Web 行動リプレイシステムに基づく Web アプリケーション動作検証システムの試作

柿元 宏晃 † 中村 正人 † 大囿 忠親 † 新谷 虎松 †
名古屋工業大学大学院工学研究科情報工学専攻 †

1 はじめに

本稿では，Web アプリケーションの自動的なテストを支援するために，検証のために必要な行動列を取得し，自動テストに必要なテストケースを生成するシステムについて述べる．

現在，Web アプリケーションの動作検証は，テストケースを人手で試みる手動テスト，またはテストツールによってテストケースを自動的に行う自動テストによって行われている．自動テストでは，繰り返し行うテストケースを命令セットとしてテストツールに与える．与える命令セットは「loadUrl('X');(ページ X を読み込む)」や「clickButton('Y');(ボタン Y を押す)」など，マウスの軌跡のような細粒度の命令ではなく，比較的粒度の高い命令から構成される．自動テストのためには，テストケースに沿った命令セットを適切に記述する必要がある．複雑な Web アプリケーションでは，命令セットを記述すること手間がかかる点が課題である．

本研究では，ユーザがテストケースを記述するのではなく，システムがユーザの行動を記録して自動テストに用いる命令セットを生成するシステムを開発している．これにより，ユーザがテストケースを記述する手間が省け，効率的なテストが可能になる．実現のためには，ユーザの Web ブラウジング時の行動を記録，解析，そして再現することが必要である．本稿では，テストセットの自動生成が可能なテストを明らかにし，その実現方法を示す．

2 Web 行動リプレイシステム

我々はこれまで，Web ブラウジング時のユーザの行動を解析し，再現を行うことが可能なシステムの開発を行ってきた．このようなシステムを Web 行動リプレイシステムと呼ぶ．本研究では，Web 行動リプレイシステムを Web プロキシとして実装しているため，ユーザの行動によって発生したイベントを JavaScript のイベントハンドラで取得することが可能である．取得したイベントをイベントの発生時間を考慮し，羅列することで，ユーザが Web ブラウジング時にとった行動を再現することができる．

2.1 Web ブラウザ上での行動記録と再現

Web ブラウザの実装や規格上の都合により，表 1 に示すような行動の記録や再現には制限がある．ファイルのアップロードは，type 属性が file である input タグの value 属性を監視することで観測できる．また，Web プロキシとして実装している Web 行動リプレイシステムでは，アップロードされた時間とそのファイルまで取得することが可能である．しかし，アップロードを再現することはセキュリティ面での制限により不可能である．Web ブラウザの HTML 描画領域外で行われる操作は，プラグインや，OS レベルでの行動追跡によってしか取得・再現を行うことはできない．Web ブラウザのプラグインを利用しようとすると，クロスブラウザでの

表 1: 再現が困難な行動

再現が困難（または不可能）な行動	行動の取得
ファイルのアップロード	可能
HTML 描画領域外で行われる操作	困難
ウィンドウサイズの変更	ブラウザ依存

Web リプレイシステムは実装できず，また，OS レベルでの行動追跡では HTML の構文解析を行うことができないという制約をそれぞれ持つ．そのため，汎用性の高い Web 行動リプレイシステムを実装しようとする場合，描画領域内に限定して操作を取得するのが妥当であると考えられる．ウィンドウサイズの変更も，HTML 描画領域外で行われる操作ではあるが，JavaScript によってウィンドウサイズを取得することができるため，行動の取得は可能である．しかし，取得できるウィンドウサイズは，Web ブラウザごとに仕様が異なっているため，取得したサイズを正確に再現できる保証がない．

本稿で述べる手法では命令セットを生成するために，Web アプリケーションのテストケースを 1 度人手で試み，発生したイベントを Web 行動リプレイシステムによって記録する．Web 行動リプレイシステムでは再現することが不可能な行動（ファイルのアップロードや描画領域外での操作）の取得は行わない．ウィンドウサイズは Web ブラウザごとに仕様が異なり，ずれが生じる可能性があるが，オブジェクトの位置に大きく影響を与える要素であるため，可能な限りの精度で取得する．本手法では，後に述べるように，オブジェクトの相対位置を基準にした行動の記録を行い，ウィンドウサイズのずれの影響を軽減しているため，高い精度でサイズを取得する必要はあまりない．その他の JavaScript で取得可能なイベントは可能な限り取得する．

3 行動記録からのテストケース生成

本手法のように，Web 行動リプレイシステムに基づいて，行動の記録・再現を行うシステムの一つに，デスクトップアプリケーションのロギングツールによってイベントの発生した位置をオペレーションシステムから取得し，記録・再現を行っているものがある．Web 上では上位のオブジェクトが下位のオブジェクトを押し出すという表示規則があるため，サイズが表示ごとに不定のオブジェクト（広告など）によって下位のオブジェクトの位置も不定になることがある．取得したイベント発生位置を絶対位置として取得している場合では，イベントの再現時にオブジェクトの位置とイベント発生位置のずれが生じ，正確に再現することができない場合がある．本手法では，Web 上で動作するシステムによってイベントの記録を行うことで，Web 上の各オブジェクトからの相対的な位置関係や，フォーカスの当たっているオブジェクトの情報を取得することができ，Web 特有の表示のずれへの対応が可能となる．

3.1 細粒度の行動記録を用いた行動分析

本手法と同様に，Web 上のシステムによってイベントの記録を行っている既存のシステムはいくつか存在する．これらのシステムではリンクやボタンなど，粒度の高い行動のみに絞ってイベントのリスニングを行い，それらに発生したイベント列を命令セットとして記録している．Web 技術が

Performance Testing of Web Applications based on Web Usage Mining

Hiroaki KAKIMOTO, Masato NAKAMURA, Tadachika OZONO, and Toramatsu SHINTANI
Department of Computer Science and Engineering Graduate School of Engineering Nagoya Institute of Technology, Gokiso, Showa-ku, Nagoya, 466-8555 JAPAN

発展した現在の Web アプリケーションでは，マウスカーソルの軌跡，mousedown，mouseup に連動するアクション（オブジェクトのドラッグ&ドロップや軌跡による描画など）や mouseover に連動するアクション（プルダウンメニューなど）が使われているものがある．既存のシステムのように，リンクやボタンのみのイベントリスニングだけでは，これら複雑なアクションを記録することはできない．本手法では，テストケースの実施よって発生したマウスイベントを全て取得し，後に相対位置での再現が可能な形式に記録を行っているため，複雑なアクションも柔軟に再現することが可能である．

ユーザの操作によってなんらかのイベントが発生した場合，本手法ではまず，イベント発生時間，イベントタイプの特定とイベントの発生したオブジェクトを特定を行う．イベントが発生した座標と，その直下のオブジェクトの左上座標の差を求める．この座標の差は，イベントの発生したオブジェクトを基準にしたイベントの相対位置を示す．イベントの相対位置の算出と同時に，イベントの発生したオブジェクトのパスを取得する．オブジェクトのパスをこのようにイベント発生時に取得することで，JavaScript で Dom の構造が変化するような場合にも柔軟に対応することが可能になる．

3.2 テストケースの生成

本手法では，相対的な位置関係でイベントを発生させるための命令セットを，主に「イベント発生時間」「イベントタイプ」「イベントが発生した座標と，その直下のオブジェクトの左上座標の差」「イベントが発生したオブジェクトのパス」から生成する．イベントを発生させる位置は命令セットの実行時にリアルタイムに決定する．前述のように，命令セットを実行する環境ごとにウィンドウサイズ，オブジェクトの表示位置に誤差が生じるため，実行時にリアルタイムに計算を行うことで，正確なイベント発生位置を決定することができる．そのため，生成する命令セットは，時系列に並んだ命令群だけでなく，命令セットの実行時に相対位置から絶対位置を計算する関数群を含んだものとなる．図 1 は，命令群を時系列に並べた命令セットの一部であり，ドラッグ&ドロップを行った場合のものである．MArep.mousemove, MArep.mouseover, MArep.mousedown, MArep.mouseup はそれぞれ，mousemove, mouseover, mousedown, mouseup イベントを発生させる関数である．MAtrace.obj はイベントの発生した位置直下のオブジェクトのパスを，イベント発生順に格納している．MAtrace.pos はイベントの発生位置を，その直下のオブジェクトからの相対位置として格納している．MArep.time はイベント間の遅延を発生させる関数である．MAtrace.time はイベント間の時間を格納した変数であり，MArep.time への入力として与えられる．

イベントを発生させるべき絶対位置は，イベントが発生したオブジェクトの絶対座標（MAtrace.obj から計算）とイベントの相対座標（MAtrace.pos）の和によって計算できる．この計算を時系列に沿ってイベントごとに行うことで，テストケースを相対座標の位置において再現が可能である．

3.3 負荷テスト機構

本手法によって生成された命令セットは，Web アプリケーションのリグレッションテストのテストケースとしてだけでなく，負荷テストにも応用ができる．Web アプリケーションにおける負荷テストで最も重要ものの 1 つとして，同時アクセス，同時作業に対する耐久性評価が挙げられる．特に不特定多数が利用するような Web アプリケーションであれば，同時アクセスは必至であり，不可欠なテストと言える．本手法によって生成された命令セットを複数のマシン，または仮想的に複数用意したブラウザ上で実行することで，同時アクセス，同時作業を行うテストセットとして利用可能である．我々は以前に，Web ページ上へリアルタイムに一斉配信を行うシステムを実装した [2]．これは，番組表と呼ばれるコンテンツ配信時間管理システムにより，Web ページ上のコンテンツやスクリプトを任意の時間帯に，指定した Web ページ上へ配信を行うことが可能になる技術である．この配信技術によって，本稿で述べた手法による命令セットを同時刻に一斉配信することで，仮想的に複数セッションによる同時作業のテストを行うことが可能である．

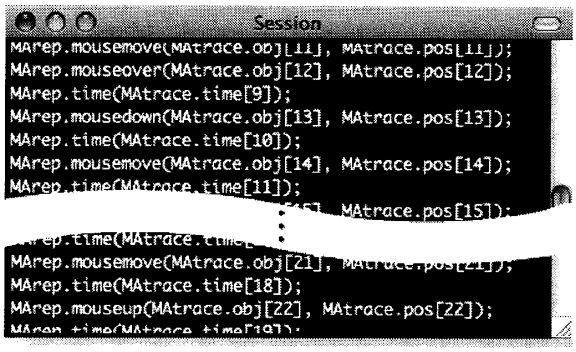

図 1: 命令セットの一部

4 評価

本システムと既存の動作検証システム Selenium IDE[1] において，Gmail での操作を行った場合の比較を行った．その結果を表 2 に示す．ファイルの添付は 2.1 節で述べたように両システムとも再現は不可能であった．本システムと Selenium IDE で大きく違う点は，用いている行動記録の粒度の差である．本システムでは，ボタンだけでなく，対象に関わらずマウスクリックイベントも取得しているため，ボタン以外で，マウスクリックイベントをリスニングして発生するような機能（受信メールの選択など）でも再現が可能であった．Gmail の本文入力欄はテキストエリアではなく，div と iframe の複合によって形成されている．このように複雑な構造をしているアプリケーションでも，細粒度の行動記録を用いることで，行動の追跡が可能になっていることが確認できた．

表 2: Gmail での操作の比較

操作	本システム	Selenium IDE
受信メールの選択	可能	不可能
メールの新規作成	可能	可能
メール本文の入力	可能	不可能
ファイルの添付	不可能	不可能
メールの送信	可能	可能

5 おわりに

本稿では，Web ブラウジング時のユーザの行動の解析および再現を行うシステム，Web 行動リプレイシステムに基づき，Web アプリケーション上で人手によって実施されたテストケースの記録から，自動テストに用いる命令セットを半自動的に生成する手法について述べた．本手法では，発生したイベントを相対座標で扱うことで，ウィンドウサイズが正確に取得できない場合や，広告などによって表示位置が変動してしまうような場合でもテストケースの再現を正確に行うことができる．

参考文献

[1] 向井康人，大囿忠親，伊藤孝行，新谷虎松，“Web における情報配信の最適化のためのユーザ行動の分析手法の提案”，第 68 回情報処理学会全国大会論文集，Mar，2006.

[2] 西健太郎，大囿忠親，伊藤孝行，新谷虎松，“Web エージェント MiSpider に基づく広告制御機構の実装”，第 19 回人工知能学会全国大会論文集，June，2005.

[1] https://addons.mozilla.org/ja/firefox/addon/2079