

3S-9

状態遷移表を用いた メニュー操作管理システムの開発

里山 元章^{*1}、三浦 英敦^{*2}、近藤 恵^{*1}

*1 (株)日立製作所 システム開発研究所

*2 (株)日立製作所 ソフトウェア工場

1. はじめに

ユーザの習熟度や過去のユーザインタフェース環境との互換性等の要因により、ユーザのユーザインタフェースに対する要求は様でない。ユーザは自分の好みのユーザインタフェースを持っており、そのユーザインタフェースに変更してプログラムを使いたいという要求がある。

しかし、GUI (Graphical User Interface) のようなユーザインタフェースの場合、その処理が複雑なことや、アプリケーションも一緒に修正しなければならないことから容易には変更できない。

例えばメニューの場合、その色や文字フォントなど外観の仕様は変更できても、操作仕様については、手続きとしてプログラム中に組み込まれていた。このため、メニューの操作仕様を変更する場合、プログラムを書き直す必要があった。

我々は、ユーザインタフェースとして最もよく利用されているメニューに着目し、その操作仕様の定義に状態遷移表を使うことを特徴とするメニュー操作管理システムMMP/MIU (Man Machine Platform / Menu Interface management Unit)を開発した。本稿では開発したMMP/MIUのメニュー操作定義方式を中心に報告する。

2. MMP/MIUの構成

MMP/MIU (以下、MIUと略す)の構成を図1に示す。MIUは、表示画面やマウスカーソルの状態を管理する表示管理部と入力データを管理するイベント管理部、ウィンドウ操作を管理するウィンドウ操作管理部、メニュー操作を管理するメニュー操作管理部からなる。

ユーザの選択操作は、ウィンドウシステムからイベント管理部をとって物理イベント(マウス押下など)に変換され、ウィンドウ操作管理部かメニュー操作管理部へ送られる。メニュー操作管理部は、物理イベントを論理イベント(コマンド)に変換し、

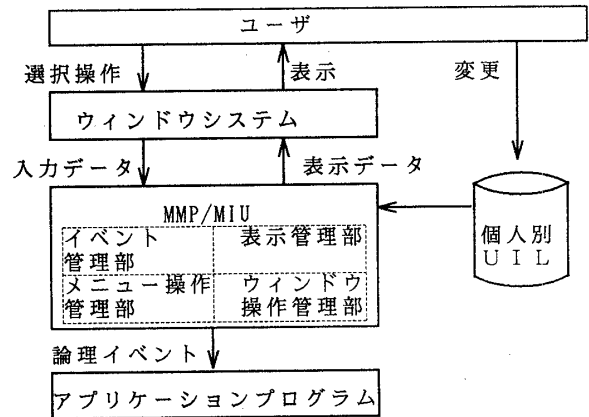


図1. MMP/MIUの全体構成

個人別に管理しているUIL (User Interface Language)の定義に従ってアプリケーションへ送る。

このため、MIU環境下のアプリケーションは、論理イベントの集合をUILを使ってMIUに登録しておき、メニューの操作や外観の仕様に依存しない処理を行う。

3. メニュー操作管理

メニュー操作とは、視覚的に表現された物を選択することでアプリケーションにコマンドを指示する操作である。我々は、メニューの仕様を、メニューを選択する時のマウスやキーボードの操作方法を決める操作仕様と、メニュー項目の位置や色などを決める外観仕様とに分け、これら2つを別々に記述するようにした。

本稿では、以下、操作仕様に記述について述べる。

3.1 操作の記述方法

メニュー操作をアプリケーションの外で実行するには、メニュー操作の定義を手続き的な言語によって記述し、実行中に定義した手続きを呼び出す方法と、メニュー操作の定義を状態遷移表で記述し、実行中にデータとして呼び出す方法が考えられた。

我々は、状態遷移表でメニュー操作を記述するこ

とにした。これは、状態遷移表が、手続き的な言語に比べ記述能力が低いものの、宣言的に記述でき、視覚的な編集系が作りやすいこと、手続き的な言語より実現が容易なことによる。

3.2 ソフトウェア構造

図2にM I Uのソフトウェア構造を示す。ユーザの選択操作によって発生する入力データは、イベント管理部によってマウスカーソルの位置（ウィンドウ内、メニュー内など）を付加した物理イベントに変換され、操作管理部に送られる。操作管理部は物理イベントと状態遷移定義表から状態遷移を行う。このとき、表示管理部に表示状態の更新を要求する。メニュー項目が確定するとコマンド表から論理イベントを作りアプリケーションに送る。

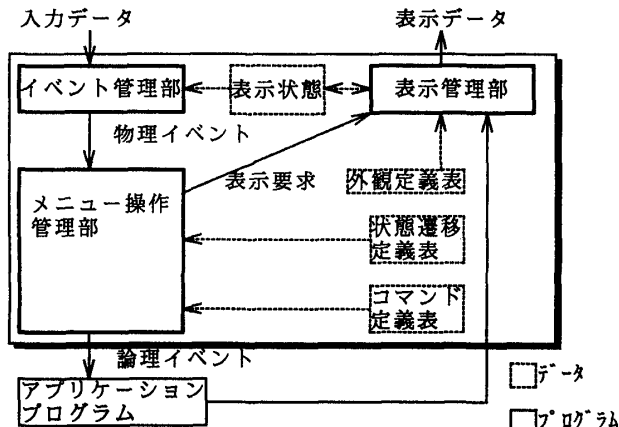


図2. ソフトウェア構造

3.3 状態遷移モデル

メニュー操作は、マウスボタンを押した状態でドラッグし、メニュー項目の上でボタンを離すことで選択する操作（Macintoshなど）と、マウスボタンをクリックすることでメニューを表示し、再度、メニュー項目をクリックすることで選択する操作（Motifなど）のどちらかがよく使われる。

上記二つのメニュー操作が、最低限表現できるように、図3のような状態遷移図でメニュー操作を表現した。

ここで、状態遷移時の出力は、遷移後の状態と常に1対1対応するようにし、遷移後の状態で出力を同時に表現する。

- (1) 非表示状態：メニューが画面に表示されていない状態。この状態に遷移したタイミングでメニュー画面を消去する。
- (2) 表示状態：メニューは表示されているがどの項目も選択されていない状態。この状態に遷移したタイミングでメニュー画面を表示する。
- (3) 選択状態：どれか1個のメニュー項目が指示

されている状態。この状態に遷移したタイミングで項目指示中の画面を表示する（どれか1つの項目の背景と文字の表示色が反転している）。

- (4) 確定状態：メニュー項目のどれか1個の選択が確定した状態。この状態に遷移したタイミングで、アプリケーションに論理イベントが送られる。

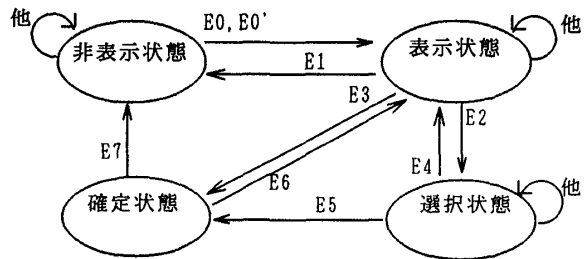


図3. メニューの状態遷移

3.4 メニュー定義の例

表1に状態遷移表の例を示す。例1はマウスボタンを押すとメニューが表示され、離すと消えるタイプのメニューである。例2はマウスボタンを押すとメニューが表示され、そのまま離しても消えないタイプのメニューである。

表1のE0～E7は図3の矢印に対応する。

	例1		例2	
E0	マウスボタン2押下	APウィンドウ	マウスボタン2押下	APウィンドウ
E0'			エスケープキー	
E1	マウスボタン2離す	メニュー外	マウスボタン2押下	メニュー外
E2	マウスボタン移動	メニュー内	マウスボタン移動	メニュー内
E3			ショートカット文字	
E4	マウスボタン移動	メニュー外	マウスボタン移動	メニュー外
E5	マウスボタン2離す	メニュー内	マウスボタン2押下	メニュー内
E6	固定指定あり		固定指定あり	
E7	固定指定なし		固定指定なし	

*) APウィンドウ：アプリケーションが使用するウィンドウ

4. おわりに

メニュー操作をアプリケーションから分離し、従来はアプリケーション開発者が開発時に決めていたメニューの仕様を状態遷移表を使って運用時に変更できるようにした。

現在、M I Uは試作運用中である。この開発で、アプリケーションの実行中においても状態遷移表を記述することで、メニューの操作仕様を変更できる本方式の実用性を確認した。

参考文献

[1] 萩谷昌己：グラフィカルユーザインタフェースとその開発環境について，bit Vol.21 No.6 pp.791-802 (1986-6)