

ユーザインタフェースの遠隔操作による機器間連携

根岸 佑也^{*1} 河口 信夫^{*1,*2}

^{*1}名古屋大学大学院情報科学研究科 ^{*2}名古屋大学情報連携基盤センター
A Rule-based Remote Macro System using Graphical User Interface

Yuya Negishi^{*1} Nobuo Kawaguchi^{*1,*2}

^{*1}Graduate School of Information Science, Nagoya University

^{*2}Information Technology Center, Nagoya University

1. はじめに

近年、日常生活の場において情報家電などのネットワークに対応した機器が増加している。また、一人で PDA やノート PC などの複数の計算機を同時に利用する場面も一般的になりつつある。このような複数の機器に囲まれた環境において、我々は、ある機器に対する操作の後に他の機器も操作するといった利用方法を日常的に行っている。例えば、電話に出たらテレビの音量を下げる操作が挙げられる。このような一連の操作は繰り返し行なわれる機会が多い。多数の機器を快適に利用するためには操作をルールとして記録し、似たような状況において自動的に機器を操作するシステムの構築が望ましい。特に、今後身の回りの機器がますますネットワーク対応となることが予測されるため、新しく買ってきた機器に対してもこれまでの機器と連携可能であることが望ましい。

このようなマクロシステムの実現手段としては、プログラミングなどの負担がなく手軽な PBE(Programming by Example)によるアプローチが適している。PBE システムは、ユーザが計算機に与える例から一般性を持つプログラムの自動生成を行う概念であり、様々な研究がなされてきた [1][9][11]。既存の PBE システムは、例示操作の記録や適用時に対象アプリケーション固有の API に依存するものが多く、未知の機器やアプリケーションへの対応には向いていない。一方、画面イメージのようなアプリケーションの視覚的な外観の利用により、API に依存しないアプローチが Potter の Triggers[10]、山本の AutoMouse[2]として提案されている。しかし、これらのシステムは全て、例示と適用の対象が同一の機器やアプリケーションに限定されており、複数の機器に対する連携操作の自動化を行うことは考慮されていない。

本稿では機器がネットワークで結ばれた環境において、ユーザインタフェースの遠隔操作により、任意の機器やアプリケーション間において連携を実現するシステムを提案する。本手法では、操作対象の機器をターゲットと呼び、その遠隔操作を行う機器をマスタと呼ぶ。ターゲットはマスタに対してユーザインタフェースを公開する。マスタはターゲット上のイベントを監視し、事前に定義されたルールに基づく操作を行う。ルールはイベントと操作の系列から定義される。イベントとしては、画面イメージからのパターン発見や、マウスやキーボード操作を用いる。ルールの定義は、マスタ上に表示された各ターゲット機器の画面に対する例示操作により行う。イベントや操作は複数の機器にまたがって定義できるため、本手法を用いれば機器間の連携操作を容易に実現できる。

本手法に基づく機器間連携システムを、遠隔操作ツールとして VNC(Virtual Network Computing)[12] を利用し、Java を用いて実装した。本システムを用いることにより、異なる OS 上の異なるアプリケーション間において、連携操作を容易に定義し利用可能にする。

本稿の構成は以下の通りである。まず第 2 節で複数機器環境におけるユーザの操作支援について検討し、第 3 節では本稿で提案するユーザインタフェースの遠隔操作による機器間連携手法について述べる。第 4 節において、第 3 節で提案した手法に基づく機器間連携システムの実装と利用例について述べ、第 5 節において提案手法について考察し、第 6 節でまとめ、今後の課題を挙げる。

2. 複数機器環境におけるユーザの操作支援

複数の機器に囲まれた環境において、手間のかかる一連の操作を高い頻度で繰り返し行うことはユーザにとって苦痛であることが想定される。例えば、(1)就寝時に照明をオフ、(2)テレビをオフ、(3)エアコンのタイマーをオンということを毎晩行っている場合、図 1 に示すように最初の照明操作をトリガーとして残りの操作を自動的に行って欲しい。

このようなルールに基づく機器間の連携操作システムの構築を行う際、対象となる機器やアプリケーションに対してユーザが行った操作内容の取得、再現を外部から行う必要がある。そのために、特定の連携 API を利用しては、任意の既存の機器やアプリケーションや新しく買ってきたものに対してシステムを適用させることが困難である。

そこで、我々はどのような機器にもユーザインタフェースは存在していることに着目し、特定の連携 API の存在を前提としない複数機器の操作記録・適用手法を考案した。対象となる機器としては PDA、PC などの計算機は当然とし、情報家電などの情報機器がネットワークによって結ばれた環境を想定する。この他に、センサネットワークなど小型制御デバイスも含む。ただし、それらは PC などを通じてそれらが持つ情報をユーザが閲覧・操作可能な GUI アプリケーションや Web インタフェースが提供されていることを前提とする。提案手法では、これらが持つユーザインタフェースを通じて任意の機器に対する連携操作を自動化する。

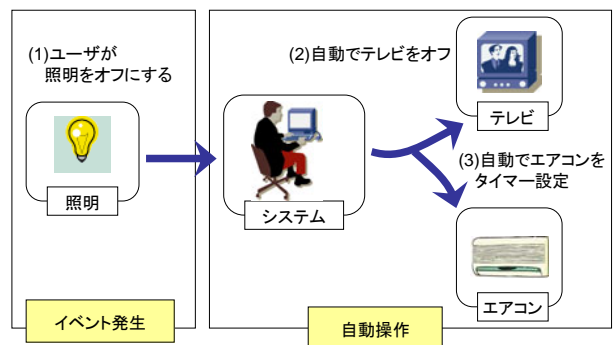


図 1: 連携操作の例

3. ユーザインタフェース経由の機器間連携システムの設計

3.1. 要求事項

前節で述べたユーザインタフェース経由での機器間連携を実現するためには、以下の2つの機能が要求される。

1. 任意の対象機器やアプリケーションに対して、依存性のない操作手順(ルール)を記述可能
2. システムが連携操作の対象となる複数の機器をまたがって監視・操作可能

3.2. ユーザインタフェースに基づく機器操作ルール

要求事項1を満たすために、具体的に以下のようなユーザインタフェースを用いて操作ルールを記述する。

- ・ 画面イメージなど視覚情報
- ・ マウスやキーボードなど Human Interface Device

これらを用いることにより、機器固有の機能を用いずにルールを定義可能になる。例えば、メール着信条件を図2中の画像パターンAの発見に、音楽プレーヤの再生を画像パターンBに対するマウス操作に置き換えることができる。また、この表現方法は対象OSやアプリケーションに制約がないだけでなく、Webページにも応用可能である。

提案するシステムでは、一連のユーザインタフェース操作ルールを Behavior と呼び、Behavior を構成する次の2つのコマンドを定義した。セマフォに関する命令は、マウスやキーボード操作を排他制御するため導入している。

- ・ **Wait Command**
指定したイベントの処理条件が満たされるまで待機する命令。詳細を表1に示す。
- ・ **Operation Command**
指定した対象機器上の Human Interface Device を操作する命令。詳細を表2に示す。

具体例として、IP電話アプリケーション Skype と音楽プレーヤ iTunes 間にて、電話をかけようとした時に音楽プレーヤを自動的に一時停止させる Behavior を図3に示す。この例では、1番目のコマンドにて Skype のアイコンに対するユーザの操作を監視し、2番目で iTunes の一時停止ボタンを探索し、3番目で発見した領域に対して左クリック操作を行う。

表1: Wait Command 一覧

イベントの種類	指定条件	説明
Image Found	イメージ	ディスプレイ上に指定イメージを発見するまで待機
Keyboard Event	キーコード	指定キーが押されるまで待機
Mouse Event	ボタン番号, イメージ	指定イメージに対してクリック操作されるまで待機
Semaphore Wait	セマフォ名	指定したセマフォに対してWait命令を実行

表2: Operation Command 一覧

操作の種類	説明
Keyboard Operation	キーボード操作
Mouse Operation	マウス操作
Clipboard Transfer	クリップボードを機器間でネットワーク転送
Semaphore Signal	指定したセマフォに対してSignal命令を実行

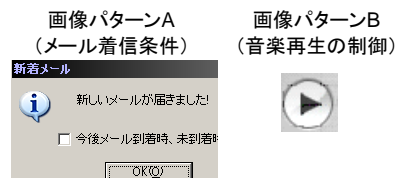


図2: 画像パターンの例

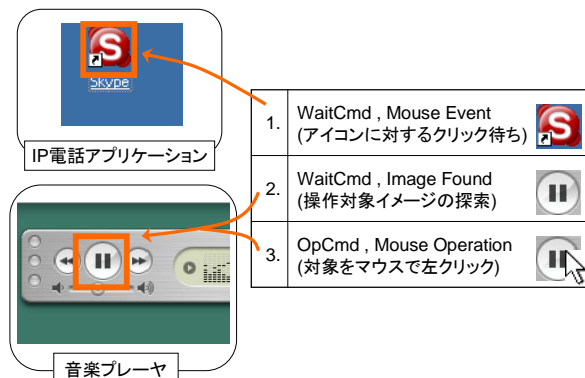


図3: IP電話と音楽プレーヤを排他利用する Behavior の例

3.3. 遠隔操作を利用した機器間の監視と制御機構

前節で設計したルールを機器間にまたがって適用するためには、3.1節で述べた要求事項2を実現しなくてはならない。本手法ではそのためにネットワーク経由の遠隔操作を利用し、機器やアプリケーションのユーザインタフェースを監視・操作することを実現する。

まず、遠隔操作を行う機器とされる側の機器として、次のように提案システムを稼働させるマスタ機器と、自動操作の対象となるターゲット機器の関係を定義する。

<マスタ機器>

- ・ ターゲット機器のユーザインタフェースを遠隔操作するツールをユーザに提供
- ・ 操作ルール(Behavior)の管理
- ・ ルールに基づき、ターゲット機器のユーザインタフェースを監視、遠隔操作

<ターゲット機器>

- ・ 外部から操作可能なユーザインタフェースの公開

具体的には VNC を利用し、Behavior で用いるユーザインタフェースの画面イメージを取得、マウスやキーボードの遠隔操作を行う。このために、各ターゲット機器に VNC サーバを設置し、マスタ機器に VNC ビューワを表示する。このため、本手法ではターゲット機器に対して VNC サーバもしくは VNC が利用する RFB プロトコルに準ずる仕組みを導入することで、マスタ機器からユーザインタフェースを遠隔操作可能にする。これは自動操作の対象とする機器のソフトウェア内部に何ら変更を要求しないので、既存だけでなく未知の機器やアプリケーションに対しても適用可能である。

また、Behavior において、ユーザインタフェースの監視の対象や、遠隔操作の対象となるターゲット機器を別々の機器として設定すれば、イベント発生元と操作対象を複数の機器間において処理可能になる。このようにマスタ機器を仲介

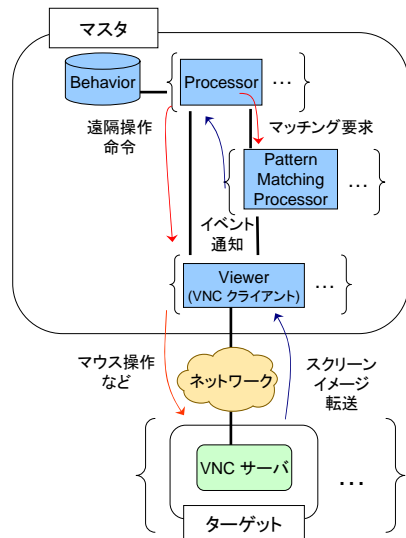


図 4: システム構成

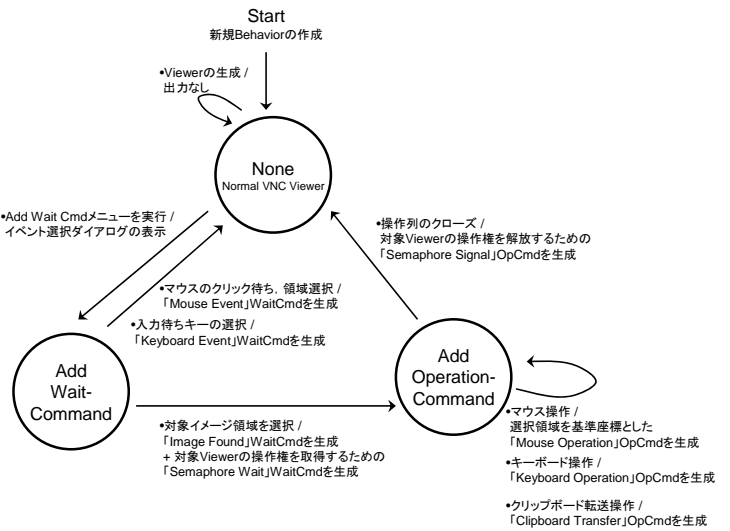


図 5: Behavior 生成の状態遷移

してイベント処理を行うことで、機器間に連携するための機構が存在しなくても、任意の機器同士に連携した自動操作を実現可能になる。

以上のことを踏まえて、提案する連携支援システムの構成を設計した。その構成を図 4 に示す。図 4 中のマスタ機器上の Processor は Behavior に記述されているコマンドを処理していくコンポーネントである。PatternMatchingProcessor は Wait Command 中にあるイメージパターンの探索命令を Processor から要求されて処理する。Viewer はターゲット機器上に設置された VNC サーバに対応する VNC ビューワであり、Operation Command が示す遠隔操作を Processor より指示されて、ユーザに代わり自動操作を行う。以下に Processor が行うイベント処理の流れを示す。

1. 読み込まれている Behavior から次のコマンドを取得。
2. WaitCommand ならば、PatternMatchingProcessor にイメージパターン探索を要求し、条件を満たすイベントが通知されるまで待機。
OperationCommand ならば Viewer 経由で遠隔操作。
3. 次のコマンドを処理するために 1 へ。
Behavior の終端に達したならば次に読むべきコマンドを先頭に戻し、1 へ。

Processor はマスタ機器上に複数個存在しても構わない。そのため、同時に並行して複数の Behavior を処理可能である。ただし、一つのターゲット機器上のマウスやキーボードは同時に複数の Processor から遠隔操作を行われると競合して誤動作を起こす可能性がある。そのため、セマフォなどを用いて遠隔操作の権利を管理する必要がある。なお、ここではユーザや他のマスタ機器の操作との競合を考慮しない。

3.4. 遠隔操作の実演によるルール of 記録

システムは、ユーザが行ったマスタ機器上の各 Viewer に対する例示操作から半自動的に Behavior を生成する。このため、Behavior の中身をユーザが直接記述する必要がなく、プログラミングなしに手軽に機器間連携のマクロ化を実現できる。図 5 に Viewer に対するユーザの操作を入力とした自動生成のための状態遷移を示す。

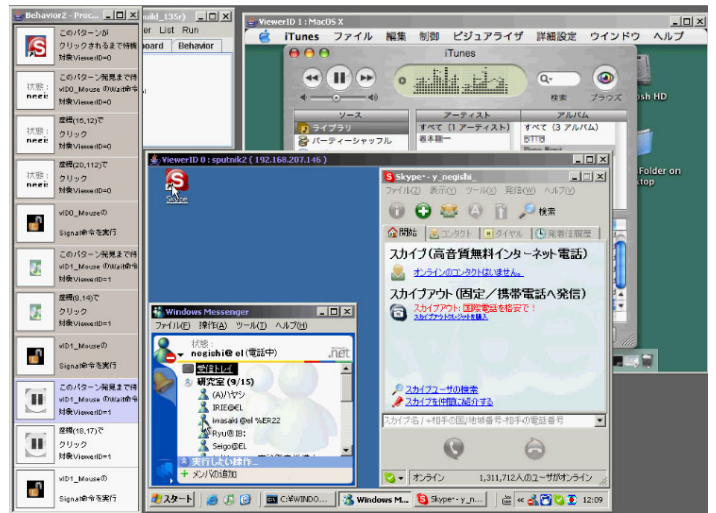


図 6: ルール適用時のマスタ機器の画面

4. 実装と利用例

3 節での設計に基づき、機器間連携システムを実装した。マスタ機器のプロトタイプ実装には Java(JDK1.5.0)と機器間連携ミドルウェア cogma[3]を利用した。cogma は、河口らが開発した動的なネットワークにも対応する機器間連携ミドルウェアである。Viewer の実装にあたっては cogma のコンポーネントの一つである EmbCodget の枠組みを利用することで、ネットワーク内のターゲット端末の発見と IP アドレスなど VNC セッション開始に要する情報を動的に自動収集する[7]。ターゲット機器側には各プラットフォーム向けの VNC サーバと、その VNC サーバの情報をマスタ機器に通知するための codget を搭載する。動作確認 OS は Windows XP, MacOS X, Linux である。

異種 OS と異種アプリケーション間での利用例として、図 3 で示した例を拡張した「(1)機器 A 上の IP 電話アプリケーションを起動したら、(2)機器 A のメッセージの状態を電話中に設定し、(3)機器 B の音楽プレーヤを一時停止する」連携のスクリーンショットを図 6 に示す。図 6 は Windows XP(機器 A)と MacOS X(機器 B)の Viewer を表示した状態の

マスタ機器の画面である。このように任意のアプリケーションや機器間に連携を手軽に定義、適用できる。

5. 考察

5.1. データ転送を伴う連携に関する制約

本手法を用いることによって、ユーザインタフェースに関するイベント駆動型の連携を実現することができる。しかし、機器間連携にはイベント通知による連携以外に、ある機器上のデータを別の機器に転送して、転送先でデータの加工を行うといったものも考えられる。例えば、カメラ内の画像データをプリンタに送信し、印刷する連携が考えられる。提案システムにおいて、この種の連携を実現するために、Operation Command にクリップボードのネットワーク転送をサポートしている(表 2)。ただし、データをクリップボードに貼り付け、取り出しを行うためには対象アプリケーションにその機能がユーザインタフェースとして提供されている必要がある。また、転送できるデータもユーザインタフェース経由で扱える形式でなければならないという制約もある。このように本手法で可能なデータ転送を伴う連携には限界が存在する。対処方法としては、OCRのように指定した画像イメージからテキストや数値データを取り出し、転送する方法が挙がる。これが実現できれば、より柔軟に連携が可能になると考えられる。

5.2. コンテキストの考慮と予測機能の必要性

本稿では、ユーザの機器操作支援を目的としてシステムを設計、構築することを述べてきたが、予測インタフェース[1]に関しては何も考慮していない。一方、状況に応じて次どのような機器操作をユーザが希望するかを過去の履歴から推定し、提示するシステム SmartMacro[6]が宮崎らによって提案されている。予測を行うことで、ユーザの意図する機器操作をより正確かつ柔軟に支援することが可能になる。しかしながら、SmartMacro が対象とするアプリケーションや機器は cogma 上で動作し、制御 codget と呼ばれる連携機構を備えている必要がある。この点で一般的なユーザインタフェースによって制御する本システムとは異なるが、ユーザ支援を進める上で、そのようなコンテキストの考慮と予測機能の追加は必要になると考えられる。そのためには、ユーザインタフェース経由による機器操作ルールに対して、どのようにセマンティックを付加するかなど、解決すべき点は多い。

5.3. 操作ルールに関する考察

前節の考察でコンテキストを考慮する必要があると述べた。現状のルールの記述方法では、場合ごとの分岐処理の記述が難しいという問題点が挙がる。設計した Behavior のように、イベントと条件に応じて処理を行うルールの記述形式にはアクティブデータベース向けに考案された ECA ルールが存在する[8]。ECA ルールを用いた機器制御としては早川らによる Ubiquitous Chip (AhroD)[4]や、宮前らによる A-WEAR[5]がある。ECA ルールのように、ルールの有効/無効化などを管理することで、柔軟な分岐処理の実現を期待できる。ただし、これらの研究で用いられている ECA ルールは、あらかじめ設計者が内容を決めて直接記述している。そこで、本稿で提案した機器間をまたがるユーザインタフェースの監視・操作機構を ECA の形式によって制御する場合、例示操作の内容から自動的に ECA ルールを生成する手法の考案や、操作の系列をどの程度の粒度で一つの ECA ルールにまとめるかを十分検討することが必要であると考えられる。

6. おわりに

本稿では機器間に対する冗長な操作の自動化を行うシステムを構築する際に、任意の機器に対応させるためにユーザインタフェースを利用した機器間の連携操作の実現手法を提案した。その特徴としては、特定の連携機構を必要としないこと、遠隔操作によるイベントの監視・操作の適用を行うこと、操作の例示による操作ルールの作成を行うことが挙げられる。そして、提案手法に基づくプロトタイプを実装し、異種 OS・異種アプリケーション間をまたぐ連携操作を手軽に実現できることを示した。

今後の課題としては、より多くのユーザインタフェースに対応できる操作ルールの検討を行いたい。また、提案したシステムを応用し、状況に応じた機器操作をユーザに提示する予測インタフェースについて検討していきたい。

参考文献

- [1] 増井俊之：予測/例示インタフェースシステムの研究，学位論文，東京大学，(1997)。
- [2] 山本格也：GUI を API として用いるプログラミング法，情報処理学会論文誌：プログラミング Vol.39 No.SIG 1 (PRO 1)，pp.26-33 (1998)。
- [3] 河口信夫，稲垣康善：cogma:動的ネットワーク環境における組み込み機器間の連携用ミドルウェア，情報処理学会コンピュータシステム・シンポジウム，pp.1-8 (2001)。
- [4] 早川敬介，塚本昌彦，寺田努，義久智樹，岸野泰恵，柏谷篤，板根裕，西尾章治郎：ユビキタスコンピューティングのための入出力制御デバイス，WISS 2002，(2002)。(http://www.nishio.ise.eng.osaka-u.ac.jp/tresearch/Ubiquitous/index.html)
- [5] 宮前雅一，寺田努，塚本昌彦，平岡圭介，福田登仁：バイクレース支援のためのイベント駆動型ウェアラブルシステム，情報処理学会研究報告 Vol.2004,No.44 MBL-29，pp.53-58 (2004)。
- [6] 河口信夫，宮崎俊和，稲垣康善：ユビキタス情報環境における履歴を用いた機器操作支援手法，映像情報メディア学会技術報告 Vol.2004,No.39 UBI-4，pp57-62，(2004)。
- [7] 根岸佑也，河口信夫：多様な接続手法に対応したデスクトップ画面共有システム，平成 16 年度電気関係学会東海支部連合大会，(2004)。
- [8] 小島功：アクティブデータベースの動向と応用へのインパクトについて，情報処理学会研究報告：ソフトウェア工学 Vol.1994, No.099，pp.73-78 (1994)。
- [9] Cypher,A.(ed)：Watch What I Do -Programming by Demonstration, The MIT Press, Cambridge, (1993)。(http://www.acypher.com/wwid/)
- [10] Richard Potter：Triggers: Guiding Automation with Pixels to Achieve Data Access, Chapter17, In Cypher[9], (1993)。
- [11] Henry Liberman：YOUR WISH IS MY COMMAND, PROGRAMMING BY EXAMPLE, Morgan Kaufmann Publishers, (2001)。
- [12] Tristan Richardson, RealVNC Ltd(formerly of Olivetti Research Ltd / AT&T Labs Cambridge)：The RFB Protocol Version3.8, (2004)。(http://www.realvnc.com/documentation.html)