STMIK AKAKOM YOGYAKARTA
Code for Life

ISRITI 2018

IEEE
IEEE Conference Number #45367

# ISRITI2018 proceeding

## International Seminar on Research of Information Technology and Intelligent Systems 2018

## "Opportunities and Trends on The Decentralized System & Future of Internet"

Yogyakarta-Indonesia, 21-22 November 2018

www.isriti.akakom.ac.id

Organized by :

STMIK AKAKOM YOGYAKARTA Code for Life

IRSI Indonesia Researcher & Scientist Institute

IEEE INDONESIA SECTION

Supported by :

EDAS

IEEE Advancing Technology for Humanity

IEEE Xplore Digital Library

IEEE Xplore® Digital Library

# Activity Diagram Similarity Measurement: A Different Approach

Reza Fauzan
Informatics Department
Institut Teknologi Sepuluh Nopember,
Politeknik Negeri Banjarmasin
Surabaya, Indonesia
reza.fauzan@poliban.ac.id

Siti Rochimah
Informatics Department
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
siti@if.its.ac.id

Daniel Siahaan
Informatics Department
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
daniel@its.ac.id

Evi Triandini
Information Systems
STMIK STIKOM Bali
Bali, Indonesia
evi@stikom-bali.ac.id

*Abstract*— **Reusing UML diagram requires a measurement to find the same UML diagram in software repositories. That is the reason why it is challenging and important in software engineering. This paper proposed an activity diagram similarity measurement in software reuse. The measurement uses the property and the flow information resided in the activity diagram. The property information contains the type of node and the value. The flow information contains the source node, flow's name, and a target node. The preliminary result shows that the semantic and structural similarity is a good parameter to measure the similarity.**

***Keywords—activity diagram, adaptive weight, semantic similarity, structural similarity***

## I. Introduction

Universal Modelling Language (UML) is a language in standard modelling for software development [1]. UML helps interaction between stakeholders. For instance, UML can help software developer to make system and user interaction models [2]. The development of UML has several issues. One of them is how to reduce the time consuming in software development. The software developer will take a lot of time if they build a design from the very beginning [3], [4]. Software reuse is one of reducing time solution. Reusing UML diagram can speed up the software development process. Besides, it can lower the risk used and the software cost [5].

Reusing UML diagram requires a measurement to find the same UML diagram in software repositories. That is the reason why it is challenging and important in software engineering [6]. In addition, if an artefact of UML diagram found in the repository, we can reuse the rest artefact of the same software models [5], [7].

There are some researches to measure the UML diagram similarity. This paper used the UML activity diagram to be measured by the similarity. Previous research measures the similarity using graph pattern [8], [9]. They found the same UML activity diagram by the type of node. For instance, the pattern is initial-action-object-action-final. They only found the activity diagram which had the same pattern without the value of the node. The output could be in a different domain. Other research measure the UML activity diagram similarity by the type and value of nodes [10], [11]. They only found the similarity of sliced part of UML activity diagram. In the beginning, they converted nodes into a directed graph. After that, they sliced by the edge/flow. The output of their research cannot compare UML activity diagram as a whole diagram.

This paper proposed a method to measure the similarity between two UML activity diagrams [12]. The method measures the semantic and structural similarity between the two diagrams. The measurement uses the property and the flow information resided in the activity diagram. The property information contains the type of node and the value. The flow information contains the source node, flow's name, and a target node. In measuring the flow information, we combine all the information in a UML activity diagram into a whole unit of similarity measurement. This study introduced an adaptive weight. The method uses weight to show the significance of each parameter within a diagram similarity measurement. The weight would be varied with respect to the availability of components within the two activity diagrams.

This new field of software engineering solution would speed up the software development process. This paper shows an approach which measures the similarity using more complete information (semantic and structural property) and a rigorous comparison (through the use of adaptive weight).

## II. Research Method

This section showed the method used in this paper. The first stage is the diagram preprocessing. The second stage is measuring UML activity diagram similarity.

### *A. Diagram Preprocessing*

Measurement similarity between two UML activity diagram requires preprocessing. The UML activity diagrams were designed by a tool. Then, the tool converts UML activity diagram into XMI-format. XMI is used to help choose the metadata that will be used in the next stage.

The metadata used can be found from the information in a UML activity diagram. UML activity diagram consists of nodes, they are action, object, and control[13]. Action node and object node consist of lexical information. They can be measured directly by finding the semantic meaning. This paper called action and object as a property information. On the other hand, a control node is a node that relates to the other node. It includes some node, they are initial node, final node, decision node, merge node, fork node, and join node. Control node can show the node flow of UML activity diagram. This

ISBN 978-1-5386-7421-5/18/USB ©2018 IEEE

paper called information in the control node as a flow information.

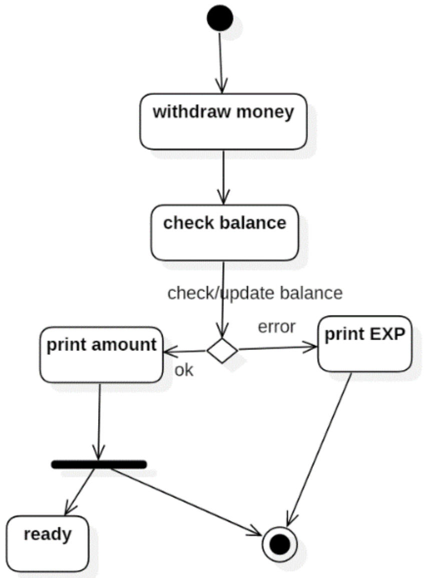

Fig. 1. UML Activity Diagram Example 1

Based on Figure 1, we can get an XMI-format file by a tool conversion. After that, we can map the information into the property and flow information. The metadata retrieval result from XMI-format are as follows:

*Property information:*

→ *action1: withdraw money*

→ *action2: check balance*

→ *action3: print amount*

→ *action4: print EXP*

→ *action5: ready*

*Flow information:*

→ *flow1: (initial: initial, name: -, action: withdraw money)*

→ *flow2: (action: withdraw money, name: -, action: check balance)*

→ *flow3: (action: print EXP, name: -, final: final)*

→ *flow4: (fork: fork, name: -, final: final)*

→ *flow5: (action: check balance, name: check/update balance, decision: decision)*

→ *flow6: (decision: decision, name: ok, action: print amount)*

→ *flow7: (decision: decision, name: error, action: print EXP)*

→ *flow8: (action: print amount, name: -, fork: fork)*

→ *flow9: (fork: fork, name: -, action: ready)*

Property information has 5 action nodes, they are withdraw money, check balance, print amount, print EXP, and ready. Flow information has 3 parts. They are source node, flow's name, and target node. Every source and target node have the information of node type.

## *B. Measurement Similarity*

The measurement similarity method used cosine similarity for every metadata found [14]. This paper also used Wu Palmer method and Wordnet to find semantic similarity between lexical information [15], [16]. If the semantic similarity cannot be measured, this paper used Levensthein distance to measure syntactically[16].

As previously informed, the similarity between two UML activity diagrams can be measured based on their metadata. The metadata has two information, namely property information (*propSim*) and flow information (*flSim*). Equation 1 shows how to measure the similarity between two UML activity diagrams (*activitySim*), i.e. $d_1$ and $d_2$.

$$activitySim(d_1, d_2) = w_{prop} \times propSim(d_1, d_2) + w_{fl} \times flSim(d_1, d_2) \qquad (1)$$

where $w_{prop}$ is the weight of the property similarity from $d_1$ and $d_2$. And $w_{fl}$ is the weight of flow similarity from $d_1$ and $d_2$. They are arbitrary weight. The next state is to measure property similarity (*propSim*) from $d_1$ and $d_2$. This measurement is described in Equation 2.

$$propSim(d_1, d_2) = w_{act} \times actSim(d_1, d_2) + w_{obj} \times objSim(d_1, d_2) \qquad (2)$$

where $w_{act}$ is the weight of the metadata action similarity from $d_1$ and $d_2$. And $w_{obj}$ is the weight of metadata object similarity from $d_1$ and $d_2$. The weight value is based on some condition as follows.

- If both UML activity diagrams have action and object, $w_{act}$ is the number of action nodes divided by the total number of action and object node. And $w_{obj}$ is the number of object nodes divided by the total number of action and object node.
- If one of them or both of them do not have the object, $w_{act}$ is 1 and $w_{obj}$ is 0.
- If one of them or both of them do not have action, $w_{act}$ is 0 and $w_{obj}$ is 1.

The next state is to measure metadata action similarity (*actSim*) from $d_1$ and $d_2$. This measurement is described in Equation 3.

$$actSim(d_1, d_2) = \frac{Max(\sum_{i,j=1}^{Max(|ACT_1|,|ACT_2|)} CosineSim(act_i,act_j))}{|ACT_1|,|ACT_2|} \qquad (3)$$

where $ACT_1$ and $ACT_2$ are a collection of action metadata lexical information from two UML activity diagrams ($act_i$, $act_j$). Then, Equation 2 showed the object metadata similarity (*objSim*). This measurement is described in Equation 4.

$$objSim(d_1, d_2) = \frac{Max(\sum_{i,j=1}^{Max(|OBJ_1|,|OBJ_2|)} CosineSim(obj_i,obj_j))}{|OBJ_1|,|OBJ_2|} \qquad (4)$$

where OBJ$_1$ and OBJ$_2$ are the collection of object metadata lexical information from two UML activity diagrams (*obj$_i$*, *obj$_j$*). Then, Equation 1 showed the similarity of flow information between two UML activity diagrams (*flSim*). This measurement is described in Equation 5.

$$flSim(d_1,d_2) = w_{init} \times initSim(d_1,d_2) + w_{fin} \times finSim(d_1,d_2) + w_{des} \times desSim(d_1,d_2) + w_{mrg} \times mrgSim(d_1,d_2) + w_{frk} \times frkSim(d_1,d_2) + w_{join} \times joinSim(d_1,d_2) + w_{aa} \times aaSim(d_1,d_2) + w_{oo} \times ooSim(d_1,d_2) + w_{oa} \times oaSim(d_1,d_2) + w_{ao} \times aoSim(d_1,d_2) \quad (5)$$

where w$_{init}$ is the similarity weight of the flow which the source node is initial node. w$_{fin}$ is the similarity weight of the flow which the target node is final node. w$_{des}$ is the similarity weight of the flow which has decision node. w$_{mrg}$ is the similarity weight of the flow which has merge node. w$_{frk}$ is the similarity weight of the flow which has fork node. w$_{join}$ is the similarity weight of the flow which has join node. w$_{aa}$ is the similarity weight of the flow which the source and target node are action node. w$_{oo}$ is the similarity weight of the flow which the source and target node are object node. w$_{oa}$ is the similarity weight of the flow which the source and target node are object and action node. And w$_{ao}$ is the similarity weight of the flow which the source and target node are action and object node. The determination of these weight values uses the same method as the weight value determination in Equation 2. The following is how to determine the amount of weight.

1. Find the part of similarity flow where both diagrams have it.
2. Count the number of occurrences of nodes in the flow according to where the weight is (only flow similarity found in number 1).
3. Each weight can be normalized by dividing the result in number 2 and total result in number 2.

Equation 5 showed the similarity of the flow which the source node is initial node (*initSim*). This measurement is described in Equation 6.

$$initSim(d_1,d_2) = \frac{Max(\sum_{i,j=1}^{Max(|INIT_1|,|INIT_2|)} CosineSim(init_i,init_j))}{|INIT_1|,|INIT_2|} \quad (6)$$

where INIT$_1$ and INIT$_2$ are the collection of type and value of target node in flow which the source node is initial node from two UML activity diagrams (*init$_i$*, *init$_j$*). Then, Equation 5 showed the similarity of the flow which the target node is final node (*finSim*). This measurement is described in Equation 7.

$$finSim(d_1,d_2) = \frac{Max(\sum_{i,j=1}^{Max(|FIN_1|,|FIN_2|)} CosineSim(fin_i,fin_j))}{|FIN_1|,|FIN_2|} \quad (7)$$

where FIN$_1$ and FIN$_2$ are the collection of type and value of source node in flow which the target node is final node from two UML activity diagrams (*fin$_i$*, *fin$_j$*). Then, Equation 5 showed the similarity of the flow which has decision node (*desSim*). This measurement is described in Equation 8.

$$desSim(d_1,d_2) = \frac{Max(\sum_{i,j=1}^{Max(|DES_1|,|DES_2|)} CosineSim(des_i,des_j))}{|DES_1|,|DES_2|} \quad (8)$$

where DES$_1$ and DES$_2$ are the collection of type and value of source or target node in flow which has decision node from two UML activity diagrams (*des$_i$*, *des$_j$*). Then, Equation 5 showed the similarity of the flow which has merge node (*mrgSim*). This measurement is described in Equation 9.

$$mrgSim(mrg_1,mrg_2) = \frac{Max(\sum_{i,j=1}^{Max(|MRG_1|,|MRG_2|)} CosineSim(mrg_i,mrg_j))}{|MRG_1|,|MRG_2|} \quad (9)$$

where MRG$_1$ and MRG$_2$ are the collection of type and value of source or target node in flow which has merge node from two UML activity diagrams (*mrg$_i$*, *mrg$_j$*). Then, Equation 5 showed the similarity of the flow which has fork node (*frkSim*). This measurement is described in Equation 10.

$$frkSim(frk_1,frk_2) = \frac{Max(\sum_{i,j=1}^{Max(|FRK_1|,|FRK_2|)} CosineSim(frk_i,frk_j))}{|FRK_1|,|FRK_2|} \quad (10)$$

where FRK$_1$ and FRK$_2$ are the collection of type and value of source or target node in flow which has fork node from two UML activity diagrams (*frk$_i$*, *frk$_j$*). Then, Equation 5 showed the similarity of the flow which has join node (*joinSim*). This measurement is described in Equation 11.

$$joinSim(join_1,join_2) = \frac{Max(\sum_{i,j=1}^{Max(|JOIN_1|,|JOIN_2|)} CosineSim(join_i,join_j))}{|JOIN_1|,|JOIN_2|} \quad (11)$$

where JOIN$_1$ and JOIN$_2$ are the collection of type and value of source or target node in flow which has join node from two UML activity diagrams (*join$_i$*,*join$_j$*). Then, Equation 5 showed the similarity of the flow which the source and target node are action node (*aaSim*). This measurement is described in Equation 12.

$$aaSim(aa_1,aa_2) = \frac{Max(\sum_{i,j=1}^{Max(|AA_1|,|AA_2|)} CosineSim(aa_i,aa_j))}{|AA_1|,|AA_2|} \quad (12)$$

where AA$_1$ and AA$_2$ are the collection of type and value of source and target node in flow which both of them are action node from two UML activity diagrams (*aa$_i$*, *aa$_j$*). Then, Equation 5 showed the similarity of the flow which the source and target node are object node (*ooSim*). This measurement is described in Equation 13.

$$ooSim(oo_1, oo_2) = \frac{Max(\sum_{i,j=1}^{Max(|OO_1|,|OO_2|)} CosineSim(oo_i,oo_j))}{|OO_1|,|OO_2|} \quad (13)$$

where OO$_1$ and OO$_2$ are the collection of type and value of source and target node in flow which has merge node from two UML activity diagrams (*oo$_i$*, *oo$_j$*). Then, Equation 5 showed the similarity of the flow which the source and target node are object and action node (*oaSim*). This measurement is described in Equation 14.

$$oaSim(oa_1, oa_2) = \frac{Max(\sum_{i,j=1}^{Max(|OA_1|,|OA_2|)} CosineSim(oa_i,oa_j))}{|OA_1|,|OA_2|} \quad (14)$$

where OA$_1$ and OA$_2$ are the collection of type and value of source or target node in flow which the source and target node are object and action node from two UML activity diagrams (*oa$_i$*, *oa$_j$*). Then, Equation 5 showed the similarity of the flow which the source and target node are action and object node (*aoSim*). This measurement is described in Equation 15.

$$aoSim(ao_1, ao_2) = \frac{Max(\sum_{i,j=1}^{Max(|AO_1|,|AO_2|)} CosineSim(ao_i,ao_j))}{|AO_1|,|AO_2|} \quad (15)$$

where AO$_1$ and AO$_2$ are the collection of type and value of source or target node in flow which the source and target node are action and object node from two UML activity diagrams (*ao$_i$*, *ao$_j$*).

Lexical information in Equation 3,4,6, and 7 can be measured directly using Wu Palmer method and Wordnet to find the semantic meaning. But, Equation 12, 13, 14, and 15 two different lexical information. They are source node and target node. We can make a rigorous comparison if source node compared to the other source node and target node compared to the other target node. Source node cannot compare to target node. It is not a rigorous comparison and will change the flow direction and the diagram structure. The measurement with all content flow (*cfSim*) is described in Equation 16.

$$cfSim(fl_1, fl_2) = w_{src} \times WuP(src_1, src_2) + w_{nm} \times WuP(nm_1, nm_2) + w_{tgt} \times WuP(tgt_1, tgt_2) \quad (16)$$

where a flow has source node (*src*), flow name (*nm*), and target node (*tgt*). w$_{src}$, w$_{nm}$, and w$_{tgt}$ is the weight of source node, flow name, and target node. The determination of these weight values uses the same method as the weight value determination in Equation 2 and 5. Besides, Equation 16 cannot be implemented directly in Equation 8-11. The lexical information contains two flow from two couple node. They are pre-flow and post-flow. We have to do a rigorous comparison to measure the similarity pre and post flow (*pairFlowSim*). Equation 17 described the measurement.

$$pairFlowSim(pair_1, pair_2) = w_{pre} \times cfSim(fl_{a1}, fl_{a2}) + w_{post} \times cfSim(flow_{b1}, flow_{b2}) \quad (17)$$

where w$_{pre}$ and w$_{post}$ are the weight of pre-flow and post-flow.

## III. Empirical Result and Analysis

The main purpose of this paper is to measure the similarity of two activity diagrams. We combine every information in UML activity diagram into an information and show the making of adaptive weight. For example, we provided two UML activity diagram (AD_1 and AD_2), we can see it in Figure 1 and Figure 2. They are semantically the same diagram. But they had different structure. AD_2 had a more complex structure than AD_1. From this example, we can measure the similarity using our proposed method.

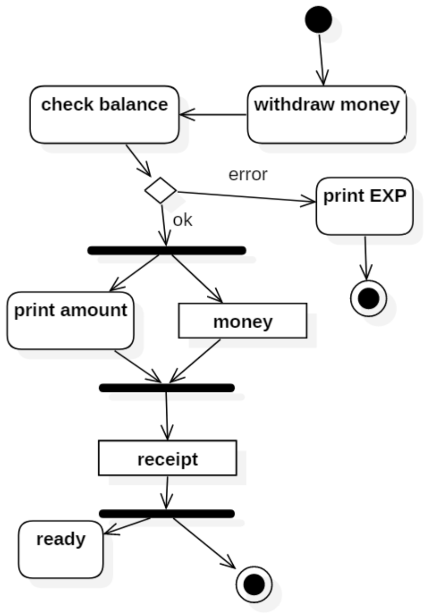

Fig. 2. UML Activity Diagram Example 2

We show the property similarity as an example of the measurement. Table I show the property similarity. The left column is the property of AD_1. The top row is the property of AD_2.

TABLE I. Property Similarity Between AD_1 dan AD_2

| | pro12 | pro22 | pro32 | pro42 | pro52 |
|---|---|---|---|---|---|
| pro11 | 1 | 0.629 | 0.707 | 0.531 | 0.662 |
| pro21 | 0.629 | 1 | 0.796 | 0.659 | 0.855 |
| pro31 | 0.707 | 0.796 | 1 | 0.694 | 0.769 |
| pro41 | 0.531 | 0.659 | 0.694 | 1 | 0.563 |
| pro51 | 0.662 | 0.855 | 0.769 | 0.563 | 1 |

As well as we knew, AD_1's action is the same as AD_2's action. And AD_1 did not have object property. So, w$_{act}$ is 1 and w$_{obj}$ is 0. We did not measure *objSim*.

Using our measurement method, we measure the similarity value is almost 1. Based on the result, this paper showed all information might affect the similarity of UML activity diagram. But, both diagrams need to have a rigorous comparison. The two things are the improvement from previous methods [9], [11]. Semantic similarity from the property and structural similarity from flow are a good parameter to measure the activity diagram similarity [12].

## IV. Conclusion

This paper introduces a method to measure the semantic and structural similarity of two UML activity diagrams. The method adopts Wu Palmer and Levensthein Distance to measure word similarity. The method also adopts cosine similarity to measure activity similarity. The proposed method consists of two parts, namely the semantic similarity of activity-pairs and flow similarity of flow-pairs. It considers various nodes and flows, of two models of activity diagram. Every detail information of UML activity diagram can determine UML activity diagram similarity. A rigorous comparison could be a good way to enhance the measurement method. So, an adaptive weight was needed for this measurement. The preliminary result shows that this measurement method could depict the various aspect of similarity of two activity diagrams.

Further research should be carried out to determine using larger dataset and more complete parameters. Thus, it is necessary to look for an alternative algorithm that is more accurate than the cosine similarity.

## AcknowledgEment

The result is in cooperation between Institut Teknologi Sepuluh Nopember, Politeknik Negeri Banjarmasin, and STMIK STIKOM Bali.

## References

[1] M. J. Chonoles, "What is UML?," in *OCUP Certification Guide: UML 2.5 Foundational Exam*, 2018, pp. 17–41.

[2] M. Chechik, S. Nejati, and M. Sabetzadeh, "A relationship-based approach to model integration," *Innov. Syst. Softw. Eng.*, vol. 8, no. 1, pp. 3–18, 2011.

[3] W. N. Robinson and H. G. Woo, "Finding reusable UML sequence diagrams automatically," *IEEE Softw.*, vol. 21, no. 5, pp. 60–67, 2004.

[4] F. M. Ali and W. Du, "Toward reuse of object-oriented software design models," *Inf. Softw. Technol.*, vol. 46, no. 8, pp. 499–517, 2004.

[5] I. Sommerville, *Software Engineering*. 2010.

[6] D. S. Kolovos, D. Di Ruscio, A. Pierantonio, and R. F. Paige, "Different Models for Model Matching: An analysis of approaches to support model differencing," *2nd Work. Comp. Versioning Softw. Model. (CVSM'09), ACM/IEEE Int. Conf. Softw. Eng.*, pp. 1–6, 2009.

[7] T. C. Lethbridge and R. Laganiere, "Object-Oriented Software Engineering : Practical Software Development Using Uml and Java," *McGraw-Hill Publ. Co.*, p. 561, 2004.

[8] P. Wohed, M. Dumas, A. H. M. Ter Hofstede, and N. Russell, "Pattern-based analysis of UML activity diagrams," no. December 2004, pp. 1–22, 2004.

[9] T. Kuschke and P. Mäder, "Pattern based Auto Completion of UML Modeling Activities," *Proc. 29th ACM/IEEE Int. Conf. Autom. Softw. Eng. - ASE '14*, pp. 551–556, 2014.

[10] D. Kundu and D. Samanta, "A novel approach to generate test cases from UML activity diagrams," *J. Object Technol.*, vol. 8, no. 3, pp. 65–83, 2009.

[11] A. Talai and Z. E. Bouras, "Software evolution based activity diagrams," *ICIT 2017 - 8th Int. Conf. Inf. Technol. Proc.*, no. 1995, pp. 82–88, 2017.

[12] D. O. Siahaan, Y. Desnelita, Gustientiedina, and Sunarti, "Structural and Semantic Similarity Measurement of UML Sequence Diagrams," in *International Conference on Information & Communication Technology and System (ICTS)*, 2017, pp. 227–234.

[13] Omg, "UML 2.4.1 Superstructure Specification," *October*, vol. 02, no. August, pp. 1–786, 2004.

[14] A. Kutuzov, A. Panchenko, S. Kohail, M. Dorgham, O. Oliynyk, and C. Biemann, "Learning Graph Embeddings from WordNet-based Similarity Measures," 2018.

[15] Z. Wu and M. Palmer, "Verb Semantics and Lexical Selection," 1994.

[16] Arthur M. Jacobs and A. Kinder, "Features of word similarity," *Comput. Lang.*, pp. 1–20, 2018.