Detecting Clones in Class Diagrams Using Suffix Array

Harjot Kaur , Manpreet Kaur

Abstract— Model Driven Engineering has become standard and important framework in software research field. UML domain models are conceptual models which are used to design and develop software in software development life cycle. Unexpected copy of model elements leads to many problem. Models contain design level similarities and are equally harmful for software maintain -ace as code clones are. So number of clones need to be detected from UML domain models. This paper introduces an approach to detect clones in class diagrams. Class diagram contains redundant element which increases the complexity and need to be removed. Firstly, class diagrams are encoded as XML files. Tokens are extracted and matched using Suffix array technique. The approach is based on finding similarities in tokens known as clones.

Index Terms-Code clones, Model Clones, Suffix Array.

I. INTRODUCTION

According to Rattan et.al. [15] and Roy and Cordy[19] copying existing code and pasting it with or without any change into other sections of programme is a popular process in software development. The copied code is called a software clone and the process is called software cloning.

Various reasons for software clones are: Programmer's less knowledge and time limits, complexity of the system, language limitations, fear of making fresh code, lack of abstraction.[15] It is very important to detect code clones because of good maintainability, compact code, good patterns and to avoid copy paste approach [19] .

Model driven development (MDD) defines domain models also called the conceptual models which mainly focuses on modeling rather than computer programming. Various UML tools are used that provides multiples views of the models. There are various UML models such as class diagram, use case diagram, activity diagram, state chart diagram, sequence Diagram etc…. [14]. The UML i.e *Unified Modeling language* gives us a standard way to define a system's view including many conceptual details such as business processes. Because of large adaptability of it by software developer it is necessary to understand the importance of modeling. Use of UML makes modeling more efficient and effective. It is very important to understand the exact definition of model clones and to derive a formal framework for model clones. Very less work is done for the clone detection in UML models [11, 15, 20].

This paper presents an approach for detecting clones in Class diagrams. As class diagram provides static view of an application and considered most efficient aspect of modeling [12]. Class diagram contains redundant elements which increase the complexity and need to be removed. In this paper we propose a technique that will find similarity between models or model elements. Class diagram is created using UML tool that is encoded as XML document. The XML document is parsed to extract the tokens. Suffix array is used for token matching and matched tokens are reported as clones.

A. Motivation

- Design and implementation of an algorithm to detect the similarity from class diagram.
- Increasing demand of model based development in software field.
- Lack of work done in model based clone detection.

The remaining part of this paper contains following: - Section 2 presents the background detail of the topic. Section 3 describes our approach of work. Section 4 presents related work done. Section 5 concludes and proposes future work in the same area.

II. BACKGROUND

A. *Model Cloning*

As Source code clone detection is a big problem for code based development, the same problem also occur for duplicated parts of models in model based development. There is a significant difference between programming languages code and models so algorithms and notations used for code clone detection are difficult to directly implement to model clone detection [15, 19, 20].

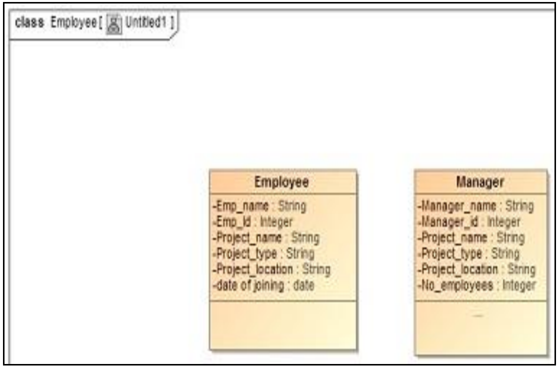
Manuscript published on 30 April 2014.
***** Correspondence Author (s)

Harjot Kaur*, Department of Computer Science and Engineering ,Doaba Institute of Engineering and Technology , Kharar,Punjab,India,+91-98881-86628,(e-mail:jotbhangu88@gmaill.com).
Manpreet Kaur , Department of Computer Science and Information Technology, Baba Banda Singh Bahadur Engineering College,FatehgarhSahib,Punjab,India,+91-97811-19498,(e-mail:manpreet.kaur@bbsbec.ac.in).

According to Storrle.[20] a *model fragment* is a set of model elements that is closed under some closure property of similarity. *Model clone* is a pair of model fragments that contains high degree of similarity or overlapping of content between the fragments.

B.Reasons for Model clones

- *Model clones due to copy/paste*:-Most of the clones are created by copy paste to reuse the content.
- *Model clones as a remains of unfinished modeling*:- Some clones are noticed as the remains of unfinished modeling by the modeler as shown in fig1.a and fig 1.b
- *Model clones through language loopholes*: - Due to some language limitations, parts of models are repeated by mistake.
- *Model clones due to time limits* :- To meet the hard time constraints most of the programmer prefer to use the existing code.
- *Model clones by intention of programmer* :- Some clones are created with purposes by the programmers to create the similar parts of models [20].



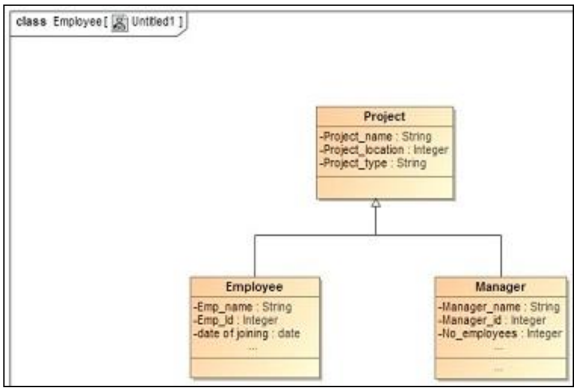


Figure:-1.a Classes with similar attributes due to unfinished modeling

Figure:-1.b Classes with removal of similar attributes

Fig1.a describes the similar attributes present in two classes that will act as duplicity for class diagram.Fig1.b is showing a new super class that contains the duplicate attributes of two classes hence reduces code size and maintainability.

C. Advantages and Applications of Detecting Model Clones

- To produce a better designed model.
- To derive a definition for understanding model clones
- To develop an algorithm to detects model clones of actual meaning.
- Resource requirements can be reduced if clones in the models are detected.
- Easy for the maintainer to maintain the model if he is aware of the presence of clones.
- Good knowledge of clones will help to introduce a effective reusable mechanism[3,10,16,20].

D. Code Clones versus Model Clones

- *Language/tool consideration*: - As code is a text file and is language dependent whereas model is integrated on some tool which is used to create or model it.
- *Identification: -* Source codes have easy identification by names and procedures whereas models use internal identifiers which are equal in definition but not identical as such.
- *Structure: -* Code is usually represented as a directory structure and models have graph like structure.
- *Syntactic representation* :- Source code is represented as a string of tokens whereas models have a dual structure, internally as a meta model class instance and externally as a set of diagrams[20].

E. Types Categorization of Model Clones

According to Storrle.[20] model clone type are following:

- *Type A: Exact Model Clone* (The copy that is identical except from layouts and internal identifiers of models e.g scope of element.)
- *Type B:Modified Model Clone* (The copy with changes to the element names, attributes and parts)
- *Type C: Rename Model Clone* (A copy with changes allowing addition or removal of parts).
- *Type D:Semantic Model Clone*(A copy , that are due to model part copying or language constraints.etc)[15,20].

III. OUR APPROACH IN MODEL CLONE DETECTION

The technique is based on token based matching. Diagrams are exported into XML files. XML is used to represent large amount of data in expressive and flexible way. Every element of XML has an XMI Id identified by "xmi:id" and a set of attributes that shows the relationship of this element with the others[14].These XMI id’s are basically used to extract the tokens. These extracted tokens are given as input to match detection techniques shown in fig: 2.There are various match detection techniques that are available eg: Suffix tree, hashing. Each technique has different clone granularity level [15].Suffix array is also introduced as new function and data structure for string comparison [8].Suffix array is a data structure used to construct and store the index of the full text which is generated in lexicographical order of strings in text. It is considered to be good for range search and fuzzy search. The suffix array of a string can be used as an index to quickly locate occurrences of a substring. Suffix array are considered to be better than suffix tree in terms of memory space and access speed [2, 6].

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

Table: 1 Multidimensional Suffix Array

Class 1	Class 2	…….	Class L
Attribute1	Attribute1	Attribute1	Attribute1
Attribute 2	Attribute 2	Attribute 2	Attribute 2
Attribute M	Attribute N	Attribute O	Attribute P
Operation 1	Operation 1	Operation1	Operation 1
Operation M	OperationN	OperationO	Operation P

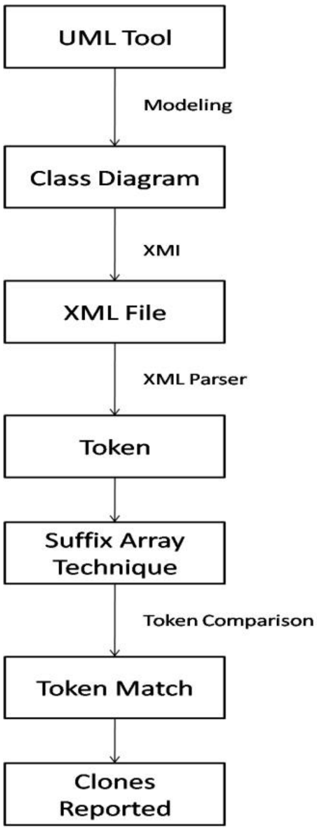


Figure: 3 Steps of our approach

A. *Steps of our approach*

- Creating Class diagram using UML tool.
- XML file is generated from class diagram.
- Analysis of XML file & detection of all the tokens (i.e. classes, operations, attributes) as shown in table: 1.
- Decryption of properties of found tokens (e.g. data type, visibility).
- Use of multi-dimensional suffix array approach to compare retrieved tokens.
- Tokens are matched with in class and outside the class.
- Matched tokens are considered to be clone.
- Clones are verified and analysed to provide related information.

➢ Number of clones
➢ Clone clusters and their instances
➢ Percentage of repetition of same clone.

IV. RELATED WORK

Harald Storrle [20] defined a framework which defines model clones, model fragments and clone detection algorithm for UML domain models. As code clones affects the source code quality, model clones affects model quality. But there are significant differences between them .He proposed MQ_{lone} tool as clone detection algorithm and provides us with various heuristics such as NAME ,NAME2,INDEX etc.

An empirical study by Rattan et.al. [15] Specified reusing software by means of copy and paste is a frequent activity in software development that leads to bug propagation and serious maintenance problems. The empirical evaluation of clone detection tools/techniques is presented. They proposed various model clone detection approaches. Future work is proposed for UML models clone detection and to enhance the speed of model clone detection.

Huang and Le Fei [8] introduced the technique of similarity measurement of source code. They proposed a quick match algorithm suitable for quick similarity of tokens extracted from the source code using suffix array. Suffix Array Based Similarity Matching algorithm finds common string among two token strings.

Rattan et.al. [16] introduced an approach to detect clones in UML models using tree structure. The technique is defining similarity measures between two model elements. According to the paper UML models are converted into XMI files. These files are represented as trees on which sub tree comparison is performed and clones are reported.

Antony et.al. [3] presented an approach for identifying near-miss interaction clones in UML behavioral models. Main goal is to identify similar pattern of interaction that are used to characterize run time behavior of interactive system. The approach is text based and focuses on the behavioral clone detection changes.

Abdul et.al. [2] presented string algorithm to find suitable data structure and algorithm for token based clone detection. RTF tool uses suffix array to detect various string matches. The techniques also give precision. Clone detection in RTF is implemented in such a way that standardized output of string matching is received.

Deissenboeck et.al. [4] proposed an approach to automatically identify duplicity in graphical models. They presented an industrial case study for BMW group that provides an efficient clone detection technique in model based quality assurance. In this paper technique mainly focus on improvement of scalability and relevancy.

Deissenboeck et.al [5] presented an approach for the automatic detection of clones in large models in model based development of control systems. The approach is based on graph theory applied on matlab/simulink models and is implemented on industrial models to show various clones.

Pham. et.al. [12] proposed ModelCD novel clone detection tool for matlab/simulink models. The tool efficiently and accurately detect exactly match and approximately match model clones. This tool is concluded to be better for high quality and running time.

Hauke Petersen [4] proposed another clone detection algorithm and technique NovelCD in his thesis. Similarity measures are defined for Matlab Simulink models. For Simulink models similarity problems can be generalized to the problem of graph isomorphism.NovelCD is compared and evaluated with other techniques to calculate the precision values.

Abdul and Jarzabek [1] presented a solution of detecting basic type of design level similarities. Patterns of co-occurring clones from different files are found by using frequent itemset mining technique. The study basically supports structural clone detection by using Clone Miner tool. Token based matching using suffix array is applied to get useful results.

Purchase et.al [12] presented a empirical study on class diagram. This paper reported an experiment that takes a human comprehension perspective on UML class diagram notational variants. The experiment specified subjects to indicate whether a supplied specification matched each of a set of experimental diagrams. The results reveal that the best performing notation may depend on the task for which it is used.

V. CONCLUSION and FUTURE WORK

Large adaptability of model based development in software field is promoting model based clone detection. In this paper we are presenting a technique to detect clones in class diagrams using suffix array with various results and interpretations. The present work reports that class diagram contains number of redundant elements. Similar attributes or operations present in two different classes are known as clones. Our result has shown that there are number of clones that occur at multiple places hence increases maintainability. The present work reported that there is cluster of elements in diagram that repeats together at multiple places. The number of clones in a class diagram provides us with percentage of clone coverage.

So we can conclude that finding redundancy or clones from the class diagram will help the developer to remove the clones, for better maintain ace and understandability of model because most of the developers interact with the system through diagrams only. Awareness of clones will help in reusable mechanism.

Our present technique can be used to explore clone detection in state chart and activity diagram. We will try to categorize clones detected into suitable category for better interpretation of results. Class diagram with large number of classes can be taken to check the working of the algorithm. . The functionality can be added to algorithm that can rate the clones relevant or non relevant for better maintaince.

REFERENCES

[1] Abdul.H.B. and Jarzabek.S.,2005 "*Detecting Higher-level Similarity Patterns in Programs*" ESEC-FSE'05,ACM,.Lisbon,Portugal.

[2] Abdul.H .B., Puglisi.S.J., Smyth.W.F., Turpin.A. and Jarjabek.S., 2007 "*Efficient Token Based Clone Detection with Flexible Tokenization*" ,ESEC/FSE'07,ACM ,Cavtat Croatia.

[3] Antony.E.P, Alafi.M.H. and Cordy.J.R,,2013 " *An-Approach to clone detection in Behavioral Models"* Queen's university,Kingston,Canada,AAC-WCRE.

[4] Deissenboeck.F, Hummel.B,Juergens.E, Pfaehler.M .and Schaetz, B.,2010"*Model Clone Detection in Practice*``, IWSC`10,Cape Town, South Africa.,pp.37-44.

[5]Deissenboeck.F.,Hummel,B.,Juergens,E.,Schatz,B.,Wagner,S.,Giard,J.F. and Teuchert,S.,2008 "*Clone Detection in Automotive model-Based Development"* ,ICSE`08,ACM,Leipzig,Germany.pp.603-612.

[6] Falke, R., Koschke, R. and Frenzel, P.,2008*." Empirical Evaluation of Clone Detection Using Syntax Suffix Trees",* Empirical Software Engineering, Vol. 13, No. 6, pp. 601-643.

[7] Hummel, B.,Juergens, E. and Steidl, D., 2011" *Index-Based Model Clone Detection*", Proceedings of 5th International Workshop on Software Clones, Honolulu, USA, pp-21-27.

[8] Lin.H.J. and Peng.L.F.,,2009 "*Quick Similarity Measurement of Source Code based on Suffix Array"*, International Conference on Computational Intelligence and Security" ,DOI 10.1109/CIS.2009.175.

[9] Liu.H, Zhiyi.M , Zhang.L. and Shao.W.*," Detecting Duplications in Sequence Diagrams Based on Suffix Trees*" Software Institute, School of Electronics Engineering and Computer SciencePeking University, Beijing , China..

[10] Kaur M.,Rattan D.,Bhatia R. and Singh M.*,"Comparison and Evaluation of Clone Detection Tools: An Experimental Approach."* CSI journal of computing, Vol 1: No of 4,Pg 44-55,2012.

[11] Kaur M.,Rattan D.,Bhatia R. and Singh M.*,"Clone detection in Models : an Empirical Study."* 3rd IBM Collaborative Academia Research Exchange(I-CARE) 2011,New Delhi, India,October 13,2011.

[12] Pham.N.H, H. A.,Nguyen, T. T., Nguyen, J.M.Kofahi and Nguyen,T.N.,2009." *Complete and Accurate Clone Detection in Graph-based Models*", ICSE'09,Vancouver,Canada, IEEE.

[13] Petresen.H,2012 "*Clone Detection in Matlab Simulink Models*" ,IMM-M.Sc,Berlin..

[14] Purchase.H.C, Colpoys.L., McGill.M., Carrington.D. and Britton.C.,2001 "*UML class diagram syntax: an empirical study of comprehension*", Australian Symposium on Information Visualization, Sydney,vol.9.

[15] Rattan.D, Bhatia.R, and Singh.M ,2013. "Software *clone detection: A systematic review*", Information and Software Technology 55 pp.1165-1199.

[16] Rattan.D, Bhatia.R and Singh.M, 2012 " *Model Clone detection based on tree comparison",*IEEE ,.

[17] Roy, C.K., Cordy J.R. and Koschke, R., 2009. *"Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach",* Science of Computer Programming , Vol.74,No. 7,pp. 470-495.

[18] Roy, C.K., Cordy J.R. and Kosher, R.,2008. *"An Empirical Study of Function clones in Open Source Software Systems",* Proceedings of 15 th Working conference on Reverse Engineering,pp-81-90.

[19] Roy, C.K., Cordy J.R. and Koschke, R.,2007*." A Survey on Software Clone Detection Resarch",* Technical Report 2007-541, Queen's University at Kingston Ontario,Canada,115pp.

[20] Storrle.H" *Towards Clone Detection in UML domain models*", DOI:10.1007/s10270-011-0217-9.

[21] Yamashina.T, H.Uwano,K.Fushida,Y.Kamei,M.Nagura,S.Kawaguchi and H.Lida,*"Shinobi: A Real Time Code Clone Detection Tool for Software Maintenance"* nara institute of science and technology.

Harjot Kaur has pursuing her M..Tech in Computer Science and Engineering and done her B.Tech in Compute Science and Engineering from Punjab Technical University. Currenty she is working as a lecturer in Doaba Institute of Engineering and Technology, Kharar.Her research intersrests are in model clone detection in class diagrams.

Manpreet Kaur has completed her M.Tech. in Computer Science and Engineering and B.Tech. in Information Technology from Punjab Technical University. Currently she is working as an assistant professor at Baba Banda Singh Bahadur Engineering College, Fatehgah Sahib. Her research interests are in software clone detection and software maintenance.