

モデル検査機構によるユーザインターフェース評価システムの設計

平石 広典[†] 溝口 文雄^{†‡}

[†]株式会社ウィズダムテック [‡]東京理科大学理工学部

1 はじめに

ユーザインターフェース評価は、専門家による経験的な評価やアンケートによる評価、実際の利用実験を通じて評価がなされるのが一般的であり、ボタンや画面のデザイン、反応性や操作順序などの様々な要因によって評価がなされる。

本研究では、ユーザインターフェースの操作系列 [3] についての評価システムを設計した。被験者による操作実験から得られた操作系列の視覚化によって、どのような状況でユーザが戸惑いスリップ [1] を起こすのかを明らかにすることができる。また、モデル検査の手法 [2] を適用することで、操作系列の検証を行い、操作フローにおける不正な操作や、デッドロックを検知することができる。それによって、操作系列におけるユーザインターフェースの変更や最適化を支援することが可能である。

2 操作系列の視覚化

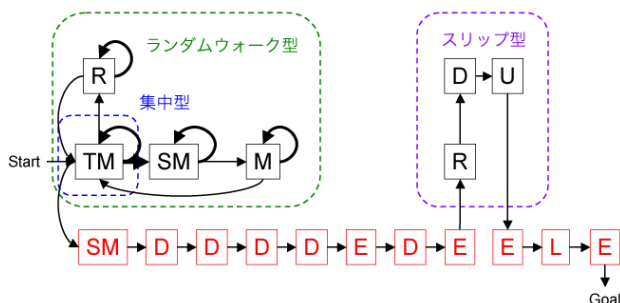


図 1: ボタン操作系列の視覚化例 (TM: トップメニュー, SW: システムメニュー, M: メニュー, U: 上, D: 下, R: 右, L: 左, E: 決定)

図 1 は、図 2 に示したビデオデッキのリモコンによる操作実験で得られた操作系列の視覚化例である。VHS, DVD, ハードディスクを搭載したビデオデッキのリモコンであり、「メニュー」と記述されたボタンが3つも存在する。それぞれのボタンにはそれぞれの機能が割り当てられており、現在のビデオデッキは様々な機能を有しているため、操作が複雑化しているのが現状である。

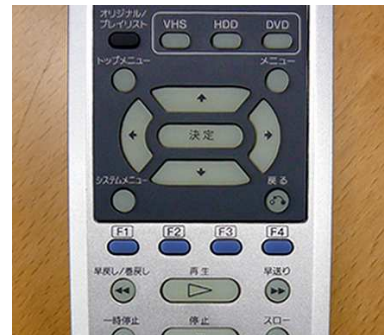


図 2: 一般的なビデオデッキのリモコン

図 1 の結果は、このビデオデッキを一度も操作したことのない被験者がマニュアルを見ないで、「DVDのファインライズ」を実行しようとした時の操作系列を示した。それぞれの記号はリモコンのボタン操作を意味しており、Start から操作を開始し、矢印の順番にボタン操作を実行したことを意味する。赤く記述された系列は、このタスクを実行するための最短系列である。最短系列にのらない操作は一つの記号にまとめられ、R → R と表現せずに R に戻る矢印として表現される。また、線の太さはその経路を通った回数を意味する。

この被験者は、「トップメニューボタン」(TM) からスタートし、「システムメニューボタン」(SM)*、「右ボタン」(R)、「メニューボタン」(M)などをランダムに操作した。被験者ははじめての操作であったため、いくつかのボタンを押して操作を理解しようとしている。このように最短系列から外れた部分にランダムウォーク型の特徴が視覚的に理解できる。また「トップメニューボタン」の操作に集中しているといった集中型の特徴も見取れる。さらに、最短系列の後半部分に「決定ボタン」(E)ではなく「右ボタン」(R)を押してしまったといったスリップ型の特徴が現れている。

視覚化によって、ユーザが操作しやすい操作、間違えやすい操作などを明確化することが可能である。

3 SPIN による操作モデルの検証

SPIN[2] は、分散、並列プロセスのモデル検査を行うツールであり、ソフトウェアの検証やセキュリティプロトコルの検証に利用されている。モデルが取り得る状態遷移を網羅的に生成して、デッドロックや不正な状態の発生、到達できない状態の検知を行うことができる。

*この機器では、トップメニュー表示中にシステムメニューを押しても、システムメニューが表示されない。

Verification of a user interface using a model checker
Hironori Hiraishi[†], Fumio Mizoguchi^{†‡}
[†]WisdomTex Inc.
[‡]Faculty of Sci. and Tech. Tokyo University of Science

3.1 操作モデルの記述

SPIN では、Promela を利用してモデルの記述を行うことで検証を実行することができる。以下は Promela で記述した図 2 の操作モデルの一部である。

```
1. /* トップメニュー */
2. active proctype top_menu(){
3.   do
4.     :: mode == MODE_VHS
5.       -> d_step{mode = MODE_HD;
6.           disp = DISP_TITLE_LIST;}
7.     :: mode == MODE_HD
8.       -> disp = DISP_TITLE_LIST;
9.     :: mode == MODE_DVD
10.      -> disp = DISP_TITLE_LIST;
11.   od
12. }
13.
14. /* システムメニュー */
15. active proctype system_menu(){
16.   do
17.     :: mode == MODE_VHS
18.       -> d_step{mode = MODE_HD;
19.           disp = DISP_SYSTEM_MENU;}
20.     :: mode == MODE_HD
21.       -> disp = DISP_SYSTEM_MENU;
22.     :: mode == MODE_DVD
23.       -> disp = DISP_SYSTEM_MENU;
24.   od
25. }
```

上の記述は、「トップメニューボタン」と「システムメニューボタン」の記述である。例えば、トップメニューの記述における 4-6 行目では、モード (mode) が VHS(MODE_VHS) の時にボタンが押された場合 (4 行目)、モードが HD(MODE_HD) に変わり (5 行目)、同時に表示 (disp) がタイトルリスト (DISP_TITLE_LIST) に変化する (6 行目) ことを意味する。つまり、ボタンが押された (操作がなされた) 時の状態変化を記述している。

また、ボタン (操作) は独立した並行プロセスとして記述されている。ユーザはどのような状態でも物理的にリモコンのボタンを押すことが可能であり、また、同時に押すことも可能である。そのため並行プロセスとして記述し、ユーザの操作を常に受け付けられるように表現している[†]。

3.2 操作モデルの検証

SPIN による操作モデルの検証によって、操作系列における以下を検出することが可能である。

● デットロックの検出

Promela で記述された操作モデルを SPIN によって実行すると、デットロックが生じた場合にはエラーを出力する。エラーが出力された場合には、トレースを取ることが可能で、どのような操作でデットロックが発生するかを確認することが可能である。

[†]ソフトウェア等の操作で、ある操作の後に画面が切り替わって新たなボタンが表示されるような場合は、操作後に起動するプロセスとして記述できる。

● 不正な操作の検出

SPIN では Never Claim といった機能があり、遷移してはならない状態を定義し、その状態を取り得る場合には、エラーとして検出することができる。想定していない操作や操作系列を Never Claim として定義することで、不正な操作が可能かどうかを検出することができる。

● 到達できない操作の検出

操作モデルに到達できない状態が存在する場合には、SPIN では unreachable といった結果出力され、どの状態に到達できないのかを明示的に表示する。これは到達できない操作を意味し、実際の操作モデルにおいては考えにくいだが、インターフェースの変更や最適化に有効である。

3.3 インターフェースの変更と最適化

図 2 のようなリモコンでは、一般的に、なるべくボタンは少ない方がいい。しかし、それぞれのボタンにはそれぞれの機能が割り当てられおり、簡単に変更することはできない。しかしながら、操作モデルを定義し検証を行うことで、あるボタンを削除した場合に生じるデットロックや到達できない操作を検出することが可能である。つまり、インターフェースの変更をおこなう場合に、SPIN を利用することで全体の操作系列の正当性を保ちつつ変更することが可能である。

4 おわりに

本研究では、ユーザインターフェースにおける操作系列の評価として、操作系列の視覚化と SPIN による操作モデルの検証といった方法を提案した。操作系列の視覚化では、実際のユーザ実験から得られた操作系列を視覚化することで、ユーザの起こしやすい操作やスリップを明確にすることが可能である。また、操作モデルの検証によって、操作モデルに存在するデットロックや不正な操作を検知することが可能である。

この視覚化と操作モデルの検証は、実際のユーザ評価の結果を明確にし、ユーザインターフェースの再デザインを実行する際に特に有効的な方法で、検証機能によって全体の正当性を保ちつつインターフェースの変更が可能である。

また、操作系列の評価としてある操作から次の操作に到達できるステップ数が重要であると考えられる。操作間の距離の算出が今後の課題である。

参考文献

- [1] Donald A. Norman, Categorization of action slips, *Psychological Review* 88, 1981.
- [2] Gerard J. Holzmann, *The Spin Model Checker: Primer and Reference Manual*, Addison-Wesley Pub. 2003.
- [3] William J. Clancey, *CONCEPTUAL COORDINATION: How the Mind Orders Experience in Time*, Lawrence Erlbaum Publishers, 1999.