

状態マシン図を用いたスマートフォンアプリのモデル化手法

宮崎 仁^{†1} 横川 智教^{†2} 落水 恭介^{†2}
佐藤 洋一郎^{†2} 有本 和民^{†2}

本稿では、UML の状態マシン図を用いてスマートフォンアプリをモデル化する手法を提案する。アプリを動作させるユーザイベントのうちタップイベントに焦点を当て、タップイベントにより内部状態を変化させながら画面移動するアプリの動作を状態マシン図でモデル化する。

A Modeling Method for Smartphone Apps using State Machine Diagrams

HISASHI MIYAZAKI,^{†1} TOMOYUKI YOKOGAWA,^{†2}
KYOSUKE OCHIMIZU,^{†2} YOICHIRO SATO^{†2}
and KAZUTAMI ARIMOTO^{†2}

In this paper, we propose a method for modeling smartphone apps using UML state machine diagrams. We focused on a tap event which is one of user operations on smartphone apps. In our method, changes of internal states and screen transitions caused by tap events are represented by state machine diagrams.

1. ま え が き

近年、スマートフォンやタブレット端末の普及により、それらの上で動作するアプリケーション（以下、アプリと呼ぶ）が広く利用されている。ユーザの増加によりアプリが多様化したことから、その信頼性および品質の向上が求められている。そこで、テストやモデル検査などのソフトウェアの高信頼化を支援する技術の需要が高まっている。これまでに、モデル検査や自動テスト生成など数多くの手法が提案されているが、これらを効果的に利用するためには、アプリのモデル化が必要となる。

これまでに筆者らは、文献¹⁾に示すような UML の状態マシン図を用いてアプリの動作をモデル化手法を提案してきた。状態マシン図で記述した設計に対してモデル検査や自動テスト生成を行うための手法は数多く提案されているため²⁾、これらの手法を利用することも可能である。しかしながら、モデル検査を利用するためには、状態爆発を回避する必要がある。本稿では、状態マシン図の状態数を削減してアプリの動作をモデル化する手法について示す。

2. アプリの構造と動作

アプリは、ユーザによる画面上のタッチ操作や、アプリ内部でのメッセージ通信などに基づいて動作するが、本稿では、画面のタッチ操作（以下、タップイベント）によって引き起こされるアプリの振る舞いのみをモデル化の対象とするものとする。アプリの構造はアプリの各画面がもつ情報と画面間によって定義される。画面はタップイベントに反応するボタンをもち、タップされたボタンと内部状態に依存して移動先の画面が決定される。よって、アプリの構造は、画面の集合 V 、ボタンの集合 B 、内部状態の集合 S 、そして画面へのボタンの割り当て $A: V \rightarrow 2^B$ により定義される。アプリの動作は、開始画面 V_0 、初期状態 S_0 、そして画面遷移の集合 $N: V \times S \times B \rightarrow V \times S$ により定義される。

タップイベントで動作する音楽再生アプリの例を図 1 に示す。このアプリは曲目リスト (List)、音楽 A 再生 (A)、音楽 B 再生 (B) の 3 つの画面をもつ。画面 List には “Song A”, “Song B”, “Play”, “Stop” の 4 つのボタンが割り当てられる。“Song A” および “Song B” のタップにより画面 A および B にそれぞれ移動し、“Play” と “Stop” のタップでは画面は変化しない。画面 A と B には “Play”, “Stop”, “Back” のボタンが割

^{†1} 川崎医療福祉大学

Kawasaki University of Medical Welfare

^{†2} 岡山県立大学

Okayama Prefectural University

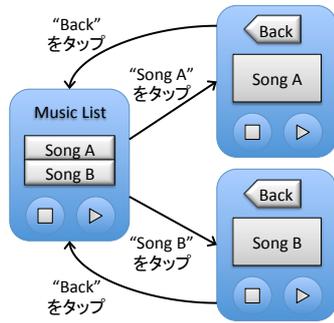


図 1 タップイベントで動作する音楽再生アプリ

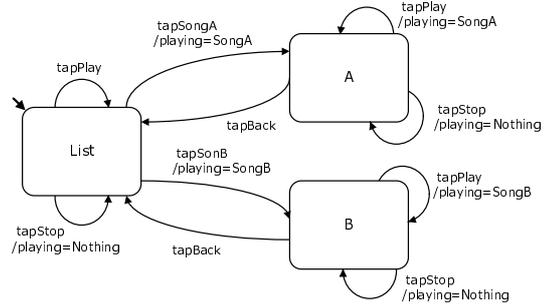


図 3 状態マシン図によるモデル

り当てられており，“Play”と“Stop”のタップでは画面は変化せず，“Back”のタップにより画面 List へと移動する。また，開始時の画面 List では音楽は停止状態であるが，画面 A および B に移動すると音楽 A および音楽 B の再生状態となる。各画面で“Stop”をタップすると音楽は停止状態となり，画面 A および B で“Play”をタップすると音楽 A および音楽 B の再生状態となる。図 2 にアプリの構造および動作の定義を示す。

$$\begin{aligned}
 V &= \{List, A, B\} \\
 B &= \{SongA, SongB, Play, Stop, Back\} \\
 S &= \{PlayingA, PlayingB, Stopped\} \\
 A &= \{\langle List, \{SongA, SongB, Play, Stop\} \rangle, \\
 &\quad \langle A, \{Play, Stop, Back\} \rangle, \\
 &\quad \langle B, \{Play, Stop, Back\} \rangle\} \\
 V_0 &= \{List\} \\
 S_0 &= \{Stopped\} \\
 N &= \{\langle (List, Stopped, SongA), (A, PlayingA) \rangle, \\
 &\quad \langle (List, Stopped, SongB), (B, PlayingB) \rangle, \\
 &\quad \langle (List, PlayingA, Stop), (List, Stopped) \rangle, \\
 &\quad \langle (List, PlayingB, Stop), (List, Stopped) \rangle, \\
 &\quad \langle (A, PlayingA, Stop), (A, Stopped) \rangle, \\
 &\quad \langle (A, PlayingA, Back), (List, PlayingA) \rangle, \\
 &\quad \langle (A, Stopped, Play), (A, PlayingA) \rangle, \\
 &\quad \langle (A, Stopped, Back), (List, Stopped) \rangle, \\
 &\quad \langle (B, PlayingB, Stop), (B, Stopped) \rangle, \\
 &\quad \langle (B, PlayingB, Back), (List, PlayingB) \rangle, \\
 &\quad \langle (B, Stopped, Play), (B, PlayingB) \rangle, \\
 &\quad \langle (B, Stopped, Back), (List, Stopped) \rangle\}
 \end{aligned}$$

図 2 アプリの構造および動作

3. 状態マシン図によるアプリのモデル化

本手法では，アプリの画面遷移を状態マシン図でモデル化する。画面遷移の関係は，画面と画面間の移動を状態と遷移とすることで表す。画面は，各画面に配置されたボタンをタップすることで移動するため，ユーザのタップイベントを画面間の移動を表す遷移の

イベントとして表す。また，アプリの内部状態は変数で表し，内部状態の更新は変数の更新として遷移のアクションで表す。ボタンをタップすることで，内部状態のみが更新され画面は移動しない場合は，変数の更新をアクションとする自己遷移で表す。

図 1 から得られる状態マシン図を図 3 に示す。例えば， $\langle (List, Playing=Nothing, TapSongA),$

$(A, Playing=SongA) \rangle \in N$ は，状態 List から状態 A へのラベル $tapSongA / playing=SongA$ をもつ遷移とすることで，画面 List でボタン SongA をタップすることをきっかけに，画面は A へ移動し SongA が再生されることを表している。また， $\langle (A, Playing=songA, TapStop), (A, Playing=Nothing) \rangle \in N$ は，状態 A からのラベル $tapStop / playing=Nothing$ をもつ自己遷移とすることで，画面 A でボタン Stop をタップすることをきっかけに，画面は移動せずに SongA が停止することを表している。

4. あとがき

本稿では，スマートフォンやタブレット端末上で動作するアプリがタップイベントをきっかけとしておこす動的な振る舞いを，状態マシン図を用いてモデル化する手法を提案した。今後はより大きな例題に適用し，その実用性を確認する予定である。

参考文献

- 1) 落水恭介, 横川智教, 宮崎仁, 佐藤洋一郎, 有本和民: 状態マシン図を用いたスマートフォンアプリのモデル化, 信学技報, vol. 113, no. 159, SS2013-21, pp. 49-54, 2013
- 2) William Chan, Richard J. Anderson, Paul Beame, David H. Jones, David Notkin, William E. Warner: Optimizing Symbolic Model Checking for Statecharts, IEEE Transactions on Software Engineering, vol. 27, no. 2, pp. 170-190, 2001