アプリケーションの実装時組み込み型操作手順提示機構による作業効率の向上について

矢田　久美子†　　白石　善明†　　毛利　公美‡
名古屋工業大学†　　岐阜大学‡

1. はじめに

業務の効率化のために情報システムの導入が広範囲に進んでおり，様々な場所で必要に迫られてPCを使用する機会が増加している．情報システムには，操作可能なボタンや入力フォーム等，様々なオブジェクトが存在するため，初めて使用する人がとまどうことも珍しくない．

業務の中には，アプリケーションを用いて所望の事項（以後，タスク）を遂行することで完了するものがある．タスク完了までの操作手順を知る方法に手引書がある．しかし，タスクの完了に手引書が必要な人と，操作手順を覚えている人とを比べると後者の方が作業効率は良い．このような格差はディジタルデバイド[1]の一形態であり，その緩和はオフィスの生産性を向上するための課題の一つである．

これを解決するために，我々は，シナリオファイルで定義した次の操作をインタフェース上に提示する機構 Operation Leader を提案している[2]．本稿では，Operation Leader をアプリケーションに組み込んだ場合，手引書を読みながら操作する場合と比べて作業効率が向上することを評価する．

2. 作業効率の向上する操作手順提示方法

手引書などのタスクの開始から完了までの操作手順を示す方法では，その説明に沿うようにアプリケーション画面と見比べながら実行するので，操作手順を覚えているときと比べて作業効率が低下する．この原因であるアプリケーション画面との見比べ行為をなくしたもの，つまりタスク完了までの操作手順をアプリケーション画面に直接的に示すインタフェースを考える．タスクはオブジェクトを決まった順序で一つずつ操作していくことで完了できる．そこで，次に操作するべきオブジェクトを縁取りしたり，背景色を変えたりと，他のオブジェクトに比べて強調表示してユーザの視線を誘導するといった操作手順の提示を逐次行うインタフェースを考える．

アプリケーションのタスクには，シーケンシャルな手順だけでなく，シーケンシャルでない手順，つまり分岐やループのある手順によって構成されるタスクもある．従って，任意のアプリケーションに対応するためには，そのような操作補助も行えるような機能を持つ必要がある．

これらのタスク完了に至るまでの操作手順は，アプリケーションの設計・開発段階において設計者・開発者によって想定されているため，その一連の手順を設計や開発の段階でシナリオとして定義することは可能である．シナリオとは，タスク完了に至るまでの操作が記述したものである．それは，プロセスという単位で構成され，各プロセスはユーザがその時点で行う操作と，オブジェクトに対する説明の付加などの処理を定めている．

このシナリオを基としてユーザがある操作を実行したとき，次のプロセスに進むことを促す対話的なインタフェースは図1のようになる．

アプリケーションは一般に複数のタスクを遂行する機能を持つので，ユーザの完了したいタスクがわからないと操作手順が提示できない．そこで，ユーザの完了したいタスクを選択してもらい，インタフェース上で操作手順を開始するようにする．

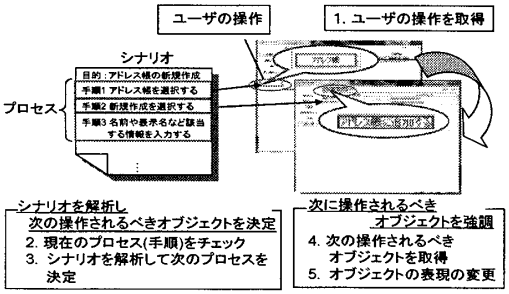

図1 インタフェース上での手順提示の流れ

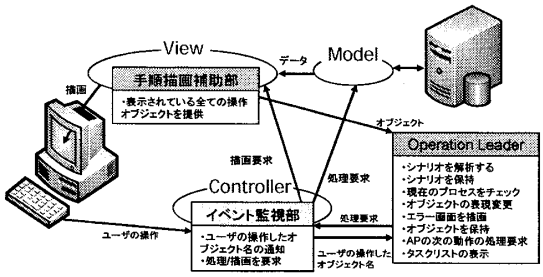

図2 アプリケーション全体の構成

3. 操作手順提示機構 Operation Leader

図1中の1.と4.だけがアプリケーションに依存しており，その他の処理はシナリオを読み込み，必要な情報を得ることができれば独立して処理ができる．この独立して処理が可能な部分である“シナリオの解析”と“オブジェクトの表現の変更”を行う機能を持つ機構を Operation Leader と呼ぶ．Operation Leader はモジュール化してその API を呼び出す形でアプリケーションプログラムに組み込み，手順提示を行うインタフェースを実装できるものである[2]．

操作手順をインタフェース上に提示するためには，ユーザの操作を取得し，アプリケーションの次の動作を実行するイベント監視部と，あらかじめオブジェクトを Operation Leader に提供する役割を持つ手順描画部をアプリケーションに組み込み，Operation Leader と連携を取るという構成になる．GUI を持つアプリケーションの多くが MVC モデル[3]に即していることから，MVC モデルに沿った形で Operation Leader を組み込んだアプリケーションの構成図を図2に示す．

操作手順をインタフェース上に提示するためには，ユーザの操作を取得するイベント監視部と，予めオブジェクトを Operation Leader に提供する役割を持つ手順描画補助部をアプリケーションに組み込み，Operation Leader と連携を取るという構成となる．

アプリケーションに Operation Leader を組み込むためには，アプリケーションプログラムにコードを追加する必要がある．追加するコードは，図3に示したように，「Operation Leader の生成」，「オブジェクトの提供」，「ユーザの操作を取得」のための3種類のコードである．

これらのコードをアプリケーションプログラムに追加・変更することで，シナリオに定義した通りにインタフェース上に手順を提示する．

On Improvement of Work Efficiency by Operation Presentation Mechanism For GUI Application
† Kumiko YADA and Yoshiaki SHIRAISHI ・ Nagoya Institute of Technology
‡ Masami MOHRI ・ Gifu University

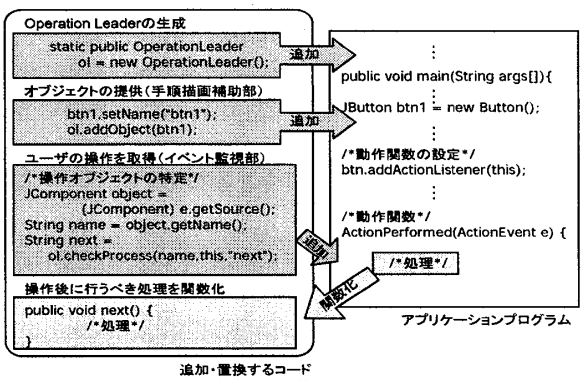

図 3 追加・置換するコード

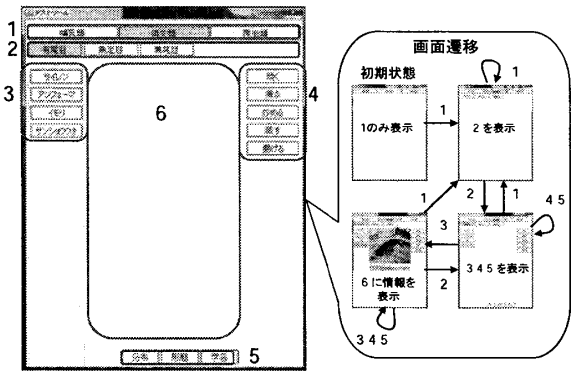

図 4 作成した評価用アプリケーション

4. 作業効率の評価と考察

操作手順を知る方法である手引書と比べて，ユーザにとって Operation Leader が有用であるか，つまりタスクを達成するのにどれだけ効率的になるかをユーザ実験によって評価する．

Operation Leader が有用であれば，操作手順を知る方法の一つである手引書と比べ，ユーザの作業効率も主観的評価も向上すると考えられる．このことから，同じようなタスクについて（A）Operation Leader によって操作手順を示される場合と（B）手引書によって操作手順を示される場合でアプリケーションを操作してもらい，それぞれの作業パフォーマンスを比較した．ここでの作業パフォーマンスとは，各タスクを実行した際の「タスク完了時間（秒）」，「マウス移動距離」，「クリック回数」である．作業パフォーマンスの値が小さい方がより作業効率の良い操作手順を知る方法であると言える．

作業パフォーマンスの測定に必要なテストユーザ数は少なくとも 10 人とされている[4]．そこで，PC の操作に慣れている 12 人を被験者とした．いずれも日常的に PC に触れており，Operation Leader の使用を想定するオフィスの環境に適していると考える．

テストタスクには，ある事柄について階層構造を持つアプリケーションを評価用に Java Swing で作成し，同程度の仕事量である 3 種類のタスクを設定した．作成したアプリケーションを図 4 に示す．ユーザには（A）（B）それぞれで 3 種類のテストタスクを連続して行ってもらった．実験実施の際，同一ユーザ内での（A）（B）の順序が作業パフォーマンスに与える影響を考慮して，その順番をユーザごとにランダムとした．ただし，（A）を先に行った人と，（B）を先に行った人とがそれぞれ 6 人ずつになるよう人数は調整した．各ユーザには，実験終了後に主観的な評価をアンケート解答してもらった．

実験の結果，得られたタスク完了時間，マウス操作距離，クリック回数のデータを表 1 に示す．表中の値はテストタスク 3 種類の作業パフォーマンスの平均値である．

表 1 中のクリック回数以外の値は（B）に比べて（A）の値が小さいことがわかる．特にタスク完了時間は，（A）は（B）の 2 分の 1 程度であることが見てとれる．

クリック回数については，Operation Leader を用いた場合，タスク開始時にタスクの一覧からタスクを選択することで操作手順の提示が開始されるので，1 回分余分にクリックされる．そのため，平均値は（B）の方が小さくなっている．正確に指示された通りに操作したかを測るためには，（A）のクリック回数の値から 1 回分差し引く必要がある．

そのことを踏まえると，作業パフォーマンスの平均値は（B）に比べて（A）の方が小さい．つまり，手引書を使用する場合に比べて Operation Leader を用いた方が作業効率が上がっていること示している．

更に（A）と（B）の作業パフォーマンスに有意差の検定が見られるかの検定を行った結果，1%有意水準において，クリック回数に関しては有意差が見られなかったが，タスク完了時間とマウス操作距離に関しては有意な差があることがわかった．

（A）と（B）を比較し，最も有意差が大きかったものはタスク完了時間である．これは，主に手引書と画面を見比べる必要がなくなったことの効果であるが，さらにユーザが現在の手順をチェックする必要がなくなることも，この結果の一因として考えられる．

表 1 作業パフォーマンスの平均値

	(A)OperationLeader	(B)手引書
タスク完了時間(sec)	8.67	16.23
マウス移動距離(dots)	1502	1828
クリック回数	7.0	6.1

クリック回数に有意差が見られなかったのは，操作手順を知る手段の優劣を比べる実験で，あらかじめ全ての操作手順を示したものをユーザに対して与えている．つまり，操作手順を間違うことは起こらないという考えの下実験を行っているので，妥当な結果だと考えられる．

ユーザのアンケートによる主観評価では，「わかりやすかった」，「使いやすかった」，「今後使っていきたいと思った」という項目で，（A）（B）どちらがよりよい方法であったのかについてアンケートを行った．その結果，12 人全てが全ての項目において（A）Operation Leader を選択した．

5. おわりに

本研究では，アプリケーション開発時にタスクを完了するための操作手順を定義したシナリオファイルをユーザの操作と連動して逐次解析することで，次に操作されるべきオブジェクトを強調し，ユーザに操作手順を提示する機構 Operation Leader を開発した．

ユーザによる評価実験から Operation Leader を組み込んだアプリケーションの方が手引書を用いる場合に比べて作業効率の向上が確認できた．また，主観評価でもわかりやすいという評価が得られた．

今後は，より柔軟な操作手順提示方法が可能なように Operation Leader の改良を行う予定である．

参考文献

[1] 富士通総研，"デジタルデバイドとは(digital divied)"，http://jp.fujitsu.com/group/fri/report/cyber/basic/words/did.html ，2010/1/14 アクセス

[2] 矢田久美子，白石善明，毛利公美，"アプリケーションの実装時組み込み型操作手順提示機構の提案と評価"，第 8 回情報科学技術フォーラム論文集，RO-002，pp.101—108，Sep.2009

[3] Steve Burbeck, "Applications programming in Smalltalk-80(TM): How to use Model-View-Controller(MVC)", Smalltalk Webring, http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html, 1992

[4] Jakob Nielsen，篠原稔和，三好かおる，ユーザビリティエンジニアリング原論 ユーザのためのインタフェースデザイン，東京電機大学出版局，1999