

「モックアップ・ソフトウェア」によるラピッド・プロトタイピング環境の実現

3H-8

(株)日立製作所 システム開発研究所  
 (株)日立製作所 システム開発研究所  
 (株)日立製作所 神奈川工場  
 (株)日立製作所 情報システム工場  
 日立京葉エンジニアリング(株)

近藤 博文  
 山下 廣太郎  
 小野田 貴  
 中村 正雄  
 竹内 秀典

1. はじめに

実システム開発以前に、実システムの操作環境を模擬して実際の利用者に直接操作してもらい、体験的な操作性の事前検証を可能にするモックアップ・ソフトウェアの研究を進めている[1][2][3]。

多種の入出力デバイスを有する専用端末では、操作者一端末間の会話形式で処理を進めるという点から、モックアップ・ソフトウェアによる要求定義支援が有効である。文献[3]では、専用端末開発以前での検証を目的に、汎用端末を用いた要求定義支援方式を提案した。モックアップ・ソフトウェアを利用することにより、要求定義のためのプロトタイピングが可能となる。本論では、モックアップ・ソフトウェア方式によるラピッド・プロトタイピング環境の実現手段について述べる。

2. モックアップ・ソフトウェア方式の概要

汎用端末を用いて、(1)仕様表現のビジュアル化、(2)仕様定義の試行錯誤の迅速化、(3)データ定義の簡素化、(4)ユーザ要求への即応化、を実現することを要求定義支援の基本方針とした。モックアップ・ソフトウェアの基本機能としては、以下がある。

(1)仕様実行機能

テーブルウェアの定義に基づく入力情報の解釈→画面遷移の実行という会話処理の定式化により、多数の入出力デバイスの動作を模擬する。

(2)仕様定義機能

標準仕様の可変要素をパラメータ化し、会話画面対応のテーブルウェアに対するデータ定義を可能にする。

(3)動的仕様変更機能

仕様実行中の状態を保持したまま仕様定義に制御を切替え、テーブルウェアの動的仕様変更を可能にする。

(4)個別処理組込機能

テーブルウェアへの固有変数の登録、変数間の演算式定義により、個別処理の簡易な組込を可能にする。

(5)テーブルウェア管理機能

上記機能を実現するのに必要なテーブルウェアを共通的に管理する。

3. ラピッド・プロトタイピング環境の実現

モックアップ・ソフトウェアによるラピッド・プロトタイピング環境の実現には、以下が課題である。

(1)実行時制御アルゴリズムの開発

仕様実行機能では、入力情報の解釈→画面遷移の

実行という形で会話処理を定式化する。そのアルゴリズムを実行時制御アルゴリズムと呼ぶ。仕様実行機能を実現する実行時制御アルゴリズムの開発が必要となる。

入力情報の解釈→画面遷移の実行を以下のように実現する実行時制御アルゴリズムを開発した(図1)。

(a) 入力情報の解釈

キー入力に対して、入力キー情報を参照し、どのような入力があったのかを判断する。次に、現画面の画面遷移情報を参照し、イベント判定を行なう。イベントが成立しなければ、入力待ち状態を保持する。イベントが成立したならば、遷移先

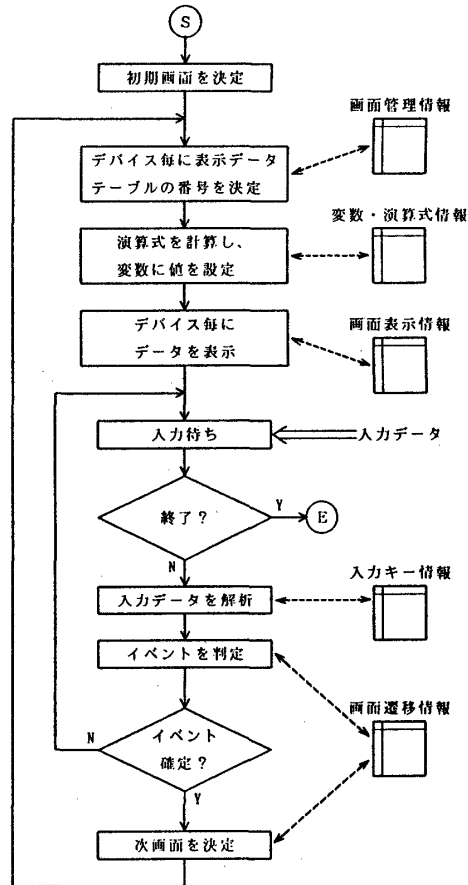


図1 モックアップ・ソフトウェアの実行時制御アルゴリズム

の画面番号を取得する。

(b) 画面遷移の実行

イベントが成立した場合の遷移先の画面番号に基づき、画面管理情報を参照する。画面管理情報には、画面番号毎に、その画面で表示すべき画面表示情報へのリンク情報がデバイス毎に設定されている。画面管理情報を参照して得た画面表示情報を表示することにより、画面遷移を引き起こす。この時、画面表示情報に含まれる変数については、変数・演算式情報を参照し、その値を得る。各デバイス毎に必要な情報を表示して、画面遷移を完了する。

(2) 仕様定義エディタの開発

仕様定義機能では、画面定義や変数・演算式定義など、仕様定義を効率よく行うための仕様定義エディタが必要となる。

以下のサブエディタ群により構成される仕様定義エディタを開発した。

(a) 画面管理エディタ

画面識別情報として、画面番号と画面番号毎の画面名、および画面表示情報へのリンク情報を定義する。

(b) 画面遷移定義エディタ

登録画面毎に、画面遷移を引き起こすイベントおよびそのイベントによる遷移先画面の番号を定義する。

(c) 画面表示定義エディタ

登録画面毎に、表示する表示情報をデバイス対応に定義する。

(d) 変数・演算式定義エディタ

登録画面毎に、変数名、変数の属性、変数の組合せによる演算式を定義する。

(e) 入力キー定義エディタ

入力情報の解析を行なうために、入力キーとイ

イベントとの対応を定義する。

(3) 仕様実行状態の保存と回復

動的仕様変更機能は、仕様実行を一旦中断し、仕様定義に制御を切替え、仕様定義終了後、中断した仕様実行状態を再現することにより実現する。そのためには、仕様実行状態の保存と回復の手段が必要となる。

以下の要領により、仕様実行状態の保存と回復を可能とした。

(a) 仕様実行状態の保存

仕様実行が中断された時、以下の情報を記憶する。

- ①仕様実行を中断した時点の画面番号
- ②前画面までの画面遷移による画面表示結果
- ③前画面までの画面遷移による演算実行結果

(b) 仕様実行状態の回復

②を表示した後、③を初期値として、①に対する演算式を実行する。その実行結果を参照し、①に対する画面表示情報を表示する。

4. おわりに

システム仕様をビジュアルな形で事前検証するモックアップ・ソフトウェア方式によるラピッド・プロトタイプリング環境(図2)の実現手段について述べた。

<参考文献>

[1] 山下、川岡：操作性事前検証システム「モックアップ・ソフトウェア」の基本構想，情報処理学会第36回全国大会講演論文集 pp.2269-2270 (1988)  
 [2] 川岡、山下：操作性事前検証システム「モックアップ・ソフトウェア」の実現方式，情報処理学会第36回全国大会講演論文集 pp.865-866 (1988)  
 [3] 近藤、山下：「モックアップ・ソフトウェア」による要求定義支援方式の提案，情報処理学会第39回全国大会講演論文集 pp.1529-1530 (1989)

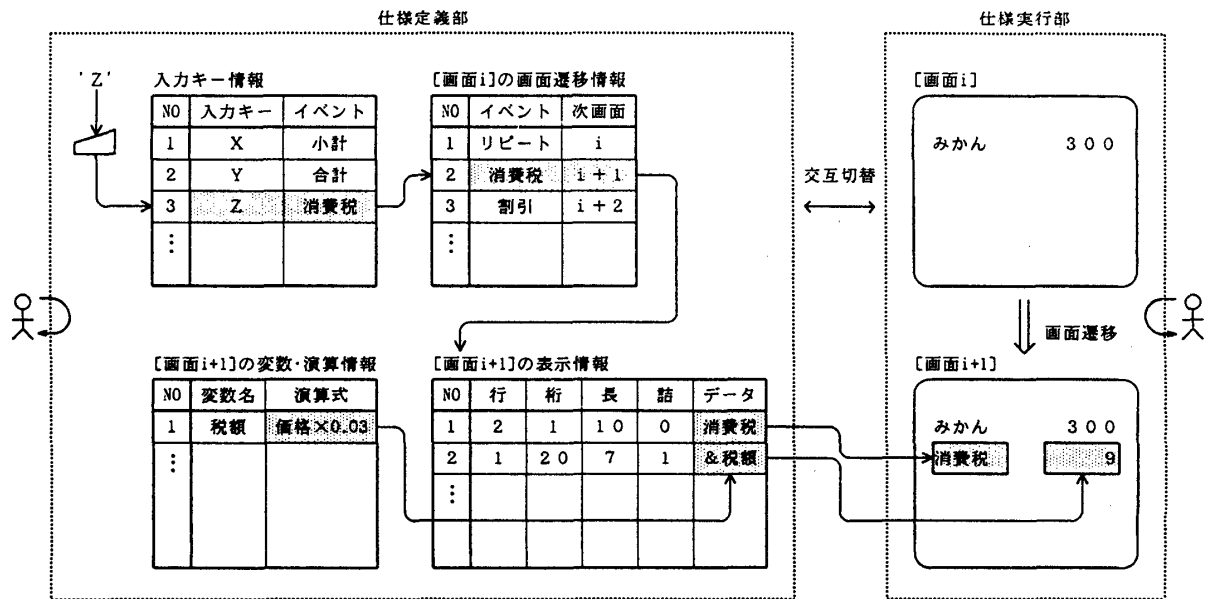


図2 モックアップ・ソフトウェアによるラピッド・プロトタイプリング環境