

テクニカルノート

ユーザインタフェース変換の方法とホストアプリケーションプログラムへの適用

相 原 達†

通常、ユーザインタフェースの変更にはアプリケーションプログラムの書換えが必要である。その書換えを少なくして、変更されたユーザインタフェースを提供できるユーザインタフェース変換の方法を述べる。アプリケーションプログラムの状態をユーザインタフェースを通じて推察し、対応する別のユーザインタフェースに変換してユーザに提供する。ユーザの操作はアプリケーションプログラムの状態の遷移に変換される。S/370 上で作成されたパネル形式のユーザインタフェースを提供するホストアプリケーションプログラムへの適用例について、ユーザインタフェース変換に必要な情報の種類と変換システムの実現方法について述べる。

User Interface Mapping Methods and Their Applications to Host Programs

TORU AIHARA †

This paper describes the methods of user interface mapping which enable applications to provide new user interfaces with no or little modifications to the application itself. The user interface mapping system infers the application states from the user interfaces, converts the user interfaces to new ones, presents them to the user, handles the user's interactions, and converts them to the transitions of the application states. This paper also describes two experiences where these user interface mapping methods are applied to the S/370 terminal emulation interfaces.

1. はじめに

グラフィカルユーザインタフェース (GUI) が普及する一方で、既存のアプリケーションプログラム (AP)、特に 3270 端末インタフェースを提供する S/370 系の AP を今後どのように扱っていくかがシステムインテグレーションの現場では問題となっている。AP が古いために書換えが困難であったり、信頼性維持のために変更が好まれなかったりする。

ユーザインタフェース (UI) を自由に変更するために UI 管理システム (UIMS) の研究がなされてきた¹⁾。また、同等の機能を提供するコントロールを自由に置き替えるしくみも提案されている^{2), 3)}。これらの研究では汎用的な切り口を定義することで AP と UI を分割することを目的としてきた。

本稿では、AP の状態遷移^{4), 5)}に注目し、UI などを監視・操作し、新しい UI を提供する UI 変換システムを提案する。そして、3270 端末インタフェースを提供す

る AP への適用事例について報告する。

2. UI 変換

2.1 変換可能な条件

UI をシステムの提供する状態の一表現であると考え、あるシステムの状態の集合を S_0 とし、そのうち変換対象となる状態の集合を S とすると、 $S_0 \supseteq S$ となる。AP の状態 s を UI i に変換できるためには以下の 3 条件を満足しなければならない。

[条件 1] 状態 s_i ($s_i \in S$) すべてについて、UI i_1 が導出できる。

[条件 2] UI i_1 上での操作 $o_{1,j}$ すべてが、状態 s_1 の遷移 $t_{1,j}$ に変換できる。

[条件 3] 状態 s_1 に遷移 $t_{1,j}$ を適用した結果のすべての状態 s_j について、 $s_j \in S$ となる。

2.2 状態の融合

状態 $s_i \in S$ ($i=1, 2, \dots, n$) の集合 S_a ($S \supseteq S_a$) がさらに以下の 2 条件を満足する時、融合可能な状態 s_a とみなすことができる。ここで、 S_a に含まれない状態と S_a に含まれる状態との関係から、 S_a への入り口

† 日本アイ・ビー・エム (株) 東京基礎研究所
IBM Research, Tokyo Research Laboratory

となる状態集合を S_{ain} , S_a からの出口を持つ状態集合を S_{aout} と呼ぶことにする. ($S_a \supset S_{ain}$, S_{aout})

[条件 4] S_a 内部での遷移が既知であり, 入り口状態 S_{ain} , 出口状態 S_{aout} も既知である.

[条件 5] すべての状態 $s_j \in S_a$ について, どの出口状態 $s_k \in S_{aout}$ への遷移が可能かつ既知であり, 中途の遷移状態はすべて S_a に含まれる.

3. 3270 端末インタフェースへの適用

3.1 用語の説明

3270 端末画面は複数のフィールドによって構成される. ホスト AP で扱う文字列データがひとつのフィールドに対応する. ホスト AP と 3270 端末は, 3270 データストリーム (3270 DS) を送受信することで UI を実現している. フィールドの情報は 3270 DS に含まれる. しかし, 多くのホスト AP は, 直接 3270 DS を扱わず, 各フィールドデータと 3270 DS との変換を行う UI サブシステム (DPS や SDF) を使用する. その場合, 画面レイアウトをマップとして事前に定義・登録する. 本稿で取り扱う 3270 端末インタフェースと GUI の構成要素との対応は参考のため付録に示す.

3.2 3270 端末画面への適用

画面の状態を AP の状態と直接対応付けた場合, キー入力が状態の遷移に対応する. UI 変換によって以下の機能追加が可能である.

[機能 A] AP の操作とは直接関係のない機能 (クリップボードへの文字列複写など) を追加する.

[機能 B] 3270 端末インタフェースで使用されるキーの配列・機能を交換する.

[機能 C] 3270 端末インタフェースで使用されていないキーやマウス操作に機能を割り当てる.

[機能 D] メニューバー, ポップアップメニュー, ツールボックスなどを提供し, それぞれのコントロールに機能を割り当てる.

この適用例の機能は, 端末エミュレーションプログラムによっても提供されている⁹⁾.

3.3 3270DS への適用

3270 DS を直接扱えば, AP の状態に対応する 3270 DS を受信し, 状態遷移にあたる 3270 DS を送信することができる. 1 文字ごとのキー入力が状態の遷移ではなく, アテンション ID (AID) キー入力から次の AID キー入力の間が一状態となる. 次の UI 変換も可能となる.

[機能 E] 3270 フィールド編集の方法 (カーソルキーの動作など) を自由に変更できる.

さらに, AP 画面ごとの特長を利用して, AP の状態

を認識することにより, 以下のようなインタフェースの変更が可能になる.

[機能 F] 画面のレイアウトの変更. 不必要情報の削除.

[機能 G] 提供するコントロールの変更. (例えば, 入力フィールドを地名選択のリストボックスにする.)

これらの UI の変換には, AP の状態ごとに以下の情報が必要となる.

[情報 1] 画面の認識手段 (左上に表示される画面 ID など).

[情報 2] 3270 端末インタフェースの構成 (フィールドごとの意味情報など).

[情報 3] 変換後に提供する UI の定義.

端末エミュレータの提供する API を利用することで, 3270 DS を解析し, UI の変換システムを実現できた⁹⁾.

この適用例では, AP の状態ごとに情報を定義する必要がある. そのため, システム全体の変換は困難で, 情報 1 はアドホックに定義することになる. しかし, 変換対象の AP をしぼり, ユーザごとに UI をカスタマイズする場合には有効であった.

3.4 マップへの適用

DPS などのマップを使用すれば, 3.3 節の機能を提供するのが確実・容易になる. 情報 1 はマップ ID によって明らかになる. また, 情報 2 も, マップに意味情報を付加することで作成可能である. システムインテグレーションのように大規模なシステム移植の場合には有効であった⁹⁾.

一方で, DPS は S/370 上で稼働するため, インタフェース変換システムから使用できないので, DPS のエミュレータを作成する必要がある. また, AP も DPS エミュレータを使用するように端末との通信部分に変更が必要となる.

3.5 AP の複数状態の融合

AP の動作に精通していれば, 画面の遷移と遷移後の画面が予測できる. 画面の遷移を UI 変換システムで監視をすることで, 複数画面の遷移を一状態とみなし, それらの画面を統合して次の機能を提供できる.

[機能 H] AP の情報提示の順番や, 選択などのユーザの操作の順番などを入れ替えて, 見かけのダイアログを変更する. (例えば, メニューをたどり機能を決めてから対象物を選択していたダイアログを, 対象物, 機能の順に決定するダイアログに変更できる. また, 画面の大きさの制限から複数画面にわたって提供している情報をスクロールする一画面で提供することもできる.)

この変換には、さらに次の情報が必要である。なお、これらは、状態融合した状態を一状態とみなして処理するために、それぞれ情報 1, 2, 3 を自然に拡張したものとなっている。

〔情報 4〕 状態融合した状態へ入ったことの認識手段。

〔情報 5〕 状態融合した状態間での状態遷移とそれを監視する仕組み。

〔情報 6〕 状態融合した状態に対応して提供される UI。

UI 変換システムは、融合状態の遷移を監視する仕組みとそれぞれの融合状態内の中間的な状態から得られる情報の記憶域を持つことで実現できた⁷⁾。

この適用例では、提供される AP の機能は、融合状態ごとに異なる。そこで、AP が提供する機能からユーザが使用するものだけを限定して、使用しない機能を削除した、カスタマイズした UI を提供するのに使用した。UI 変換で扱う機能を限定することで、状態融合の条件が満たされやすくなり、また、ダイアログの変更により、ユーザにとって必要な機能をユーザ好みの UI で提供することができた。一方で、一般的に適用すると、監視装置および情報記憶域の構成が複雑となったり、また、状態融合が不可能になる傾向がある。

4. ま と め

UI 変換の方法と適用条件とを論じ、3270 端末インタフェースへの適用事例について考察した。ユーザに提供する UI を元の UI から大きく変更するためには、より多くの UI の知識と AP の知識が必要となる。そのため、本稿の方式は、変換対象となる AP が提供する機能を限定できる場合には特に有効である。今後の課題としては、AP の状態、UI、操作を記述する汎用表現の整備と、すでに GUI を提供する Windows AP に対する UI 変換の適用方法とを考えている。

参 考 文 献

- 1) 橋本 治：ユーザインタフェース管理システムの研究動向と将来，情報処理，Vol. 33, No. 11, pp. 1331-1339 (1992)。
- 2) 増田英孝，笠原 宏：アプリケーション実行時 GUI レイアウト変更機能，情報処理学会論文誌，Vol. 35, No. 9, pp. 1794-1806 (1994)。
- 3) 増田英孝，笠原 宏：実アプリケーションを利用したユーザ自身による GUI レイアウト変更実験，情報処理学会論文誌，Vol. 36, No. 1, pp. 129-138 (1995)。
- 4) 横井尚子，兼子 毅，飯塚悦功：状態遷移を考慮

した新しいソフトウェア設計法，電子情報通信学会論文誌 D-I, Vol. J 76-D-I, No. 5, pp. 200-209 (1993)。

- 5) 小野康一，丸山勝久，深澤良彰：プログラム変更に対する正当性検証技法と分割技法の適用，電子情報通信学会論文誌 D-I, Vol. J 77-D-I, No. 11, pp. 747-758 (1994)。
- 6) 日本アイ・ビー・エム (株)：IBM パーソナル・コミュニケーションズ/3270 バージョン J 4.0-Windows 導入と使用の手引き，SC 88-3107-00 (1994)。
- 7) Ogata, M., Aihara, T., Kishi, N. and Lien, Y.-C.: Tools for Mapping User Interfaces, *Proceedings of the 3rd International Conference on Human-Computer Interaction*, pp. 635-642 (1989)。
- 8) 相原 達：システム・インテグレーションにおけるユーザ・インタフェースの検討，第 48 回情報処理学会全国大会論文集 (5)，2 J-9, pp. 233-234 (1994)。

付録 3270 端末インタフェースと GUI の対応

3270 端末インタフェースは、ホスト AP から送信された 3270 DS を解釈して画面を生成し、ユーザが操作して最終的にアテンション ID (AID) キーと呼ばれるキー (Enter, Clear, PF キー, PA キー) を押すことで、それまでの操作によって変更されたフィールド情報、カーソル位置が AID キーの種類とともに 3270 DS として AP に送信される。これらの情報はフィールドによって構成され、それぞれのフィールドはフィールド属性、拡張属性によって性質が決定される。これらのフィールドは以下のような GUI のコントロールに対応している。

- (a) テキスト・フィールド (変更不可)
- (b) 数字入力フィールド
- (c) シングルバイト文字 (SBCS) 入力フィールド
- (d) ダブルバイト文字 (DBCS) 入力フィールド
- (e) SBCS, DBCS 混合文字入力フィールド
- (f) 非表示 SBCS 入力フィールド
- (g) 非表示 DBCS 入力フィールド
- (h) 非表示 SBCS, DBCS 混合文字入力フィールド

非表示は、パスワードなど機密のため表示を避ける場合に使用される。その他のコントロールは、青・緑・白・赤・桃・黄・空の 7 色の色情報を持つ。

ただし、本稿では 80 文字 24 行の画面サイズを持ち、グラフィックスやライトペンや罫線は使用しない標準的な環境を対象とする。また、メッセージの到着などユーザ操作とは非同期なシステムのメッセージについて

でも議論しない。

さて、8100 情報システムの DPPX (Distributed Processing Programming eXecutive) 分散サービスでは、ユーザが入出力で使用する画面のレイアウトは DPS (Distributed Presentation Service) マップとして定義されている。DPS のインタフェースでは、3270 DS の代りに、より抽象化された情報 (マップ ID とフィールド情報とカーソル情報) を使用する。これは、CICS/BMS (Customer Information Control System /Basic Mapping Support) での SDF (Screen Definition Facility) が提供するオンラインパネルでも同様である。

(平成 7 年 4 月 25 日受付)

(平成 7 年 6 月 12 日採録)



相原 達 (正会員)

1960 年生、1983 年東京大学工学部電気工学科卒業、1985 年同大学院電子工学修士課程修了。同年日本アイ・ビー・エム (株) 入社。マイクロ・メインフレーム結合の研究開発、PC によるユーザインタフェースの改善・カスタマイズに従事する。現在は、ノート PC のユーザインタフェースについて関心を持っている。IEEE, ACM 各会員。