

操作履歴と DOM 構造に基づく Web 行動分節化システム

柿元 宏晃[†] 伊藤 太樹[†] 佐野 博之[†] 平田 紀史[†]

白松 俊[†] 大園 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

アプリケーションのオートメーションは、アプリケーションの操作をプログラムによって自動的に行うことであり、頻繁に行う作業や定期的に行う一連の作業を自動化することで作業の効率化を図るために行われる。また、アプリケーションの操作を自動で行い、デモとしてプレゼンテーションに利用するシーンがある。

オートメーションは、人手でプログラムを記述して行う方法と、ユーザの操作を一度記録し (以下、キャプチャ) 操作履歴からプログラムを自動生成し、再実行する (以下、リプレイ) 方法 (以下、キャプチャリプレイ型) の 2 通りがある。近年ではアプリケーションの動作が複雑になってきており、オートメーションを行うプログラムを人手で記述することはコストがかかるため、キャプチャリプレイ型のプログラム生成が主流になりつつある。

アプリケーションのオートメーションでは、リプレイを行う PC の環境により操作の再生タイミングが異なり、エラーが生じてしまう点が課題である。これは、アプリケーションの動作が操作の再生に追いついていないことが原因であり、既存のオートメーションシステムでは操作の再生タイミングを経験則に基づいて調整することで解決が試みられている。また、アプリケーションのオートメーションは、デスクトップアプリケーションを対象としたものと、Web アプリケーションを対象としたものがある。Web アプリケーション上でのキャプチャリプレイではセッション情報のキャプチャリプレイを行う研究もある [2] が、本稿では Web アプリケーション上でのマウスカーソル移動やキーボード入力といった操作 (以下、Web 行動) におけるキャプチャリプレイを対象とする。Web アプリケーションではデスクトップアプリケーションと異なり、フォントサイズやウィンドウサイズなどの表示環境がユーザによって様々であり自由度が高いため、Web アプリケーション上でキャプチャを行う際の表示環境とリプレイを行う際の表示環境が異なり、Web 行動のリプレイを正確に行うことができない点が課題である。課題のため、Web アプリケーションを対象とした既存のオートメーションシステムでは座標を基準とした、Web 行動のリプレイを行うことができないため、再現可能な Web 行動が限定されている。

本稿では、DOM 構造に変化がみられた点で Web 行動の分節化を行い待機処理を付加することで、アプリケーションの動作とリプレイのタイミングを調整し、閲覧環境に低依存なキャプチャリプレイを可能にする。また、相対座標による操作の発生座標の表現によって、さまざまな表示環境において座標を基準としたオートメーションの実行を可能にする。

2 Web 行動キャプチャ

キャプチャリプレイ型のオートメーションでは、リプレイ時に実行する命令群 (以下、オペレーションセット) の生成

Web Usage Segmentation System based on the DOM Structure and Usage Logs in a Web Page

Hiroaki KAKIMOTO, Taiki ITO, Hiroyuki SANO Norifumi HIRATA, Shun SIRAMATSU, 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

を行うため、Web アプリケーション上での Web 行動を 1 度人手で試み、発生した JavaScript イベント (以下、イベント) を Web 上のシステムによって記録する [1]。Web 上ではページ上部に配置された要素が下部に配置された要素を押し出すという表示規則があるため、フォントサイズの変化によって下部の要素座標が変動する課題がある。また、ウィンドウサイズの変化によっても同様であり、取得したイベント発生座標を絶対座標として取得している場合は、リプレイ時に前述の課題により要素の座標にずれが生じ、正確に再現することができない場合がある。このような課題のため、座標情報が必要なイベントのリプレイを行うことが困難であるが、Web 技術が発展した近年の 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) が使われているため、座標情報を考慮することはキャプチャリプレイにおいて重要な課題であるといえる。本手法では、Web 上のシステムによってイベントの記録を行い、イベントの発生座標をイベントの発生要素の XPath と要素からの相対座標によって表現することで要素座標の変動に頑健なリプレイを行うことが可能になる。

イベントが発生した場合 Web ページの左上を基準とした座標としてイベントの発生位置を取得する。本研究ではこの座標をイベント絶対座標と呼ぶ。イベント絶対座標をイベントが発生した座標直下のオブジェクトを基準として表現する (図 1)。イベントの絶対座標直下の要素をイベント直下要素、イベント直下要素の左上座標とイベントの絶対座標との差をイベント相対座標と呼ぶ。イベント絶対座標はイベント直下要素の左上座標とイベント相対座標の和で表すことができる。本手法ではイベント直下要素の位置を座標の代わりに XPath で表現する。この表現により、再現時に可変長のオブジェクトによってイベント直下要素の位置が変動した場合でも、XPath に変更がなければ、取得時と同様に再現することが可能になる。

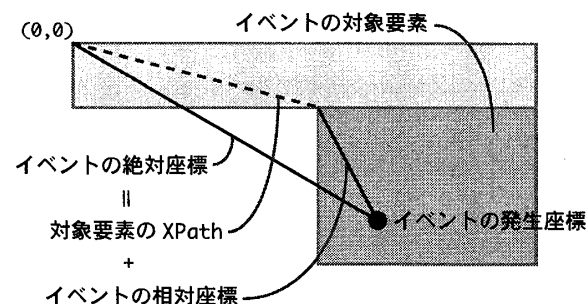


図 1: イベント発生座標の相対座標表現

3 Web 行動の分節化

キャプチャ機構ではイベントの取得に加え DOM 構造の情報も取得する。取得はイベントの発生ごとに行い、DOM 構造に変化が見られた場合のみ DOM 構造情報として記録する。

分節化は DOM 構造の変化が発生したイベントを基準に行う。イベントが発生した時点で以前の状態と DOM 構造が

変化している場合、そのイベントがトリガとなって DOM 構造に変化が発生した可能性が高い。また、DOM 構造の変化が発生する前の他のイベントがトリガとなっている可能性もある。図 2 にイベントの発生と DOM 構造情報の取得タイミングについて示す。イベント A によって要素 X が追加され DOM 構造に変化が起きる。キャプチャ機構はイベント A を取得した時点で DOM 構造に変化は起きていないので、DOM 構造の情報は記録せず、イベント A の記録だけ行う。イベント B が発生したときはじめて、要素 X によって DOM 構造に変化が起きているという情報が得られるため、イベント B の発生時にはイベント B の記録と DOM 構造情報の記録を行う。イベント C は要素 X を参照する。図の例において待機処理が必要になるのは、イベント B とイベント C の間である。イベント C は要素 X を参照しているため、イベント C が発生する前までに要素 X の追加が完了している必要がある。また、イベント C が要素 X を参照していない場合でも、イベント C が間接的に要素 X を参照する可能性も考えられるため、イベント C の発生前までは要素 X の追加が行われている必要がある。Web 行動の分節化システムでは、DOM 構造の変化が確認された点を Web 行動の節とし分節化を行う。

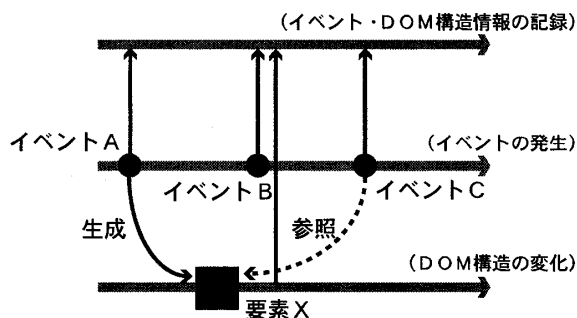


図 2: イベント発生と DOM 構造情報の取得

4 Web 行動リプレイ

Web ブラウザの実装や規格上の都合により Web 行動はリプレイに制限がある。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キャプチャすることができる。さらに、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、ファイルをアップロードする行動を再現することはセキュリティ面の制限により不可能である。また、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 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。

本手法では相対座標によってイベントを発生させるために、オペレーションセットの引数としてイベント発生時間、イベントタイプ、イベントが発生した要素の XPath、相対座標を定義する。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 (オブジェクトのパスから計算) とイベントの相対座標 (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) の和である。イベントの発生座標はオペレーションの実行ごとに XPath と相対座標から計算を行う。事前に計算を行ってしまうと、要

素の追加や削除によって座標が変動する可能性があるためである。

5 Web 行動キャプチャリプレイ機構の応用

Web 行動分割を用いた Web 行動キャプチャリプレイ機構の応用として、Web 行動の記録を遠隔地に送信し、ムービーのように実行する Web 行動デモシステムを試作した。図 3 に Web 行動デモシステムの構成を示す。Web ページ上での操作を記録し、本稿で述べた Web 行動キャプチャリプレイ機構がオペレーションセットを生成する。オペレーションセットを Web ページごとのブックマークレットとして分割し、それぞれをメールに添付して遠隔地の相手に送信する。遠隔地の相手は、添付されたオペレーションセットを Web 行動デモシステムにアップロードすることでリプレイを見ることが可能である。これは Web ページの操作のムービーを遠隔地の相手に送信する場面に利用可能である。オペレーションセットはムービーより軽量のファイルとなるため、メールへの添付に適している。

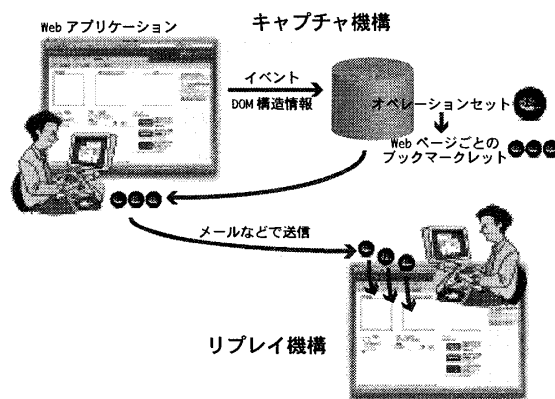


図 3: Web 行動デモシステム

6 おわりに

本稿では、キャプチャを行う環境とリプレイを行う環境の違いによってアプリケーションの動作とオペレーションの実行のずれが発生する課題に対して、DOM 構造の監視というアプローチによって解決を行った。DOM 構造に変化がみられた点で Web 行動の分節化を行い待機処理を付加することで、アプリケーションの動作とリプレイのタイミングを調整し、閲覧環境に低依存なキャプチャリプレイを可能にした。

また、フォントサイズやウィンドウサイズなど、多様な表示環境によって座標情報を伴うイベントのキャプチャリプレイが困難であるという課題に対し、イベントの発生座標を相対座標と XPath によって表現することで解決を行った。これによりこれまで扱うことができなかった、座標情報を伴うイベントのリプレイが可能になり、複雑なインターフェースを持つ Web アプリケーションでの Web 行動キャプチャリプレイを可能にした。

参考文献

- [1] 柿元宏晃, 中村正人, 大園忠親, 新谷虎松: “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”, 第 71 回情報処理学会全国大会講演論文集, vol.71, no.1, pp.517-518, (Mar. 2009)
- [2] I. J. Nino, B. de la Ossa, J. A. Gil, J. Sahuquillo, A. Pont: “CARENA: A tool to capture and replay web navigation sessions”, in Proceedings of the Third IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON '05), May 2005.