

FrUITeR: A Framework for Evaluating UI Test Reuse

Yixue Zhao
yixue.zhao@usc.edu
University of Southern California
USA

Marcelo Schmitt Laser
marcelo.laser@gmail.com
University of Southern California
USA

Mark Harman
mark.harman@ucl.ac.uk
University College London
UK

Justin Chen
justin.chen@columbia.edu
Columbia University
USA

Jie Zhang
jie.zhang@ucl.ac.uk
University College London
UK

Nenad Medvidovic
nen@usc.edu
University of Southern California
USA

Adriana Sejfa
sejfa@usc.edu
University of Southern California
USA

Federica Sarro
f.sarro@ucl.ac.uk
University College London
UK

*Proceedings of the 28th ACM Joint Meeting on European Software Engineering
Conference and Symposium on the Foundations of Software Engineering, Association
for Computing Machinery, 2020, 1190–1201*

背景

- UIテストは単調で生産性が悪い
- 自動UIテスト手法の研究が進められている
- 自動UIテスト手法を効率的に評価する手法が確立していない

テスト再利用による自動UIテスト

- ユーザの操作による一連のイベントをテスト
- あるアプリケーションの一連のイベントのテストを再利用する
 - 例) Wishの一連のイベント{a1-1, a1-2, a1-3}はEtsyの各イベントに対して{a1-1→b3-1, a1-2→b3-2, a1-3→b3-3}のようにマッピングできるのでEtsyの一連のイベント{b3-1, b3-2, b3-3}でテストを再利用する

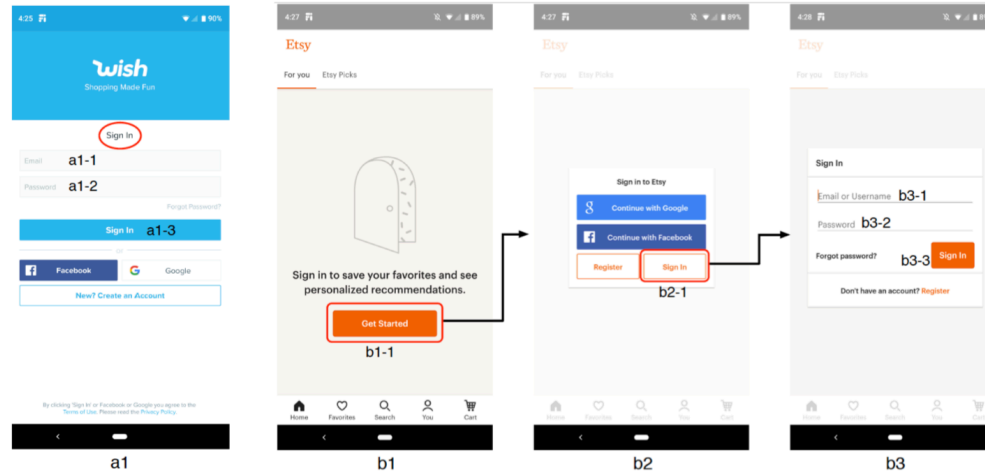


Figure 1: sign-in tests for Wish (a1) and Etsy (b1-b3).

自動UIテスト手法

- 機械学習ベースの手法
 - GUIで用いられるテキスト，画像などの要素を学習
 - アプリ固有のGUIイベントを他のテストで使用できる正規GUIイベントにマッピングする
- 類似度ベースの手法
 - テストの複製元アプリから複製先アプリのイベントと類似するものを抽出
 - 類似イベントのテストを再利用する

各手法の評価に関する問題点

1. 評価用メトリクスが標準化されていない
2. 手動操作が多い
3. 操作が標準化されていない
4. コンポーネントごとに性能を評価していない
5. 評価対象プロジェクトが異なる

目的

- 自動的にUIテスト再利用を評価するフレームワークFrUITeR(Framework for evaluating UI Test Reuse)を構築
- 問題点1～3, 5と4の一部を解決する

FrUITeRの要件定義

1. 評価に用いるメトリクスは標準的かつ実用的である
2. 手動操作を減らす
3. 再現性を確保する
4. 評価対象テスト再利用機能は、モジュール化する
5. 再利用可能なベンチマークを提供する

FrUITeRで用いられるメトリクス

- 要件1, 2, 3を満たすために必要となるメトリクスを定義
- 正確さメトリクス
 - イベントのマッピングの正確さを測る
- 実用度メトリクス
 - 転送されたテストの有効性を測る

正確さメトリクス

Table 1: Fidelity metrics as used in AppFlow [23], CraftDroid [25], ATM [21], GTM [20], and FrUITeR.

	True Pos. (TP)	False Pos. (FP)	True Neg. (TN)	False Neg. (FN)	Accuracy	Precision	Recall
AppFlow	<i>anon</i>	<i>anon</i>	<i>anon</i>	<i>anon</i>	Accuracy	<i>dnc</i>	<i>dnc</i>
CraftDroid	TP	FP1	<i>none</i>	FN	<i>none</i>	Precision	Recall
ATM/ GTM	Correctly Matched	Incorrectly Matched	Unmatched (!exist)	Unmatched (exist)	<i>dnc</i>	<i>dnc</i>	<i>dnc</i>
FrUITeR	Correct	Incorrect	NonExist	Missed	Accuracy	Precision	Recall

- **anon** : 報告されていないが, 他のメトリクスを計算する際に使用されている
- **dnc** : 計算されていないが, 他のメトリクスを使うことで定義することができる
- **none** : 計算できない

実用度メトリクス

- effort

- 転送されたテストと手動で作成されたテストの内容がどれ程近いかを測定
- レーベンシュタイン距離を測定する

- reduction

- 転送されたテストにより手動でテストを作成した場合に比べてどれほど作業量を減らせるか測定
- $\text{Reduction} = (\#gtEvents - \text{effort}) / \#gtEvents$
- $\#gtEvents$: 手動で作成されたテスト数

FrUITeRの処理内容

- 要件2, 3と要件4の一部を満たすために必要となる処理
- 2つの重要な側面
 - 入力と生成物の均一な表現
 - 評価結果を自動的に出力するカスタマイズ可能なコンポーネントのセット

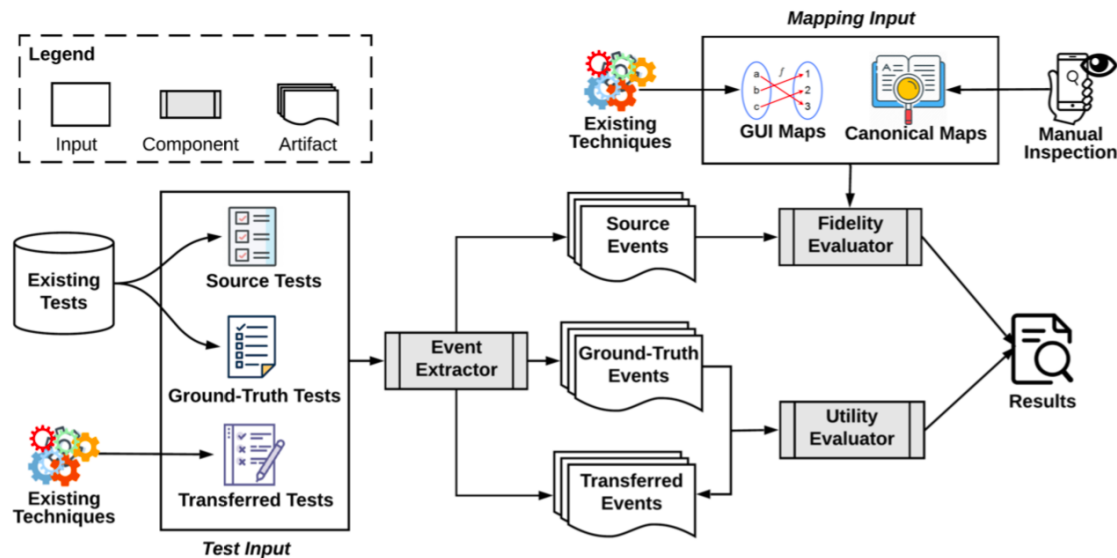


Figure 2: Overview of *FrUITeR*'s automated workflow.

Event Extractor

- テストコードから一連のイベントを抽出
- 一連のイベントの抽出は以下の手順で行う
 - テストコードを解析してイベントが発生するコードの位置を特定
 - 各イベントの呼び出し元を特定

Fidelity Evaluator

Algorithm 1: FIDELITY EVALUATOR

Input: EventList *srcEvents*, GUIMap *guiMap*,
CanonicalMap *srcCanMap*, *tgtCanMap*

Output: Sets *correct*, *incorrect*, *missed*, *nonExist*

```
1 correct = incorrect = missed = nonExist =  $\emptyset$ 
2 for i = 1 to srcEvents.size do
3   src  $\leftarrow$  srcEvents.GET(i)
4   trans  $\leftarrow$  guiMap.GETMAPPED(src)
5   srcCan  $\leftarrow$  srcCanMap.GETCANONICAL(src)
6   transCan  $\leftarrow$  tgtCanMap.GETCANONICAL(trans)
7   if trans  $\neq$  null then
8     if transCan == srcCan then
9       correct.PUT(src)
10    else
11      incorrect.PUT(src)
12  else
13    if tgtCanMap.CONTAINS(srcCan) then
14      missed.PUT(src)
15    else
16      nonExist.PUT(src)
17 return correct, incorrect, missed, nonExist
```

再利用元で遷移先のイベントがある場合,
再利用対象appのイベント遷移が
同じであれば*correct*(TP)
異なれば*incorrect*(FP)

再利用元で遷移先のイベントがない場合,
再利用対象appのイベント遷移が
存在していれば*missed*(FN)
存在していなければ*nonExist*(TN)

Utility Evaluator

- **Event Extractor**で作成されたイベントと正解イベントから実用度メトリクスを計測
- 実用度メトリクス（再喝）
 - **effort**
 - 転送されたテストと手動で作成されたテストの内容がどれ程近いかな
 - **reduction**
 - 転送されたテストにより手動でテストを作成した場合に比べてどれほど作業量を減らせるかな

Base Line Technique

- **Naïve Base Line**

- テスト複製先のアプリのイベント遷移を複製元アプリを基にランダムで選択する
- 正確さ，実用度の下限を測り，他の手法を評価するために構築

- **Perfect Base Line**

- 正解集合のテストを基にイベント遷移を選択
- 手作業で作成した正解集合のテストと他の手法で作成されたテストの実用度の違いを見るために構築

FrUITeRのベンチマーク

- 要件5を満たすために構築
- データセット
 - ショッピングアプリ：10件
 - ニュースアプリ：10件

Table 2: Summary information of benchmark apps.

	App ID	App Name	#Downloads	#Tests	#Events
Shopping	S1	AliExpress	100M	15	76
	S2	Ebay	100M	13	48
	S3	Etsy	10M	13	55
	S4	5miles	5M	12	78
	S5	Geek	10M	13	85
	S6	Google Shopping	1M	15	72
	S7	Groupon	50M	14	66
	S8	Home	10M	14	98
	S9	6PM	500K	14	63
	S10	Wish	100M	14	85
News	N1	The Guardian	5M	13	76
	N2	ABC News	5M	9	31
	N3	USA Today	5M	11	28
	N4	News Republic	50M	10	40
	N5	BuzzFeed	5M	11	50
	N6	Fox News	10M	11	28
	N7	SmartNews	10M	9	20
	N8	BBC News	10M	9	22
	N9	Reuters	1M	10	37
	N10	CNN	10M	9	24

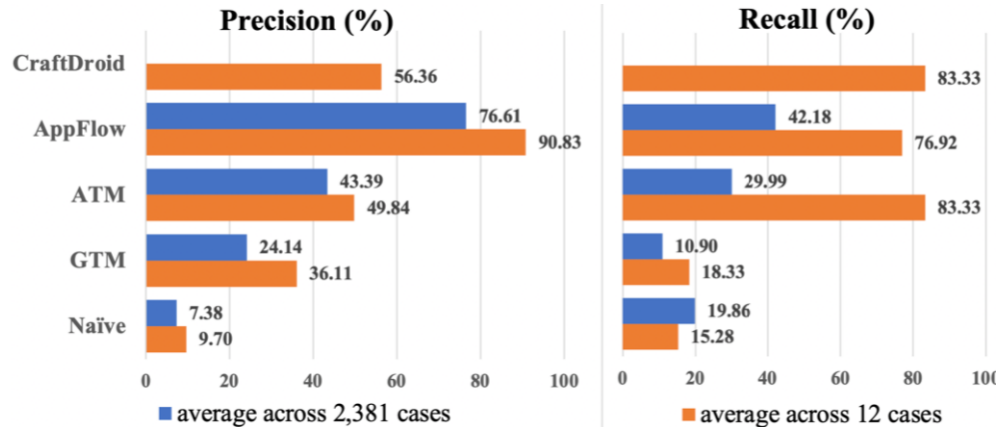
FrUITeRのベンチマーク

- AppFlow[23]で使用されたテストケースをさらに追跡してテストケースを作成

Table 3: Benchmark test cases in shopping (TS) and news (TN) categories.

Test ID	Test Case Name	Tested Functionalities
TS1/TN1	Sign In	provide username and password to sign in
TS2/TN2	Sign Up	provide required information to sign up
TS3/TN3	Search	use search bar to search a product/news
TS4/TN4	Detail	find and open details of the first search result item
TS5/TN5	Category	find first category and open browsing page for it
TS6/TN6	About	find and open about information of the app
TS7/TN7	Account	find and open account management page
TS8/TN8	Help	find and open help page of the app
TS9/TN9	Menu	find and open primary app menu
TS10/TN10	Contact	find and open contact page of the app
TS11/TN11	Terms	find and open legal information of the app
TS12	Add Cart	add the first search result item to cart
TS13	Remove Cart	open cart and remove the first item from cart
TS14	Address	add a new address to the account
TS15	Filter	filter/sort search results
TN12	Add Bookmark	add first search result item to the bookmark
TN13	Remove Bookmark	open the bookmark and remove first item from it
TN14	Textsize	change text size

イベントマッピングの正確さ



- 青色バー：ショッピングアプリとニュースアプリの両方で転送されたテストケースで計測された平均
- オレンジ色バー：CraftRoid[]の手法を他の手法と比較するためにCraftRoidで用いられたテストケース12件で計測された平均

Figure 3: Comparison of average *precision* and *recall*.

※AppFlowが機械学習ベース
それ以外は類似度ベース

- 機械学習ベースの手法の方が全体的に精度がいい
 - 学習に用いるデータの種類が類似度ベースより多いため
- Naïve BLよりもrecallが劣る手法がある

実用度

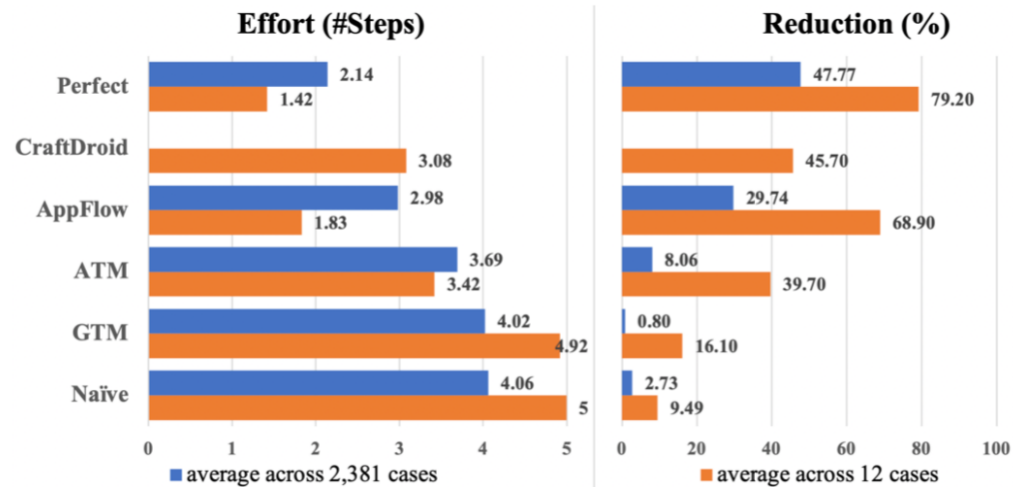


Figure 5: Comparison of average *effort* and *reduction*.

- Reductionではほとんどの手法がPerfect BLに劣る (Perfect BLも良い結果ではない)

まとめ

- 自動テスト手法を標準的に評価できるようにした
- 自動UIテスト手法を評価を半自動的にできるようにした
- 既存手法を評価した
 - 正確さは機械学習ベースの方が勝る
 - 実用度は手動よりも劣る結果が見られた

所管

- 手法構築にあたって要件定義を明確にされていて分かりやすかった
- 既存手法の解決すべき点が明らかになったためさらに研究が進められるかもしれないと思った