

# 製品使用ログからの可変性マイニングによる 運用プロファイル生成手法の提案

佐藤 孝一<sup>†</sup> 岸 知二<sup>†</sup>

<sup>†</sup>早稲田大学大学院 創造理工学研究科 経営システム工学専攻

## 1. 研究背景・目的

ソフトウェアプロダクトライン(SPL)開発は、類似ソフトウェア製品を効率的に開発するための手法である。SPL 開発では多くの製品が導出できるため、全製品のテストが現実的に不可能である。そのため、一部製品を代表製品としてコア資産のテストを行う手法があり、代表製品の選択方法として運用プロファイル(OP)を活用する手法がある。しかし、現在提案されている SPL 製品群全体の使われ方を表す OP は、別途それに対応する製品情報に関わるモデルが必要になるという制約がある。本研究では、対応するモデルを必要としない SPL 製品群の OP を提案するとともに、それを既存の製品使用ログのマイニングによって生成する手法、ならびに生成された OP を活用した代表製品の選択手法を提案する。

## 2. 関連研究

### 2.1. Devroey ら[1][2]の手法

この手法は、遷移をフィーチャでラベル付けた状態遷移系であるフィーチャドトランジションシステムズ(FTS)と SPL 製品の OP を用いて SPL 製品群のテストにおける製品優先順位付けを行うものである。まず、テスト技術者に指定された確率範囲内の遷移列を抽出し、その遷移列を FTS によってフィルタリングすることで対応する製品が存在する遷移列が得られる。製品が含む遷移列の確率から製品優先順位付けを行う。

## 3. 研究目的

### 3.1. SPL 製品群の OP 生成時の問題

個々の SPL 製品の OP を単純に統合して SPL 製品群の OP を作成した場合に起こる問題の例を図 1 に示す。図 1 は、ある SPL 製品 1 及び 2 と、それらを統合し両者の平均確率を付与した SPL 製品群の OP (1+2) である。SPL 製品群の OP から A から始まる長さ 4 の遷移列を抜き出すと、 $\{A \rightarrow B \rightarrow E \rightarrow F \rightarrow H\}$  と  $\{A \rightarrow B \rightarrow E \rightarrow G \rightarrow H\}$  という 2 つの遷移列が最

も確率の高い遷移列となるが、 $\{A \rightarrow B \rightarrow E \rightarrow G \rightarrow H\}$  という遷移列はどちらの製品にも存在しない。このように各 OP を単純に統合した場合、実際には存在しない遷移列が現れてしまう問題がある。

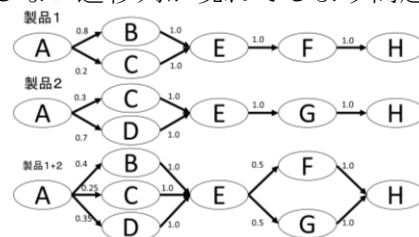


図 1. SPL 製品群の OP 生成時の問題の例

### 3.2. 関連研究の課題

関連研究では、上記の問題を SPL 製品群の振舞いを表す FTS によって解決していた。したがって、FTS に相当するモデルの存在が前提となる。しかし、全製品の振舞いを示す FTS の構築は高コストであり、その存在を前提とすることは現実的でない場合が多い。

## 4. 提案手法

### 4.1. 目的

本研究の目的を以下に挙げる。

1. 製品使用ログ情報のみから生成可能な SPL 製品群の OP の提案及びその生成手法の提案
2. 上記の OP を活用したテスト手法の提案

### 4.2. モデルの定義

扱うモデルを以下のように定義する。

- 製品使用ログ
  - ▶ 個々の製品に対して一つ定義される。
  - ▶ 遷移前と遷移後の状態が記載される複数回の使用シーケンスの集合からなり、製品の使用で状態遷移が起こる度に蓄積される。
- SPL 製品の OP
  - ▶ SPL 製品に対して一つ定義される。
  - ▶ 状態がノード、遷移がエッジで表される状態遷移系で、エッジに遷移の生起確率が付与される。
  - ▶ 製品の重みを任意に付与できる。

同じ SPL の製品でも、製品ごとに重要度が異なる場合がある。それを SPL 製品群の OP に反映さ

An usage profile generation method based on variability mining from products usage log  
KOICHI SATO<sup>†</sup> KISHI TOMOJI<sup>†</sup>  
<sup>†</sup>Department of Industrial and Management Systems Engineering CSE Graduate School, Waseda University

せるために製品ごとに重みを与える。重みは SPL 製品群の OP の遷移の生起確率計算に利用する。

- SPL 製品群の OP
  - ▶SPL に対して一つ定義される。
  - ▶状態がノード、遷移がエッジで表される状態遷移系で、エッジに遷移の生起確率と現れる製品名の集合が付与される。

既存の SPL 製品群の OP[1][2]とはフィーチャに対するブール式ではなく製品名集合でラベル付けしている点が異なる。遷移を製品名集合でラベル付けすることで、後述の方法により対応するモデルがなくても遷移列の除去が可能になる。

### 4.3.全体像

提案手法の流れは以下のとおりである。

- Step1-1.各 SPL 製品に製品使用ログを蓄積
- Step1-2.製品使用ログから遷移生起確率を算出
- Step1-3.算出結果から各 SPL 製品の OP 生成
- Step2.生成した各 SPL 製品の OP を統合し、SPL 製品群の OP を生成
- Step3-1.SPL 製品群の OP を利用し、遷移列の抽出・削減

### Step3-2.遷移列を利用し、製品優先順位付け

Step1 を各 SPL 製品の OP 生成部分、Step2 を SPL 製品群の OP を生成部分、Step3 を製品優先順位付け部分と呼称する。Step1 と 2 が 4.1 節の 1、Step3 が 4.2 節の 2 に対応する。

### 4.4.各 SPL 製品の OP 生成部分(Step1)

各製品使用ログから SPL 製品の OP を生成する。遷移確率は、その遷移が行われた回数をその遷移の遷移前の状態から出る遷移の合計で除算して計算する。この方法で各 SPL 製品の全遷移に確率を付与し、各 SPL 製品の OP を生成する。

### 4.5.SPL 製品群の OP 生成部分(Step2)

各 SPL 製品の OP を統合し、SPL 製品群の OP を生成する。複数の SPL 製品の OP に共通した遷移がある場合、遷移確率に重みを乗算したものの合計から遷移前の状態がある製品の重み合計で除算したものをその遷移確率とする。共通した遷移がない場合は、遷移確率に重みを乗算したものをその遷移前の状態がある製品の重み合計で除算したものを遷移確率とする。全遷移は、その遷移が存在する SPL 製品名でラベル付けされる。

### 4.6.製品優先順位付け部分(Step3)

遷移列抽出では、任意の始点の状態と遷移列の長さを指定し、その条件から SPL 製品群の OP からグラフを定義し、グラフ走査によって導出可能な遷移列を全て導出する。その後、ラベル情報から実際に存在しない遷移列を取り除き、実行可能な遷移列集合を得る。実行可能な遷移

列には、その遷移列を実行できる製品名の集合を付与する。実行可能な遷移列の生起確率の合計を各 SPL 製品のスコアとし、スコアが高い製品を選択する。選択された製品で実行可能な遷移列を取り除き、スコア付けと選択を繰り返す。全ての実行可能な遷移列に対応する製品が選択されるまでこれを繰り返す。

## 5.評価実験

提案手法によって SPL 製品群の OP を生成し、製品優先順位付けができることを確認するため、自作した Web アプリケーション 3 つ(製品 0、製品 1、製品 2)を対象に実験を行う。

### 5.1.OP 生成(Step1,Step2)

生成された SPL 製品群の OP を図 2 に示す。SPL 製品群の OP が生成できることを確認した。

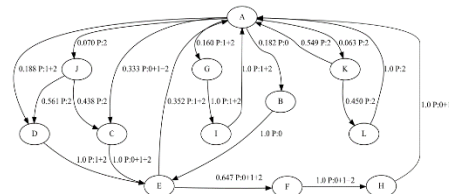


図 2.SPL 製品群の OP

### 5.2.製品優先順位付け(Step3)

導出された全ての遷移列数は 45 で、削減後の遷移列数は 34 となった。遷移列の抽出・削減ができることを確認した。実行可能な遷移列から優先順位付けを行った結果、2→0 が選択できた。製品選択時の累積スコアを優先順位付け時のスコアとランダム選択時の平均スコアの間で比較したグラフを図 3 に示す。優先順位付けの方が早期に高いスコアを得ることができた。

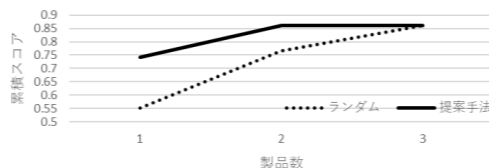


図 3.優先順位付け結果

## 6.結論

本研究では、製品使用ログから SPL 製品群の OP を生成し、それを用いた代表製品の選択手法について提案した。今後、実例に対応する等して、手法を洗練したい。

### 参考文献

[1] X. Devroey, et al, "Coverage Criteria for Behavioural Testing of Software Product Lines," ISoLA 2014, pp.336-350, 2014.  
 [2]X. Devroey, et al, "Statistical prioritization for software product line testing: an experience report,"Software System Modeling, pp.153-171, 2017.