

# マイコン用CASEツール：testCASE (2) ～要求分析ツール～

7D-2

中島 毅\*、石川 洋\*、小林 剛\*、別所 雄三\*、広田 和洋\*\*

三菱電機（株）情報技術総合研究所\* 三菱セミコンダクタソフトウェア株式会社\*\*

## 1. はじめに

マイコン用CASEツールのtestCASEの中で、要求分析ツール(testCASE/RAと呼ぶ)は、最上流の要求分析段階を支援する。testCASE/RAは、顧客の要求をビジュアルに記述し、顧客の要求を誤りなく抽出することを支援する。さらに、抽出されたビジュアルな要求仕様を論理的なデータとして、テスト項目生成ツールへ渡す。

## 2. 要求分析作業の流れ

図1は、要求分析段階における作業プロセスとそこでツールがどのように使用されるかを示す模式図である。要求分析段階は、要求仕様作成と外仕作成の2つの作業からなる。要求仕様作成作業は、仕様入力と静的エラーのチェックと顧客レビューの3つの作業からなり、要求仕様の正しさが確認されるまで繰返えされる。

仕様入力作業では、開発者が、状態遷移図/表エディタ、入出力デバイス定義エディタを用いて、図的に（図及び表を用いて）仕様を入力する。静的エラーのチェックでは、検証系が、入力された要求仕様を論理的に評価し、入力した仕様の完全性、無矛盾性をチェックし、その結果を開発者にレポートする。開発者は、レポートされたエラーを仕様エディタ群を用いて修正する。顧客によるレビューでは、顧客にシステムの動くイメージをってもらい仕様の正しさを検証してもらう。開発者は、プロトタイプを用いてシステムのMMIを定義することができ、その画面の上の操作を入力として、要求仕様を直接実行することができる。レビューの結果、仕様の誤りや変更があった場合、（その変更が小さい場合）開発者は直ちに仕様を編集し再実行してその場で再レビューすることができる。

外部仕様書作成作業においては、要求分析ツールが入力済みの図表を出力する。開発者は、その図表をISV製品のワープロに取込み、仕様書を完成させることができる。

さらに、要求分析ツールは、入力された入出力デバイス定義からプログラムのマクロ定義をテキストとして出力する。このマクロ定義は、内部設計およ

びコーディング・デバッグの中で使用することができる。

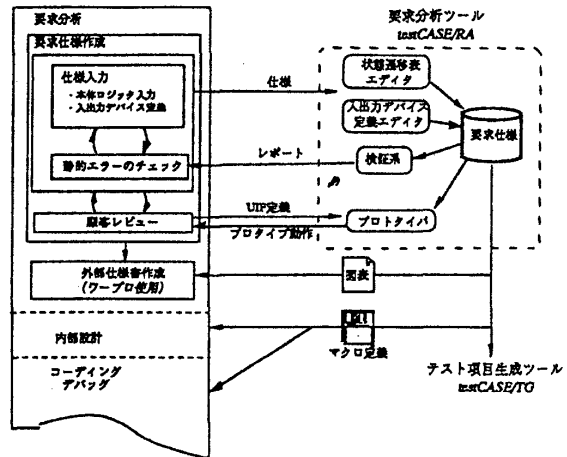


図1. 要求分析の作業の流れ

## 3. 仕様記述方式

ここでは、testCASE/RAにおける仕様記述方式の設計方針について述べ、仕様記述方式の2つの主要部分であるデバイス定義と状態遷移表について説明する。

### 3.1 仕様定義方式の設計

#### (1) 従来の作業とのギャップが小さくなること

マイコンS/W開発では、デバイスあるいはデバイスドライバの単位でまとめて外部仕様を記述する機会が多い。このため、デバイス定義から分析を始めることができるようにした。

また、MMIの仕様は状態遷移表を用いて定義していることが多い。潜在的にHarelの拡張状態遷移図<sup>1</sup>への需要はあるがテストを考えると最終的な仕様としては表を用いるのが標準的である。このため、基本的には状態遷移表を用い、図は補助的に用いて図から表への変換機能を提供することとした。

#### (2) 簡単であること

上流ツールでは特に注意しなければならないが、新しい機能(testCASE/RAではプロトタイピング機能)のために難解な作業が増えたのでは、ツールの

testCASE : CASE Tools for Microcomputer Software (2) - Requirements Analysis Tool -

Tsuyoshi NAKAJIMA\*, Hiroshi ISHIKAWA\*, Tsuyoshi KOBAYASHI\*, Yuzo BESSHO\*, Kazumi HIROTA\*\*

\*Mitsubishi Electric Corporation

\*\*Mitsubishi Electric Semiconductor Software Corporation

普及上問題点がある。そのため、入出力デバイス定義では、従来外部仕様として書いていたようなデバイスの仕様と、プロトタイピングにおける（基本的なふるまいが組み込まれている）ウィンドウ部品要素と、テストのためのプローブとをパッケージ化して扱う。入出力デバイスは、testCASE/RAのビルトインの仕様部品として提供し、カスタマイズが必要な場合は、パラメータ設定/選択程度で行えるようにする。

### 3.2 入出力デバイス定義

#### (1) 見かけ/動作/ポート接続

主に仕様として入力する部分である。入力デバイスの構成要素の数、配置、背景等を決める。また構成要素毎に形状と色を決める。さらに、デバイス固有の動作に関する記述や、マイコンポートとの接続の情報なども記述する。

#### (2) 発生/駆動イベントの定義

入力デバイスに対しては、構成要素に対する操作を操作イベントとしてを定義する。操作イベントに対応して、発生イベントを定義する。発生イベントとは、入力デバイスに対する操作イベントから、本体ロジックに出力されるイベントのことである。

表1では、キーマトリックスからの発生イベントはKEYだけであり、Valueによって違いを識別できるように定義している。ここで、Valueはイベントに伴って値を伝搬する特殊な変数として定義される。

表1. 発生イベントの定義

操作イベント	発生イベント名	Value
K00	KEY	0
K01	KEY	1
..	..	..
K33	KEY	15

出力デバイスに対しては、システムからの出力を受ける駆動イベントを定義する。

### 3.3 状態遷移表

状態	イベント	KEY_MODE	KEY_NUM(No)
通常モード		時間設定モード	
時間設定モード Enter: 時間設定(Time) 0 Exit: (Time, to Value) 0 時間設定(Enter)	分設定 Enter: LED(4)BLINK; LED(5)BLINK Exit: LED(4)INORM; LED(5)INORM	分設定	1
	秒設定 Enter: LED(4)BLINK; LED(5)BLINK Exit: LED(4)INORM; LED(5)INORM	秒設定	1
	通常モード Enter: LED(1)BLINK; LED(1)BLINK Exit: LED(1)INORM; LED(1)INORM	通常モード	1

図2. 状態遷移表

・状態遷移表内で用いる一時変数は、表エディタに付随した情報として、定義する。状態遷移表から参

照する外部状態値も定義できるようにする。

- ・状態の階層を記述することができる。階層を上下できるI/Fを用意する（太い下矢印ボタンを押すとより下位の階層を展開できる）。

### 4. testCASE/RA上での作業

次にtestCASE/RAを用いて行う作業の流れを示す。

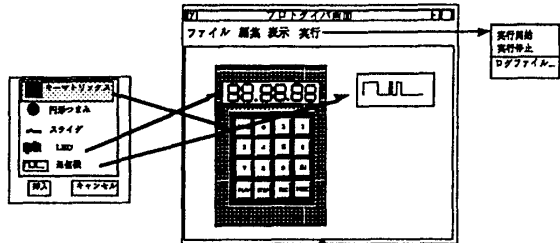


図3. プロトタイプ画面

#### 1) 初期画面の作成

プロトタイプ上でシステムを構成する入出力デバイスをひな型から選択して張り付けることでシステムイメージを作る（図3）。これによって、システムを構成するデバイスを決める。背景あるいはレイアウト等はこの画面においていつでも修正可能である。

#### 2) 仕様の定義

プロトタイプ画面上からデバイスを選択し、順次定義を行っていく(3.2節)。並行して本体ロジック定義を行っていく(3.3節)。

#### 3) 静的チェック

状態遷移表上の文法、デバイス定義記述の完全性、イベント参照関係などのチェックを行う。

#### 4) プロトタイプの実行

プロトタイプの「実行開始」メニューを選択することでスタートする（図3）。本体ロジックが、状態マシンとして実行される。入力デバイス上のイベント入力が、発生イベントとして生成され、本体ロジックへの入力となる。その実行の結果が出力デバイスに表示される。

### 5. おわりに

testCASE/RAは、マイコンS/W開発の外部仕様作成作業を支援し顧客レビューのためのプロトタイピング機能を提供するツールである。このツールを利用することによってMMI仕様の早期決定が可能になる。

#### 参考文献

- 1) Harel, D.: On Visual Formalisms, *Comm. of the ACM*, Vol. 31, No. 5, pp. 514-530 (1988).