人の振る舞いをもとに，インタラクション手法の組み合わせを設計した．図 3.2 にインタラクションの分類を示し，以下に各インタラクションの概要を詳述する．

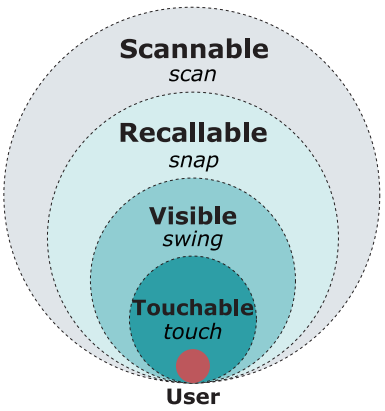


図 3.2　Distance-based Interaction Method

Touch Interaction for Touchable Distance

Touchable な距離とは，人が手を広げた範囲内を指し，対象物との距離が最も近い．例えば，椅子に座って作業中に机の上にあるフォトフレームを選択する距離である．この状況下では，対象物を目視できるため指定の直感性が十分に担保されている．よって，携帯端末の画面を確認すること無く素早い選択ができることが望ましい．

よって，私たちは Touchable な距離に応じたインタラクションとして touch Interaction を提案

する．対象物と人が近接なインタラクションを行う動作として，Touch は一般的である．例えば，IC カードを用いて駅の改札を Touch し，通過をするシステムが日本では普及している．よって，携帯端末に RFID リーダを接続し，機器にかざすことで指定を行う Touch Based Interaction を提案する．

Swing Interaction for Visible Distance

Visible な距離とは，同じ部屋にある機器など目に見える範囲にある対象物がある距離を示す．例えば，椅子から 5 メートル先のプリンタを指定するなどの状況である．この距離は，機器への接近などの身体的努力が少ないインタラクションが望ましい．本論文では，矢印が描かれてあるコードどおりに利用者が携帯端末を振ることで指定を行う Swing インタラクションを提案する．

この距離では，カメラや指差し動作など様々な手法がある．しかし，振る動作は利用者の活発な動作に対応できる．例えば，カメラに比べて，一度矢印の方向を認識すれば見続ける必要がなく，着座やくつろいでいる体勢でも指定が可能である．また，ビジュアルマーカなどに比べ，距離や明るさに影響を受けない．

Snap Interaction for Recallable Distance

Recallable な距離は，利用者が対象物を目視や接触出来ないが，機器の存在を認識できる認知的距離である．オフィスや家にある機器を思い出し特定する状況がこの距離に当てはまる．この距離は，対象物が見えない時があるため，指定の直感性を補完しなければならない．

この距離に適したインタラクションとして，写真から機器を選択する手法を提案する．この手法は，IP アドレスなどの情報が紐付けられたビジュアルマーカと一緒に撮影された写真を選ぶ方法である．利用者は使う可能性がある機器をあらかじめ撮影し，機器選択時には写真ライブラリから対象物が撮影されている機器を選ぶ．これにより，アドレスや機器の場所を覚える必要がなく，また遠隔地にある機器も指定が可能である．写真を見て機器を視覚的に確認しながら選択することで，指定の直感性を補える．

Scan Interaction for Scannable Distance

Scannable な距離は，上記の 3 つに該当しない利用したことがなく，存在を知らないが検索できる状況である．機器の検索を行う方法として，Bluetooth スキャンがよく使われている．この手法を Distance Based Interaction にも適用し，初めて出かけた場所やどこかにあるはずの機器を探す状況で適した手法として提案する．本手法は，同じネットワーク内の機器を検索し，携帯端末上で一覧表示された中から機器の名前を選択する手法である．ローカルサーバが機器情報を一元管理し，携帯端末から問い合せることで実現する．

3.2.2 シームレスなインタラクションの自動的切り替え

本研究では，複数のインタラクションを提案するだけではなく，これらをシームレスに切り替える手法を提案する．これにより，コントローラが利用者の意図を汲み取ることで，対象物を素早く，かつ容易に選択することができる．これを実現するために，利用者の意図を予見したインタラクションの切り替えとマルチインタラクションコードを提案する．

利用者の意図を予想したインタラクションの切り替え

距離に応じて，各インタラクションを用意するだけではなく，このインタラクションをユーザが意識すること無く切り替えが可能であることが望ましい．つまり，ユーザが利用したいサービスがあり，機器を指定する時，ユーザは特定の機器を指定したいという欲求にのみ関心があり，どのインタラクションを利用したという願望は無い．

よって，利用者の意図をもとに行った動きを検出し，インタラクション手法を遷移させる．この遷移のトリガは，各手法に対応している．この遷移とトリガを図 3.3 に示す．例えば，振って機器を指定する時はメイン画面を振ると Swing インタラクションが自動的に開始され，触れて指定したい時は機器を触る動作で Touch インタラクションが開始される．

マルチインタラクションコード

シームレスに機器を選択するため，Collaboration-Ready の機器にコードを貼り付ける手法を提案する．これにより，Collaboration-Ready の機器を利用者が一目で認識できる．図 3.4 にこのマルチインタラクションコードを示す．これは，ARToolkit が描かれているビジュアルマーカであり，この四角い枠の中に矢印が描かれている．そして，矢印の下に利用者がコードを貼るための向きが描かれている．コードの裏に RFID タグが貼られ，RFID リーダが利用可能である．これにより，コード一枚だけで，振る，写真を撮る/選ぶ，触るなどの複数のインタラクションが可能になる．また，四方向の矢印を組み合わせることで，上下左右のサンプル加速度データだけで複数のコードが生成可能であり，管理が容易である．また，RFID タグプリンタからコードを印刷すれば，容易に発行ができる．

3.3 サービス発見型機器接続手法

第 2.6 節で述べた機能要件のうち，利用者の認知的負荷が少ない接続指示を達成するため，本論文ではサービス発見型機器接続手法を提案する．第 3.2 節で述べた手法により，機器選択が容易になったとしても，連携サービス発見や利用までに多くの手順を踏んだり，操作が困難であれば利用者に負荷を与えてしまう．よって，本手法は機器選択の手間を減らし，かつ容易にサービスを発見する．以下に具体的なアプローチを述べる．

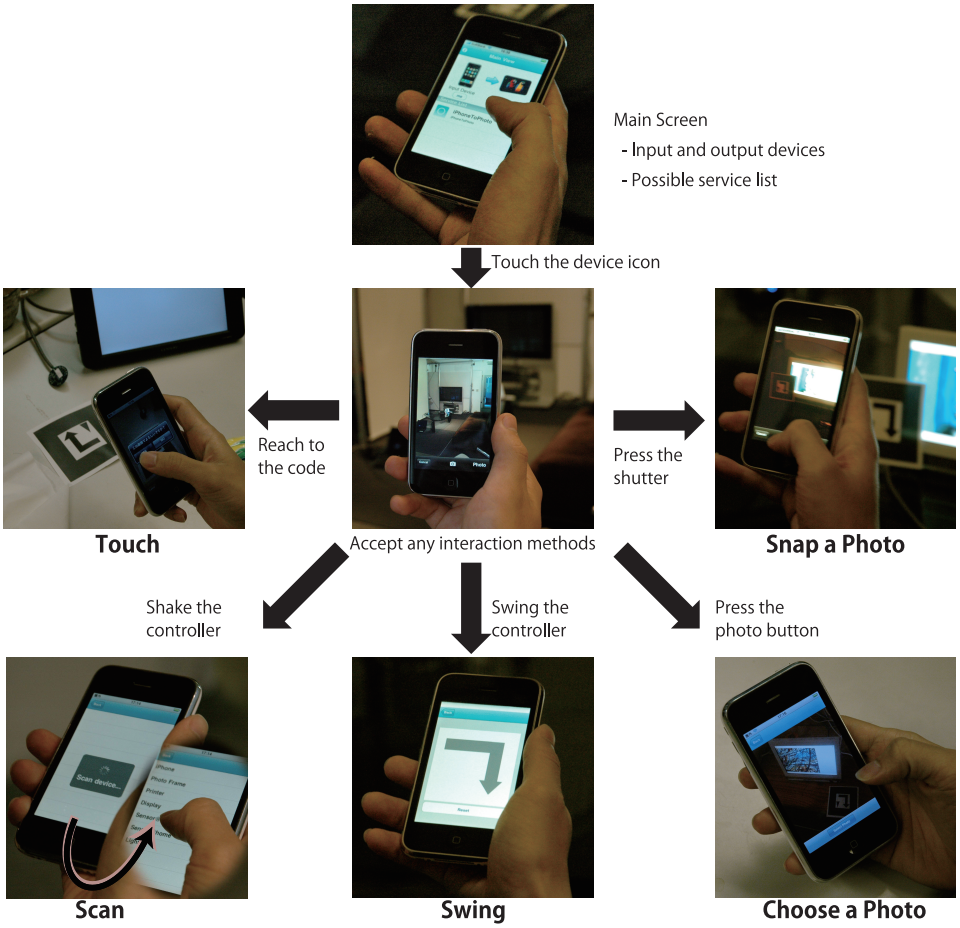


図 3.3　インタラクションの利用者の動きによる動的切り替え

3.3.1　連携状態閲覧型操作インタフェース

ブラビアなど家庭に普及している機器連携を行う手法は，接続や連携している機器をブラックボックス化し，利用者に見せないことで使用時の混乱を与えないようにしている．また，連携サービス自体も非可視化しており，どの連携サービスが使えるのかは利用者が想像する必要がある．この手法は，利用者が機器連携の仕組みを理解出来ないため，連携中の機器を組み替えが困難であるため柔軟な連携サービスを実現するためには適していない．Chameleon は，利用者に対して接続中の機器と利用可能なサービスを表示することで，容易に連携中の機器を組み替えたり現在接続中の機器が理解できる．Chameleon の操作インタフェースを図 3.5 に示す．

操作インタフェースは，上部に入力側と出力側の機器が表示され，下に入出力の機器間で生成可能なサービスが表示されている．利用者は，サービスを変更したいときには，機器のアイコンをクリックすると，機器選択の画面へと移行する．

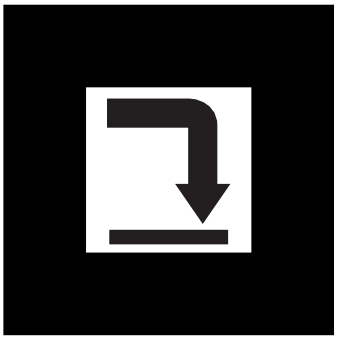


図 3.4　マルチインタラクションコード

図 3.5　操作インタフェース

3.3.2　サービス発見型機器選択手法

連携サービスの実行までには，１）連携対象機器の特定，２）接続，３）連携サービスの選択，４）連携サービスの利用までを利用者が行う必要がある．本項では，実行までの手順数と利用者の認知的負荷の観点から，サービス利用手法を比較し，本手法として機器選択型サービス利用手法を述べる．サービス発見型機器選択手法は，サービスが紐付けられた機器を利用者が選択することでサービスの発見，利用をするものである．

コンピュータの進化に合わせて，サービスを利用するインタフェースも変化している．この変化におけるパラダイムとして，CUI (Character User Interface)，GUI (Graphical User Interface)，Real World Interface がある．

CUI は，アプリケーションをはじめに指定しサービスの利用を行う．例えば，ファイル編集をするには，*vi sample.txt* のようにアプリケーションの後にコンテンツを明記する．これは，あらかじめファイルに紐付けられたアプリケーションを知る必要がある．これをサービス発火型と呼ぶ．これに対して，**GUI** はマルチモーダルな利用手法ができ，アプリケーションを起動しコンテンツを選ぶ方法と，コンテンツを選び自動的に紐付けられたアプリケーションが動作する手法がある．後者をコンテンツ発火型と呼び，GUI はサービス発火型とコンテンツ発火型が可能である．

Real World Interface でも，GUI と同じくサービス発火型とコンテンツ発火型が考えられる．つまり，連携サービスを選択する方法と，コンテンツである機器を選択する方法である．この観点から，Real World Interface における連携サービスの利用手法を考察する．

サービス発火型サービス利用手法

サービス選択型は，サービス発火型であり利用時にサービスをまず選択する手法である．例えば，部屋にいる時にコントローラに，写真表示サービスや音楽再生サービスが表示され，利用者が選択することでサービスが開始される．この実現のためには，システムが空間内にある機器をもとに連携サービスをあらかじめ生成し，利用者に対して提供を行う．本手法は，連携サービスを利用したいと考えたときに，すぐサービスを選択すればいいため，利用者の認知的負荷が少ない．しかし，２，３種類の機器であれば考えられるサービスが限られているが，機器の数が多くなるとシステムの計算量が多くなってしまう．また，数多くのサービスが表示されるために，サービス選択時の認知的負荷が多くなる．

コンテンツ発火型サービス利用手法

コンテンツ発火型サービス利用手法は，コンテンツである機器を選択し，その後サービスを利用する手法である．例えば，フォトフレームに携帯端末に保存されている写真を表示するときは，フォトフレームと携帯端末の２つを選択し，その後写真表示サービスを利用する．本手法は，利用者が利用したいサービスを思い描いた後に，サービスを生成可能な２つの機器を選択する必要があるため，利用者の手間が増える．また，利用者は連携サービスの認知モデルが出来上がっていないために，サービスを利用したいにも関わらず，機器をあらかじめ選択する必要があり混乱を与える可能性がある．

集中管理型機器選択サービス利用手法

上記に挙げた以外の手法として，集中管理型機器選択サービス利用手法がある．本手法は，機器間接続を一つの機器が集中管理し，この管理している機器に対して利用者が連携指示を送り，サービスを利用する．現在多くの家電メーカーが取り組んでいる手法であり，集中管理する機器として一般的にテレビが用いられている．どのサービスを利用するにしても，テレビに対して命令を送れば良いため，認知的負荷は低くなるが，その一方でサービスの柔軟性が欠如する．例えば，携帯端末とオーディオ機器により音楽利用サービスを実行するには，複雑な設定が必要となってしまう．

サービス発見型機器選択手法

本論文では，上記に述べた利用手法の利点と欠点を考慮した上で，サービス発見型機器選択手法を提案する．本手法は，サービス発火型とコンテンツ発火型を合わせたものである．具体的には，ある特定の機器を 1 つ選択すると，選択された機器と空間内にある機器により生成出来る連携サービスをシステムが計算し，提案するものである．つまり，サービス発火型における欠点である計算量と膨大なサービス量に対して，1 つの機器を選択するという制限を設けることで，システム側の負担を減らし，かつ利用者の認知的負荷を下げる手法である．例えば，メディアサーバに保存されている写真をプリンタから印刷するためには，プリンタを選ぶ．プリンタが選択されると，それに伴ない入力の機器を検索し，自動的にサービスのリストを表示することで，サービスを容易に実行することができる．

3.4 統一インタフェースによるサービス制御手法

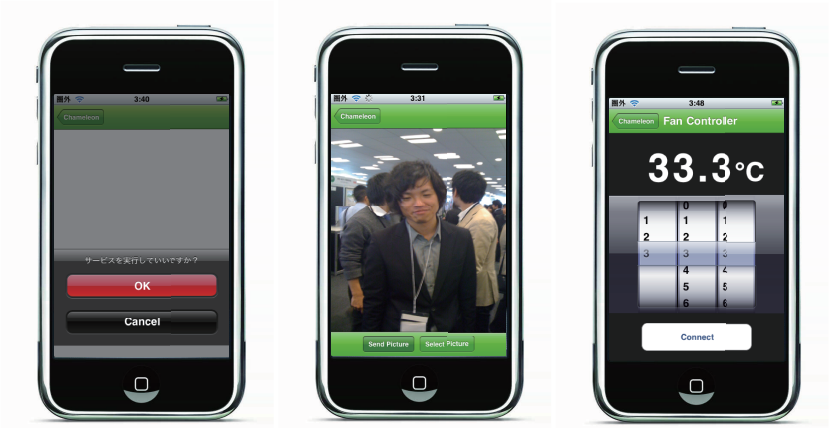
サービスを制御するには，ハードウェアにはパソコンやリモコン，ソフトウェアとして CUI や GUI など様々ある．しかし，サービスによりインタフェースが異なると利用可能なサービスを減少させてしまう．Chameleon は，複数のインタラクションを動作させるための計算能力とセンサを兼ね備えているシステム的な観点と，利用者が日常的に持ち運び，特別なデバイスを持つ必要が無い利用者側の観点から携帯端末上で動作するアプリケーションとして，インタフェースを設定した．加えて，異なるサービスを一つの画面上から動作可能にした．

連携サービスには，a) データ移動，b) 遠隔制御，c) 機器間連携の 3 種類が考えられる [30]．データ移動は，ある機器から画像や音楽などのデータを異なる機器への移動（再生），遠隔制御は携帯端末からパソコンの操作などの異なる機器による制御，機器間連携は，温度センサが 20 度以下になればエアコンを動作させるなどである．本論文では，この分類に基づき，b) 遠隔制御を除き，サービスの雛形であるコントローラを作成した．

次節で記述する各機器で定義される，デバイス記述言語の内容に合わせて各コントローラを切り替えることで，同じ機能を有する異なるメーカの機器にも対応出来る．a) データ移動には，固定されたデータの送信と，データを選択し送信する，2 つがある．例えば，前者はディスプレイに表示している写真を印刷するなどであり，後者は，携帯端末に入っている写真を選び印刷するなどのシナリオが考えられる．前者のコントローラを図 3.6a，後者を図 3.6b に示す．また，機器間連携におけるコントローラの例を図 3.6c に示す．

3.5 機器非依存型サービス記述手法

柔軟な連携サービスを実現するためには，異なるメーカや製品を共通のプラットフォームで動作させる必要がある．家電メーカが現在提案している機器連携は，メーカ独自の仕様に基づいているため，異なるメーカ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



a) データ移動型コントローラ（フォトフレームの写真を印刷）　b) データ移動型コントローラ（写真の転送）　c) 機器間通信型コントローラ（センサと扇風機）

図 3.6　サービス利用コントローラの雛形例

サービス記述手法を用いる．この手法として，徳田研究室が研究を行っている USDL（Universal Discription Language）[12] を拡張し，デバイス記述 XML として利用した．デバイス記述 XML を図 3.7 に示す．デバイス記述 XML には，機器の ID や名称と，機器の Resource，Function が記述されている．Resource，Function は第 2.3.2 項で述べたものである．図の例では，フォトフレームに対して，Resouce には output タグとして対応する image/jpeg が記述され，Function として対応する input タグに image/jpeg が記述されている．連携サービスは Input 側の Resource と，Output 側の Function をマッチングさせ，マッチングできれば連携サービスが利用できると判定される．表 3.1 に記述例を示す．このデバイス記述 XML は，機器を管理するサーバが保持する想定である．

$< device >$ ··· 機器の情報全体を表すタグ．他の全てのタグを子要素に持つ．
$< property >$ ··· 機器のプロパティが記述されるタグ．子要素として，機器の名前，アイコン，機器の詳細が記述される．
$< resource >$ ··· 機器が持つリソースが記述されるタグ．子要素として，リソースの id，名前，output，mode が記述される．
$< function >$ ··· 機器の機能が記述されるタグ．子要素として，ファンクションの id，名前，input，mode が記述される．
$< output >$ ··· リソースに対応する出力先が記述されるタグ．$< resource >$ タグの子要素として，記述される．
$< input >$ ··· ファンクションに対応する入力先が記述されるタグ．$< function >$ タグの子要素として，記述される．
$< mode >$ ··· 機器間の通信形式が記述されるタグ．$< resource >$，$< function >$ タグの子要

素として，記述される．

```
<device>
 <property>
  <id>photoframe</id>
  <name>Naoya's Photoframe</name>
  <icon>photoframe.gif</icon>
  <description>Naoya's Photoframe</description>
 </property>

 <resources>
  <resource id="current-picture">
   <name>写真</name>
   <output>image/jpeg</output>
   <mode>packet</mode>
  </resource>
 </resources>

 <functions>
  <function id="display-picture">
   <name>写真を表示する</name>
   <input>image/jpeg</input>
   <mode>packet</mode>
  </function>
 </functions>
</device>
```

図 3.7　デバイス記述 XML

表 3.1　デバイス記述例

機器名	Resource（input）	Function（output）
iPhone	music, image/jpeg	
PhotoFrame	image/jpeg	image/jpeg
Printer		image/jpeg
Sensor	Context	
Light		Context

3.6　関連研究

本節では，連携サービスを利用するための研究として，ミドルウェア，機器指定手法，機器連携手法の３つの観点から関連研究を述べ比較をする．

3.6.1 連携ミドルウェア

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[16] や DLNA[13] が現在普及しつつある．また，Bluetooth 機器と UPnP 機器など異なる通信プラットフォーム間においてもシームレスに連携させる技術として，uMiddle[29] が提案されている．このように，スマート環境構築の為に，多くの基礎技術が整備されつつある．しかし，これらは主にコンピュータや家電の世界で開発されているが，実空間との結びつきについては議論されていない．そのため，どのようにこれらのミドルウェアをインタフェースから利用して行くかについて議論を進めて行く必要がある．

3.6.2 機器指定手法

現在，人と機器のインタラクションを実現する様々な手法が提案されている．以下に詳述する．

現在，人と機器とのインタラクションとして，携帯端末や人の動作で指定や制御を実現する技術がある．以下に詳述する．携帯端末により周辺環境と連携する研究が行われている [8][3]. UbiCity[35] では，街に設置されたパブリックディスプレイを携帯端末により，情報の取得や操作を行っている．この携帯端末を使い，実世界の身体動作により利用や指定手法も様々提案されている．RFID を利用した Touch インタラクションとして，Riekki[38] は携帯端末をタグにタッチし，サービスを実行するトリガとしている．u-Photo[41] は，実空間のセンサや機器を直感的に利用するために，写真のメタファを利用している．ビジュアルマーカを貼った機器を撮影し，機器の情報を写真に保存し，写真上から機器操作を可能にする．Snappy[18] は，利用者の振る動作により，機器の連携を支援している．矢印が描かれた SRCode 通りに携帯端末を振り，機器を指定する．これらの手法は，特定の環境では効果的であるが，ネットワーク越しの機器や柔軟なサービス制御までは行われていない．

Enrico[39] は身体的なモバイルインタラクションの技術として，Touching, Pointing, Scanning の 3 つの手法を比較している．比較の結果，利用者は位置や行動状態，動機がインタラクションの選択に影響を与えると述べている．しかし，この結果をもとにインタラクションをどう共存させるかや，機器の操作までしか行われていない．. Chameleon は，複数のインタラクションを統合し一つの環境で動作することで，各インタラクションの短所を補完しあうことができる．

3.6.3 機器連携支援インタラクション

以下に機器連携を実現する研究を示す．

Pick-and-Drop

Pick-and-Drop[37] は，コンピュータ間での情報移動を実世界のペンにより実現している．Pick-and-Drop を図 3.8 に示す．利用者は，GUI における Drag-and-Drop のように，デスクトップや PDA などの端末に表示されているファイルなどをペンにより掴み，目的の機器に落とすことで

ファイルの移動ができる．会議でのファイル移動やデータを大きなディスプレイに表示するときに特に有効である．しかし，遠隔の機器への移動や，ファイル移動などにサービスが限られており，柔軟な連携サービスは実現することが出来ない．

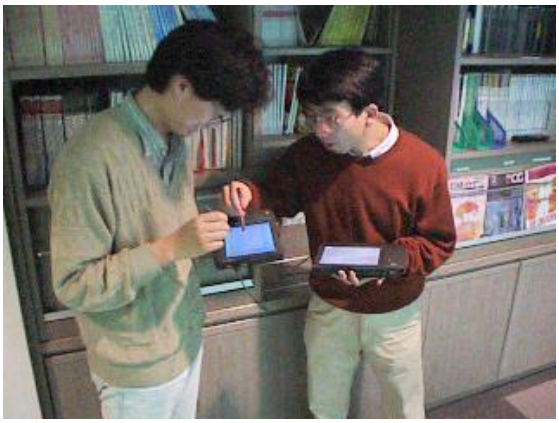


図 3.8　Pick-and-Drop

Touch-and-Connect

Touch-and-Connect[19] は，接続をしたい機器のボタンを押し，直接機器に触れることで，機器間の接続を実現している．Touch-and-Connect を図 3.9 に示す．各機器には，接続元を示すプラグボタンと接続先を示すソケットボタンが設置されている．利用者は接続したい機器のプラグボタンをまず押し，次に接続先のソケットボタンを押すことで機器が接続される．また，状態を表示可能なボタンを使うことで，利用者の排他制御し，複数利用者においても誤接続を防ぐことができる．Touch-and-Connect は，接続状態が可視化でき，かつボタンを押すだけで連携が実現できるため，利用者にとって理解が容易である．しかし，連携可能な状態にするまでの設定や，あらゆる機器に対してボタンを設置することは困難である．

tranStick

tranStick[2] は，メモリースティックなどの「鍵情報」が保存された物理的なストレージメディアを，機器に挿し込むことで仮想的なネットワークを構築する手法である．tranStick を図 3.10 に示す．機器を接続するために，ケーブルを繋ぐことは利用者にとっても熟練されており，理解が容易である．これを仮想的なネットワーク構築に応用し，同じ色のメモリースティックを 2 つの機器に指すだけで構築ができる．具体的なアプリケーションとして，PC とプロジェクタを繋ぐ tranScreen や音楽を再生する tranSound を提案している．ストレージメディアを指すだけなので，利用者の負担が少ないが，PC 同士を接続したときなど複数のサービスが考えられる時には優先度によりサービスが自動的に決定されるため，利用者が選ぶことができない．

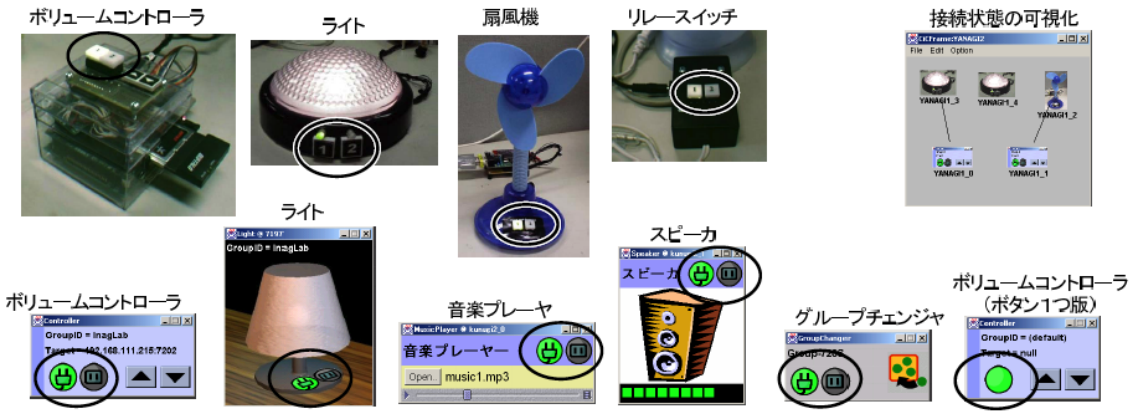


図 3.9　Touch-and-Connect

Gaze-Link

Gaze-Link[1] は，カメラを備えた計算機を使って，「見る」ことで通信相手を特定し，接続を実現する手法である．Gaze-Link を図 3.11 に示す．Gaze-Link は，CyberCode と呼ばれる二次元バーコードを機器に添付し，カメラでこの対象を認識することで接続を開始する．また，画像を用いるので，テレビ会議など遠隔地の環境との接続も行うことができる．しかし，連携元がカメラが接続された計算機に限定されるため，オーディオシステムとスピーカなどカメラがない機器が接続元となった連携は実現することができない．

図 3.10　tranStick

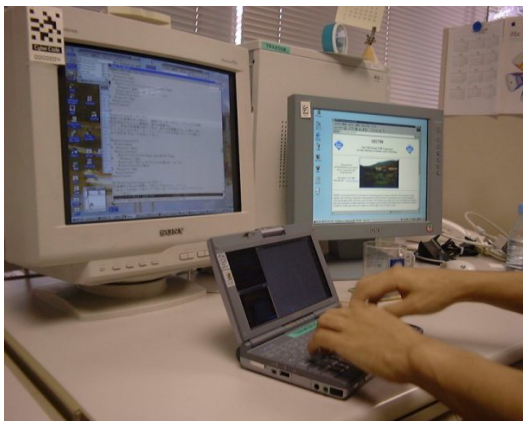


図 3.11　Gaze-Link

3.7 まとめ

本章では，はじめに，本研究における Chameleon システムの概要を述べ，ついで第 2 章で述べた機能要件を達成するためのアプローチとして，1）機器との距離に応じた機器指定手法，2）サービス発見型機器接続手法，3）統一インタフェースによるサービス制御手法，4）機器非依存型サービス記述手法を提案した．その後，関連研究をまとめ，既存研究との比較，考察により本研究の意義を示した．

第 4 章

Chameleon の設計

本章では，マルチモーダルなインタラクション支援システムである Chameleon を設計する．Chameleon は，インタラクションにより機器選択を行う機器選択機構，Resource と Function の機器が連携出来るかの解決を行うサービス解決機構，機器に連携命令を送るためのサービス制御機構から構成される．はじめに，Chameleon 全体の設計をハードウェア，ソフトウェアの双方について述べる．ついで，Chameleon の動作概要を述べる．

4.1 Chameleon の設計概要

本節では，Chameleon の設計をハードウェア，ソフトウェア双方の観点から述べる．ハードウェア構成では，Chameleon コントローラと連携対象機器，及び機器を管理するサーバとの関係性に関して述べる．ソフトウェア構成では，Chameleon を構成する各機構を挙げ，それらを構成する各モジュールの機能を述べる．

4.1.1 ソフトウェア構成

本項では，Chameleon のソフトウェア構成について述べる．Chameleon コントローラは，機器選択機構，サービス解決機構，サービス制御機構の 3 つから構成され，各モジュール間でデータの受け渡しを行っている．図 4.1 に，ソフトウェア構成図を示す．以下に，各機構について詳述する．

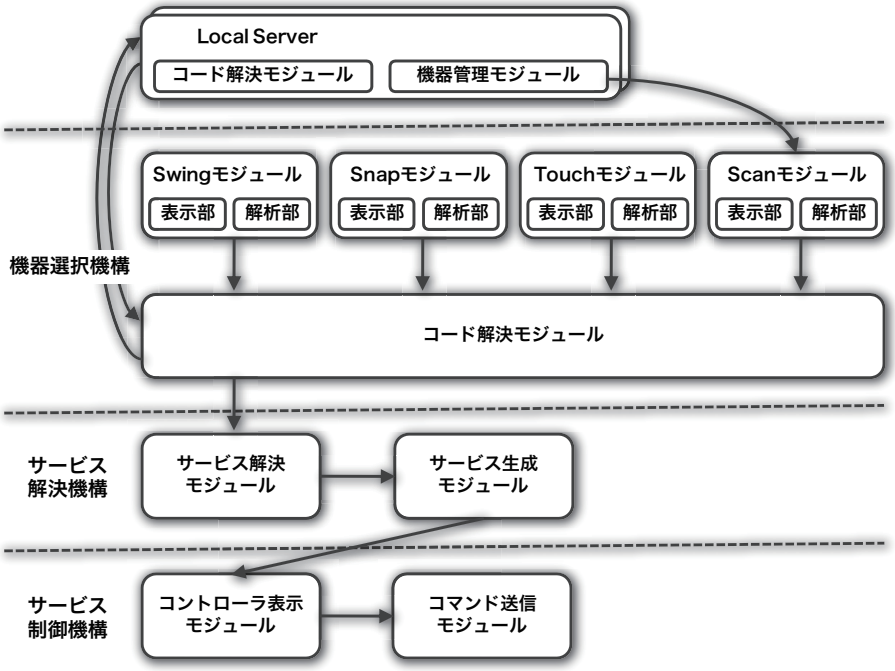


図 4.1　ソフトウェア構成

機器選択機構

機器選択機構は，各インタラクションモジュールと，コード解決モジュールにより構成される．インタラクションモジュールでは，操作状態を利用者に見せる表示部と機器に添付してあるコードをもとに解析を行う解析部に分かれる．コードをもとに解析が終了後，コード解決モジュールに対してマルチインタラクション ID が送られる．マルチインタラクション ID は，Swing であれば，

Up-Down，Touch であれば，RFID tag の ID など各インタラクションにより異なる．コード解決モジュールは，マルチインタラクション ID を Global Server に問い合せる．Global Server は，現在起動している機器のリストと，マルチインタラクション ID に対応した機器の情報を保持している．この情報をもとに，コード解決を行った後に，機器情報がコード解決モジュールに送られる．最後に，機器情報をサービス解決機構に送信する．

サービス解決機構

サービス解決機構は，Resource，Function の機器が連携可能か判断し，可能であれば，２つの機器情報からサービスを生成する．よって，サービス解決モジュールとサービス生成モジュールにより構成される．サービス解決モジュールは，Function 機器側の機能（表示，音楽再生など）に引数として，Resource 機器側の Resource（画像，音楽など）を取ることができるか判断する．サービス生成モジュールは，解決モジュールの結果に基づき，各機器にコマンドを送るためのサービス情報を生成する．

サービス制御機構

サービス制御機構は，コントローラ表示モジュールと，コマンド送信モジュールにより構成される．コントローラ表示モジュールは，各サービスに応じて，コントローラを動的に生成する．例えば，iPhone からフォトフレームに写真送信時には，コントローラは，はじめに iPhone に保存されている写真の選択画面へと移行する．コマンド送信モジュールでは，各機器にサービス実行命令を司る．

4.1.2 ハードウェア構成

本項では，Chameleon のハードウェア構成について述べる．図 4.2 に示すように，Chameleon は，連携が動作する機器群と，機器のネットワーク情報やどのようなリソースとファンクションを持つかを管理するサーバと，機器連携を取り扱う携帯電話から成る．各ハードウェア間は，IP ネットワークに基づいた通信により，データの送受信をする．そのため，携帯電話は Wi-Fi の機能を持つ携帯電話の利用を想定している．また，IP ネットワークによる通信を行うために，機器群は IP ネットワークが利用できることが求められる．そのため，ネットワーク機能を持たない機器は中間ノードとして計算機を挟み，計算機と機器との間でシリアル通信や USB，Bluetooth などの無線通信により接続されることが求められる．

4.2 動作概要

本節では，Chameleon の動作概要について述べる．はじめに，連携サービス実行における，コントローラ，Global Server, 各機器との間における動作概要を示す．次に，Scan インタラクション時におけるデバイス発見の流れを示す．

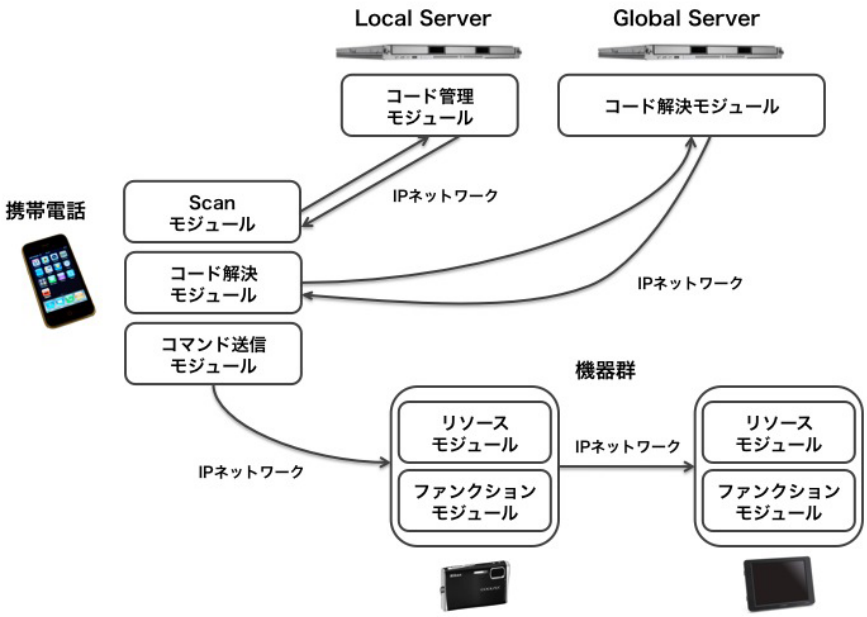


図 4.2　ハードウェア構成

4.2.1　連携サービス実行の動作概要

本項では，連携サービス実行時の動作概要について述べる．動作概要図を図 4.3 に示す．各機器が起動し，ネットワークアドレスを取得後に，Global Server に対して，デバイス情報を送信する．このデバイス情報には，IP アドレスや送受信を行うためのポートなどが含まれる．携帯端末は，機器を各インタラクションで選択後，マルチインタラクション ID をもとに，機器の詳細情報を Global Server に問い合わせる．この詳細情報は，各機器がどんな Resource と Function を持っているかが記述された XML 形式で返答が返ってくる．携帯端末側で，Input/Output の機器指定が終了後，携帯端末でサービスの解決を行う．この解決後，コマンドを送るための XML が生成される．そして，コントローラは，各デバイスに対してサービス実行要求を行う．

4.2.2　デバイス発見の動作概要

本項では，コントローラ側でデバイス発見を行う際の動作概要に関して詳述する．動作概要図を 4.4 に示す．前述したように，各機器は起動時に Global Server に機器情報を送るようになっている．Global Server は，現在のネットワークにおいて，どの機器が起動しているかを管理している．携帯端末は，デバイス発見である Scan Request を Global Server に対して送信すると，Global Server は現在起動しているすべてのデバイス情報が含まれたデバイスリストを携帯端末に

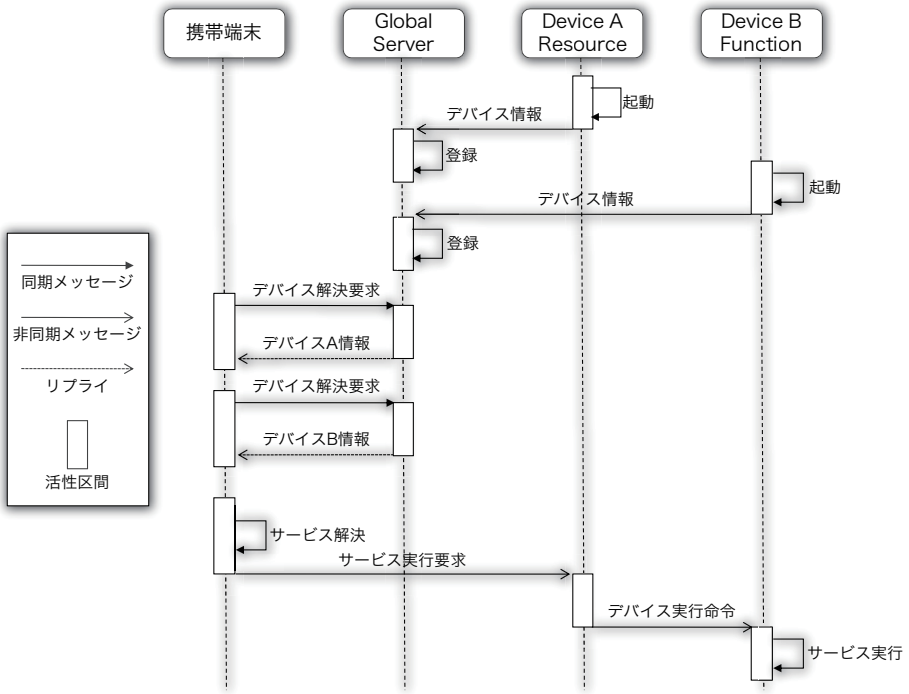


図 4.3　連携サービス実行の動作概要図

送信する.

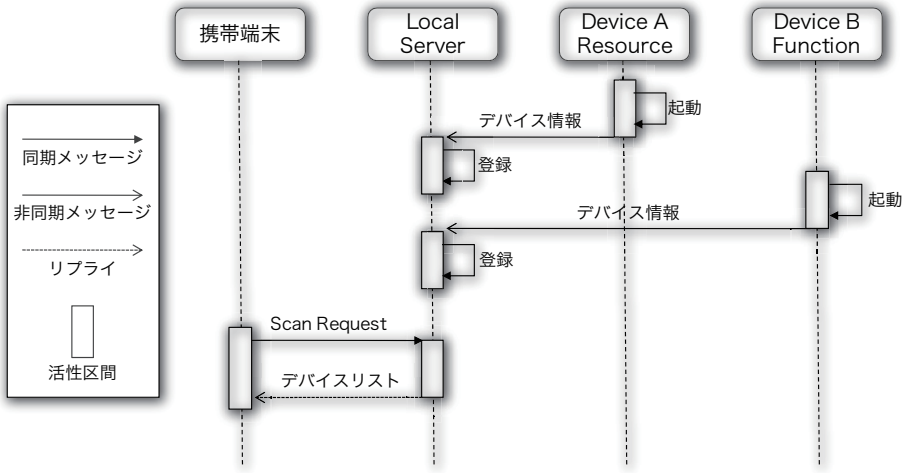


図 4.4　デバイス発見の動作概要

4.3 まとめ

本章では，マルチモーダルなインタラクション支援システム Chameleon を設計した．はじめに，Chameleon のソフトウェア設計として，Chameleon がインタラクションにより機器選択を行う機器選択機構，Resource と Function の機器が連携出来るかの解決を行うサービス解決機構，機器に連携命令を送るためのサービス制御機構から構成されることを述べた．ついで，Chameleon のハードウェア構成として，連携命令を司る携帯端末と，連携を行う機器群，機器の状態を管理するためのサーバから成ることを述べた．その後，Chameleon の動作概要を述べた．

第 5 章

Chameleon の実装

本章では，Chameleon のプロトタイプ実装について述べる．プロトタイプ実装では，連携指示を行うための携帯端末として iPhone 3GS を用い，objective-c 言語を使用してソフトウェアを実装した．はじめに，ハードウェアの実装について述べ，ついで，ソフトウェアの実装について述べる．

5.1 ハードウェアの実装

Chameleon のハードウェア構成は，機器の状態や情報を管理するためのサーバ，連携指示を行うための携帯端末，連携が実際に行われる機器群から成る．本節では，プロトタイプ実装で利用したハードウェアについて述べる．図 5.1 に示す．

図 5.1　プロトタイプ実装

5.1.1 機器管理サーバ

プロトタイプ実装では，機器管理サーバとして，計算機に 3 台の Mac Book Pro を利用した．Java 実行環境にはサン・マイクロシステムズ株式会社が開発した VM を用いた．主な計算機のハードウェア，および，ソフトウェア仕様を表 5.1 にまとめる．プロトタイプ実装では，コード解決モジュール，機器管理モジュールを動作させた．

表 5.1　プロトタイプ実装に用いた計算機の仕様

項目	Mac Book Pro
CPU	2.8GHz
Memory	8GB
OS	Mac OS 10.5.8
JavaVM	JRE1.5.022

5.1.2 コントローラ

プロトタイプ実装では，コントローラとして iPhone 3GS を用いて，機器連携の指示を行った．コントローラでは，図 4.1 に示したソフトウェアを動作させた．仕様を 5.2 に示す．

表 5.2　プロトタイプ実装に用いたコントローラの仕様

項目	iPhone 3GS
CPU	600MHz
Memory	256MB
OS	iPhone OS 3.1.2
カメラ画素数	3MPx
センサ	加速度センサ，近接センサ，環境音センサ

5.1.3 機器群

プロトタイプ実装で用いた機器群，機器名，Resource, Function を図 5.3 に示す．機器群は IP ネットワーク，USB，Bluetooth いずれかの方法で，機器管理サーバと接続ができる機器である．

表 5.3　プロトタイプ実装に用いた機器群

項目	機器名	Resource	Function
プリンタ	Pixus ip100	なし	印刷
ライト	なし	なし	警告
スピーカー	不明	なし	再生（音楽）
ディスプレイ	FlexScan L997	表示している画像	表示（画像）
フォトフレーム	plus one LCD-8000U	表示している画像	表示（画像）
デジタルカメラ	COOLPIX S52c	画像	なし
扇風機	なし	なし	回転
センサ	SunSpot	コンテクスト	なし

5.1.4 RFID リーダ

本論文では，Touch インタラクションを実現するために，RFID を利用した．しかし，iPhone 3GS には，RFID リーダの機能が無く，また外付けの RFID リーダを作成するには特別なライセンスが必要である．そのため，本論文では iPhone に RFID リーダを取り付けるのではなく，iPhone に RFID のタグを貼り，機器側にリーダを設置した．リーダは，機器に触れた瞬間にデータを送る必要があるため，リーダが iPhone との通信手段を備えることが求められる．よって，サン・マイクロシステムズが開発した SunSpot に，小型 RFID リーダ Skyetek M1-Mini を接続しリーダとして利用した．作成した RFID リーダを図 5.2 に示す．また，RFID タグの作成には，ブラザーの RFID プリンタ P-Touch RL-700S を用いた．RFID プリンタを図 5.3 に示す．

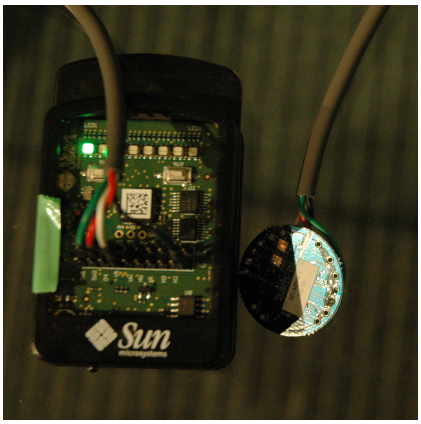


図 5.2　SunSpot と Skyetek M1-Mini による RFID リーダ

図 5.3　RFID プリンタ P-Touch RL-700S

5.2　ソフトウェアの実装

本節では，第 4.1.1 項で述べたソフトウェア構成のうち，コード解決モジュール，サービス解決モジュール，コマンド送信モジュールのソフトウェアについて述べる．本ソフトウェアは，iPhone 上で動作するために，objective-c の言語を用いて実装した．

5.2.1　コード解決モジュール

コード解決モジュールの主な役割は，各インタラクションモジュールにより，解析が行われた機器の ID を，機器の詳細情報に変換することである．まず各インタラクションは，マルチインタラクションコードを解析し，マルチインタラクション ID の形に変換を行う．マルチインタラクション ID は，各インタラクションにより異なり，Touch では RFID タグの ID，Swing では降る方向，Snap では ARToolkit のマーカ ID となる．表 5.4 にマルチインタラクション ID を示す．各インタラクションモジュールで，マルチインタラクション ID が解決されると，この ID は **mid:INTERACTION:INTERACTION_ID:SERVER_TYPE** のフォーマットでサーバに送信される．mid は接頭語であり，INTERACTION は各インタラクションの名前（touch, swing, snap）が入る．INTERACTION_ID には，表 5.4 で述べたマルチインタラクション ID が入る．SERVER_TYPE には，Resource，Function どちら側の機器を解決しているかが入る．例えば，Swing インタラクションを用いて，Resource 側のフォトフレーム（マルチインタラクション ID: up-down）を解決する場合には， **mid:swing:up-down:resource** のリクエストをサーバに送信する．送信が終わると，サーバ側でマルチインタラクション ID に応じた，機器の IP アドレス，ポート番号，デバイス記述 XML をコントローラは受信する．デバイス記述 XML を図 3.7 に示す．

表 5.4　マルチインタラクション ID

インタラクション	マルチインタラクション ID	ID の例
Touch	RFID タグの ID	69F1
Swing	矢印の組み合わせ	up-down
Snap	ARToolkit のマーカ ID	1

各インタラクションの解析手法を以下に述べる．

Touch インタラクションの解析

Touch インタラクションは，SunSpot に付属した RFID リーダで読み取った機器の id を，iPhone に送信することで実現する．SunSpot と iPhone は，SunSpot のベースステーションを経由し，TCP/IP のソケット通信により行われる．SunSpot は，コードと共に各機器にバインドされるために，id に各 SunSpot に固定に割り振られた id が入る．

Swing インタラクションの解析

Swing インタラクションは，iPhone に内蔵する加速度センサの値の変化に基づき，id を解析する．Swing は，上下左右の 4 方向を 2 回組み合わせた矢印からなる．加速度の解析手法として，振る方向に対する加速度データの変化の特徴に着目をした．iPhone を上下に振ると，加速度 Z の値が著しく変化するが，加速度 X，Y の値はあまり変化しない．上下どちらに振ったかは，加速度 Z の値がはじめに，プラスかマイナスどちらの方向に増加するかを観察する．例えば，上に振る動作は，加速度 Z の値がはじめに，マイナス方向に転じ，端末を止める動作でプラス方向へと変化する．この変化を検出し，振った方向を検出する．携帯端末が左右に触れると，上下とは異なり，加速度 X の値が変化する．上下の検出と同じく，左右のどちらに触れたかは，加速度 X の値がプラスかマイナスどちらの方向に転じているかにより判断する．

Snap インタラクションの解析

Snap インタラクションは，iPhone のカメラを使い，ビジュアルマーカの解析により実現する．本論文におけるビジュアルマーカは，図 3.4 で示したマルチインタラクションコードである．iPhone 上では，ARToolkit のライブラリを利用し，マーカの解析を行う．また，機器を撮影すると，一度ローカルサーバにデバイス記述 XML を問い合せる．その後，撮影された JPEG 形式の画像ファイルのコメント領域にデバイス記述 XML を保存する．写真選択時には，コメント領域に保存されているデバイス記述 XML を読みだすことで，一度撮影した機器を何度でも写真から選択出来る．

Scan インタラクションの解析

Scan インタラクションは，ローカルサーバに現在動作している機器のデバイス記述 XML を問い合せて，機器の選択を行う．Scan インタラクションが実行されると，サーバに TCP/IP のソ

ケット通信で，リクエストを送る．その後，ローカルサーバは，iPhone に対して動作中のデバイス記述 XML が含まれた XML を返す．

インタラクションの遷移

利用者は，機器選択画面から全てのインタラクションが利用できる．各インタラクションの解析機構が，全て動作することにより実現する．しかし，Scan インタラクションは，携帯端末をシェイクし，実行が行われる．このシェイクの動作は，加速度の変化を観察し判別する．シェイク時における加速度の値は変化の波が大きいために，分散値の合計をもとに検出する．

5.2.2 サービス解決モジュール

サービス解決モジュールは，Resource で指定された機器と Function で指定された機器が連携可能か判断を行う．連携可能かは，Resource で選択されたデバイス記述 XML に書かれている output タグの中身と，Function で選択されたデバイス記述 XML の input タグの中身が一致するかで判断を行う．一致したときには，利用者に対して function タグの name をサービス名として表示する．また，一致時にはコマンド送信時に使うための XML を生成する．この XML は次の項で述べる．

5.2.3 コマンド送信モジュール

コマンド送信モジュールでは，Function 側に設定されている機器に対して，連携実行 XML を送信する．連携実行 XML を図 5.4 に述べる．連携実行 XML は，Resource と Function のデバイス記述 XML により生成される．device タグの中身には，function に入る機器の情報が入る．

$< resourceTransgerRequest >$ $\cdots$ 送信命令全体を表すタグ．他の全てのタグを子要素に持つ．
$< to >$ $\cdots$ 送信先の情報を表すタグ．子要素として，図 3.7 で示したデバイス記述 XML を持つ．

$< call >$ $\cdots$ 送信先で実行される機能を示すタグ．子要素として，$< arg >$ 持ち，機能における引数として利用する．functionId には，図 3.7 における function の id，resourceId には，resource の id が記述される．
$< arg >$ $\cdots$ 送信先で実行される機能の引数を表すタグ．画像や音楽などの特定のファイルを指定する時に記述される．

5.3 まとめ

本章では，Chameleon のプロトタイプ実装について述べた．はじめに，ハードウェアの実装として，機器管理サーバ，コントローラ，機器群の詳細について述べた．その後，ソフトウェアの実

```
<resourceTransferRequest>
  <to>
    <device>
      <property>...</property>
      <resources>...</resources>
      <functions>...</functions>
    </device>
  </to>

  <call functionId="..." resourceId="...">
    <arg name=" ... " >...</arg>
  </call>
</resourceTransferRequest>
```

図 5.4　連携実行 XML

装として，Chameleon を構成するコード解決モジュール，サービス解決モジュール，コマンド送信モジュールについて，それぞれの実装の要点を述べた．

第 6 章

評価実験

本章では，Chameleon のユーザビリティ評価を行う．本手法の評価として，Distance Based Interaction Method と，シームレスなインタラクションの自動的切り替え手法を複数の被験者に対して行う．本実験は，まずはじめに，利用者が複数のインタラクションをどのように使い分けるかを調査した．そのうえで，その結果に基づき実装された Chameleon システムで，Distance Based Interaction Method の評価を行い，結果を考察する．

6.1　機器指定におけるマルチインタラクションの必要性

本節では，まずはじめに本論文で提唱するマルチインタラクションの必要性を評価する．そのために，利用者と機器との距離の変化に応じて，異なるインタラクションをどう使い分けるかを観察する．はじめに，本実験の被験者と実験環境を延べ，ついで，実験結果と実験手順を詳述する．

6.1.1　実験手順

本実験の実験環境として，慶應大学徳田研究室の実験空間である t41 を利用し，複数の機器を利用者から異なる距離に配置した．図 6.1a に実験空間を，図 6.1b に利用者と機器との距離を示す．また，センサを利用者から目視できない位置に配置した．被験者は，20 歳前半のプログラミング経験があるコンピュータリテラシーを有した学生 4 名である．被験者は，各インタラクションの使い方を聞いた後，1 分間インタラクションの練習をした．その後，実験の開始位置としてソファーに座り，6 種類の機器を 2 回ずつ計 12 回，自分の好きな手法で選択し，どのインタラクションを用いたかを記録した．実験終了後に，利用者に対して実験に対する感想をインタビュー形式により尋ねた．

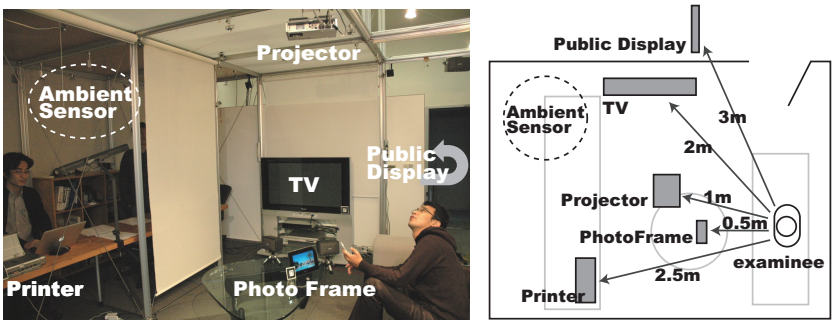


図 6.1　a) 実験環境と b) 利用者と機器との位置関係

6.1.2　実験結果

本実験の実験結果を表 6.1 と図 6.2 に示す．表 6.1 は，利用者と機器の距離，各機器をどのインタラクションにより選択したかの割合を表している．図 6.2 は，各機器の選択に用いられたインタラクションの割合である．Snap には，撮影による選択と，保存された写真を選択するの 2 種類がある．前者を Snap，後者を Picture と実験結果と記述している．図 6.2 から見て取れるように，

赤色で描かれている Snap（撮影による指定）は，一度しか利用されていない．これは，対象物にフォーカスを当てる必要があるため，利用者の負荷が大きいためであると考えれられる．

表 6.1　機器選択時のインタラクション手法の使い分け

	フォトフレーム	プロジェクタ	テレビ	プリンタ	パブリックディスプレイ	センサ
距離	0.5m	1m	2m	2.5m	3m	空間内のどこか
詳細	テーブルの上	天井	壁沿	机の上	部屋の外	目に見えない
Touch	0.625	0.25	0.25	0.25	0.125	0.0
Snap	0.125	0.0	0.0	0.0	0.0	0.0
Swing	0.125	0.25	0.5	0.5	0.0	0.0
Picture	0.125	0.25	0.125	0.125	0.5	0.25
Scan	0.0	0.25	0.125	0.125	0.625	0.75

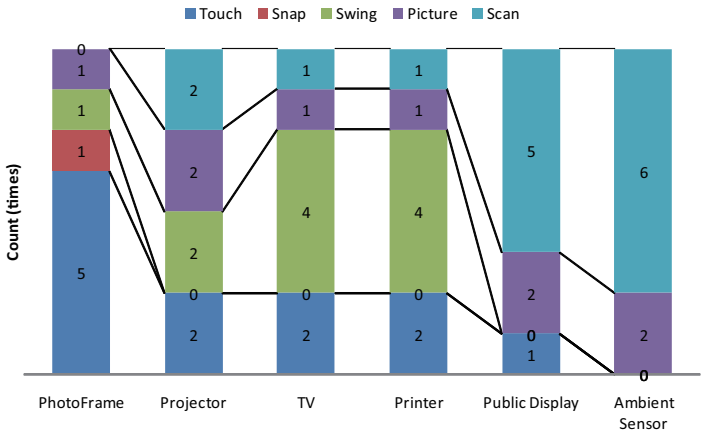


図 6.2　機器選択時のインタラクション手法の使い分けの割合

6.1.3　考察

実験結果から，利用者は自らの状況に応じて，インタラクションを適切に切り替えていると分かった．実験終了後のインタビューでは，インタラクションの選択理由に，多くの被験者が対象物との距離や間にある障害物の有無で挙げていた．例えば，最も近い距離にあるフォトフレームは，Touch が最も採用され，一方で視覚に入らない場所にあるパブリックディスプレイやセンサは，Scan や Picture の利用頻度が高い．実験から，距離だけではなく，利用者の活動状態や機器の認知の度合いが選択に影響を与えていると分かった．例えば，Swing は一度コードの矢印を認識すれば，対象物の注視が求められないため，快適であり負担が少ないとの意見があった．これに対して，Snap は画面上でマーカが認識出来ているかを確認するため，利用者の負荷が高いと分かった．結果から，人は自分の置かれた状況を解釈し，インタラクションを選択するため，距離に応じた複数のインタラクションを提供する手法は正しい方向性であると分かる．その他の意見として，マルチインタラクションコードは，矢印が枠内に描かれているため，Snap や Scan に比べ振る動作を促していると感じる．Touch できるか不安になるとの意見があった．

6.2 インタラクション切り替え手法

本節では，第 6.1 項の結果を受け，シームレスにインタラクションを切り替えをする手法を評価する．複数インタラクションは，状況に応じて適切な手法の選択が可能だが，一方で切り替えの手間を与えてしまう．よって，本実験はこの手間を減らすための利用者の意図を予想したインタラクションの自動的切り替え手法を評価する．はじめに，本実験の被験者と実験環境を延べ，ついで，実験結果と実験手順を詳述する．

6.2.1 実験手順

本実験では，インタラクションの切り替え行うための，利用者のアクションをトリガとして自動切り替え手法と手動切り替え手法の 2 種類を用意した．図 6.3a,b に本手法のスクリーンショットを示す．図 6.3b は画面に，触る，振る，撮る，スキャンする，の各インタラクション選択のためのボタンが表示されており，利用者が押すことで各インタラクションの画面へと移行する．図 6.3a はカメラが最初から起動してあり，この画面から直接，各インタラクションが行える．例えば，振るインタラクションで機器選択するには，この画面のまま直接端末を降り始めると，振るインタラクションの画面へと推移する．実験では，被験者に対してインタラクション手法をランダムに 20 回指定し，指定されたインタラクションで機器選択が完了するまでの時間を測定した．実験の手順は，簡単な説明（目的，使い方，実験に関する諸注意），インタラクション選択作業，アンケート記入の 3 段階である．実験終了後のアンケートは，リカート法 [26] に基づく，質問内容と評定方法である．実験では，ユーザビリティの指標を表す学習しやすさ，効率性，記憶しやすさ，エラーがあるかまたは起こした際の回復，ユーザの主観満足度の 5 つに関して，実験を行い評価した．被験者は，6.1 と同じ 20 代前半の学生 4 名である．

6.2.2 実験結果

タスクを終了までの所要時間を表 6.4 に示す．縦軸には，選択終了までに要した時間の平均値が描かれている．赤で示した棒グラフが自動的切り替え手法でのタスク実行までの所要時間であり，青が手動でインタラクションを選択した時の所要時間である．黒い線で表された範囲は，各結果の標準偏差を示している．

Snap インタラクションを除いて，自動切り替え手法と手動切り替え手法の間に大きな所要時間の変化はなかった．Snap インタラクションは自動切り替え手法時に時間がかかっているが，これは写真撮影時にバックグラウンドプロセスで動いている加速度解析による誤作動が原因であると思われる．

Snap を除いて，自動と手動切替手法には，大きな所要時間の変化はなかった．Snap は，自動切り替えの際に，時間がかかっているが，これは写真撮影時に背後で動作する加速度解析による誤作動が原因だと思われる．

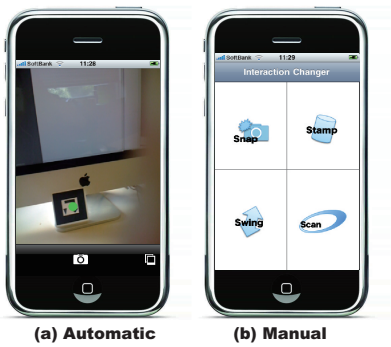


図 6.3　自動インタラクション切替と手動インタラクション切替手法のスクリーンショット

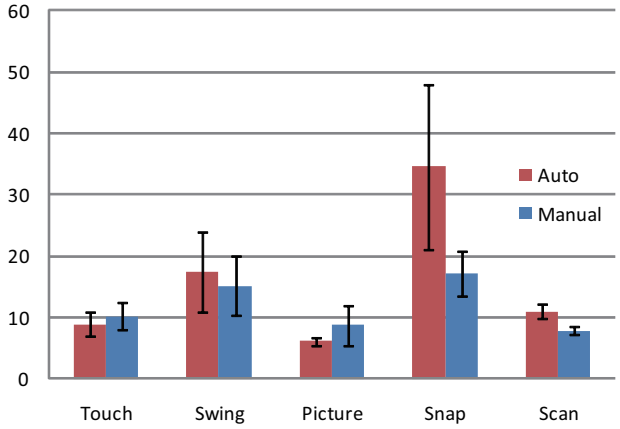


図 6.4　タスクを終了するまでの所要時間

表 6.2 は，自動切り替え時の，被験者が意図したインタラクションが起動しなかった認識エラー率である．Swing に対する誤認識が 0.56 と大きくなる傾向にあるが，この誤認識の多くは，同じくジェスチャを遷移トリガとする Scan との誤認識が多かったためである．よって，円を描いたり，手前に引くなどの振る動作を考察しつつ，再度解析アルゴリズムを検討する必要がある．

表 6.2　シームレスインタラクション切替手法の誤認識

	Touch	Swing	Picture	Snap	Scan
認識エラー率	0.06	0.56	0	0.06	0.06

表 6.3　質問結果：シームレスインタラクション切替手法

質問	Manual	Automatic
使い方を習得するのは簡単だ	4.0	3.75
やっていた非常にイライラする	3.0	3.0
機器の選びやすさの生産性があがるとおもう	3.75	4.25
したことはほとんど間違っていたのではないかと思う	1.25	1.5
必要だと思ったことが全部できる	3.5	3.75
やっていて楽しい	3.0	4.0

表 6.3 は，ユーザビリティ評価の 5 つの指針 [32] をもとに行った評価実験の結果である．第 6.2.2 項で述べた，Snap インタラクションにおける誤認識や所要時間にも関わらず，“やっていて楽しい” の項目は，自動切り替え手法における点数は 4.0 と，手動切り替え手法よりも 1.0 点高かった．また，アンケートの自由記述欄からも，自動切り替え手法は，手動切り替え手法よりも楽しいとの意見を得た．“機器の選びやすさが効率的に行えると思う” の項目では，4.25 と自動切り替えが手動切り替えよりも 0.5 点高かった．アンケートの自由記述欄では，手動切り替えはインタラクションとボタンのマッチングを頭の中でする必要があるが，自動は直接機器を選択することができるとの意見があった．

6.2.3 考察

結果から，シームレスなインタラクション切り替え手法のユーザビリティに関する各項目について考察する．利用者の動作をトリガにインタラクションを切り替える自動切り替え手法は，解析のために複数のプロセスが動作するため，所要時間を多くかかり，またインタラクションの認識エラーが手動に比べ発生しやすいと分かった．これは，利用時に煩わしさを与えるため，利用の負担へとつながる．このような欠点にも関わらず，ユーザビリティ評価の結果からは，利用者は自動切り替えを好むことが分かった．自由記述で得た意見を元に考えると，自動切り替えはボタンがあるため，ボタンの選択やインタラクションとの対応付けをすることが求められる．一方で，自動切り替え手法は，あるインタラクションを行いたいと考えると，その動作を直接始めることができる．よって，自動切り替え手法を利用者が好むと考えられる．そのため，本論文で提唱するシームレスなインタラクションの自動的切り替え手法の方向性は正しいと分かる．切り替え時の誤動作や所要時間は，Swing と Scan の切り替えにおけるしきい値の変更や，画像解析アルゴリズムの改良により改善されて行くと考えられる．

6.3 機器選択型とサービス発見型の機器接続手法の評価実験

本節では，第 3.3.2 に述べた，機器接続手法であるサービス発見型機器選択手法における作業負荷を評価する．そのため，実験の比較対象として，入出力２つの機器を選択する，機器選択型と比較する．はじめに，本実験の被験者と実験環境を述べ，ついで，実験結果と実験手順を詳述する．

6.3.1 実験手順

本実験は，リビングルームを想定して作られた実験環境である，徳田研究室の t41 にて行った．実験では，携帯端末を含めた，５種類の機器を用意し，４種類の機器連携シナリオを各手法で一回ずつ実践してもらう．本実験における評価シナリオを表 6.4 に示す．実験の手順は，説明（目的，使い方，実験に関する諸注意），1 番目の手法による機器連携作業，アンケート記入，２番目の手法による機器連携作業，アンケート記入の５段階である．アンケートは，NASA-TLX を用いて，利用者の主観的作業負荷を評価した．機器連携作業は，作業の慣れによる誤差を防ぐ為に，機器選択型とサービス発見型を交互に 1 番目の手法とした．機器選択型は，入出力両方の機器を選択し，その後サービスを選択する手法である．サービス発見型は，入出力どちらか一方の機器を選択すると，選択された機器に応じて利用可能なサービスがリストアップされる手法である．また，機器選択時のインタラクションは，Swing に固定し，インタラクションによる作業負荷の誤差を無くした．本実験の被験者は，１０代から２０代の大学生１０名である．

表 6.4　機器接続手法における評価実験時の連携機器

順番	入力機器	出力機器
S1	iPhone	フォトフレーム
S2	フォトフレーム	ディスプレイ
S3	フォトフレーム	プリンタ
S4	センサ	ライト

6.3.2　実験結果

本実験では，NASA-TLX を用いて被験者の主観的作業負荷を評価した．本実験の計測結果として重み付けされた作業負荷の平均値（WWL：weighted workload），６つの尺度項目ごとの重み付けされた平均値について図 6.5 に示す．

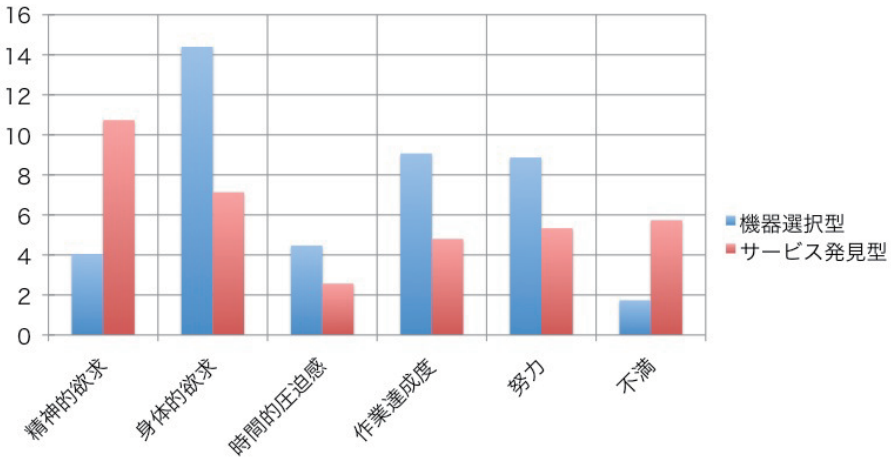


図 6.5　NASA TLX による作業負荷の平均値

機器選択型の６つの評価尺度を合わせた平均値は，42.567 であり，サービス発見型が 36.300 であった．

6.3.3　考察

実験結果から，機器接続における作業負荷の考察を行う．まず，サービス発見型の機器接続手法は，２つの機器を選択する機器選択手法に比べて，平均の作業負荷が低い事が分かった．特に作業負荷の要因である，身体的欲求が低く済むために，全体的な作業負荷が軽減されていると思われる．また，２つの接続手法では，求められる作業負荷の要素が異なる事が分かった．例えば，機器選択型は，入出力２つの機器をインタラクションにより選ぶ手間が必要なため，身体的欲求が高くなる．これに対して，サービス発見型は，入出力どちらか一方の機器を選ぶため，手間が少なく済むため，身体的欲求が低い．しかし，多数のサービスがリストアップされると，サービスを選択す

る手間が増える為に，精神的欲求が増える．また，環境内に機器が少ない時には，サービス発見型は有効である．しかし，機器が増えると，発見されるサービスが多くなる為に，サービス選択の負荷が掛かってしまう．このことから，サービス発見型において，より精神的負荷を減らす為のインタフェース設計が今後求められる．

6.4 Distance Based Interaction Method を用いた機器連携の評価実験

上述した２つの評価実験から，複数のインタラクションを統合した一つの画面から動作させることで，機器に対する命令を容易にすることが分かった．本節では，この手法を用いて，機器連携を実現する Chameleon の評価実験を行う．そのために，利用者が実際に連携を行うシナリオを想定し，実験環境を用意し，被験者に利用してもらう．はじめに，本実験の被験者と実験環境を述べ，ついで，実験結果と実験手順を詳述する．

6.4.1 実験手順

機器連携が同じ空間内だけでなく，異なる空間との間でも行われることを考慮し，実験環境として，仮想の自宅とオフィスを用意した．仮想の自宅には，リビングルームを想定して作られた本研究室の実験室 6.1 を使い，オフィスとして学生が研究を行う研究室を利用した．実験では，家からオフィスまでの移動時に考えられる５つの連携シナリオを用意した．実験の手順は，説明（目的，使い方，実験に関する諸注意），Distance Based Interaction Method による機器連携作業，アンケート記入の三段階である．本実験の被験者は Distance Based Interaction Method の利用経験がない 20 代から 30 代までの 8 名である．そのうちコンピュータをほとんど使わないコンピュータ初級者が 1 名，日常的に使っている中級者が 5 名，OS やネットワークの知識，プログラミング経験のある上級者が 2 名である．ユーザが選ぶインタラクションは，指定せず，好みに応じて選択してもらい，各作業においてどのインタラクションを利用したかを記録した．

図 6.6 に，評価で実施したシナリオを示す．また，表 6.5 はシナリオの詳細を表している．自宅を模した部屋には，テレビ，プロジェクタ，ディスプレイ，PC，センサ，フォトフレーム，プリンタなど複数の機器がある．また，仮想のオフィスには机の上に PC，ディスプレイがあり，少し離れた机の上にライトがある．シナリオは，機器の場所や所有者が異なり，また機器が各々違うリソースとファンクションを持ち，それにより生成されるサービスの種類を複数考え構成した．

表 6.5　評価シナリオの詳細

順番	概要	場所
S1	iPhone に保存されている写真をフォトフレームに表示	自宅
S2	フォトフレームに表示されている写真をプリンタで印刷	自宅
S3	環境内にあるセンサの値を iPhone に表示	自宅からオフィスへの移動中
S4	家のフォトフレームの写真をオフィスのディスプレイに表示	オフィス
S5	家のセンサからのアラートをオフィスのランプで表示	オフィス

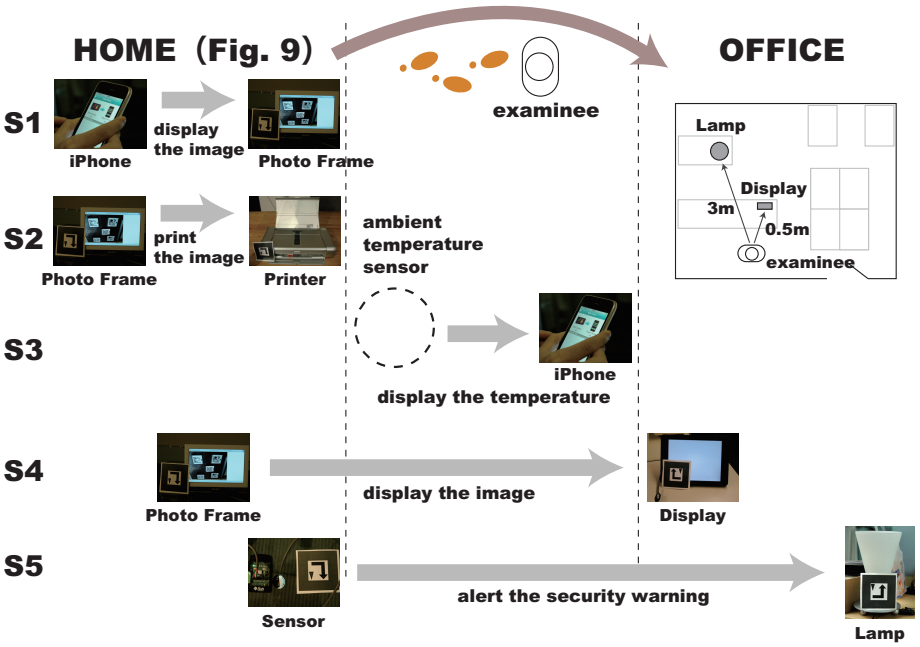


図 6.6　評価シナリオ

6.4.2　実験結果

表 6.6,6.7 に実験結果を示す．表 6.7 はシステム全体のアンケート結果であり，表 6.6 は環境ごとのアンケート結果を表しており，自宅を近傍，オフィスを遠隔，移動中の廊下を未知の場所と記述している．

表 6.6　Distance Based Interaction を用いた機器連携手法における距離別の評価結果

質問	近傍	遠隔	未知の場所
選択をするのが面倒だ	1.38	1.13	1.5
使い方を習得するのは簡単だ	4.5	4.38	3.88
やっていて非常にイライラする	1.75	1.5	1.63
したことはほとんど間違っていたのではないかと思う	1.5	1.5	1.75
必要だと思ったことが全部できる	4.25	4.5	4.0
やっていて楽しい	4.75	4.5	4.75

表 6.7　Distance Based Interaction を用いた機器連携手法における全体の評価結果

質問	結果
使い方を習得するのは簡単だ	4.75
やっていて非常にイライラする	2.38
したことはほとんど間違っていたのではないかと思う	1.38
必要だと思ったことが全部できる	4.25
やっていて楽しい	4.88
機器の連携は直感的に行えた	4.63
インタラクションは直感的に行えた	4.5

6.4.3 考察

実験結果から，Chameleon のユーザビリティに関する各項目について考察する．

学習しやすさ

プログラミングやネットワークの知識のある利用者は，簡単な説明だけで Chameleon を使い，連携サービスの利用ができた．しかし，コンピュータ初級者は機器を連携させることに関して，経験が無く，利用を戸惑っていた．表 6.7 のアンケート結果においても，「使い方を習得するのは簡単だ」の項目に対する被験者の平均は 4.75 であり，非常に学習しやすい印象を持っていることが分かる．しかし，表 6.6 の未知の場所における項目の平均値は，他と比べて 3.88 と低かった．これは，見えない機器を連携させることが求められるため，認知的負荷が大きかったと考えられる．

記憶しやすさ

Chameleon は，連携サービスの取得には，1）機器の選択，2）サービスの選択，3）サービスの利用，3つの手順が必要である．しかし，多くの被験者は，一回の説明で手順を学習することができた．よって，時間をおいても忘れにくいと考えられる．

エラーがあるかまたは起こした際の回復

実験では，機器選択を何らかのインタラクションで行うときに，各インタラクションが解析を誤った，インタラクションの移行が思い通りにいかなかったの２つのエラーが特に起きた．第 6.2 節での評価実験でも，Swing と Scan への移行の遷移が誤作動し，結果タスクを終了するまでの時間を多く要していることが分かっている．このエラーは，表 6.7 の「やっていて非常にイライラする」という質問に対する平均得点 2.38 に影響を与えていると考えられる．一方で，サービスの選択や利用に関するエラーは発生しなかった．

利用者の主観満足度

Chameleon による連携サービス利用に関するアンケート結果で，設問の平均値は 4.88 であった．リカートの個人的満足度の中間点は 3.6 であることが知られており，4.66 という数値は，平均より満足度の高いインタフェースであることを示している．システムのアフォーダンスを向上することで，さらに高い満足度につながると考えられる．また，表 6.6 から分かるように，距離に関わらず高い満足度を得ることができた．自由記述欄では，「技術（インタラクション）の押し付けが無い」，「自分の状況に応じて，選択方法の変更が出来てよい」との意見が目立った．

また，利用者の選択手法は，ばらつきがあった．例えば，S2 におけるフォトフレームの指定は，Touch が２人，Snap が 1 人，Swing が 1 人である．この結果は，連携においても，利用者が距離や自分の活動状態に応じて，柔軟にインタラクションを選択し，複数のインタラクションを提案する Chameleon の方向性が正しいと分かった．

6.5 まとめ

本章では，ユーザビリティ評価を行った結果の考察について整理した．ユーザビリティ評価では，シームレスなインタラクションの自動的切り替え手法，Distance Based Interaction Method に関して評価を実施した．次章で，本研究の課題について整理する．

第 7 章

結論

本論文では，機器連携におけるマルチモーダルなインタラクション支援システム Chameleon の設計と実装を行った．本章では，実験において見えてきた課題を簡潔に述べ，ついで本論文のまとめを述べる．

7.1 結論

家電や携帯端末などのネットワーク化やデジタル化にともない，機器間が相互接続し，単一の機器ではできなかった柔軟なサービスを実現できるようになった．また，無線 LAN の機能を持つ携帯端末が普及し，ネットワーク化された機器が家庭，オフィス，公共空間へと普及することで，人と機器との関係性が変化するなかで，動的に連携が発生しつつある．しかし，利用者が機器連携のサービスを取得するには，IP アドレスや機器名の特定など複雑な設定や操作が必要であった．実世界のインタラクションにより，直感的に機器へと命令を送る手法があるが，動的に機器が変化する中では，単一のインタラクションでは不十分である．また，機器間の連携命令をどう利用者が扱うかは，利用者の連携に対するメンタルモデルの欠如により，利用者の負担が大きかった．

本論文では，複数のインタラクションにより，機器との距離が近傍から遠隔へと変化しても，直感的に連携命令の送信が可能なマルチインタラクション手法を提案した．また，利用者が着座している状態や機器が見えない状況でも，適したインタラクションにより機器選択が可能である．加えて，複数のインタラクションを可能にするマルチインタラクションコードを作成し，コードの管理や発行の手間を軽減した．さらに，環境内にある様々な連携サービスの発見を支援するため，サービス発見型機器選択手法を提案した．これにより，サービス実現性のある機器を 1 つ特定するだけで，利用可能なサービスを見ることができ，利用者の負担が軽減される．プロトタイプ実装のユーザビリティ評価から，以下の 2 つが分かった．はじめに，利用者が機器との距離に応じて望ましいインタラクションを選択し，複数インタラクションの必要性と機器選択における高い主観満足度との知見を得た．ついで，連携サービス利用時に，サービス発見型の機器選択手法に基づいて，利用者の負荷を下げ，かつ高い主観満足度を得た．

7.2 今後の展望

Chameleon の実装は十分ではなく，更なる改善が求められる．本節では，Chameleon の設計及び実装における問題点と解決策を述べ，今後の展望とする．

7.2.1 アフォーダンスの向上

Chameleon は，利用者に対するアフォーダンスを向上する必要がある．Norman[33] は，ユーザインタフェースにおけるアフォーダンスを行為の可能性と述べている．Chameleon は，特に携帯端末とマルチインタラクションコードのアフォーダンスに関して改善点がある．

第 6.2 項で述べた評価実験での，アンケートの自由記述欄には，システムが提供するインタラクションが分かりづらい，カメラが最初に起動した状態では，撮影を促す印象を受けるとの意見があった．複数のインタラクションを統合し，一つの画面から起動すると，機器選択の効率性が向上するが，選択時の混乱を与える時がある．よって，画面に利用可能なインタラクションの表示やイ

ンタラクションを促すインタフェースの再設計が必要である．

また，マルチインタラクションコードは，矢印は振る動作を促す印象を受ける，現在のコードは，Touch できるか分かりづらい，との意見があった．ビジュアルマーカや，QR コードなど普及しているコードは，機能とインタラクションが一対一対応である．よって，複数の機能をコードが持つと，混乱を与えると分かった．Rukzio[5] は，一つのコードに，Touching，Pointing，Direct Input を促す絵を描き，複数のインタラクションをまとめている．また，この研究をもとに MultiTag Project[6] が進められている．しかし，コードに関する発行や管理の手間に関しては議論されていない．先行研究を参考に，コードの再設計が求められる．

7.2.2 インターオペラブルな機器連携サービスの実現

今回の実装では，連携を IP アドレスを持つ機器に限定していたが，今後 Bluetooth や Zigbee など異なる通信規格の機器との連携により，さらに柔軟なサービスが実現できる．そのために，関連研究で述べた機器間のインターオペラビリティな相互接続を支援する uMiddle などのミドルウェアとの親和性を高めていく必要がある．これにより，外出先にある Bluetooth を搭載したオーディオシステムで携帯端末にある音楽を再生するなどの即興的な連携サービスの構築が可能である．

7.2.3 連携時のネットワーク状態の可視化

現在の Chameleon は，接続中の機器とサービスの名前は視覚化され，利用者が理解できる．しかし，背後で動作しているサーバ間や異なるネットワーク間通信などのシステムの流れは閲覧できない．バックグラウンドで動作するシステムの流れや状態を利用者が確認しながら，連携を実行することで，連携に対する理解を深め柔軟なサービス構築が可能となる．例えば，異なるネットワークセグメント間に対して，VPN を構築し安全な接続を確立の支援などが可能である．

謝辞

本研究の機会を与えてくださり，絶えず丁寧なご指導を賜りました，慶應義塾大学環境情報学部教授徳田英幸博士に深く感謝致します．また，貴重なご助言を頂きました慶應義塾大学環境情報学部安村通晃教授，慶應義塾大学環境情報学部准教授高汐一紀博士に深く感謝致します．

慶應義塾大学徳田・高汐・中澤研究室の諸先輩方には，貴重な御助言を頂き，多くの議論の時間を割いて頂きました．特に，政策・メディア研究科特別研究助教伊藤昌毅博士には，本研究の草稿時から多くの助言と励ましを頂きました．また，環境情報学部専任講師中澤仁博士，政策・メディア研究科特別研究助教由良淳一氏，青木崇行氏には，本研究をすすめるにあたって多くの励ましとご指導を頂きました．ここに，深い感謝の意を表します．

くわえて，５年間もの長い間研究生活をともに過ごした，伊藤友隆氏，河田恭兵氏，生天目直哉氏，また徳田義幸氏，山本純平氏をはじめとした HORN 研究グループ，KMSF 研究グループのメンバー，私生活において多くの助言と罵倒を頂いた梅野剛平氏をはじめ，その他多くの友人には深く感謝します．

最後に，６年間の学生生活を常に励まし続け支えてくれた家族に深く感謝し，謝辞と致します．

2010 年 2 月 10 日

橋爪 克弥

参考文献

[1] Y. Ayatsuka, N. Matsushita, and J. Rekimoto. Gaze-link: A new metaphor of real-world oriented user interface. *IPSJ Journal*, Vol. 42, No. 6, pp. 1330–1337, 2001.

[2] Y. Ayatsuka and J. Rekimoto. tranSticks: physically manipulatable virtual connections. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 251–260. ACM New York, NY, USA, 2005.

[3] Rafael Ballagas, Jan Borchers, Michael Rohs, and Jennifer G. Sheridan. The smart phone: A ubiquitous input device. *IEEE Pervasive Computing*, Vol. 5, No. 1, p. 70, 2006.

[4] C. Beckmann, S. Consolvo, and A. LaMarca. Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. *Lecture notes in computer science*, pp. 107–124, 2004.

[5] G. Broll, J. Hamard, M. Paolucci, M. Haarlnder, M. Wagner, S. Siorpaes, E. Rukzio, A. Schmidt, and K. Wiesner. Mobile interaction with web services through associated real world objects. In *MobileHCI '07: Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*, pp. 319–321. ACM, 2007.

[6] Gregor Broll, Enrico Rukzio, Massimo Paolucci, Matthias Wagner, Albrecht Schmidt, and Heinrich Hussmann. Perci: Pervasive service interaction with the internet of things. *IEEE Internet Computing*, Vol. 13, pp. 74–81, 2009.

[7] Steven Feiner, Blair Macintyre, Dor ＼'eeSeligmann. Knowledge-based augmented reality. *Commun. ACM*, Vol. 36, No. 7, pp. 53–62, 1993.

[8] H.W. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mobile Networks and Applications*, Vol. 7, No. 5, pp. 341–351, 2002.

[9] Katsuya Hashizume, Kazuhiro Imura, Kyohei Kawada, Naoya Namatame, Tomotaka Ito, Masaki Ito, Jin Nakazawa, Kazunori Takashio, and Hideyuki Tokuda. Swing, snap and stamp it!: Device interaction with fun. In *11th International Congerence On Ubiquitous Computing*, 2009.

[10] http://panasonic.jp/viera/link/index.html. パナソニック株式会社 viera link.

[11] http://www.apple.com/jp/support/bonjour/. Bonjour.

[12] http://www.cubiq.jp. ユビキタスサービスプラットフォーム技術の研究開発　usdl.

[13] http://www.dlna.org/. Digital living network alliance.

[14] http://www.jini.org/. Jini.org.

[15] http://www.toshiba.co.jp/digital/regzalink/. 東芝 regzalink.

[16] http://www.upnp.org. Universal plug and play forum.

[17] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 234–241, New York, NY, USA, 1997. ACM.

[18] Tomotaka Ito, Katsuya Hashizume, Kyohei Kawada, Naoki Nakagawa, Naoya Namatame, Masaki Ito, Jin Nakazawa, Kazunori Takashio, and Hideyuki Tokuda. Snappy: A snap-based human interaction for multiple device collaboration. Advances in Pervasive Computing. Adjunct proceedings of the 7th International Conference on Pervasive Computing (Pervasive 2009)., 5 2009.

[19] 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.

[20] N. Kawaguchi. Cogma: A middleware for cooperative smart appliances for ad hoc environment. In *Proc. of International Conference on Mobile Computing and Ubiquitous Networking (ICMU2004)*, pp. 146–151, 2004.

[21] J. Kela, P. Korpip, J. Mntyjrvi, S. Kallio, G. Savino, L. Jozzo, and S.D. Marca. Accelerometer-based gesture control for a design environment. *Personal Ubiquitous Comput.*, Vol. 10, No. 5, pp. 285–299, 2006.

[22] C.D. Kidd, R. Orr, G.D. Abowd, C.G. Atkeson, I.A. Essa, B. MacIntyre, E. Mynatt, T.E. Starner, W. Newstetter, et al. The aware home: A living laboratory for ubiquitous computing research. *Lecture notes in computer science*, pp. 191–198, 1999.

[23] Naohiko Kohtake, Jun Rekimoto, and Yuichiro Anzai. Infopoint: A device that provides a uniform user interface to allow appliances to work together over a network. *Personal Ubiquitous Comput.*, Vol. 5, No. 4, pp. 264–274, 2001.

[24] Manu Kumar, Andreas Paepcke, and Terry Winograd. Eyepoint: practical pointing and selection using gaze and keyboard. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 421–430, New York, NY, USA, 2007. ACM.

[25] Jennifer Lai, Anthony Levas, Paul Chou, Claudio Pinhanez, and Marisa Viveros. Bluespace: personalizing workspace through awareness and adaptability. *Int. J. Hum.-Comput. Stud.*, Vol. 57, No. 5, pp. 415–428, 2002.

[26] M.J. LaLomia and J.B. Sidowski. Measurements of computer satisfaction, literacy, and aptitudes: A review. *International Journal of Human-Computer Interaction*, Vol. 2, No. 3,

pp. 231–254, 1990.

[27] P. MAES. Intelligent software agents vs. user-controlled direct manipulation : A debate. *ACM SIGCHI97 Extended Abstracts*, 1997.

[28] Rodolfo La Maestra. Hdmi - a digital interface solution. *HDTV Magazine*, 2006.

[29] Jin Nakazawa, W. Keith Edwards, Hideyuki Tokuda, and Umakishore Ramachandran. A bridging framework for universal interoperability in pervasive systems. In *in Pervasive Systems", ICDCS 2006*. IEEE Computer Society, 2006.

[30] Jin Nakazawa and Hideyuki Tokuda. Smart space computing. コンピュータソフトウェア, Vol. 21, No. 3, pp. 213–223, 20040525.

[31] J. Nielsen. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pp. 152–158. ACM New York, NY, USA, 1994.

[32] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[33] D.A. Norman. *The design of everyday things*. Basic Books New York, 2002.

[34] D.A. Norman, 野島久雄. 誰のためのデザイン. pp310-312, 新曜社, 1990.

[35] Timo Oaja and Hannu Kukka. A digital city needs open pervasive computing infrastructure. In *Digital Cities 6: Concepts, Methods and Systems of Urban Informatics Workshop at the 4th International Conference on Communities and Technologies*, 2009.

[36] Tadashi Okoshi, Shirou Wakayama, Yousuke Sugita, Soko Aoki, Takeshi Iwamoto, Jin Nakazawa, Tomohiro Nagata, Daichi Furusaka, Masayuki Iwai, Akihiko Kusumoto, Noriyuki Harashima, Nobuhiko Nishio, Yoshito Tobe, Yasushi Ikeda, and Hideyuki Tokuda. Smart space laboratory project: Toward the next generation computing environment. In *in International Workshop on Networked Appliances*, 2001.

[37] Jun Rekimoto. Pick-and-drop: A direct manipulation technique for multiple computer environments. In *In Proceedings of UIST'97*, pp. 31–39, 1997.

[38] Jukka Riekki, Timo Salminen, and Ismo Alakarppa. Requesting pervasive services by touching rfid tags. *IEEE Pervasive Computing*, pp. 40–46, 2006.

[39] Enrico Rukzio, Karin Leichtenstern, Vic Callaghan, Paul Holleis, Albrecht Schmidt, and Jeannette Chin. An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. *UbiComp 2006: Ubiquitous Computing*, pp. 87–104, 2006.

[40] Ben Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, pp. 33–39, New York, NY, USA, 1997. ACM.

[41] 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.

[42] K. Tsukada and M. Yasumura. Ubi-finger: Gesture input device for mobile use. In *Proceedings of APCHI*, Vol. 1, pp. 388–400. Citeseer, 2002.

[43] Takuro Yonezawa, Hiroshi Sakakibara, Jin Nakazawa, Kazunori Takashio, and Hideyuki Tokuda. Spot & snap : An interaction for associating sensor nodes and everyday objects to realize diy smart object service(ubiquitous computing,¡special issue¿interaction technologies : From principles to applications). *Transactions of Information Processing Society of Japan*, Vol. 48, No. 3, pp. 1381–1392, 20070315.

[44] 暦本純一. 実世界指向インタフェース : 実空間に拡張された直接操作環境 (¡特集¿仮想と現実の融合). 情報処理, Vol. 43, No. 3, pp. 217–221, 20020315.

[45] 南正輝, 森川博之, 青山友紀. ネットワークサービスシンセサイザのデザイン. 電子情報通信学会技術研究報告. SSE, 交換システム, Vol. 100, No. 670, pp. 1–8, 20010309.

[46] 増井俊之. 実世界 gui による情報家電プログラミング. 情報処理学会研究報告. HI, ヒューマンインタフェース研究会報告, Vol. 2000, No. 61, pp. 1–6, 20000706.

付録 A　評価用紙

複数のインタラクションによる機器連携サービスについて

名前：

年齢

性別：

コンピュータの知識（初級者，中級者，上級者）

機器連携によるサービス利用について

使い方を習得するのは簡単だ	1　2　3　4　5
やっていて非常にイライラする	1　2　3　4　5
したことはほとんど間違っていたのではないかと思う	1　2　3　4　5
必要だと思ったことが全部出来る	1　2　3　4　5
やっていて楽しい	1　2　3　4　5
機器の連携は直感的に行えましたか	1　2　3　4　5
インタラクションは直感的に行えましたか	1　2　3　4　5

近傍の機器を連携させることに関して

選択をするのが面倒だ	1　2　3　4　5
使い方を習得するのは簡単だ	1　2　3　4　5
やっていて非常にイライラする	1　2　3　4　5
したことはほとんど間違っていたのではないかと思う	1　2　3　4　5
必要だと思ったことが全部出来る	1　2　3　4　5
やっていて楽しい	1　2　3　4　5

（遠隔の）移動先で（元の部屋の）機器を連携させることに関して

選択をするのが面倒だ	1　2　3　4　5
使い方を習得するのは簡単だ	1　2　3　4　5
やっていて非常にイライラする	1　2　3　4　5
したことはほとんど間違っていたのではないかと思う	1　2　3　4　5
必要だと思ったことが全部出来る	1　2　3　4　5

やっていて楽しい　1　2　3　4　5

知らない機器を探して連携させることに関して

選択をするのが面倒だ　1　2　3　4　5
使い方を習得するのは簡単だ　1　2　3　4　5
やっていて非常にイライラする　1　2　3　4　5
したことはほとんど間違っていたのではないかと思う　1　2　3　4　5
必要だと思ったことが全部出来る　1　2　3　4　5
やっていて楽しい　1　2　3　4　5

良かった点

悪かった点

気づいたこと

ご協力ありがとうございました。