Candidate Analysis of Change Impacts on the Web Application

Ade Sukendar, Ayi Purbasari, Anggoro Ari Nurcahyo

Abstract: ***Web-based applications will generally change due to bugs, new requirements, fresh environmental migration, or structural changes as part of software evolution. Continuous changes to web applications cause the program to become bigger and more complicated, so it makes difficulties in identifying which parts of the app will be changed. This study propose a technique of impact analysis in the candidate impact set stage in web application of Java-based. This technique can identify different source code file between web pages and typed of class. Source code of application will transform to intermediate format and then to object-oriented data dependency graphs format before doing identify. Identify source code take on program structure has transform based on request of change. The analysis of the impact of these changes produces the set of candidates affected by the change. The prototype is built to test candidate impact analysis results in Java web applications. Testing with case studies shows the technique can analyze the impact of program code changes on Java-based web applications.***

Keywords: Software evolution; candidate impact set; change impact analysis; Java web applications; object-oriented

I. INTRODUCTION

The software has been built will continue to change. Change occur due to bugs, new needs, migration to new environments and structural changes. Continuous changes make software bigger and more complicated. Developers of software have trouble understanding and searching for the source to be changed [1]. The process of identifying potential changes needs to be done to understand the impact that occurs if the software is changed. The process of identifying potential changes software is called change impact analysis [2].

Change Impact Analysis or CIA can be used to determine the scope of work [3], cost estimates, scheduling [4] and developer recommendations required according to ability and experience in developing prior systems [5]. CIA is doing before the implementation of change is complete, it aims to understand the object of the program and reduce the risk of implementation of change. CIA research commons is vertical domain that focus in source code with the same language [6]. In reality a software consists of various program source codes like web technology.

Revised Manuscript Received on May 22, 2019.

Ade Sukendar, Department of Informatic, Pasundan University, Indonesia

Ayi Purbasari, Department of Informatic, Pasundan University, Indonesia

Anggoro Ari Nurcahyo, Department of Informatic, Pasundan University, Indonesia

Web applications have different characteristics compared to regular applications. They are complex multi-tiered, heterogeneous architectures, and borders on the web environment [7].Moreover, web applications implement many technologies, paradigms, and programming languages, e.g. CSS, HTML, XHTML and so on [8]. Based on characteristic, the identification of source code program affected by the changes will be difficult and insufficient with program execution [9]. CIA on web applications is proposed by research [10] and [9] with identifying changes to different source code program in software repositories.

Web platform of Java-based implement several development component for web application, one of web application component and enterprise [11]. Developer using that component with create a web pages and source code based of class. Source code based of class can implemented with object-oriented structure and others with non object-oriented structure. Identify between different source code will be challenge for software maintainer.

This study propose a technique of impact analysis in the candidate impact set (CIS) stage in web application of Java-based. This technique can identify different source code file between web pages and typed of class. Source code of application will transform to intermediate format and then to object-oriented data dependency graphs format before doing identify. Identify source code take on program structure has transform based on request of change. The prototype is constructed using Java programming to show the results of an impact analysis of changes.

II. LITERATURE REVIEW

Change Impact Analysis

Analysis of the change impact plays an important role in software change activity because the process of analyzing is not an easy thing and is a challenging activity to do. The analysis consists of SIS, CIS, and AIS [12]. CIS generates the affected candidate object before the implementation. The impact analysis research on web applications is done by research [10] and [9]. The method uses a static method, limited to the clustering process based on simultaneous changes to the software repository called opportunistic change rather than the structure of the program code. So the result of the analysis of program code relation cannot be by the actual program code.

The method of static analysis on object-oriented programming has been performed by [3], [13], [14] and [15]. The analytical method proposes an impact change analysis algorithm with the graph as the representation of

the inter-method data dependency and inter-method data dependency. In the study [3] can only identify the relation between program code in the C++ language. The code analysis of the program performed is not limited to the program code but also to the bytecode representation of Java programs, conditioned to identify the java code relation and to represent it in various forms of the graph [16].

Impact analysis on Java-based web applications can identify non-object web pages and object-oriented program classes. Extract web pages uses a web page class model from [17] while extracting a programming class is a class that has been compiled with a bytecode format [16]. Extract reference program statements on each class method and impact analysis process use the method from research [3].

Object-Oriented Impact Analysis Method

The object-oriented programming paradigm has different characteristics from the procedural. The object-oriented paradigm has the characteristics of encapsulation, inheritance, polymorphism and dynamic binding. These characteristics will have difficulty in analyzing the impact and ripple effect on the changed system components.

The study by [14] creates the model and type of changes in object-oriented system library changes. The model and type of change are done to understand the combination of relationships and dependencies on a more complex class than the procedural language. Research by [4] performs an impact analysis on object-oriented systems by proposing the concept of Object-Oriented Data Dependency Graphs (OODDG), impact analysis algorithms and standard calculations.

While research by [13] uses three different object-oriented graph-dependent algorithms to calculate the transitive closure to identify the affected elements. They propose the concept of intra-method data dependency graphs to calculate the entity's impact within the method body, inter-method data dependency graphs to calculate the dependencies between the methods and object-oriented system dependency graphs to calculate the impact of changes at the system level. Furthermore, he also proposes the type of change to distinguish between related entities, namely contaminated, clean, semi-contaminated and semi-clean.

Java Web Application

Web applications are positioned on web containers consisting of several web components. The scope of web applications that will be studied is within the scope of the components of JSF and EJB. The source code of those two components will be analyzed for their dependencies and effects if there are changes. The JSF component consists of:

- Components and Pages components
- Converter components
- Validator components
- Backing bean components

The EJB object is a Java class that is above its class declared annotation EJB specification. EJB is responsible for handling application transactions. Dependencies on EJB classes consist of:

- Class dependencies, they are class to class, class to the method, class to a variable, a method to a variable, and method to method
- Web pages dependencies
- Class with web pages dependencies, they are dependency classes to web pages and web pages dependencies to classes

Research Questions

Based on the background that has been described above, as for the formulation of the problem to be discussed are:

- How is the impact change analysis technique at the candidate impact set stage capable of identifying web pages and classes of programs in Java-based web applications?
- How to design and implement a prototype by using the change impact analysis technique on a web application based on the change criteria?

III. METHOD

Analysis and Design

Analysis of the impact of CIS phase changes receives input in the form of Java-based web application program code and change criteria. Code of the program that will be in the process of Java web application program code that is wrapped in web archive format (WAR). The program code consists of web pages and program classes. The code of the program will be extracted from its structure, relation, and dependencies into the form of the graph stored in the repository. The extracted program code results in the class structure of the web page and the structure of the program class that can be selected as the change criteria. The desired change criteria are at the class member level attributes and methods. Analysis of impact using a concept of Object-Oriented Data Dependency Graphs (OODDG), impact analysis algorithms and standard calculations from research [4].

Based on the impact change analysis model on the web application, it can be designed the program needs to handle the function. Mapping the analytical function into the design as follows:

- The extract function will be handled by the extractor module.
- The analyze function will be handled by the analyzer module.
- The view function will be handled by the viewer module.

Implementation and testing

Implementation and testing are the prototypes and testing based on the proposed technique. Implementation shows the physical part of the system with various artifacts. Then it will be tested by using a small application section class with web pages and web pages with web pages.

IV. RESULTS

The impact analysis of the Candidate Impact Set (CIS) stage functions to get the candidate set to be affected. Input process is the result of the process of starting the impact set (SIS) in the form of an initial program object that will be affected if there is a change.

The initial object of this program in the CIS process is the change criteria.

Component of Programs

Model of web page class

The class model for the web page applied in this study is a class model based on [16]. The web page is transformed to UML class standard consisting of name, attribute and class operation. In the web page class, the class name represents the name of the web page. Attributes represent dynamic information content and as an input data on a web page.

Meanwhile, operations represent actions or events that occur on a web page.

The naming of the web page class attribute is taken from Expression Language (EL) statements on the value attribute of the input and output text tag of the web page. If # {*bean.field1*} means the attribute name is *field1*. The naming of the web page class method of the EL statement is the *action* and *actionListener* attribute of the *tag commands* of the web page. If EL in the commands tags:

- # {*bean.action*} then the method name is *action.*
- # {*euroConverter*} then the method name is *euroConverter*. This naming also applies to *validators*.

Meanwhile, the naming of the *method* of web pages that call another web page is the name of the web page added by the word *linker*.

Relation of program objects

The structure of the web page program is different from the class structure. The structure of a web page consists of head and body. The head section covers a set of tags to initialize the program object of the web page, while the body part covers program instructions for manipulation as well as displaying data and information. The program instruction tag on the body of the web page in the Java web application involves calling to a programming class that has a specific purpose such as a backing bean class, a validator, and a converter.

- Backing bean web pages. Backing bean web pages are tags on web pages that are identified as related to the backing bean class. Tags that refer to the backing bean class are as presented in Table 1. The tag attribute calls the backing bean class method.

Table. 1 Relation of web pages with a backing bean

No	Tag	Attribute
1	*h:commandButton*	*action, action Listener*
2	*h:commandLink*	*action, action Listener*

- Web page to web page. The web page can be called directly by another web page using the h: button and h: link as in Table 2. Both tags define the web page to be called using the outcome attribute. Calling a web page to another web page means there is a relationship between the web pages.

Table. 2 Relation of web pages with web pages

No	Tag	Attribute
1	*h:button*	*outcome*
2	*h:link*	*outcome*

- Web page validator. The web page validator describes the relationship between the web page and the validator class. Validator serves to validate the input data received from web pages. The handled validator is a self-made validator by implementing the validator interface. The web page validator tag can be seen in Table 3 below.

Table 3. Relation of web pages with validators

No	Tag	Attribute
1	*h:inputText*	*validator*
2	*f:validator*	*validatorId*

- Web page converter. The web page converter describes the relationship between a web page and a class converter. Converter serves to convert values from String to other data types and vice versa of certain data types to the String data type received from the web page. The converter handled is a self-made converter by implementing an interface converter. The web page converter tag can be seen in Table 4 below.

Table. 4 Relation of web pages with converter

No	Tag	Atribut
1	*h:outputText*	*converter*
2	*f:converter*	*converterId*

Impact Analysis Process

The impact analysis of the CIS phase change in this study receives input of Java-based web application program code and change criteria. The program code consists of web pages and program classes. The code of the program is extracted from its structure, relation, and dependencies into the form of the graph stored in the repository.

The extracted program code results in the class structure of the web page and the structure of the program class that can be selected as the change criteria. System analysis of the impact of changes in web applications can be seen in Figure 1.

The change impact analysis process consists of three processes: extracting, analyzing and viewing. The extracting process works to extract the Java-based web application program code into graph form in the repository. The analyzing process works to process the analysis of the affected object changes. Meanwhile, the viewing process serves to display the results of the object code analysis program.

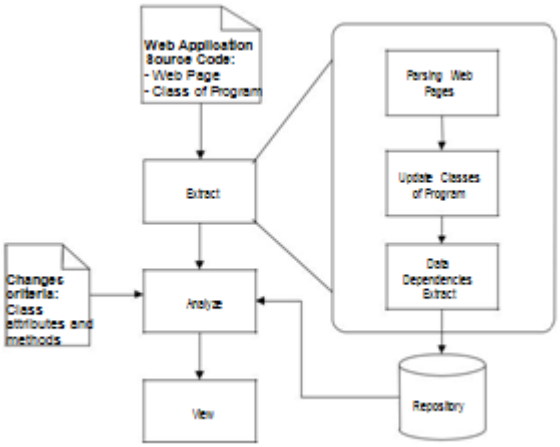

Fig. 1 CIS technique on Java web application

Extract

This process extract source code web application to graph form on repository. Web pages parsing to class then adding to that source code. If in web page have backing bean class call web page then will add as implementation in that method class. Both class will identify class structure, relation and dependency relationship.

Analyze

This process will analyzing source code object take impact of changes with using algorithm from research [3]. Processing this step needs input data criteria changes of source code object.

View

This process will show source code object that take a impact analysis from analyze processing. Sourcecode object showing in class structure view.

Extracting process

Sources of information required for the extract process of Java-based web application program code. The program code contains the compiled web pages and program classes program code. The compiled program code is formatted bytecode or an intermediate representation. In the process of extracting the source code, there is a process of parsing web pages, bytecode class manipulation, and identification of the structure and class relation of web pages and programs.

Web pages

Extracting web pages is a process to extract a web page program object into a web page classstructure. The class structure of web pages is formed based on the structure of the program object web page consisting of various tags. The process requires data input in the form of a web page as in Figure 2.

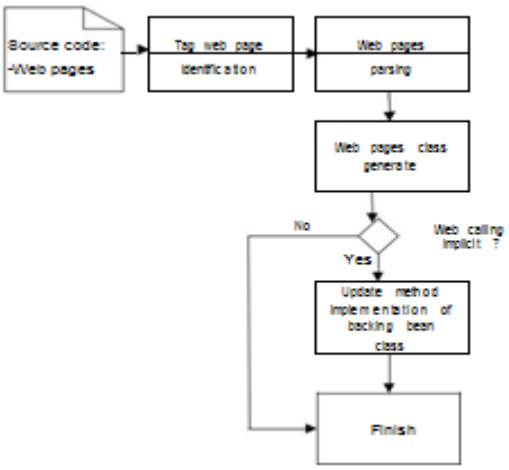

Fig. 2 The process of extracting web pages

The steps to extract web pages are as follows:

a. Identify web page tags will take attributes class and methods.

b. Parsing web pages to class structure like attributes, methods and implementation. Default method in this class have name render method. Web page classes have category that consist of *pages*, *validator*, *converter* and *backing*.

c. Web pages that have generated to class structure then save as file with class extension. That file then adding to other class in source code application in same package with web pages. If that class have method backing bean to call web pages then adding implementation to method implicitly.

Class Program

Extracting the class object program is a process to extract the class program structure, including the class of web pages that have been raised. The object of the program is the result of compilation in the form of bytecode. The extracted class structure consists of packages, class names, attributes, and methods. The output of the program class extract process is stored in the repository. The process can be seen in Figure 3.

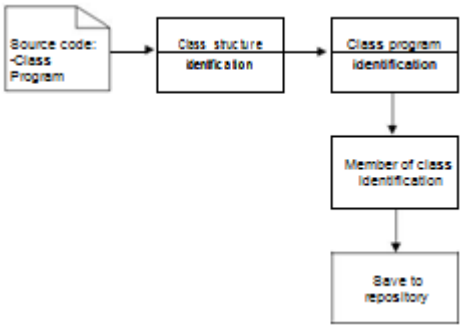

Fig. 3 The process of extracting the program class

The steps to extract source code of class are as follows:

a. Identify class structure like file name and file directory.

b. Identify source code of class like class name and package.

c. Identify property class like attribute and method.

d. Result of identification then save in repository.

Data dependency. Extracted data dependency is a process to extract dependencies of program class objects. The process of extracting data dependencies uses the graph dependency method proposed in [3] i.e. object-oriented data dependence graph (OODDG). The process can be seen in Figure 4.

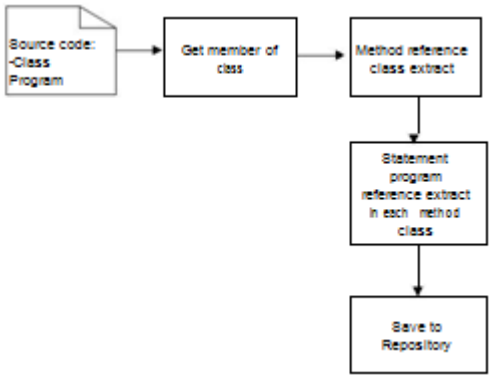

Fig. 4 The process of extracting data dependencies

The steps to extract data dependencies of class are as follows:

a. Take member of class (attribute and method) from repository.

b. Member of class has a taken, each class method will identify attribute and method references from self or others class.

c. Member of class extracted again based on program statement consist of expression statement, call of function, selection statement, looping and switch.

d. Result of process b and c then save in repository.

Analyzing and viewing the process

Analyzing process is a process that serves to find other objects that will be affected by changes. This process uses object-oriented impact analysis algorithms based on proposals from research [3] i.e. *TotalEffect*, *SetInit*, *FindEffectInClass* and *FindEffectAmongClass* algorithms.

Viewing process is a process that serves to display the results of the impact analysis of changes generated in the analyzing process by giving a presentation in the form of dependencies between objects affected by the changes. This process will show the dependencies of the class object in the form of a tree structure.

Module

Modules of prototype consist of repo, util and graph modules. The three modules are accessed by two extractor modules and analyzer as shown in Figure 5.

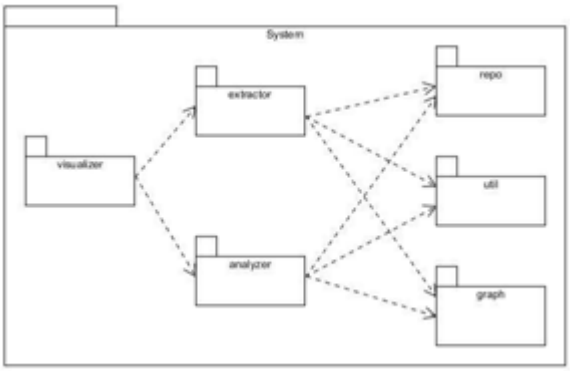

Fig. 5 Global prototype design

Extractor module have a task for extracting source code to graph. They have three main method for doing extract, method for extract pages, class and dependencies. Then Analyzer module have task for identifying source code object that has taken impact by changes. The steps to identifying are initialization criteria of changes, identifying reference dependencies attribute and method in scope inner of class and outer.

The generation of implementations on the bytecode classes is handled by the *GenerateZipFile* class. The implementation of web page method is divided into four types:

- Pages
- Converter
- Validator
- Backing

V. DISCUSSION

Implementation of the proposed CIS technique in Java web application has been made a prototype. The CIS prototype is named as Java Web Impact Analysis (JWIA). The main function of the prototype is the extracted process, choosing change criteria and analytical function. The JWIA prototype interface can be seen in Figure 6.

Tests on this research using some cases of small program code Java web applications. The first example is testing on web pages that refer to the validator class, the converter class, and the backing bean class. The backing bean class refers to the EJB class named A. The source code of the web application is extracted; the change criterion selects the method found in the validator class, the converter class and class A. The change candidate's analysis function will identify the affected program object according to the change criteria inserted. Illustration first test as shown below in Figure 7. A second example of testing like for the example first but adding one web page that has a relation to backing bean class. A third example of testing is a relation between web pages.

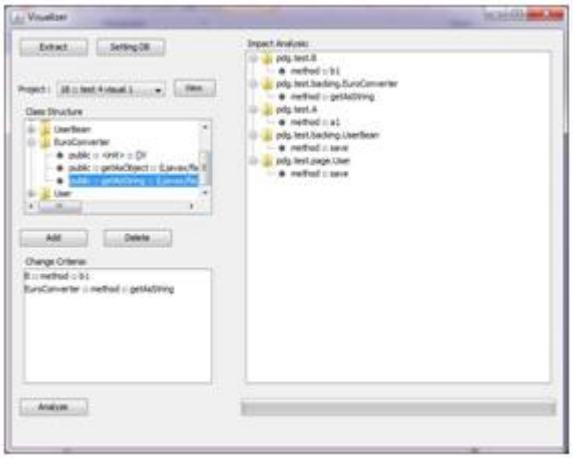

Fig. 6 JWIA prototype interface

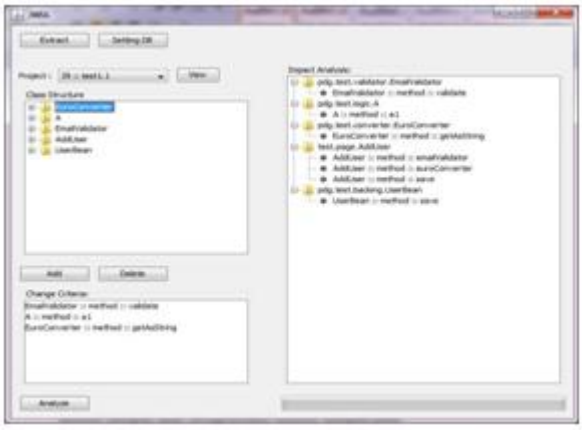

Fig. 7 Test execution results

VI. CONCLUSION AND FUTURE RESEARCH

Based on the results of analysis of the changes impact in the CIS stage in the Java-based web application program code, it can be concluded CIS technique can identify web page and program class. The identification process consists of extract, analyze, and view process. Extract performs three steps they are extracting web page, adding the implementation of the backing bean class, and class structure and its relation. Analyze is analyzing dependencies between the program class and the web page class at an intermediate or compiled level. The prototype is made for test the proposed technique, which is called Java Web Impact Analysis (JWIA). Result the test is approved with a small test with the web application.

Here are some suggestions that can be input for further research:

a. Identification of CIS with applying the concept of class *inheritance* and *polymorphism* and web page *templates*.

b. Class-based identification using annotation (sign @) that has become standard in java-based web applications.

c. Type implementation of impact analysis changes such as add, update and delete on the criteria that will change and the weight factor on each dependency of the affected object.

REFERENCES

1. Rovegard, P., Angelis L., dan Wohlin, C., *An Empirical Study on Views of Importance of Change Impact Analysis Issues*. IEEE Transaction on software engineering, Vol. 34, No. 4. 2008.
2. Bohner, S. A., Impact analysis in the software change process: a year 2000 perspective, in *Proceedings of the 12th International Conference onSoftware Maintenance (ICSM'96)*, Monterey, CA, November 1996, pp. 42–51.
3. Lee, M. L., Change impact analysis of object-oriented software. A Dissertation Submitted to the Graduate Faculty of George Mason University, Fairfax, Virginia. 1998.
4. Tripathy, P. dan Naik, K., *Software Evolution, and Maintenance*. Willey. 2015.
5. Kagdi, H., Hammad, M., dan Maletic, J. I., *Who Can Help Me with this Source Code Change?*. IEEE. 2008.
6. Lehnert, S., *A review software change impact analysis*. Technische Universität Ilmenau, Ilmenau. 2011.
7. Mansour, N. dan Baba, N., *Ripple Effect In Web Application*. Dept. of Computer Science and Mathematics, Lebanese American University, Beirut. 2010.
8. Chung, S. dan Lee, Y., Modeling Web Applications Using Java and XML Related Technologies. *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03*). IEEE. 2002.
9. Jashki, A., Zafarani, R., dan Bagheri, E., Towards a More Efficient Static Software Change Impact Analysis Method. ACM. 2008.
10. Sherriff, M. dan Williams, L., Empirical software change impact analysis using singular value decomposition. First International Conference on Software Testing, Verification, and Validation, 0:268–277. 2008.
11. Goncalves, A., *Beginning Java EE 7*. Apress. New York. 2013.
12. Bohner, S. A., Software Change Impacts: An Evolving Perspective. *Proceedings of the International Conference on software maintenance(ICSM)*, Phoenix, Arizona, IEEE Computer Society Press, Los Alamitos, CA, pp. 263–271. 2002.
13. Lee, M., Offutt, A., J., dan Alexander, R., T., Algorithmic analysis of the impacts of changes to object-oriented software, in *Proceedings of the34th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS 34)*, Santa Barbara, CA, USA, pp. 61–70.2000.
14. Kung, D., Gao, J., Hsia, P., Wen, F., Toyoshima, Y., and Chen, C., Change impact identification in object-oriented software maintenance, *inInternational Conference on Software Maintenance*, Victoria, BC, Canada, pp. 202–211. 1994
15. Ryder, B. dan Tip, F., Change impact analysis for object-oriented programs. In *Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering (PASTE '01)*, Snowbird, Utah, USA, June 2001, pp. 46–53
16. Shu, G., Sun, B., Henderson, T. dan Podgurski, A., JavaPDG: A New Platform for Program Dependence Analysis. *IEEE Sixth InternationalConference on Software Testing, Verification and Validation*. 2013.
17. Designing JSF Applications: a Storyboard Approach (DJA) part one and part two by Steven L. Murray in 2009, data obtained through the site internet: ttp://jsfcentral.com/articles/storyboard_1.html. Download at 14 Agustus 2016.