

状態遷移表による ソフトウェア開発技法のための教材作成

青森職業能力開発短期大学校 川守田 聡

Creation of the Teaching Materials for the Software Development Using a State Transition Matrix

Satoshi KAWAMORITA

要約 近年のソフトウェア開発においてソフトウェアの開発規模の拡大によりプログラムのモレ、ヌケなどによる不具合の増加は製品開発に大きなダメージを与える。このプログラムのモレ、ヌケなどの防止には状態遷移表を使用することにより、不具合のプログラムのモレ、ヌケなどの低減化をすることができる。産業界ではソフトウェア開発に状態遷移表を利用することが一般的になっている。しかし、教育現場においては状態遷移表によるソフトウェア開発技法はほとんど行われていない。本報告では、状態遷移表によるソフトウェア開発のための教材とその活用結果について述べる。

I はじめに

近年、電子機器の高度機能化により電子機器を制御するソフトウェアの開発規模が拡大している。そのためソフトウェアによる不具合も増加し、ソフトウェア業界に限らず、ハードウェアの製造業においても電子機器を制御するプログラムの不具合をなくすために状態遷移表を利用した機器開発を行い始めた。

状態遷移表によるソフトウェア開発技法のための教材はあまりなく、今回、状態遷移表の原理を理解し、応用するための教材開発を行ったので報告する。

II 組込みソフトウェア開発の現状

家庭用DVDレコーダに組み込まれているプログラムの行数は平成2年で20万行であったが、平成5年で180万行となり3年間で9倍に増加している。携帯電話では平成2年で100万行であったが、平成5年で500万行となり3年間で5倍の増加で、この行数は1980年代後半の第3次オンラインシステムなみの規模となっている。

経済産業省 2007年版組込みソフトウェア産業実態調

査によれば、製品の出荷後の不具合の原因は製品企画・仕様の不具合が17.9%、ソフトウェアの不具合が43.8%で不具合の6割以上は仕様・ソフトの不具合が原因となっている。

これらの不具合をなくすために状態遷移表を使うことによって、フローチャートや状態遷移図に出てこない部分を知ることができ、仕様に定義されていない、ある状態において事象が実行されたときの処理プログラムのモレやヌケを早期に発見し、検討することができる。

III 状態遷移表とは

表1に状態遷移表を示す。状態遷移表は①事象（イベント）、②状態（ステート）、③アクション、④遷移の4つから構成されている。斜線「/」は事象が起きてもアクションは「何もしない」を表し、交差斜線「×」はその状態において、そのような「事象が起こることはありえない」ことを表す。

表2に「電話をする」という一連の行動・状態を状態遷移表に表す。「切れている」状態で「電話をする用事」事象が発生すると、「電話をかける」アクションを起こし、「電話中」状態に遷移する。この段階でアクティブな状態は「電話中」となり、「相手の声」が発生すると、

「用事を伝える」「電話を切る」アクションを起こし、「切れている」状態に遷移する。この段階でアクティブな状態は「切れている」状態である。

表1 状態遷移表

	S	状態A	状態B	状態C
E		0	1	2
事象A	0	遷移先 アクション		
事象B	1		遷移先 アクション	
事象C	2			遷移先 アクション

表2 「電話をする」の状態遷移表

	S	切れている	電話中
E		0	1
電話する 用事	0	1 電話をかける	
相手の声	1		0 用事を伝える 電話を切る
話中の音	2		0 電話を切る

IV 状態遷移表の利点

図1に普段よく使う状態遷移図を示す。状態1から事象1が実行されることにより処理1が実施され、状態2になる。次に事象2が実行されることにより処理2が実施され、状態1になる。あるいは状態2において、事象3が実行されることにより処理3が実施される。これを見るとなにもモレやヌケが無いように思われる。

しかし、これを表3に示す状態遷移表に書き表す。これを見ると状態1において事象2あるいは事象3が実行されたとき、どう処理されるかが示されていない。さらに状態2において事象1が実行されるときもどう処理されるかがしめされていない。これが不具合の元となる(表3色が付いている部分)。これによって早期にプログラムのモレやヌケを発見することが可能となり、プログラムの不具合をなくすことができる。

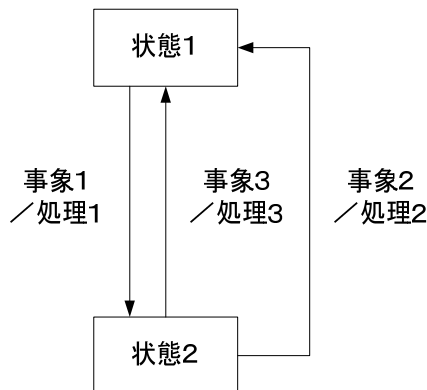


図1 状態遷移図

表3 図1の状態遷移表

	S	状態1	状態2
E		0	1
事象1	0	1 処理1	
事象2	1		0 処理2
事象3	2		0 処理3

V 状態遷移設計による実装

図2に状態遷移設計による実装方法の概略図を示す。main関数で初期化処理関数を実行後、イベント解析関数に移動する。イベント解析関数で該当する事象を見つけ状態遷移表(STM: State Transition Matrix)を表すSTM操作関数に移動する。STM操作関数では現在のイベント番号と状態番号から状態遷移表に示すアクションに該当する処理プログラム(処理関数)をまとめている関数ポインタテーブルにアクセスし、関数ポインタテーブルの中で該当する処理関数を見つけ、処理関数に移動し実行する。その後遷移処理を実行して状態番号を現在のものに変更し、イベント解析関数に戻り次のイベントを待つ。という構成になっている。

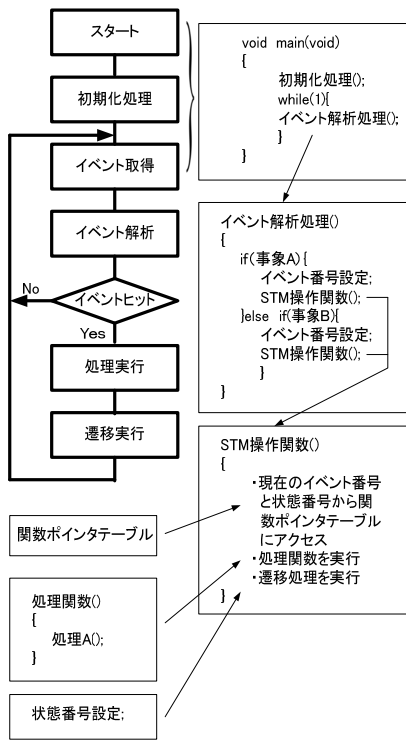


図2 状態遷移設計による実装

VI 教材の構成

教材はマイコン扇風機を想定している。図3に今回製作した教材の回路図を示す。言語にはC言語を用いて

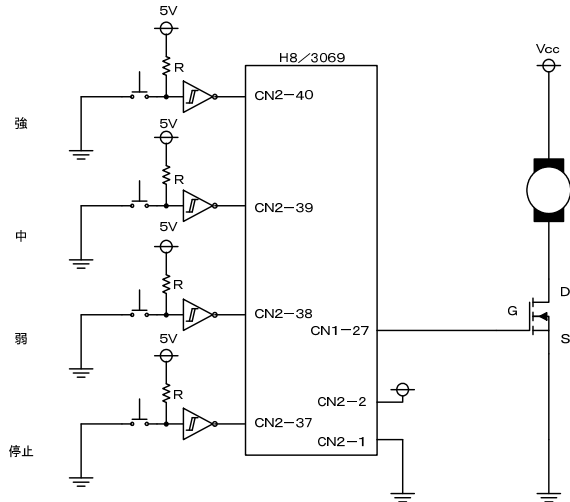


図3 教材の回路図

いる。図4に実習教材を示す。モータが扇風機とし、白いタクトスイッチが扇風機の風量を「強」「中」「弱」「停止」を設定するボタンである。

下位プログラムとしてはDCモータのPWM速度制御の理解が必要でデューティ比の違いにより風量の変

化を行っている。

最初にPWM速度制御の例題を解き、演習問題を解くことによって下位プログラムの理解を深める。そのあと上位プログラム(状態遷移表を使用)による下位プログラムを制御する例題を理解し、演習問題を解く構成となっている。

演習問題1は以下の通りである。

- (1) 4つの風力切換えボタンがある扇風機を動作させる場合の状態遷移表を書きなさい。
 - (2) 状態遷移表より上位プログラムを作成しなさい。
- 演習問題1における状態遷移表を使用する上位プログラムの要求仕様は次の通りとなっている。①扇風機には「停止ボタン」「風力弱ボタン」「風力中ボタン」「風力強ボタン」の4つがある。②「風力弱ボタン」を押すと風の強さは「弱」となる。③「風力中ボタン」を押すと風の強さは「中」となる。④「風力強ボタン」を押すと風の強さは「強」となる。⑤「風力停止ボタン」を押すと送風を停止する。⑥「送風OFF」状態で、「送風停止ボタン」を押しても反応しない。⑦「風力弱」状態で、「風力弱ボタン」を押しても反応しない。⑧「風力中」

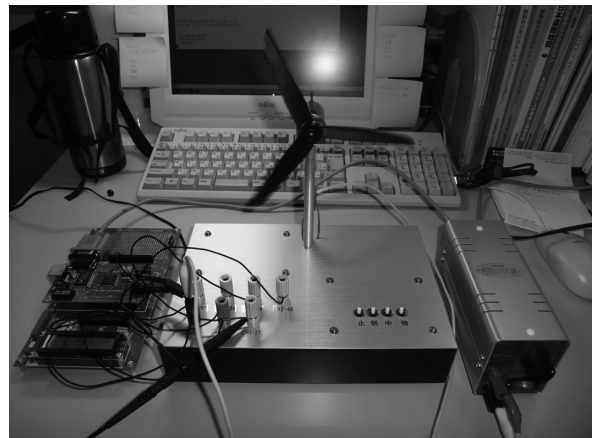


図4 実習教材

表4 演習問題1の状態遷移表

	送風OFF	送風ON		
		風力弱	風力中	風力強
送風停止ボタン 押下	/	送風OFF	送風OFF	送風OFF
	送風を停止する	送風を停止する	送風を停止する	送風を停止する
送風弱ボタン 押下	送風ON:風力弱	/	送風ON:風力強	送風ON:風力弱
	風力を弱にする	風力を弱にする	風力を弱にする	風力を弱にする
送風中ボタン 押下	送風ON:風力中	送風ON:風力中	/	送風ON:風力中
	風力を中にする	風力を中にする	風力を中にする	風力を中にする
送風強ボタン 押下	送風ON:風力強	送風ON:風力強	送風ON:風力強	/
	風力を強にする	風力を強にする	風力を強にする	風力を強にする

```

■ void (* const
func[event_NUM][state_NUM])(void) = {
■     {NULL,TEISHI,TEISHI,TEISHI},
■     {JYAKU,NULL,JYAKU,JYAKU},
■     {TYU,TYU,NULL,TYU},
■     {KYOU,KYOU,KYOU,NULL}
■ };
    
```

図5 演習問題1の状態遷移表のプログラムコード

状態で、「風力中ボタン」を押しても反応しない。⑨「風力強」状態で、「風力強ボタン」を押しても反応しない。⑩4つのボタンの中で複数のボタンを同時に押すことはない。

表4に演習問題1の状態遷移表を、図5に関数ポインタテーブルで表現したプログラムコードを示す。

図5の関数ポインタテーブルを見ると表4の状態遷移表に示されるアクションが表の通りに並べられていることがわかる。ここで、「NULL」は「何もしない」処理をするプログラム、「TEISHI」は「送風を停止」処理をするプログラム、「JYAKU」は「送風を弱」処理をするプログラム、「TYU」は「送風を中」処理をするプログラム、「KYOU」は「送風を強」処理するプログラムである。これらの処理プログラムはモータの回転数を変化させるものである。関数ポインタテーブルのコマンド func [A] [B] (void)のAは「行」、状態遷移表の「事象」を示し、Bは「列」、状態遷移表の「状態」を示す。今、func [3] [2] (void)であれば、表4の状態遷移表において、事象が「送風中ボタン押下」で、状態が「送風ON、風力弱」で、アクションが「風力を中にする」を示し、「TYU」処理プログラムを実効する。void (*const func [A] [B] (void) = { }; の { } の中に、状態遷移表のアクションを実効するプログラム関数名を配列させる。

演習問題2は以下の通りである。

- (1) 状態遷移表を書きなさい。
- (2) 状態遷移表より上位プログラムを作成しなさい。

演習問題2における状態遷移表を使用する上位プログラムの要求仕様は次の通りとなっている。①扇風機には「送風開始ボタン」と「送風停止ボタン」の2個がある。②「送風開始ボタン」を押下すると、風の強さが「弱」の状態を送風を開始する。③風の強さは「3段階(弱・中・強)」に変更できる。④「送風開始ボタン」を押下することで、風力を「弱→中→強→弱→中→…」と切り替えられる。⑤「送風ON」状態で「送風停止ボタン」

表5 演習問題2の状態遷移表

	送風OFF	送風ON		
		風力弱	風力中	風力強
送風停止ボタン押下	/	送風OFF 送風を停止する	送風OFF 送風を停止する	送風OFF 送風を停止する
送風開始ボタン押下	送風ON:風力弱 風力を弱にする	送風ON:風力中 風力を中にする	送風ON:風力強 風力を強にする	送風ON:風力弱 風力を弱にする

```

■ void (* const
func[event_NUM][state_NUM])(void) = {
■     {NULL,TEISHI,TEISHI,TEISHI},
■     {JYAKU,TYU,KYOU,JYAKU}
    
```

図6 演習問題2の状態遷移表のプログラムコード

を押下すると送風を停止する。⑥「送風OFF」状態で「送風停止ボタン」を押下しても反応しない。

表5に演習問題2の状態遷移表を図6に関数ポインタテーブルで表現したプログラムコードを示す。

演習問題3は以下の通りである。

- (1) 状態遷移表を書きなさい。
- (2) リズム送風の下位プログラムを作成しなさい。
- (3) 状態遷移表により上位プログラムを作成しなさい。

演習問題3は演習問題2の仕様にリズム送風を追加したものである。要求仕様は次の通りとなっている。①「リズムボタン」を押下すると「リズム送風」を開始する。②「リズム送風」は「送風OFF」「送風ON(風力弱/風力中/風力強)」状態にあるときに、「リズムボタン」を押下することで「リズム送風」状態になる。③「リズム送風」中に「リズムボタン」を押下しても反応しない。④「リズム送風」中に「送風停止ボタン」を押下すると「リズム送風を停止」し、「送風OFF」状態になる。⑤「リズム送風」中に「送風開始ボタン」を押下すると

表6 演習問題3の状態遷移表

	送風OFF	送風ON			
		風力弱	風力中	風力強	リズム送風
送風停止ボタン押下	/	送風OFF 送風を停止する	送風OFF 送風を停止する	送風OFF 送風を停止する	送風OFF 送風を停止する
送風力ボタン押下	送風ON:風力弱 風力を弱にする	送風ON:風力中 風力を中にする	送風ON:風力強 風力を強にする	送風ON:風力弱 風力を弱にする	送風ON:風力弱 風力を弱にする
リズム送風ボタン押下	送風ON:リズム送風 リズム送風を開始する	送風ON:リズム送風 リズム送風を開始する	送風ON:リズム送風 リズム送風を開始する	送風ON:リズム送風 リズム送風を開始する	/

```

■ void(*const func[event_NUM][state_NUM])(void)={
■     {NULL,TEISHI,TEISHI,TEISHI,TEISHI},
■     {JYAKU,TYU,KYOU,JYAKU,JYAKU},
■     {RIZUMU,RIZUMU,RIZUMU,RIZUMU,NULL}
■ };
    
```

図7 演習問題3の状態遷移表のプログラムコード

「リズム送風を停止」し、「送風 ON：風力弱」状態になる。

表6に演習問題3の状態遷移表を図7に関数インタフェースで表現したプログラムコードを示す。このように状態遷移表を作成することによって、プログラムのモレやヌケを見つけそれらをなくし、表の通りに実行プログラム（アクション）を動作させることによって不具合の無いプログラムを作成することができる。

この後に演習問題1、2、3の状態遷移表を用いないで、下位プログラムのみによって作成させることにより状態遷移表の効果を十分に感じ取ることができ、下位プログラムのみによるプログラム開発の難しさ（スバゲティプログラム）も理解することができる。

プログラムは関数インタフェースにアクションとなるプログラムを状態遷移表の通りに並べることによって簡単に作成できる。そのため、一番のポイントは状態遷移表がきちんと作成できることである。プログラム作成前に状態遷移表の作成演習を行う。演習1はナビゲーションシステムである。要求仕様は次の通りである。①カーナビはGPS、テレビチューナ、スピーカと液晶画面で構成されている。②ボタンは「電源」「現在地」「TV」の3つがある。③「電源」は、カーナビ本体をON/OFFするためのボタンである。ただし、自動車が走行中のときは、電源OFFにはできない。④「現在地」は、カーナビ画面の地図上に現在の車の位置を表示する。また自動車が走行中でも操作は有効である。⑤「TV」は、画面にテレビを映し、スピーカによりテレビの音声を流す。自動車が走行中のときは、テレビの音声を流すが、画面にはテレビ放映を非表示にし警告文字を表示する。またカーナビ画面に切り替わると音声は消える。⑥カーナビの電源がONのとき、自動車本体により走行中か停止中かの信号を受け取ることができる。

この状態遷移表を表7に示す。

表7 演習1の状態遷移表

	電源OFF	電源ON			
		停止中		走行中	
		カーナビ画面	テレビ画面	カーナビ画面	テレビ画面
電源ボタン	電源ON:停止中:カーナビ画面 電源を入れる	電源OFF 電源を切る	電源OFF 電源を切る	/	/
現在地ボタン	/	- 現在地を表示する	電源ON:停止中:カーナビ画面 テレビの音声を消す、カーナビ画面に切り替える、現在地を表示する	- 現在地を表示する	電源ON:走行中:カーナビ画面 テレビの音声を消す、カーナビ画面に切り替える、現在地を表示する
TVボタン	/	電源ON:停止中:テレビ画面 テレビ画面に切り替える、テレビの音声を消す	/	電源ON:走行中:テレビ画面 走行中警告を表示する、放映を非表示にする、テレビの音声を流す	/
走行通知	x	電源ON:走行中:カーナビ画面 /	電源ON:走行中:テレビ画面 走行中警告を表示する	/	/
停止通知	x	/	/	電源ON:停止中:カーナビ画面 /	電源ON:停止中:テレビ画面 テレビ画面に切り替える

演習2はこのナビゲーションシステムにCDプレーヤの機能を追加するものである。追加の要求仕様は次の通りである。①メディアのCDが挿入されていることを前提とする。②CDは1曲目から再生し、画面に曲一覧を表示する。CD選曲画面以外に遷移したときはCD再生は停止する。③CD選曲画面はタッチパネル形式になっており、曲一覧から曲名に触れることにより選曲再生を行う。ただし、自動車が走行中のときは安全を考慮し、選曲は行えない。

この状態遷移表を表8に示す。演習2はナビゲーションシステムにCDプレーヤを追加する問題であるが、状態遷移表を用いることにより簡単に機能追加を行うことができる。プログラムも追加機能の下位プログラムを作成し、上位プログラムである関数インタフェースにアクションを追加するだけである。

このように状態遷移表はある状態において事象が実行されたときに処理するプログラムのヌケやモレを防ぐためだけでなく、機能追加を容易にできるメリットもある。

表8 演習2の状態遷移表

	電源OFF	電源ON					
		停止中			走行中		
		カーナビ画面	テレビ画面	CD選曲画面	カーナビ画面	テレビ画面	CD選曲画面
電源ボタン	電源ON:停止中:カーナビ画面 電源を入れる	電源OFF 電源を切る	電源OFF 電源を切る	電源OFF 電源を切る	/	/	/
現在地ボタン	/	電源ON:停止中:カーナビ画面 現在の地を表示する	電源ON:停止中:カーナビ画面 テレビの音声を消す、カーナビ画面に切り替える、現在の地を表示する	電源ON:停止中:テレビ画面 CDを停止する、カーナビ画面に切り替える、現在の地を表示する	電源ON:走行中:カーナビ画面 現在の地を表示する	電源ON:走行中:カーナビ画面 テレビの音声を消す、カーナビ画面に切り替える、現在の地を表示する	電源ON:走行中:カーナビ画面 CDを停止する、カーナビ画面に切り替える、現在の地を表示する
TVボタン	/	電源ON:停止中:テレビ画面 テレビ画面に切り替える、テレビの音声を消す	/	電源ON:停止中:テレビ画面 CDを停止する、テレビ画面に切り替える、テレビの音声を消す	電源ON:走行中:テレビ画面 走行中警告を表示す、テレビの音声を流す	/	電源ON:走行中:テレビ画面 CDを停止する、カーナビ画面に切り替える、現在の地を表示する
CDボタン	/	電源ON:停止中:CD選曲画面 CD選曲画面に切り替える、CDを再生する	電源ON:停止中:CD選曲画面 テレビの音声を消す、CD選曲画面に切り替える、CDを再生する	/	電源ON:走行中:CD選曲画面 CD選曲画面に切り替える、CDを再生する	電源ON:走行中:CD選曲画面 CD選曲画面に切り替える、CDを再生する	/
CD選曲	x	x	x	CDを選曲、再生する	x	x	/
走行通知	x	電源ON:走行中:カーナビ画面 /	電源ON:走行中:テレビ画面 走行中警告を表示する	電源ON:走行中:CD選曲画面 /	/	/	/
停止通知	x	/	/	/	電源ON:停止中:カーナビ画面 /	電源ON:停止中:テレビ画面 テレビ画面に切り替える	電源ON:停止中:CD選曲画面 /

VII おわりに

状態遷移表によるプログラム作成はある状態において事象が実行されたときに処理するプログラムのモレやヌケを無くす方法として非常に有効な手段であることがわかった。状態遷移表によるソフトウェア開発は製造業等ではよく使用する手法であり、能開大の応用課程やセンターの離職者訓練における組込み教科では必要な分野であり今後積極的に導入されることを期待する。

[参考文献]

- (1) 渡辺政彦、拡張階層化状態遷移表設計手法
Ver2.0-Embedded SE のための設計手法、東銀座出版社、2008年、pp.4-5
- (2) 組込みプレス編集部、組込みプレス Vol.13 (組込みシステムシリーズ)、技術評論社、2008年、pp.115-122