## <u> Title: 웹 소프트웨어의 신뢰성 18</u>

- ✓ Instructor: Gregg Rothermel
- ✓ Institution: KAIST
- ✓ Dictated: 강단비, 김주현, 김지현, 주다은

[00:00]

As in the paper, JavaScript errors in the wild: an empirical study.

This paper is originally written by Ocariza and presented in ESRE, 2011.

The motivation of this paper is simple.

In modern web applications, 97% of web applications uses JavaScript to make the application dynamic.

So, error of JavaScript is very important, but on state of the art, the errors of JavaScript is not enough studied.

There are some reasons such as JavaScript engine is ?[00:48].

Even though there exist errors in JavaScript, we can not see the errors.

The authors study errors of JavaScript, from error message which is hide on the web application, which can be exposed by some profiling tools by console message.

So the authors focus on the error message but they do not consider consequence of the JavaScript errors.

The contributions of this paper is said following.

The author develop systematic methodology to execute web application with their own based cases.

And second one is they study errors of JavaScript from 50 real web applications.

And they categorize studied errors.

They could make 4 categories which are permission denied, undefined symbol, exception, and syntax error.

The authors gave some implications from the study.

So... uhm...

There are several conditions that make JavaScript error to be exposed such as interaction speed.

And second one is the errors and JavaScript static and dynamic characteristics are correlated.

However, there are some limitation of this paper.

First one is, the author do not study the consequence of the errors.

So they just look error message of JavaScript applications.

So they don't know which error affect how the web application works.

So if there is functionality fault, they do not know.

Even though there is an error, the application may work fine.

So, the author study that aspect not at all.

So, that is one limitation.

Second one is they use just error messages.

So, there might be hidden errors that affect program behaviour, but do not produce error message.

Third one is, the authors target real web application which may not be same during the experiment.

So, as you know, we don't know when the JavaScript in real web application is changed.

So, during the experiment, the target JavaScript code maybe changed or not, but the author cannot know.

So there is one limitation.

And last one is, they include errors caused by external component such as advertisement.

But such advertisement is out of cover of the developers, so these errors should not be considered to this study.

But they could not distinguish them, so they just included it.

Here is overall experimental methodology.

First, they choose target web applications.

So they choose 50 web pages in Alexa Top 10, excluding duplicated pages.

[05:00]

Which means there are several language version of Google, Google Korea, and Google U.S.

So they ignore such duplication.

So they choose 50 web pages, and they made test suite.

Each test suite contain 15 test cases and there are 3 test suites for each application.

They made this test suite manually.

And then they run the test suite using existing tool,

So they learn each test suite with 3 test mode.

In here, test mode means interaction speed.

So they differenciate the difference of each adjacent event, so they use 3 different test mode and for each test mode, they learn 3 times.

So total, they learn each test suite 9 times.

And then they collect and categorize the error message.

So they use 2 tools, selenium for creating and running test cases and firebug for collecting error messages.

And they develop another tool but I think there is, hard to say the tool., they just...

Pathing script, the make pathing script.

So, I don't agree with that is tool, so I do not include this slide.

So I will explain how they execute web application in detail.

A test case represent an interaction with a web application.

So A test case may contain multiple events.

In here, this interaction mean user and web application, interactio between user and web application, which is an event.

So its test case used in the experiment is manually made based on normal interaction.

So the author, do not aims to break the web application behaviours.

So they manually made each test case and each test case contain 2.66 events on average.

So they make test suite with 15 test cases.

Each test suite visits 29 web pages.

(Student speaking) ... they say, their test case are based on normal interactions.

(Student speaking) The minimum normal number of interactions is ambiguous.

Yeah, in paper they say very ambiguously but in my interpretation, if you are targeting Google, then open the webpage and search something, like that.

You don't try to enter search message and then link to another page like that.

If you are targeting Amazon, you search a product and then move to find page, do not try to break the web application own purpose.

So there is normal, meaning of normal.

(Student speaking) That's how I ...

(Professor speaking) So they're not, they're not acting like real, real strong testers trying to, try all the strange things...

Yeah.

(Professor speaking) So it's not robust as testing a wall.

Yeah, they also try to just, realistic behavior of web application, not corner cases.

(Professor speaking) And so, we just have to keep in mind that their whole taxonomy of errors is based on normal interactions and we might get a different taxonomy if they were looking for really robustness in errors.

Yeah so...

(Professor speaking) Strange behaviors.

So [?9:54] of this paper instead, even though the normal interaction, there are many errors.

[10:00]

That is being claim of this paper.

So we don't know how the authors create test case in the paper because there is no mention about it.

They just say it is normal test cases.

(Student speaking) I think maybe they use a [?10:27] to record the interaction between the browser and user, manually, so they present as normal interaction because two ways, two cannot be automated.

(Student speaking) Some weird click actions or...

Maybe they couldn't make weird click manually but yeah, anyway, they didn't.

Anymore question about it?

(Student speaking) The errors are awkward usually in abnormal situations?

Yes.

(Student speaking) So they needed to consider that, I'm still curious about it...they used in normal situations.

I think so because this paper claim there are several errors in JavaScript even though the users use the web application on purpose with normal interaction.

So there is main claim of the main author so I think this approach is, makes sense, approach makes sense.

Is that enough to answer your question?

Then, let's move on.

And then the authors replay the each test suite with three testing mode.

There are three testing mode; one is fast mode, medium mode, and slow mode.

Fast mode is there is no delay between the event, and medium mode, there is 500 millisecond between the events, and slow mode is 1000 millisecond between the events.

So there is reasonable testing mode.

And they're replayed three times for each testing mode.

And then, the authors find error using JavaScript console error message.

So an error has two attribution, attribute; one is a text description of the errors such as null exception at blah-blah.

And second is location of the code that triggers the errors.

Prio Js.s at line 11, something like that.

So the author assumed that if two errors are same, if all the attributes of the errors are same.

I think that is reasonable assumption.

So the authors makes 6 research questions.

First one is are JavaScript errors prevalent in web applications and if so, do these errors share common characteristics?

Second one is does the speed of interaction affect the frequency of JavaScript errors?

Third one is do non-deterministic JavaScript errors occurs?

In here, non-deterministic JavaScript errors means that a error that is occur in some execution but

does not occurs other executions with same test case.

Fourth one is are there any correlations between a web application's static and dynamic characteristics and the number of errors in the web application?

Fifth one is that are there inter-category correlation among the different error categories?

And last one is is the number of errors in a web application affected by the framework used in its construction?

[15:00]

(Student speaking) I have a question, in the static and dynamic characteristics of web applications...

The characteristics just...are... I'm not gonna define characteristic but I give you some example.

For static characteristic, number of domains, the web application use, or the JavaScript code size, number of JavaScript.

And dynamic characteristic, for example, number of function codes...

(Student speaking) During the test case?

Yes.

Is it enough to your question?

(Student speaking) Yes.

Okay.

Were the Javascript ?, number of Javascript,

And dynamic charactoristic for example.

Number of ?..

(student questioning)

Yes.

Is it enough to your question?

So.

Are we explain a result of the paper.

The author's answer is question 1.

The answer is, yes there are any errors in improvement prevalent in web applications and there is common characteristic so they could make for categories.

And the category sees that permission denied, null exception, undefined symbol, syntax error.

So most of errors, almost high most of errors are permission denied errors and undefined symbol errors are next one, and null exception and syntax error.

Permission denied errors occur when Javascript code from one domain attempt to access an object or property in another domain.

So this broken same origin property, so the situation may permission deny errors.

So this error often caused by advertisement.

And second one is null exception errors.

Null exception errors occur, null value is used to access property.

So this error often arises due to missing or mistype DOM elements.

Third one was undefined symbol errors.

This error occurs Javascript code calls a function that has not been defined what accesses a property of a object that does not belong to the object.

This error often cross by update of web application.

Which means in previous version of web application there is pointer definition, property definition. But during updating, so the property or permission is removed, but each reference are not fully removed so this errors occur.

And last one is syntax error.

I'm very surprised because even though they are disployed real web application but still there are syntax errors and portion of syntax error is not very small.

Next I'll show the answer of the paper about research question 2.

Does the speed of interaction(event) affect the frequency of JavaScript errors?

The answer is, yes, look at this table.

The author choose random file errors for showing the result.

In test mode, for error warning in test mode there are 4 errors on average but in medium and slow mode there're 2.33 errors.

Different errors, error is similar result.

The error frequency are different among the test mode.

So this means the tester or developer should consider the speed of interaction.

Second research question is there is non-deterministic JavaScript errors?

## [20:00]

So the answer is also yes.

Look at this table.

You know in same error, in same test case, there are some different number of the errors.

So the author concludes that there is no deterministic errors in JavaScript.

Here is answer for research question 4.

First slide shows correlation between the number of JavaScript and static characteristic.

So they consider 4 static characteristic.

First turn is Alexa Rank, which means this web application how frequently used by the users.

And second one is JavaScript size.

The third one is domains the web application use.

And first one is domains with JavaScipt.

So excepting this property, JavaScript size, before that this number is squarement correlationcoefficient.

So you can interpret that if the correlation is more than 3 than there is high correlation between them.

So look 4 letters ?[21:47] low.

There is correaltion between JavaScript errors and Alexa Rank domains and domains with Javascript but there's no significant correaltion between JavaScript and size.

(Student questioning)

In the more of domain used by web application and number of domain with JavaScript the web application use.

(Student questioning)Number of domains caused by...

Web applicaiton.

This is domain, and just count domains with JavaScript.

This means with and without.

This means with JavaScript.

If there is a JavaScript code so only domains are counted and this all domains are counted.

Is it okay?

And this slide shows correlation between the number of JavaScript errors and dynamic characteristics.

So the author considered 4 dynamic characteristics.

First one is function calls, and second one is eval function calls is special function.

And third one is object property deleted.

And first one is inhebitance overridings.

So they conclude except for function of number 4 is no significant correaltion between the number of JavaScript errors and dynamic characteristics.

(Student questioning)

The authour cosider only this law or the number of matters or java script so this correlation coefficiency is quiter than pointer 3 conclude that threre is correlation between function codes and number of matters but this number is very close to pointer 3 so it is hard to say there are high correlation between them.

(Student questioning)

Javascript allow delete the property of objects so this characteristic means the application how many time delete the object property.

[25:00]

(Student questioning)

Last research question was, there inter-category correlations among the different types of errors.

The answer is yes, syntax errors and all other non-miscellaneous erros are highly correlated so this is correlation, coefficient between syntax error and prmission denied, undefined symbol null exception.

So all of them is greater than point 3, and especially undefined symbol and syntax errors are very highly correlated with each other.

And second correlation that non-deterministic null exceptions and non-deterministic undefined symbol errors are highly correlated.

This correlation, coefficient means all the errors are either non-deterministic or deterministic error at all but this coefficient is only consider non-deterministic errors.

So you can see the correlation, coefficient smaller than this one but still very high.

And the last question was are there any correlation with frameworks used and the number of Javascript errors.

In this table shows the frameworks and average number of errors and how many web application used the framework.

So J-Query is most polular framework in Javascript web application developing.

Each error switch number of errors, this almost same as over average errors.

But they found that if the developer use multiple framework, the number of errors is higher than average.

So they require that do not use multiple framework when new develop new application.

(Professor questioning)

Yes, the error is for...

(Professor questioning)

Reliable stable?

(Professor questioning)

Number of site, I so named.

(Professor questioning)So aren't there any sites.

Yes, they construct 15 sites, half of them use J-Query, and one-fifth do not use any frameworks.

(Professor questioning)

Actually I doubted but I just presented the just negotious plan.

And authors did some implications.

First implications is for programmers, web application developer.

So the autor says that some programming patterns are suggested from the experimental result.

First one is using fewer domain is good for reduing errors.

And second one is do not use multiple frameworks.

There is one interesting observation that is some errors are arise due to other component which is which cannot controlled by the programmer.

So the autor says that some programming patterns are suggested from the experimental result.developer should consider effect of external component.

So they recommend to developer to consider the external component.

## [30:00]

And search impication is that most of web pages are functionally fine although they have errors.

It means that very tolerant for its behaviors, thus, the programmers need to be more careful to ensure the reliability.

This is very common implication.

And second implication is for tester.

Testing should be done in multiple mode, so as you can see there are some differences among the test mode so the tester need to test same case in different speed.

Or they need to define test case including the event script.

Second implication is since there are non-deterministic errors which portions are 7% or more errors, for distinct errors though this is huge portion so application needs to be tested multiple times.

And someone is that there are errors related with external components, so integrated test is needed.

So my presentation is done so let's discuss the content of thie data.