

タスク実施支援システムの構築

6 B-5

山田季史、福原綾介、長崎等、東基衛
早稲田大学大学院理工学研究科 早稲田大学理工学部

1. はじめに

現在のアプリケーションソフトウェアは多機能、複雑化の傾向を示している。かつ、その機能を使いこなせるよう支援する目的のオンラインヘルプやマニュアルの評価はいま一つである。

本研究では、あるタスクを実施しているユーザの、操作履歴を取得、分析することによって、その実施に合わせた適切な情報を提示し、ナビゲーションを行う方法を提案し実装した。

2. 現状の問題点

多くの場合、ユーザはソフトウェアの使用の初期段階においては、成果物の見栄えなどをもとに新たな機能を検索する努力を行うが、ある一定の成果物が得られてしまうと、その成果物に満足してしまい、さらに良い機能があるにも関わらず、それらの機能を探しに行こうという意欲を見せなくなってしまい多機能性の恩恵を受けているとは言い難い。

3. 解決案

現状の問題点を解決する手段として、本研究では、ユーザのタスク実施に合わせて、適切な情報を提供する一種のナビゲーションシステムを構築することで解決することを目指している。

ユーザの操作に合わせた適切な情報として、以下の2つが挙げられる。

- ・ある操作を実施したユーザに、行うことの出来るアクションの候補及び、その操作方法を提示する。
- ・あるアクションを実施するのに2種類以上の方法がある場合、その実施に合わせてその方法を提示する。

4. システムの概要

3で述べた解決案を実現するため、アプリケーションにセンサーを付加することを提案する。この結果システムは図1のようなシステム構成となる。

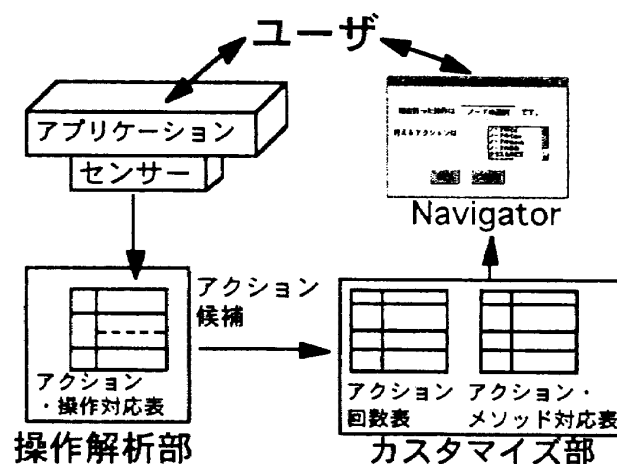


図1 システム概念図

本システムは、あるアプリケーションに対して、センサー、操作解析部、カスタマイズ部、Navigatorで構成される。センサーはユーザの操作履歴を取得する。操作解析部では、履歴を解析し、ユーザの状態を把握する。カスタマイズ部では、その結果を踏まえて、Navigatorに表示する内容を決定する。ユーザはNavigatorの画面をもとに操作を行いシステムを活用する。以下にシステムの具体的な説明を行う。

5. アプリケーションとセンサー

4節のシステムを構築するためには、ユーザが行ったユーザ操作の取得が出来ることが最低限必要である。

5.1 対象ソフトウェア

このシステムは[1]の提案モデルに基づいてJAVA1.1言語で実装されたアプリケーションを対象とする。[1]の提案モデルは以下の図2で示される。このモデルを採用する理由は、イベントレベルでの履歴だけでなく、タスクレベルの履歴も取得できるからである。

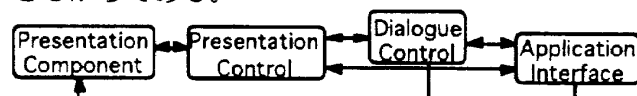


図2 提案モデル

5.2 ユーザ操作の取得

ユーザ操作要素を取得するために、アプリケーションにセンサーを取り付ける。

5.2.1 イベントレベルの操作履歴の取得

Presentation Component と Presentation Control 間の通信には Event Delegation Model が使われている。したがって、イベントの処理をするクラスが <EventType>Sensor を継承するようにする。

そして各 <EventType>Sensor の中で取得したい情報を得るためのメソッドを定義することによってイベントレベルのユーザ操作履歴の取得が可能となる。

5.2.2 タスクレベルの操作履歴

Presentation Control から Dialog Control へメッセージが送られて来た時点で、そのメッセージのタイプを取得することによりタスクレベルのユーザ操作履歴の取得が可能となる。

6. 操作解析部

6.1 アクション・操作対応表の作成

ユーザ操作履歴列とそれに対応するアクション名の対応表を作成するために、タスク分析、機能分析によって得られた機能の各々に対して開発者が実施をし、操作履歴を取得する。そして得られた操作履歴列に対してアクション名を付加しておく。

ノードを作る	Node MOUSE_RELEASED,(82,78),mods=16,clickCount=1 ADDDATANODE GETONENODE CLASSWIZARD SELECTMODE MOUSE_CLICKED,(82,78),mods=16,clickCount=1 Cancel
別方法のユーザ操作要素列	

図3 アクション・操作対応表

6.2 アクションの同定

アクションを確実に同定することはかなり困難である。なぜならユーザが途中で機能の実行をやめたり、機能の実行の間に別の機能を実施してしまう場合が多々存在するからである。しかし本研究では、タスクレベルでの履歴が取れるため、タスクレベルでのメッセージの内容とイベントレベルの内容からアクションの同定が可能となる。

7. カスタマイズ部

ナビゲーションは、ユーザのタスク実施に合わせて適切な情報を提供するのだが、そのナビゲーションは、「目標」、「理由」、「手順」、「予測・失敗」の4つの構成要素から成り立たなければならない。なぜなら、既存のマニュアルやヘルプなどは、目標と手順のみの表示がほとんどであるが、ユーザはなぜその目標状態を提示したのかが不明であり、

その過程でおこるかもしれない失敗等を考えると、ナビゲーションの指示に従おうという意欲が減退するからである。

本研究では「目標」として、ある操作を行ったユーザに、その操作を含むアクション候補を、アクション・操作対応表から検索、提示することで提供する。

「理由」は、色々な要素が考えられるが、本研究では、アクションの同定により把握できるアクションの実施回数に限定した。各アクション回数を保存しておき、その回数によって表示の内容を変える。

「手順」は、各アクションに対して、自然言語で記述したメソッドを対応づけておき、アクションの選択と共に表示するようにする。

「予測・失敗」は、ユーザがアクションを選択した際に、常にシステム側が、ユーザの操作履歴とアクション・操作対応表を監視し、もし違う操作をしたさいにはユーザに知らせることで提供する。

8. 考察及び今後の課題

本論文で示したシステムによってユーザの状態を把握することが可能となった。

また、それを解析することにより、ユーザが行うことの出来るアクションの候補を動的に示すことが可能となり、目的がはっきりしているユーザにはその具体的な操作を提示でき、はっきりとしないユーザにも使用しているアプリケーションにはこんな機能もあるということが示せ、使用の幅、成果物の幅が広がることを可能とした。

現在、適応例として Object Designer というアプリケーション上で実装を進めている。

今後の課題として、現在は各アクション間の関係を記述し、ナビゲーションが行えるようにすることや、今回は JAVA 言語上に特化したシステムであったが他の言語で作られたアプリケーションへの対応方法などが挙げられる。

謝辞 本研究の一部は早稲田大学特定課題研究助成費 96-A-148 及び 97A-163 の補助によるものです。

[1]長崎 等他、適応型ユーザインタフェースを実現するためのシステムアーキテクチャ、情報処理学会第 52 回全国大会