

タイトル	表計算ソフトによるユーザインターフェース状態遷移仕様の表現とデジタルモックアップ用プラットフォームへの統合
著者	菊地，慶仁
引用	北海学園大学工学部研究報告，35：143-149
発行日	2008-02-00

# 表計算ソフトによるユーザインターフェース状態遷移仕様の表現とデジタルモックアップ用プラットフォームへの統合

菊地 慶仁\*

## Representation of User Interface State Transition Specification Utilizing Spreadsheet Software and Integration to Digital Mockup Platform

Yoshihito KIKUCHI\*

### Abstract

In previous report, we proposed a formal description methodology and validation about embedded electronics instrument's user interface state transition specification. However, from a difficulty of data input operation, our methodology was not welcomed by designer at hearing stage to electric maker. To resolve this problem, we propose a representation method about state transition specification of user interface. And, proposed methodology is integrated to BehaveView software platform.

### 1. はじめに

本研究は、デジタルカメラに代表される小型電子機器のユーザインターフェース設計初期段階を対象としている。これらの製品ライフサイクルが短い機器で開発期間やコストを短縮させるためには、設計初期段階でユーザビリティや仕様一貫性などの検証を予め行うことで、開発途中での仕様変更を減らし下流のソフトウェア開発工程で仕様情報を再利用する必要がある。本研究では、このような利用のための製品仕様の記述方法と検証を目的としている。

前報までに、小型電子機器のユーザインターフェース仕様を状態遷移機械として代数仕様の形で形式的に表現し、予め規定した公理系を用いて仕様の妥当性を検証する方式についての提案、Mathematicaによる試作システムの開発、Visual Wireless Communicatorの仕様を対象として検証を行った結果について報告してきた<sup>1)2)</sup>。

しかしながら、一般企業の設計現場におけるヒアリングでは、データ入力の煩雑さについて

---

\* 北海学園大学工学部電子情報工学科

\* Department of Information and Electronics Engineering, Faculty of Engineering, Hokkai-Gakuen University

の問題点が指摘された。このため本報告では、一般的なオフィスで広く用いられている表計算ソフトを状態遷移仕様の入力用ソフトウェアとして用いる方法について提案を行う。第2章では、指摘された問題点について分析し課題を整理する。第3章では、具体的な表の構成について報告し、考察及び結論とする。

## 2. 設計現場でのヒアリングで判明した問題点

D. Harelによって提唱されたStatecharts<sup>3)</sup>は、多くの研究でGUI設計への応用が検討されている<sup>4)5)</sup>。我々も同様の提案を行ってきた。実際の設計担当者に対してヒアリングを行う機会を持つことができ、我々が提案してきたGUI設計の初期における仕様の形式的な表現と検証の必要性に対しては合意を得ることができた。しかしながら、設計担当者がUMLツールを使ってGUIの状態遷移仕様を記述表現することは不可能との意見を得た。主な指摘は次の2点である。

- 1) 提案した設計仕様記述や検証の方式は、設計対象を3次元CADモデルとして表現し、ユーザビリティテストなどを仮想的に行うデジタルモックアップでのユーザビリティテストに主に用いられている。しかしながらStatechartなどの状態遷移仕様は、引き続きファームウェア設計では用いられることがない。
- 2) 大多数の設計者は非常に多忙であり、Statechartのような図式仕様やコンピュータツールの習得及び入力作業に時間を割く余裕が無い。

指摘された問題点に対して、以下の対応が課題となる。

- 1) GUIの状態遷移仕様は、検証のみに用いるのではなく、検証に引き続くソフトウェア開発においても、一般的な製品仕様として用いられなくてはならない。
- 2) 状態遷移仕様は、特別な習得期間を必要とせず簡単に定義できなくてはならない。一般的なオフィス作業者が仕様を扱えるような単純な構造を持たなくてはならない。

## 3. 表計算ソフトウェアの活用の提案

### 3.1 表計算ソフトの使用によって期待されるメリット

上記の課題への対応として、我々は表計算ソフトウェアを入力用に活用する方式を取った。表計算ソフトウェアを用いるメリットを以下にまとめる。

- 1) 仕様が表の形式を取っていれば、一般的なドキュメントとして他の設計資料と同様に用いることができる。
- 2) 表計算ソフトウェアは、すでに一般のオフィスで広範囲に用いられている。EXCELが高いシェアを得ているが、このソフトウェアと代替可能なオープンソースソフトウェアも存在するので、使用に際して追加的なソフトウェアの導入などの経済的な問題は無いと考えられる。

- 3) 表形式は非常に一般的かつ視覚的なデータ構造であり、図式表記などに比べて設計者が容易に見方を習得できると考えられる。また表計算ソフトウェアの使用方法も一般化しており、データの入力において特別な技能が必要とされることはない。従って、設計者直接ではなく補助作業員のレベルでも入力の支援を行うことができると考えられる。
- 4) 表計算ソフトをカスタマイズするためのプログラム環境としてVisualBasicが用意されており、入力作業の支援や必要な形式でデータを出力させることもできる。また、LCDパネルへの出力画像を表の要素として張り込んでおき、状態遷移情報と一緒に出力してデジタルモックアップでのLCDパネル出力とする、と言った活用も可能である。

### 3.2 表形式の概要














市場で売られているデジタルカメラをリバースエンジニアリングして、表の形式及び入力を支援するマクロプログラムの機能の検討を行った。また、すでに開発しているBehaveViewデジタルモックアップシステムへの入力としても用いるための検討も合わせて行った。当初は、単一のシートでの表現を試みたが、VisualBasicによる情報の抽出段階では、目的毎に分かれている方が望ましいので後者の方式を採用した。EXCELでは、一つのファイルをBookという単位で扱っており、複数の表=Sheetを含む形式になっている。本研究で試作した状態遷移仕様は、一つのBookで一つの機器の全状態遷移仕様を表現することとし、機能毎に一つ一つのシートに情報を記載する形式とした。

図1に状態とイベントの相関を示す表と、その一部の拡大を示す。最も基本的なシートは、ユーザもしくは機器自体によって起こされるイベントの列挙と、状態遷移の関係を記述したものである。状態遷移シートでは、1行がシステム中の状態の一つを意味し、この行に交錯する縦列が全てのイベントの列挙となる。この表では、ある状態と発生するイベントの全ての組み合わせがマスで表現され、なおかつ一つの状態と一つのイベントの組み合わせは一つのマスにだけ対応する。従って他の状態及びイベントの組み合わせとは排他的な関係となる。イベントに対応して状態遷移が起こる場合には、遷移する先の状態番号と該当する出力を出力表の番号で指定する。

設計対象からの出力は、LCDパネルからの画像出力、LED点灯、音声出力など多岐に渡って考えられるので、出力指定の表は、出力の組み合わせに識別番号を割り当てて指定する。我々は、デジタルモックアップのためのインターフェースと、入力支援用のマクロプログラムを開発した。インターフェースは、Visual BasicでプログラムされBehaveViewが読み込める形式で表の情報を出力する。出力指定表中で用いたLCDパネルからの出力画像は、そのままBehaveViewで使用することが可能となっている。

表の作成は、幾つかの状態と出力しか持たない非常に単純な内容から始まる。設計者はBehaveViewを併用して設計対象の振舞いを確認しながら状態遷移仕様を詳細化することがで

State transition table for normal shooting

No.	Start state	Event	Event								
			Push Power Button	Half Push Shutter Button	Push Rec. Button	Push Select Button	Push Set Button	Push Play Button	Push Shutter Button	Release Shutter Button	Auto Transition 1
1	Power Off	Behaviour	to Power On 1		to Rec Mode 1				to Play Mode 1		to Power On 0
		Guard condition	Power Button Interval								Power
		Action sequence	Activate Power		Activate Power				Activate Power 2		Activate Power
		End state	Power On						Play Mode		Power On
											
2	Power On	Behaviour	to Power On 1								to Rec Mode 2
		Guard condition									to Power On 0
		Action sequence	Shutoff Power								Shutoff Power
		End state	Power Off								Rec Mode
											
3	Play Mode	Behaviour			to Rec Mode 3	to Play Mode 3	to Play Mode 4			to Recording	
		Guard condition									
		Action sequence			Display Rec	Display Next Picture	Play Mode			Shoot	Recording
		End state			Rec Mode	Play Mode				Recording	
											
4	Rec Mode	Behaviour		to Focus						to Play Mode 2	
		Guard condition									
		Action sequence		Focus LED							Display Play
		End state		Focus							Play Mode
											

State transition table for normal shooting


No.	Start state	Event	Event
			Push Power Button
1	Power Off	Behaviour	to Power On 1
		Guard condition	Power Button Interval
		Action sequence	Activate Power
		End state	Power On
			

表1 状態遷移表及びその一部の拡大

き、デジタルモックアップで満足されたものが、最終的な仕様となり、引き続きファームウェア開発工程へと渡される。また、これらのツールは設計対象とその競合製品の比較においても有効と考えられる。

### 3.3 表の具体的な構造

表2に、一つのBookに含まれる表の種類を示す。BehaveView用にデータを出力するためにVisualBasicスクリプトを用いるには、全てのシートの命名規則を決めておかなければならない。表の各シートの名前は“@keyword + xxxxx”の形式をとる。“@keyword”までは固定的に用いる。もし表中の項目数がnの場合には同種のシートを複数用いることが可能で、この場合には“xxxxx”の部分シート毎に定義することになる、例えば“@State Normal Shooting”は、状態定義目的のシートで“Normal Shooting”の名称を持つ。

Term	Sheet Name	Number
Cover	@Title	1
Sheet Construct	Optional	1
Function List	Optional	1
State Transition Table	@StateTable xxxxx	n
State Hierchal Structure List	@State xxxxx	n
Event List	@Event xxxxx	n
Effectively List	@Effectivity xxxxx	n
Action Sequence List	@ActionSequence xxxxx	n
Action/Animation List	@Action xxxxx	n
Constituent Parts List	@Parts xxxxx	n
Related Document List	@Document xxxxx	n
Images List	@Image xxxxx	n
Sound List	@Sound xxxxx	n

Additional sheet should not named with "@" n : More than one sheets are available

表2 シートの構成とシートの命名規則

### State Hierchal Structure List

Level1	Level2	Level3
Power_Off		
Power_On	Rec_Mode	Focus
		Recording
	Play Mode	
DATA END	DATA END	DATA END

Hierchal structure is modeled as this. Left side one is super state.

### State List

State name	Declaraiton	Type	Image	Entry action	Do action	Exit action
Power_Off		INITIAL				
Power_On		NORMAL				
Rec_Mode		INITIAL				
Focus		NORMAL				
Recording		NORMAL				
Play_Mode		NORMAL				

表3 状態名と階層構造の定義

Statechartにおける並行サブ状態を表現するためにもシートを用いる。“@State”シートがこの目的のために用意されている。表3は状態名および状態の階層構造の表現例である。レベル1では2つの状態が存在する。“Power\_Off”及び“Power\_On”である。他の状態は、この2つの状態のサブ状態となる。項目“Type”は、階層中での開始/終了状態であるかどうかを示している。

表1では、状態遷移を示している。テーブルサイズが巨大となるために、一部分の拡大図も

添付する。この表は、デジタルカメラの電源投入及び切断に対応する“Power\_Off”，“Power\_On”と、一般的な撮影に関する他の状態についてまとめている。この表は、各々の状態と、その状態が許容できるイベントの関係を示している。表の一行が一つの状態に対応し、一列がイベントに対応する。ある状態でイベントが発生することが可能であり、かつ何らかの状態遷移や出力が行われるのであれば、表中に記載を行う。表中の升が空白であることは、イベントが発生しないか発生しても無視されることを意味する。LCDディスプレイから画像出力が行われる場合は、その画像イメージを表中に張り込み、デジタルモックアップに送って実際にモックアップ内モデルのLCDに表示させることが可能である。画像は“@Image”シートで指定する。

拡大された表の画像は、より詳細な記述の内容を示している。この図は、状態“Power\_Off”においてイベント“Push\_Power\_Bottun”が発生可能であり、この場合には振舞い“to\_Power\_On\_1”が起動される。“Power\_Button\_interval”と名付けられたガード条件が、状態遷移に用いられる。この際に実施されるシーケンスは“Activ\_Power”と名付けられており、“Power\_On”と名づけられた状態が終了状態となる。ガード条件、シーケンスは、個々のテーブルで記述される。

シーケンスは、状態遷移の際に呼び出される一連の処理を定義するもので、状態遷移表中ではどのシーケンスが呼ばれるかを定義する。シーケンスは、予め規定されたアクションを複数まとめて実施するための定義である。アクションは、画像の変更、LEDの点灯もしくは消灯、本体部品の動作など一つ一つ厳密に定義される。しかしながら、特定のイベントに対しては、これらのアクションを複数順番に実施することが非常に多いので、シーケンスとしてまとめる方法が用意されている。

表4では、シーケンス“Activate\_Power”が7つのアクション“Open\_Lens 1”から“Light\_On\_LED”までで構成されている。アクションはシステム中で発生する最小限の内容である。繰り返しはアクション及びシーケンスの両方で定義することができる。我々は、シーケンス内で繰り返し回数を増やすことはできるが、アクション内部で実施を再分割することはできない。

#### 4. 考察

提案した表について他の事業所でのヒアリング及びワークショップで以下のような示唆をえることができた。これらは、将来的なテーマとして考えられる。

1) 入力のための方式として図式表記は受け入れられていないために、我々は、図式表記の入力を回避する方法を提案した。一方、図式表記の理解し易さについては常に了解を得ることができた。従って、表形式の仕様入力からレポートとして図式表記を出力する機能が必要と考えられる。

2) 状態遷移表の升が空白になっている場合に、それが単に未入力のまま残っているのか、仕

Action Sequence List

Action sequence name	Declaration of Action sequence	Repetition condition				Action name	Synchronization of action		
		Repetition name	Declaration of condition	Condition	Interval		Name of synchronization	Action number to be synchronized	Declaration of synchronize
1 Shoot						1 Display_Recording			
						2 Blink2_LED			
2 Play_Movie						1 Play_Movie			
3 Display_Next_Picture						1 Display_Next_Picture			
4 Focus_LED						1 Focus_LED			
5 Activate_Power						1 Open_Lens_1			
						2 Open_Lens_2			
						3 Open_Lens_3			
						4 Open_Lens_4			
						5 Open_Lens_5			
						6 Display_Rec			
						7 Light_On_LED			

Action/Animation List

	Action name	Declaration of action	Action type	Related parts	Executable time	Repeat condition				Relationship	
						Repeat condition name	Declaration of repeat	Condition	Interval	Relation name	Declaration of relation
1	Open_Lens_1		START	Cylinder1	0						
2	Open_Lens_2		START	Cylinder2	0						
3	Open_Lens_3		START	Cylinder3	0						
4	Open_Lens_4		START	Lcover_b	0						
5	Open_Lens_5		START	Lcover_u	0						
6	Display_Rec		START	LCD1	0						
7	Light_On_LED		START	LED1	0						
8	Close_Lens_1		START	Cylinder1	0						
9	Close_Lens_2		START	Cylinder2	0						
10	Close_Lens_3		START	Cylinder3	0						

表4 出力毎に実施されるシーケンスとアクション/アニメーション定義

様としてイベントを無視することを支持しているかが分かりにくいとの指摘を受けたので、セルが空白のままの場合にどう扱うかを検討しなおす必要がある。

5. 結論

我々の個別の成果は、過去の発表機会に報告されている。本研究では、我々のプロジェクト中での状態遷移仕様の入力機能に絞って以下の項目を報告した。

- 1) BehaveViewへの状態遷移仕様の入力として、従来の図式表記に変わる表形式の入力方法を提案した。
- 2) 表の個別の仕様を策定し、状態遷移仕様を表現できることを示した。

参考文献

- 1) 菊地慶仁, 金井理, 岸浪健史, 「UI挙動仕様の代数的表現と検証」, ヒューマンインターフェースシンポジウム (2004)
- 2) 菊地慶仁, 「ユーザインターフェース仕様の検証機能のデジタルモックアップ用プラットフォームへの統合」, 2005年度北海学園大学工学部研究報告
- 3) Harel, D. : Statecharts : A visual formalism for complex systems. Amsterdam, Science of Computer Programming 8 North-Holland, 1987, p.231-274.
- 4) Horrocks I. : Constructing the User Interface with Statecharts, Boston, Addison-Wesley, 1999.
- 5) Paterno, F. : Model-based Design and Evaluation on Interactive Applications. Springer-Verlag, 2000.