状態遷移表検査技術について

福岡県産業・科学技術振興財団
福岡知的クラスター研究所　研究員
松本　充広

九州大学大学院
システム情報科学研究院　教授
福田　晃

１．はじめに

状態遷移表とは、横軸に状態名を記述し、縦軸にイベント名を記述し、状態名とイベント名で指定されるセルに、指定した状態において指定したイベントを受理した時のアクションと遷移先状態を記述した表である。指定した状態において指定したイベントが発生しないと設計者が判断した場合には、このセルに×印をつけ、このセルを不可セルと呼ぶ［1］。

複数のサブシステム（タスクなど）が組み合わさって出来ているシステムの場合、システムの全体像の把握が困難になり、設計者が不可セルと判断して×印を付けたとしても、組み合わせた全体のシステムでは、指定した状態に指定したイベントが発生してしまうという問題が発生することがある。

本稿では、この問題を発見する状態遷移表検査技術について議論する。

２．状態遷移表

本稿で対象とするシステムは、サブシステム（タスクなど）が組み合わさって出来ているシステムである。本稿では、図１の構成を持つシステムを例題に用いて議論する。

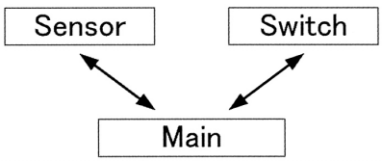

図１　例題システムの構成

２.１．状態遷移表に関する用語定義

状態遷移表とは、横軸に状態名を記述し、縦軸にイベント名を記述し、状態名とイベント名で指定されるセルに、指定した状態において指定したイベントを受理した時のアクションと遷移先状態を記述した表である。

図２はSensorの状態遷移表、図３はSwitchの状態遷移表、図４はMainの状態遷移表（但し、後述するように問題が存在する）である。

	State1	State2
!Catch	event(Main, Get); => State1	=> State2
Off	disable => State2	
On		enable => State1

図２　Sensorの状態遷移表

	State1
!Touch	event(Main,Change); => State1

図３　Switchの状態遷移表

	State1	State2
Get	proc-get => State1	proc-get => State2
Change	event(Sensor, Off); => State2	event(Sensor, Off); => State1

図４　Mainの状態遷移表

Sensorの状態遷移表（図２）において、‘ State1 ’、‘ State2 ’が状態で、‘ !Catch ’、‘ Off ’、‘ On ’がイベントである。イベントには、Sensorによる刺激の捕捉のように常に発火可能な能動的イベントと他のサブシステムのアクションとして発火される受動的イベントとがある。‘ !Catch ’は能動的イベント、‘ Off ’、‘ On ’は受動的イベントであり、両者は‘ ! ’の有無で区別する。

図２のセル（State1, !Catch）（State1と!Catchとで指定されるセル）に書かれている‘ event（Main,Get）; ’はアクションである。また、セル（State1, Off）に書かれている‘ disable ’もアクションである。アクションは、イベント発火、内部処理に分類され、event（）で囲まれた‘ event（Main,Get）; ’はイベント発火、囲まれていない‘ disable ’は内部処理である。

図２のセル（State1, !Catch）に書かれている‘ => State1 ’は遷移先状態である.

イベント発火と受動的イベントの受理とは同期して発生すると仮定する。例えば、図２のセル（State1, !Catch）に書かれている‘ event（Main,Get）; ’と図４の縦軸に書かれている‘ Get ’は同時に発生すると仮定する。

図２のセル（State1, On）に書かれている×印は、設計者が不可セルと判断したこと、つまり、‘ State1 ’において‘ On ’を受理することがないと判断したことを示す。

２.２. 状態遷移表の意味

状態遷移表に対し、以下の１～５の方法に従って生成した状態遷移図で意味を与える。

１．横軸の一番左の状態を初期状態とする。
２．各状態に対し、イベントの受理を待つ状態を用意する。この状態を受理可能状態と呼ぶ。
３．セル（状態名，イベント名）が不可セルでない場合、状態に相当する受理可能状態を始点とし、イベント名を名前とする状態遷移を用意する。このセルがアクションを持たない場合、この状態遷移の終点は、遷移先の状態に相当する受理可能状態とする。アクションを持つ場合は、この状態遷移の終点となる状態を新たに用意する。
４．３のセルがアクションを持つ場合、新たに用意した状態を始点とし、先頭のアクションの名前を名前とする状態遷移を用意する。以降にアクションを持たない場合、この状態遷移の終点は、遷移先の状態に相当する受理可能状態とする。アクションを持つ場合は、この状態遷移の終点となる状態を新たに用意する。
５．先頭以降のアクションに対し、一つ前のアクションに対応する状態遷移の終点を始点とし、このアクションの名前を名前とする状態遷移を用意する。終点は方法４と同様にする。

図２のSensorの状態遷移表に１～５の方法に従って意味を与えた状態遷移図は図５である。

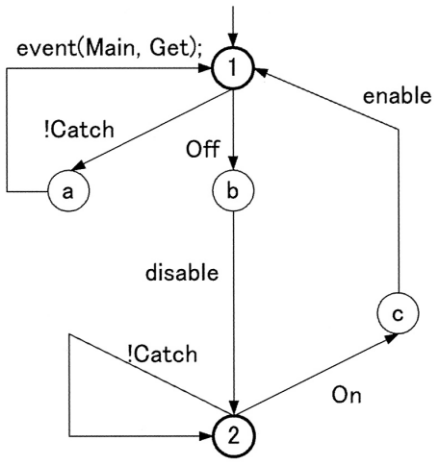

図５　図２へ意味を与える状態遷移図

図２の‘ State1 ’、‘ State2 ’に対応する受理可能状態は、それぞれ図５の‘ 状態1 ’、‘ 状態2 ’である。また、初期状態は‘ 状態1 ’である。
図２において、‘ State1 ’の列で不可セルにはならないセルは、セル（State1, !Catch）とセル（State1, Off）である。図５には‘ 状態1 ’を始点とし、それぞれのイベント名を名前とし、それぞれの終点を‘ 状態a ’、‘ 状態b ’とする状態遷移が存在する。

図２において、セル（State1, !Catch）のアクションは‘ event（Main, Get）’で、図５には‘ 状態a ’から‘ 状態1 ’への‘ event（Main, Get）’という名前の状態遷移が存在する。

図２において、セル（State1, Off）のアクションは‘ disable ’で、図５には‘ 状態b ’から

‘状態2’への‘disable’という名前の状態遷移が存在する。

３．状態遷移表検査技術

状態遷移表検査技術は、システムを構成する全サブシステムの状態遷移表を入力とし、組み合わせた全体のシステムにおいて不可セルと記述したセルが本当に不可セルであるかどうかを検査する技術である。更に、不可セルでない場合には、初期状態から異常状態（不可セルと記述したが不可セルではなかったセル）に到達するイベントのシーケンスを発見し、問題を修正するためのヒントを与える技術である。

状態遷移表検査技術を実現する状態遷移表検査ツールの構成は図６である。

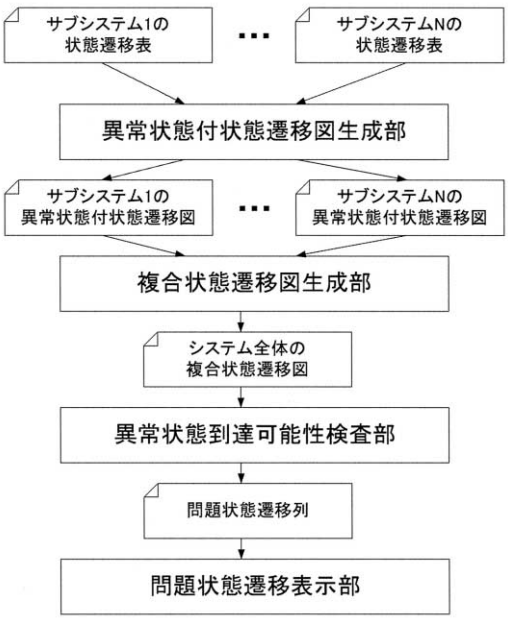

図６　状態遷移表検査ツールの構成図

３.１．状態遷移表検査ツールの動作原理

本節では、状態遷移表検査ツールの構成の内、異常状態付状態遷移図生成部、複合状態遷移図生成部、異常状態到達可能性検査部の動作原理について議論する。問題状態遷移表示部については３.２節にて議論する。

３.１.１．異常状態付状態遷移図生成部

異常状態付状態遷移図生成部では、サブシステム（タスク）の状態遷移表を入力とし、まず、２.２節の１～５の方法に従って意味を与える状態遷移図を生成する。次に、以下の方法に従って異常状態付状態遷移図を生成する。

１．異常状態を用意する。
２．セル（状態名，イベント名）が不可セルの場合、状態に相当する受理可能状態を始点とし、イベント名を名前とし、異常状態を終点とする状態遷移を用意する。

Sensorの状態遷移表へ意味を与える状態遷移図（図５）に１～２の方法に従って異常状態を追加した異常状態付状態遷移図は図７である。同様に、Switch、Mainの異常状態付状態遷移図はそれぞれ図８、図９である。

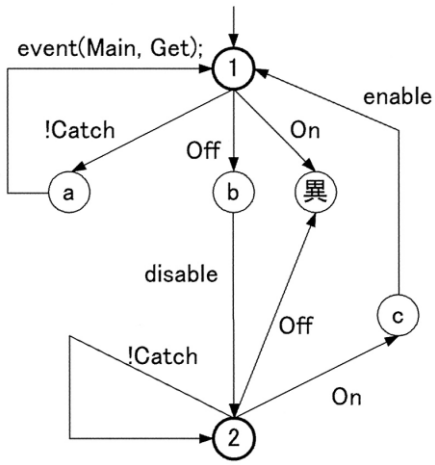

図７　Sensorの異常状態付状態遷移図

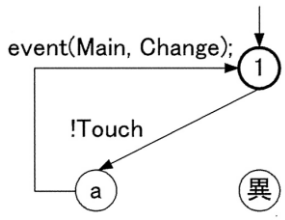

図８　Switchの異常状態付状態遷移図

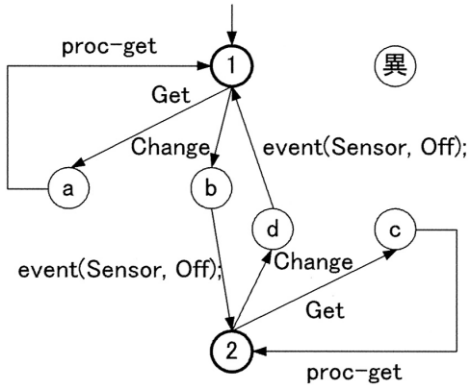

図９　Mainの異常状態付状態遷移図

図７の‘状態異’は異常状態である。

図２において、セル（State1, On）とセル（State2, Off）が不可セルである。従って、図７は図５に‘状態異’、‘状態1’を始点とし‘状態異’を終点とする‘On’の名前を持つ状態遷移、‘状態2’を始点とし‘状態異’を終点とする‘Off’の名前を持つ状態遷移の３つを追加した状態遷移図である。

３.１.２．複合状態遷移図生成部

複合状態遷移図の状態は、各サブシステムの状態の組である。複合状態遷移図生成部は、以下の１～４の方法に従って各サブシステムの異常状態付状態遷移図を組合せ、システム全体の複合状態遷移図を生成する。なお、本節では、状態（サブシステムの状態の組）の構成要素のことを、要素状態と呼ぶことにする。

１．初期状態の組を初期状態とする。

２．任意の状態Ａ（要素状態の組）において、ある要素状態を始点とする能動的イベント又は内部処理を考える。状態Ａ（要素状態の組）の内、この要素状態のみをイベント又は内部処理の終点に変更した状態（要素状態の組）を状態Ｂと呼ぶことにする。状態Ｂは存在しなければ用意する。このとき、状態Ａを始点、状態Ｂを終点とし、能動的イベント又は内部処理の名前を名前とする状態遷移を用意する。

３．任意の状態Ａ（要素状態の組）において、ある要素状態１を始点とするイベント発火に対し、別の要素状態２が存在し、要素状態２を始点とする同名の受動的イベントが存在する場合を考える。状態Ａ（要素状態の組）の内、要素状態１と要素状態２のみをそれぞれのイベントの終点に変更した状態を状態Ｂと呼ぶことにする。状態Ｂは存在しなければ用意する。このとき、状態Ａを始点、状態Ｂを終点とし、このイベント名を名前とする状態遷移を用意する。

４．要素状態として異常状態を含む状態を異常状態とする。

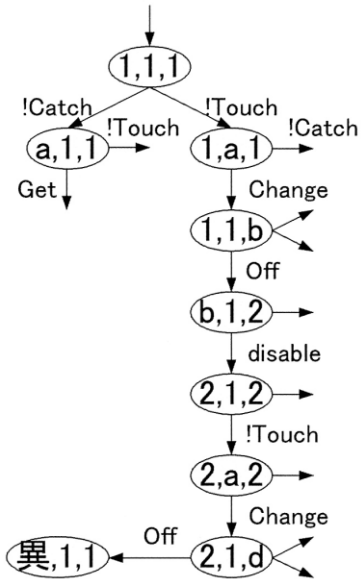

図10　例題システムの複合状態遷移図

図７、図８、図９を組み合わせた複合状態遷移図は図10である。図10の状態は、(Sensorの状態, Switchの状態, Mainの状態）である。

図10において、‘状態（1，1，1）’は初期状態で‘状態（異，1，1）’は異常状態である。‘状態（1，1，1）’を考える。Switchにおいて‘状態1’を始点とし‘状態a’を終点とする能動的イベント‘!Touch’が存在する（図８）。従って、方法２より、‘状態（1，a，1）’を用意し、‘状態（1，1，1）’を始点とし‘状態（1，a，1）’を終点とする‘!Touch’の名前を持つ状態遷移を図10に用意する。

次に‘状態（1，a，1）’を考える。Switchにおいて‘状態a’を始点とし‘状態1’を終点とするイベント発火‘event（Main, Change）;’が存在する（図８）。又、Mainにおいて‘状態1’を始点とし‘状態b’を終点とする受動的イベ

ント‘ Change ’が存在する（図９）。従って、方法３より、‘ 状態（1，1，b）’を用意し、‘ 状態（1，a，1）’を始点とし‘ 状態（1，1，b）’を終点とする‘ Change ’の名前を持つ状態遷移を図10に用意する。

3.1.3．異常状態到達可能性検査部

初期状態から、グラフ探索手法を用いて全異常状態（不可セルと記述したが不可セルではなかったセル）を発見するのが異常状態到達可能性検査部である。

3.2．状態遷移表検査ツールの使用方法

状態遷移表検査ツールは、図11の画面を持つツールである。この画面の表示部が、問題状態遷移表示部である。

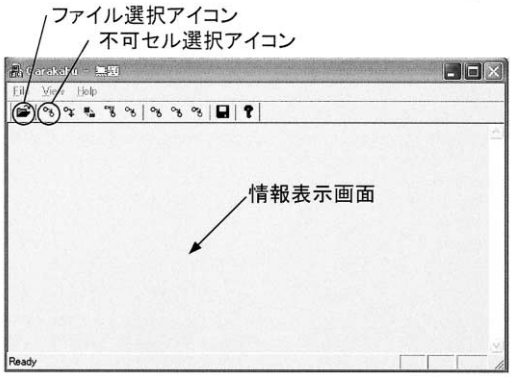

図11　状態遷移表検査ツールの画面

ファイル選択アイコンを押下することで、ファイル選択ダイアログが表示され、入力する状態遷移表（全サブシステムの状態遷移表）を選択する。

不可セル選択アイコンを押下することで、不可セル選択ダイアログが表示され、不可セルと記述したが不可セルではなかったセルが表示される（図12）。セルを選択すると、後述するシーケンスを強調表示する状態遷移表の選択画面が表示される（図13）。状態遷移表を選択すると、初期状態から異常状態（不可セルと記述したが不可セルではなかったセル）までのイベントのシーケンスが逆順（異常状態が上）に表示される（図14、図15）。

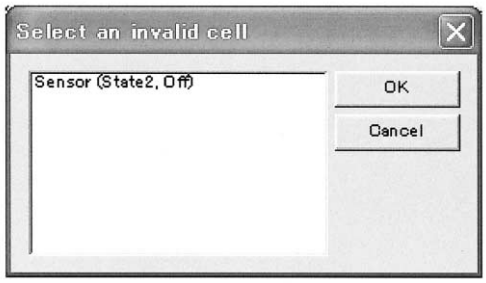

図12　不可セルではなかったセルの表示

図13　強調する状態遷移表の選択画面

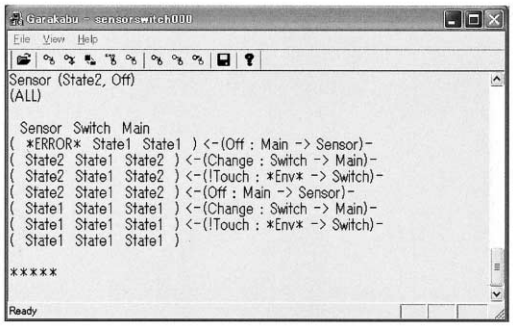

図14　シーケンスの表示（ALLの場合）

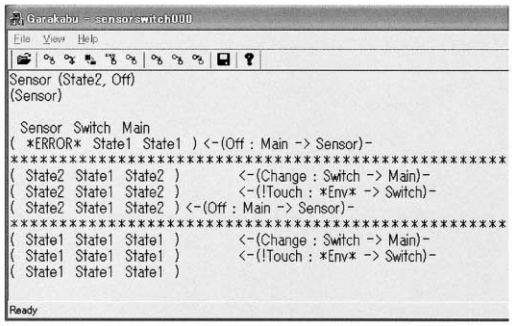

図15　シーケンスの表示（Sensorの場合）

図12より、不可セルと記述したが不可セルではなかったセルは、Sensorのセル（State2, Off）である。

不可セル選択アイコンでSensorのセル（State2, Off）を選択し、強調する状態遷移表の選択画面でALLを選択すると図14が、Sensorを選択すると図15が表示される。

図15では、Sensorの状態が変わるごとに‘＊＊＊＊’で区切られ、Sensorが受理するイベントが左寄りに、その他のイベントが右寄りに表示されている。

図15上のシーケンスを辿っていくと、‘Off’イベントをSensorが２度連続して受理していることに気づく。図２を見ると、Sensorは本来‘Off’イベントと‘On’イベントを交互に受理することを想定した設計になっている。従って、２度目の‘Off’イベントの発火、すなわち、Main（図４）のセル（State2, Change）のアクションは、‘event（Sensor,Off）;’ではなく‘event（Sensor,On）;’であるべきだと予想できる。実際、ここを上記のように変更すると、不可セルと記述したが不可セルではなかったセルは無くなる。

このように、状態遷移表検査技術を用いることで、不可セルと記述したが不可セルではなかったセルを発見するだけでなく、その問題を修正するためのヒントを与えることができる。

４．まとめ

複数のサブシステム（タスクなど）が組み合わさって出来ているシステムの場合、システムの全体像の把握が困難になり、設計者が不可セルと判断して×印を付けたとしても、組み合わせた全体のシステムでは、指定した状態に指定したイベントが発生してしまうという問題が発生することがある。

本稿では、状態遷移表検査技術を用いてこの問題を発見する方法と、更に、この問題を修正するヒントを与える方法について議論した。

５．謝辞

本研究は、文部科学省知的クラスター創成事業の一環として実施しました。各関係機関、関係者の方々に感謝します。

６．参考文献

［1］渡辺政彦,“拡張階層化状態遷移表設計手法Ver.2.0”, 東銀座出版社, 1998.