

Webアプリケーション用テストデータの生成 および優先度順位付けに関する研究

齋藤 響*, 高田 眞吾*, 丹野 治門†, 生沼 守英†

* 慶應義塾大学

† NTT ソフトウェアイノベーションセンタ

1 背景

膨大な数の Web アプリケーションが存在し、また作られ続けている昨今において、それらに対して行うテストの効率化はとても重要な課題である。

過去のテスト情報を用いてテストの効率化を図る研究は、Kim らの研究 [1] をはじめ様々なものが存在する。しかし、このような研究は同じソフトウェアに関する過去のテスト情報を用いており、異なるソフトウェア間でその情報を活かしていないものが多い。

テストデータの自動生成に関しては Spawner[2] などのツールが存在する。しかし、このような既存のツールではデータをランダム生成するため、「存在しない住所」のような意味を持った値を生成するのは難しい。

本研究では、他のソフトウェアの過去のテスト情報を活かしながらテストデータを生成する機構を提案する。

2 提案

本研究では、Lacanieta らの研究 [3] を拡張することにより Web アプリケーション用のテストデータの生成および優先度順位付けの機構を提案する。

Lacanieta らの研究は、知識ベースとして過去のテスト情報が蓄積されたデータベースを用意し、その情報を用いて関連シナリオと呼ばれるテスト用のシナリオを生成した。

本研究の提案機構の概要を図 1 に示す。

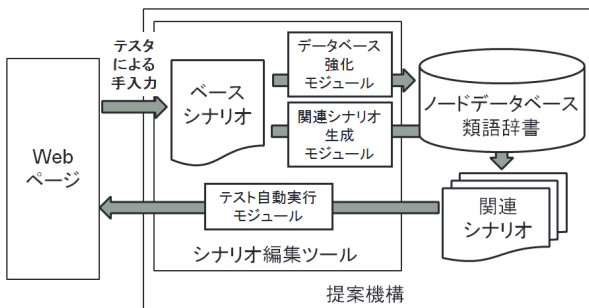


図 1: 提案機構

以下では提案機構の詳細について述べる。

Research on Generation and Prioritization of Test Data for Web Applications

Hibiki SAITO*, Shingo TAKADA*, Haruto TANNO†, Morihide Oinuma†

*Keio University, †NTT Software Innovation Center

2.1 シナリオ編集ツール

本研究におけるテストシナリオとは、図 2 のような Web ページ上での処理の流れを表すものである。これはベースシナリオと関連シナリオの 2 つに大別できる。

ベースシナリオとは、ページの処理が正しく行われるシナリオのことを指す。図 2 の左側はベースシナリオの例であり、これはユーザ ID とパスワードが正しく入力されログインが正しく行われるシナリオである。

これに対し関連シナリオとは、ベースシナリオの中の 1 つのノードを異常なものとしたシナリオのことを指す。すなわち、異常系のシナリオである。図 2 の右側は関連シナリオの例であり、これはユーザ ID に対応しないパスワードが入力されログインが正しく行われないシナリオである。

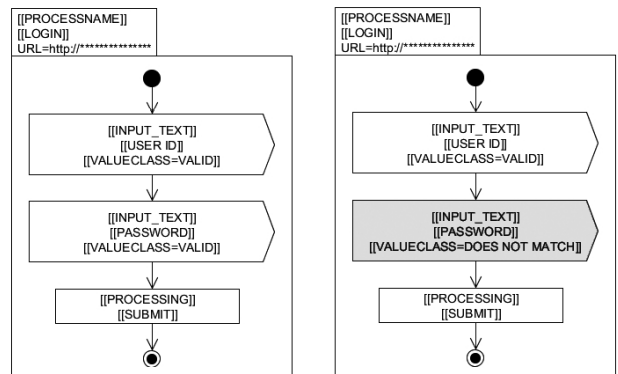


図 2: ベースシナリオ（左）と関連シナリオ（右）

提案機構ではオープンソースアプリケーションの UMLet[4] を拡張することにより実装したシナリオ編集ツールを用いてテストシナリオを編集する。テストは図 2 のように、Process Name (ログインなどの処理の名前) を含むシナリオの外枠を作り、その中でノードを組み合わせることでシナリオを作成する。シナリオを構成するノードとしてテキスト、ラジオボタン、セレクトボタン、チェックボックスの各入力ノードや、送信などの操作を表す「PROCESSING」ノードなどが用意されている。

ノードにはプロパティがあり、ここに Attribute Name (属性名) などの情報を記述する。入力ノードに関しては、ラジオボタンの場合どの選択肢をクリックするか、テキストの場合どのような具体値を入力するかなどの情報も記述する。また、テキスト入力ノードに関して

は文字数制約や文字種制約を記述できる。

2.2 知識ベース

提案機構では知識ベースとして、ノードデータベースと類語辞書の2つを用意する。なお、それぞれのデータベースはMySQLで実装した。

ノードデータベースは過去のテスト情報を蓄積するためのものである。ここでは、関連シナリオにおける異常なノードの情報が蓄積されていると言い換えることができる。データベースには各ノードの情報として、ID, Process Name, Node Type, Input Type, Attribute Name, Value Class, Concrete Value, Frequencyの8つの情報が含まれる。

類語辞書はAttribute Nameの値として使われている各単語の類義語を保存した辞書である。データベースの各レコードには、ID, Process Name, Attribute Name, Synonymsの4つの情報が含まれる。

提案機構のデータベース強化モジュールでは、関連シナリオを用いてノードデータベースを更新することができる。シナリオ内の異常なノードについて、ノードデータベースに同様のノードが存在する場合そのFrequencyをインクリメントする。このFrequencyは優先度順位付けに用いられ、過去に多くの登録があるほど優先度を高くしている。ノードデータベースに同様のノードが存在しない場合そのノード情報を追加する。

2.3 関連シナリオ生成モジュール

本モジュールでは、ベースシナリオとノードデータベースの情報から関連シナリオを自動生成する。ベースシナリオ内に存在する各テキスト入力ノードには、以下の2つの方法で関連シナリオを生成する。

2.3.1 データベースを参照する方法

まず、類語辞書を参考にしてノードのAttribute Nameの類語を決定する。次に、ノードデータベース内にProcess NameとAttribute Name（またはその類語）が一致するノードがあるか検索する。もし存在する場合はそれを入れ替えノードとし、置き換えることで関連シナリオを生成する。例えば図2の左側のベースシナリオに対し、ノードデータベースにProcess Nameが「LOGIN」、Attribute Nameが「PASSWORD」、Value Classが「DOES NOT MATCH」というノードが存在する場合、図2の右側の関連シナリオが生成される。

具体値に関しては、入れ替えノードのConcrete Valueにある値、元ノードの具体値を一部変更した値、手入力のいずれかを選択する。

また、入れ替えノードのFrequencyを利用してシナリオのスコア付けを以下の式(1)に従って行う。

$$Score = Frequency * \alpha \quad (1)$$

ここで α の値は、入れ替えノードのAttribute Nameが類語の場合は $\alpha=75$ 、そうでないものは $\alpha=100$ とする。

スコアの大きいものが優先度の高いシナリオということになる。

2.3.2 制約を参照する方法

制約情報が記述されているノードに関しては、記述されている制約それぞれに対してそれに反する関連シナリオを生成する。例えば「6文字以上」という制約の存在するノードの場合、そのノードを「5文字以下」のものに置き換えた関連シナリオを生成する。

具体値に関しては、自動生成データか手入力かのいずれかを選択する。自動生成データに関しては、文字数制約と文字種制約を満たすようにランダム生成される。ただし、制約に反する部分はそちらが優先される。先述の例の場合、「6文字以上」よりも、「5文字以下」が優先される。

スコア付けに関しては、ノードデータベースに同様のノードが存在する場合は式(1)に従い、存在しない場合は0とする。

2.4 テスト自動実行モジュール

提案機構には、テスト自動実行ツールであるSelenium WebDriver[5]を利用した関連シナリオの自動実行機構も組み込む。生成された関連シナリオそれぞれに対し、ブラウザの起動、具体値の入力やクリック、ブラウザの終了の操作が自動的に行われる。

3 結論

本研究では、知識ベースを用いた関連シナリオおよび具体値生成と、その優先度順位付けの手法を提案した。また、提案機構を実装し、関連シナリオや具体値の出力、シナリオの自動実行までを行うことができた。

今後の課題としては、ノード間の関係性を考慮した関連シナリオ生成や、テストの評価までの自動化などが挙げられる。

参考文献

- [1] J. Kim, A. Porter, "A history-based test prioritization technique for regression testing in resource constrained environments", Proc. of ICSE 2002, pp.119-129 (2002).
- [2] Spawner, <http://sourceforge.net/projects/spawner/>
- [3] R. Lacanienta, et al, "A Knowledge-based Approach for Generating Test Scenarios for Web Applications", Proc. of SEKE 2013, pp.166-171 (2013).
- [4] UMLet, <http://www.umlet.com/>
- [5] Selenium, <http://www.seleniumhq.org/>