Abstract
(論 文 の 内 容 の 要 旨)

論文題目　Interaction-Based Preventive Maintenance of Ajax Web Applications
（相互作用に着目したAjax Webアプリケーションの予防保守）
氏　　名　前澤　悠太

The challenge of maintaining Web application is preventing actual errors in a user environment. Today, users demand rich user experience of applications; hence, client-side asynchronous JavaScript and XML (Ajax) technologies are increasingly important. Developers build modern Web applications using Ajax technologies (Ajax Web applications) which can handle user events and asynchronously retrieve update data from servers. Thus, Ajax technologies make Web applications responsive, enhancing user experience. However, Ajax event-driven and asynchronous features make the contexts of running applications unpredictable. Despite concerted efforts by developers, the unpredictable contexts might conceal faults in development and testing environments but they will be exposed in a user environment.

Many researches have been conducted on state-based analysis and testing for finding faults in Ajax Web applications. States of the applications correspond to Document Object Model (DOM) representing Web pages. Since the applications dynamically manipulate DOMs by handling user events or server responses, some have succeeded in leveraging dynamic analysis techniques for extracting a finite state

machine (FSM) based on DOM instances captured at runtime. Although existing DOM-based testing techniques effectively and efficiently detect executable faults, they do not help finding “potential faults” that are not easily produced in a testing environment but will be exposed in a user environment.

We first propose a static method for extracting an FSM from Ajax Web applications. Since DOMs cannot be determined in a static manner because they are dynamically manipulated, we focus on interactions with Ajax Web applications (e.g., mouse clicks and server responses) which act as triggers to change the application states. These interactions can be statically distinguished at event handlers in the source code. Hence, our proposed method extracts an interaction-based FSM representing all possible application behaviors. Developers can use the extracted FSM to find the potential faults; however, the cost may not be negligible to manually and carefully determine the correctness of the extracted FSM, which does not enable developers to exhaustively find faults in the applications.

Secondly, we propose a method for automatically verifying the correctness of the extracted FSM. Model checking techniques are useful for automated verification against given invariants representing correct behaviors; however, there are no generic behavior invariants relevant to the interactions. Therefore, we define interaction invariants from correct and incorrect interaction-based behaviors described in Ajax design patterns, which is a catalog of Ajax Web application development know-how. The model checker leverages the interaction invariants to identify faulty interaction sequences in the extracted FSM. Although the identified faulty interaction sequences should contain suspected faults, they might be spurious counterexamples in the extracted FSM. Therefore, executable evidences that the suspected faults cause errors in Ajax Web applications are required.

We thirdly propose a method for validating the applications by revealing actual errors due to the potential faults. Although testing all possible scenarios in every environment should reveal these errors, such a method would be unrealistic. Under an assumption that unexpected network latency may make potential faults executable, we implement mutation operators to manipulate the timing of the applications handling server interactions. Our mutation operators are able to produce subtle network delays, which can reveal errors due to delay-dependent potential faults.

From results of our case studies using real-world Ajax Web applications, our proposed methods could reveal errors relevant to vulnerabilities in Web

applications, which existing analysis and testing techniques could not detect. Therefore, we conclude that our proposed methods can help developers find and debug potential faults, i.e., preventive maintenance of Ajax Web applications.

見本 Sample

論 文 の 内 容 の 要 旨

論文題目　　□□□□□□□□□□□□□□□□□□□□□□□□□□□
（□□□□□□□□□□□□□□□□□□□□□□□□□□□）
↑
（※論文目録の記載と同じにしてください。）
（※論文題目が外国語の場合には、和訳を括弧書きで付けてください。）
(* The title typed here must be identical to that shown in the Thesis Table of Contents.)
(* Add a Japanese translation in parentheses if the thesis title is written in a non-Japanese language.)

氏　　名　　○○　○○
↑
（※学位記に記載される氏名と同じにしてください。）
（※漢字圏以外の外国人は、カタカナ表記となります。）
(* Type your name in the same manner as you want to have shown on your degree certificate.)
(* Type your name in katakana if you are a non-Japanese without a kanji name.)

　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

１．「論文の内容の要旨」は、紙媒体を２部提出してください。
また、「PDF ファイル」及び「文書ファイル（Word 等で作成したもの）（省略可。）」の電子データも併せて提出してください。
２．論文博士は日本文で記入してください。（課程博士は英文でもよい。）
横書き、片面刷りとしてください。

３．大きさはＡ４判とし４ページ以内、１０ポイント程度の活字で印刷したものとしてください。
（日本語の場合は 4, 000 字以内（英語の場合は 2, 000 語以内）とする。）
４．第１ページ上部に、タイトルを「論文の内容の要旨」とした上で、論文題目及び氏名を記入し、その下から内容の要旨を記載してください。

1. **Two copies of your thesis summary must be submitted in paper form.**
   Electronic data of the thesis summary must also be submitted: a PDF file is mandatory, while submission of the original document file (MS Word or other) is optional.

2. **If you are obtaining your Doctorate degree by submitting a thesis (as a Ronpaku),** your thesis summary **must be written in Japanese. (**If you are obtaining your degree by completing the course requirements of a Doctorate program, a thesis summary in English is acceptable.)
   The thesis summary is formatted with **horizontal writing and single-sided print**.

3. The thesis summary is to be printed on **A4-size paper** and digested into **four pages or less** using **approximately a 10 point type**.
   (The restriction is **4,000 characters** for a Japanese summary and **2,000 words** for an English summary.)

4. **In the upper part of the first page, the text “論文の内容の要旨” is typed and the title of the thesis and the name of the applicant are typed on subsequent lines. The main text of the thesis summary begins below the above heading section on the same page.**