

## ポップアップダイアログの認識に基づく対応支援手法

上島 愛史 †1 梶 克彦 †2 河口 信夫 †2 竹中 光 †3 武内 重樹 †3 岡本 学 †3

†1 名古屋大学工学部

†2 名古屋大学大学院工学研究科

†3 日本電信電話株式会社 NTT サービスエボリューション研究所

## 1 はじめに

パソコンを操作する過程では、様々な種類のポップアップダイアログ（以下ダイアログ）が表示される。それらはわかりにくい表現や専門用語で情報を提示していたり、ユーザに複数の操作選択を強いたりする場合が多い。また、それらのダイアログが表示されている間は、他の操作を受け付けられないことが多いために、ユーザはその対応を煩雑だと感じることもある。特に普段からパソコンをあまり操作しないような、いわゆるパソコン初心者には、ダイアログの操作は難易度の高いものである。

これまでも、ダイアログに対する研究として、ユーザに対して親和性の高い、ストレスレスな表示方法等の研究 [1][2] が行われている。本研究では、ダイアログ出現時に、ユーザに操作案内を提示するという方法で、ストレス軽減や作業効率の向上を狙う手法の確立を目指す。

図 1 に本手法の概要を示す。本手法ではまず、複数のサンプルデータ提供者からダイアログと操作の情報を収集する。さらに、同じダイアログに対して収集された複数のデータを比較して、ダイアログが発生してから閉じられるまでの、異なる操作パターンを単一のデータとして纏める。最終的に支援対象ユーザには、そのデータから案内文を生成して提供する。

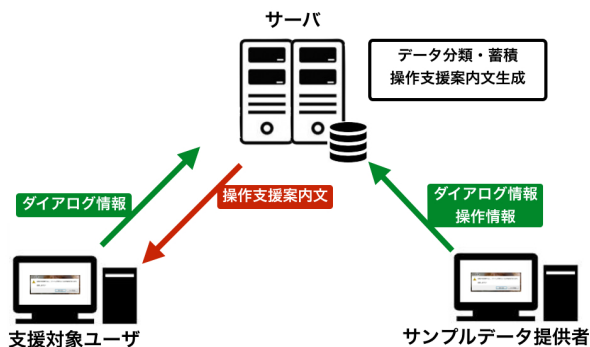


図 1: 本研究システムの概略図

## 2 ダイアログ含有情報の収集と管理

ダイアログの情報の収集を行うために、ダイアログの出現の検知と、ダイアログが持つ情報の抽出、及びユーザの操作情報の収集を行う。また、サーバ側では、支援対象ユーザに提供する案内文の生成のために、収集した情報を管理する必要がある。

## 2.1 ダイアログ発生検出

ダイアログの出現を検知するために、WindowsAPI[3]を用いて、全ウィンドウハンドルの生成を監視する。Windows 上においてダイアログはすべて、#32770 というクラス名によって管理されているため、このクラス名を用いてフィルタリングを行えば、ダイアログの発生を検出できる。

## 2.2 ダイアログの含有情報の収集

ダイアログの含有情報とは、ダイアログ内に表示されているテキストやボタン等のオブジェクトと、ダイアログ本体のビットマップデータである。

まず、それぞれのテキストやボタンなどのオブジェクトは、ダイアログ本体を親とするウィンドウハンドルを保持している。また、テキストであれば *Static*、ボタンであれば *Button* などの、特定の名称を持つクラスに属する。本手法では、ウィンドウハンドルの親子関係とクラス情報の、2つのメタデータを保持したままデータを収集する。これは、後にダイアログの同一性の判定を容易にするためである。そのために、ダイアログ内のオブジェクトごとにテキストとして抽出し、JSONデータの形式に従ってメタデータを付与して、ダイアログの表示情報を収集する。図 2 はその一例である。

また、ダイアログそのもののビットマップデータは、ウィンドウハンドルから座標を特定してキャプチャし、JPEG形式に圧縮する。

このJSONデータと画像データには、ダイアログの生成時刻をタイトルに付けて管理する。また、これら2つのデータはダイアログが閉じられたタイミングでサーバに送信する。

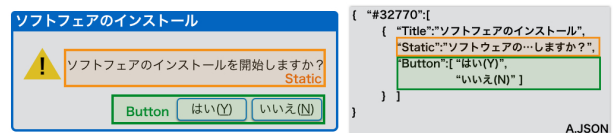


図 2: ダイアログ情報のJSONデータ化の例

## 2.3 ユーザ操作の監視とダイアログの遷移

支援対象ユーザに対して最適な操作案内を提供するには、サンプルデータ提供者がダイアログに対して行う操作の情報と、それによるダイアログの表示内容の遷移を収集する必要がある。

まず、グローバルフック [4] を用いて、クリックされたボタンの特定を行う。各ボタンのウィンドウハンドルに紐付けられる座標情報より、選択されたボタンを特定することが可能である。また、ユーザの操作によって遷移したダイアログの情報も、前節の方法を用いて収集し、JSONデータに追記する。例えば、図 3 のように、ボタンの選択によりダイアログが遷移した場合、JSONデータ内の対象オブジェクトの要素に対して、遷移後のダイアログ情報を収めるJSONデータ名を追記する。

A Method of Response Support Based on Recognition of Pop-Up Dialog  
Aiji Uejima †1 Katsunori Kaji †2 Nobuo Kawaguchi †2 Hikaru Takenaka †3 Shigeki Takeuchi †3 Manabu Okamoto †3  
†1 School of Engineering, Nagoya University †2 Graduate School of Engineering, Nagoya University †3 Nippon Telegraph and Telephone Corporation, NTT Service Evolution Laboratories

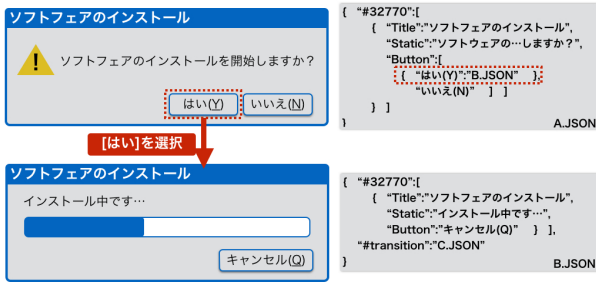


図 3: ダイアログの遷移が起こった場合の例

## 2.4 データ管理方法

2.1 節のダイアログ検出方法では、本研究で取り扱うポップアップダイアログの他に、ファイル選択やオプションウィンドウといった、モーダルウィンドウも検出することになる。これらのモーダルウィンドウとポップアップダイアログの識別をするために、ダイアログ内のアイコンを利用する。確認や警告などの通知ダイアログのほとんどは、特定のアイコンを保有している。そこで、ダイアログの画像データに対して、OpenCV[5]によるテンプレートマッチングを行い、ポップアップダイアログ以外の物を除外する。また、後のデータベースに対する検索を容易にするために、保有しているアイコンごとにディレクトリを分けて管理する。

続いて、ダイアログのタイトルでもディレクトリを分ける。もし、すでに同じタイトルのダイアログのデータが、ディレクトリに保存されていれば、JSON データ内の配列構造とオブジェクトを比較し、完全一致していれば同一であるとみなす。

データが同一のダイアログの情報であると判定できれば、JSON データのマージを行い、ひとつのダイアログに対して異なる選択を行った場合の、ダイアログ情報の遷移を補間していく。その際、マージされた回数を JSON データ内に追記しておき、案内文の生成時に、他のユーザの選択の統計を可視化する。

## 2.5 説明情報の付与

ダイアログがボタン選択以外の操作を要求するような場合、案内文上に説明や注釈があるのが望ましい。ダイアログデータの管理者は必要に応じて、JSON データ内に #note という要素名と任意の文章を追記しておく。これは、案内文生成時に説明文として認識され、案内文上に反映される。

## 3 ユーザに対する操作支援

支援対象ユーザの使用する PC 上で、ポップアップダイアログが発生した場合、その検知及び情報の収集は 2.1-2.2 節の通りを行う。

支援対象ユーザから送信されたデータは、アイコンのテンプレートマッチングによって分類し、同一のダイアログの情報を検索した後、操作支援のための案内文を生成する。予め用意された JSON データをスクリプトによって解析し、案内文となる HTML 文を自動生成する。図 4 に案内文の一例を示す。

案内文内では、各ボタン選択後のダイアログの遷移を可視化する。そして、2.4 節で JSON データに追記された、マージの回数情報を用いることで、他のユーザが一般的にどのような選択を行っているかを、統計として提示する。また、2.5 節で追加入力された、#note のオブジェクトより説明文が提示される。

案内文の生成はリアルタイムで行われるため、ダイアログの遷移状態に応じた案内の提示が可能である。例

えば、何フェーズにも亘るような、複雑なソフトウェアのインストール時では、ブラウザで手順情報等を表示しながら行うが、この手法を用いれば、自動的に提示する情報を更新して、シームレスな操作方法の確認が可能である。

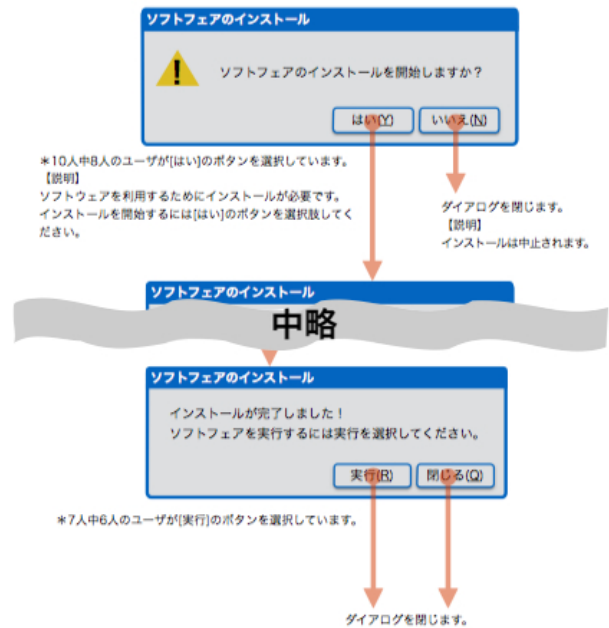


図 4: 案内文の一例

## 4 今後の課題

今後の課題として、ダイアログの同一判定方法の改善が挙げられる。本手法では現在、ダイアログを構成するオブジェクトの情報が完全一致している場合のみに、ダイアログが同一であると判定している。しかし、オブジェクトの情報がシステム環境に依存する場合があります。その場合には本手法では同一判定ができない。そのため、今後同一判定の方法について検討する必要がある。

また実際に、本手法に基づく操作支援サービスを、PC 初心者などに使用してもらい、アンケートをとって、本手法の有効性の評価を行いたい。

## 参考文献

- [1] 三好 史隆, 倉本 到, 渋谷 雄 [他], タスク集中度と認知時間を指標とした周辺表示法の評価, 電子情報通信学会論文誌, 一般社団法人電子情報通信学会, pp.831-839, 2006 年.
- [2] 三好 史隆, 倉本 到, 渋谷 雄 [他], マウス使用環境において周辺表示法がユーザの情報認知に与える影響, 電子情報通信学会技術研究報告, 一般社団法人電子情報通信学会, pp.65-70, 2006 年.
- [3] WindowsAPI, <http://msdn.microsoft.com/en-us/library/cc433218.aspx/>
- [4] SetWindowsHookEx, <http://msdn.microsoft.com/ja-jp/library/cc430103.aspx>
- [5] OpenCV, <http://opencv.org/>