

令和元年5月24日現在

機関番号：32612

研究種目：基盤研究(C) (一般)

研究期間：2015～2018

課題番号：15K00104

研究課題名(和文) コンテキスト情報に基づいたモバイルアプリケーションのテストケース生成に関する研究

研究課題名(英文) Generating Test Cases for Mobile Applications based on Context Information

研究代表者

高田 眞吾 (Takada, Shingo)

慶應義塾大学・理工学部(矢上)・教授

研究者番号：60273843

交付決定額(研究期間全体)：(直接経費) 3,200,000円

研究成果の概要(和文)：携帯端末向けのアプリケーションは、画面を通したユーザの意図的な入力以外に様々なコンテキスト情報を入力として受け付ける。コンテキスト情報には、GPSによる位置情報や電話の受信などが存在する。これがそのアプリに必須でない場合、テスト対象としないことが一般的だが、テストを怠ると、アプリケーションに悪影響を及ぼす可能性がある。本研究では、コンテキスト情報に基づいたモバイルアプリケーションのテストケース生成支援環境を構築した。支援環境は、アプリケーションの振る舞いモデル、コンテキスト情報に対応するイベントをパターンとしてまとめるなどして実現した。

研究成果の学術的意義や社会的意義

アプリケーションに対する直接的な入力以外のコンテキスト情報もテスト対象にする点に学術的意義がある。また、可能性が無数にあるため、完璧なテストができないと一般的に言われているが、組み合わせ爆発を抑えようとし、少しでも完璧に近づけようとしている点にも学術的意義がある。昨今、誰でもスマホを使用する世の中において、モバイルアプリケーションのバグは非常に悪影響をもたらす。このようなバグを減らし、高品質のモバイルアプリケーションを目指している点に、社会的意義がある。

研究成果の概要(英文)：Mobile applications can take input other than those that the user intentionally provided. Such input, which we call context information, include GPS information and phone call reception. When these information are not directly part of the application, they are often not targeted for testing. However this may lead to failures. This research thus proposes an environment for generating test cases for mobile applications based on context information. We realize this by incorporating application behavior model, defining patterns for events that correspond to context information, etc.

研究分野：ソフトウェア工学

キーワード：ソフトウェアテスト モバイルアプリケーション コンテキスト情報 イベント

様式 C - 19、F - 19 - 1、Z - 19、CK - 19 (共通)

1. 研究開始当初の背景

スマホやタブレットなどの携帯端末が昨今普及しており、それらを対象とした様々なアプリケーション(以下、モバイルアプリケーション)が開発されている。モバイルアプリケーションは従来のアプリケーションと異なるため、新しいテスト手法が必要である(引用文献)。特に、従来のPCなどを対象としたアプリケーションと異なり、画面を通じたユーザの意図的な入力以外に様々なコンテキスト情報を入力として受け付けうる。コンテキスト情報には、GPSによる位置情報、バッテリーの残存量、携帯端末の向きなど多種である。これらコンテキスト情報は常に発生しており、ユーザが意図していなくてもアプリケーションが入力を受け付ける可能性がある。そのため、モバイルアプリケーションのテストを行う際、様々なコンテキスト情報を意識して行う必要がある。

さらに、それぞれのコンテキスト情報の取り得る値は広範囲である。その結果、コンテキスト情報を網羅的にテストで扱おうとした場合、アプリケーションに対する入力の組み合わせ爆発は簡単に起きる。

Google社は、スマホの基本OSのAndroid上でアプリケーションのテストを支援するためのツールや方針を提供している。しかし、現実には、上記のコンテキスト情報に関わるテストを完全に支援しているとは言い難い。また、モバイルアプリケーションのテストの大半は手動で行っており、自動的なテスト支援ツールが望まれている(引用文献)。この現状を反映して、モバイルアプリケーションのテストに関する研究が近年活発に行われており、適応的ランダムテスト(引用文献)、コンコリックテスト(引用文献)、パターン(引用文献)、モデルベーステスト(引用文献)など、従来からある様々な手法をモバイルアプリケーションテストに応用したものが多数提案されている。しかし、これら手法におけるコンテキスト情報の扱いは不十分である。

2. 研究の目的

本研究は、コンテキスト情報に基づいたモバイルアプリケーションのテストケース生成支援環境の構築を目的とする。モバイルアプリケーションとして、Androidアプリケーションを対象とする。提案手法は、携帯端末が入力として受け付けうる様々な情報を対象とする。これにはユーザが直接入力した情報のみならず、GPSによる位置情報や携帯端末の向きなどのコンテキスト情報を含む。本環境を実現する際の主要な課題は次の4つである：(1) 対象とするコンテキスト情報の整理、(2) コンテキスト情報のテストケースへの導入方法、(3) 組み合わせ爆発の対処、(4) テストケースの自動実行。

3. 研究の方法

テストケース生成支援環境の構築のために、次の項目に基づいて研究を進めた。

- (1) テスト支援環境が対象とするコンテキスト情報の調査
- (2) 知識ベースの策定
知識ベースにはテストシナリオと呼ばれるアプリケーションの使い方における各ステップを保存し、利用することができる。本研究では、上記(1)であげたコンテキスト情報をどのような形式で知識ベースに保存するか検討する。
- (3) テストケースの生成手法の提案と支援環境の実装
組み合わせ爆発が容易に起こるため、テストケースの優先順位づけや選択など、組み合わせ爆発に対処する方法を検討する。また、テストケースは、自動実行できるようにする。

4. 研究成果

研究成果は、コンテキスト情報の調査結果、二種類のテストケース生成支援環境、今後の展望の大きく三つに分けて示す。テストケース生成支援環境については、二つの方針に基づいてプロトタイプを作成を行った。一つ目は、画面遷移モデル、ユーザによる使用モデル、プログラム変更情報の三つの観点から情報を取得し、それを利用してテストケースを作成する方針であり、もう一つはコンテキスト情報をイベントととらえた方針である。

(1) コンテキスト情報

本研究は、Androidアプリケーションを対象としている。そのため、最終的にはAndroidのプラットフォームがAPIを提供しているものに、コンテキスト情報が限定される。これには次のようなものがあげられる。

- GUIを通して得られるユーザイベント(ボタン押下など)
- デバイスのセンサにより取得される、外部からのイベント(GPS、温度など)
- デバイスのハードウェアからのイベント(電池など)
- スマホ特有のイベント(電話など)

上記のイベントは、モバイルアプリケーションによって、明示的に扱っているものもあれば、暗黙的に扱われるものもある。前者を明示イベント(explicit event)と呼び、後者を暗黙イベント(implicit event)と呼ぶ。前者は通常のテストが対象としているイベントであり、モバイルアプリケーションテストの多くの研究が対象としている。後者は本研究が目指している種類のイベン

トである． を特に下記研究成果(3)において，外部イベントと呼ぶ．

(2) 振舞いモデルに基づいたテストケース生成支援環境

テストケース生成環境のベースとして，モデルベースに基づいた支援環境を作成した．本環境のキーポイントは次の通りである．

- 手動テスト手法を部分的に取り込むことにより，人間のドメイン知識を利用し，コード網羅率を上げる．
- 二つのアプリのバージョンの差異に注目し，テストケースの削減を行う．

テストケース生成の具体的な手順は次の通りである．

既存ツール(GUIRipper)を用いて，GUI のモデルを作成する．

人間にアプリケーションを実際の Android デバイス上で実行させ，使用ログを取得する．

GUI モデルと使用ログを組み合わせる振る舞いモデルを作成する．

振る舞いモデルから遷移パスを取得する．

Eclipse JDT を用いてコールグラフを作成する．

振る舞いモデルやコールグラフを解析し，イベントシーケンスを作成し，テストケースを生成する．

アプリケーションの以前のバージョンのコードがある場合，現在のバージョンと比較し， で生成したテストケースのうち，変更の影響を受けるテストケースのみを残し，最終的なテストスイートする．

本手法は，三つのアプリケーションに対して評価を行った．その結果，コード網羅率は，既存ツールの GUI Ripper と比較して 4～35%の向上があった．また，上記手順 を実施することにより，テストケースの数をおおよそ半分に減らすことができた．この成果は，学会発表 で発表している．

上記の手法では，手順 で人間が利用しなかった機能をテスト対象にできない．そこで，振る舞いモデルを作成する前に，Android の Hierarchy Viewer を用いて，各 Activity 上の GUI 部品を取得し，木構造を得る．振る舞いモデルを作成するとき，Activity 上のすべての GUI 部品が扱えるイベントをすべて網羅できていなかった（人間が利用しなかった機能があった）場合を考える．例えば，図 1(a)では，Activity B から行先のない点線の矢印があるが，これは使用ログに現れなかったイベントを指す．この場合，Kneser-Ney 手法を用い，その未知のイベントに対して実行される確率を与える（図 1(b)）．実際にそれが実行された場合，その確率値を更新する（図 1(c)）．提案手法を既存ツール(Android Monkey と Dynodroid)と比較して評価したとき，一つのアプリケーションを除いて，すべてに対してコード網羅率が 2～17%の向上があった．さらに，発見したエラーは Monkey と Dynodroid と同等であった．この成果は，雑誌論文 と学会発表 で発表した．

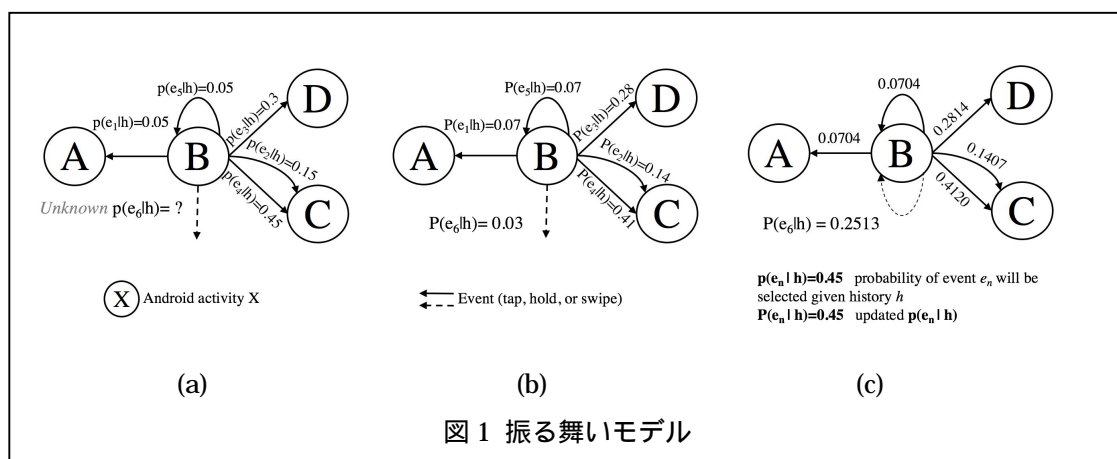


図 1 振る舞いモデル

(3) イベントパターンに基づいたテストケース生成支援環境

二つ目のテストケース生成支援環境では，外部イベントを組み合わせたパターンをあらかじめ作成し，それを利用するという基本方針をとった．36 個のイベントを支援環境の対象とし，それらを 25 個のカテゴリに分類した．例えば，Phone Call というカテゴリには，Idle, Off-hook, Ringing などのイベントがある．任意の数のイベントを組み合わせることにより，イベントパターンを作成する．組み合わせを作成するとき，無効な組み合わせは排除した．例えば，Phone Call の Idle イベントは，初期イベントにはなりえず，Ringing や Off-hook のあとでないといけない．また，Ringing は最後のイベントにはなりえない．作成したパターンは，知識ベースに登録する．最終的に，2047 個のイベントパターンを作成し，知識ベースに登録した．

テストケース生成は，2 段階で行われる．第 1 段階では，アプリケーションにある GUI 部品が対応できるイベントに基づき，基本テストケースを生成する．各基本テストケースは，イベ

ントのシーケンス (e1, e2, e3, ..., en) として表される。第2段階では、知識ベースから二種類のイベントパターンを取得する。一種類目は、基本テストケース ei と同じカテゴリの外部イベントがないか確認する。あれば、その外部イベントが含まれるイベントパターンを知識ベースから取得する。二種類目は、テストが特にテスト対象としたいイベントカテゴリを指定することにより、そのイベントパターンを取得する。最後に、基本テストケースと二種類のイベントパターンを組み合わせることにより、拡張テストケースを生成する。例えば、図2において、(1. Open menu, 2. Back)の二つのイベントからなる基本テストケースがあり、テストが Phone Call カテゴリを指定することにより、(A. Call State Ringing, B. Call State Idle)の二つのイベントからイベントパターンを取得したとする。これらを組み合わせることにより、ATC1, ATC2, ATC3 の三つの拡張テストケースを生成することとなる。三つの Android アプリケーションに対して提案手法を適用した結果、135～825個のテストケースを生成し、新たなバグを見つけることもできた。この成果は、学会発表 で発表した。

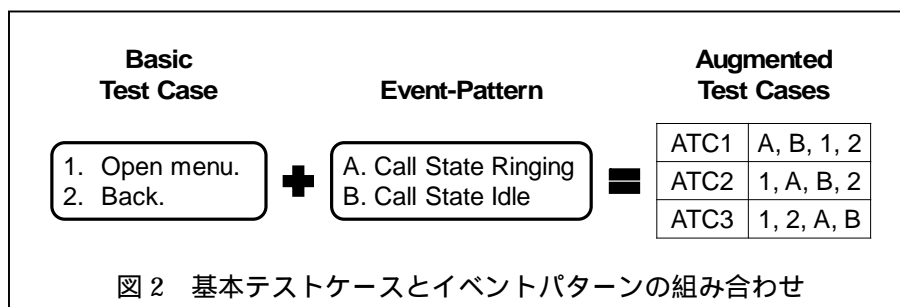


図2 基本テストケースとイベントパターンの組み合わせ

上記の手法では、簡単に組み合わせ爆発が発生しやすいため、sequence covering array を導入することにより、組み合わせ爆発を抑制した。Sequence covering array は、イベントの t-way の組み合わせを最低でも一つ含むように作成したシーケンスである。例えば、a, b, c, d の4つのイベントがあった場合、ありうるすべての組み合わせの数は 24 (=4!)である。しかし、3-way sequence covering array は、(a, d, b, c) (b, a, c, d) (b, d, c, a) (c, a, b, d) (c, d, b, a) (d, a, c, b)の6つですむ。Sequence covering array を導入するにあたり、イベント数が3～10の間のものを対象にした。また、Android 開発キットの Espresso フレームワークを利用することにより、テストケースの自動実行を実現した。評価の結果、生成したイベントシーケンス数を大幅に抑え、既存ツールと比較してコード網羅率が約18%向上した。この成果は、学会発表 で発表した。

(3) 今後の展望

本研究でコンテキスト情報を対象にしたテストケースを生成する手法を提案した。技術が発展するにつれ、コンテキスト情報の種類が増えることが想定される。さらに、より複雑なモバイルアプリケーションが開発されることが想定される。この場合、より多くの種類のイベントに対処する必要が発生するのみならず、アプリケーションの隅から隅までテストするためには、より長いイベントシーケンスを扱える必要性が出ると考えられる。今後はこれらを想定して、スケーラビリティのあるテストができるようにする必要があると考えられる。

<引用文献>

- H. Muccini, et al: “Software Testing of Mobile Applications: Challenges and Future Research Directions”, Proc. of AST 2012, pp.29-35 (2012).
- M. Joorabchi, et al: “Real Challenges in Mobile App Development”, Proc. of ESEM 2013, pp.15-24 (2013).
- Z. Liu, et al: “Adaptive Random Testing of Mobile Application”, Proc. of ICCET 2010, Vol.2, pp.297-301 (2010).
- S. Anand, et al: “Automated Concolic Testing of Smartphone Apps”, Proc. of SIGSOFT/FSE 2012, pp.1-11 (2012).
- D. Amalfitano, et al: “Considering Context Events in Event-Based Testing of Mobile Applications Domenico”, Proc. of TESTBEDS 2013, in ICSTW, pp.126-133 (2013).
- T. Griebel, et al: “A Model-Based Approach to Test Automation for Context-Aware Mobile Applications”, Proc. of SAC 2014, pp.420-427 (2014).

5 . 主な発表論文等

[雑誌論文](計1件)

Woramet Muangsiri, Shingo Takada: “Random GUI Testing of Android Application Using Behavioral Model”, International Journal of Software Engineering and Knowledge Engineering, Vol.27, Issue 09n10, pp.1603-1612, 査読あり, DOI: <https://doi.org/10.1142/S0218194017400149> (2017).

〔学会発表〕(計 9 件)

Maria Azriel Therese Eala, Shingo Takada: “Applying Combinatorial Strategies in GUI Testing Incorporating Context Events in Android Applications”, 日本ソフトウェア科学会第 35 回大会, <http://jssst.or.jp/files/user/taikai/2018/FOSE/fose1-3.pdf> (2018).

Maria Azriel Therese Eala, Shingo Takada: “GUI Testing incorporating Context Events in Android Applications”, ソフトウェアエンジニアリングシンポジウム(SES2017), ポスターセッション (2017).

Woramet Muangsiri, Shingo Takada: “Random GUI Testing of Android Application Using Behavioral Model”, Proc. of 29th International Conference on Software Engineering and Knowledge Engineering (SEKE 2017), <https://doi.org/10.18293/SEKE2017-099> (2017).

Siena Yu, Shingo Takada: “Mobile application test case generation focusing on external events”, Proc. of the 1st International Workshop on Mobile Development (Mobile! 2016), pp.41-42, <https://doi.org/10.1145/3001854.3001864> (2016).

Jose Lorenzo San Miguel, Shingo Takada: “GUI and Usage Model-Based Test Case Generation for Android Applications with Change Analysis”, Proc. of the 1st International Workshop on Mobile Development (Mobile! 2016), pp.43-44, <https://doi.org/10.1145/3001854.3001865> (2016).

Jose Lorenzo San Miguel, Shingo Takada: “Generating Test Cases for GUI-based Android Applications by Modeling and Change Analysis”, 情報処理学会第 78 回全国大会, 1J-8 (2016).

Siena Yu, Shingo Takada: “Mobile Application Testing Focusing on External Event”, 情報処理学会第 78 回全国大会, 1J-5 (2016).

Jose Lorenzo San Miguel, Shingo Takada: “Generating Test Cases for Android Applications through GUI Modeling, Usage Modeling, and Change Analysis”, Proc. of 8th International C* Conference on Computer Science & Software Engineering (C3S2E'15), pp.146-147, <https://doi.org/10.1145/2790798.2790823> (2015).

Siena Yu, Shingo Takada: “External Event-Based Test Cases for Mobile Application”, 8th International C* Conference on Computer Science & Software Engineering (C3S2E'15), pp.148-149, <https://doi.org/10.1145/2790798.2790822> (2015).

〔図書〕(計 0 件)

〔産業財産権〕

出願状況 (計 0 件)

取得状況 (計 0 件)

〔その他〕

6 . 研究組織

(1)研究分担者 (なし)

(2)研究協力者

研究協力者氏名 :

ローマ字氏名 :

科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。