Sequence Diagram Similarity Measurement: A Different Approach

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

Reza Fauzan
Departement of Computer Science
Institut Teknologi Sepuluh Nopember,
Politeknik Negeri Banjarmasin
Surabaya, Indonesia
reza.fauzan@poliban.ac.id

Daniel O Siahaan
Departement of Computer Science
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
daniel@its.ac.id

Siti Rochimah
Departement of Computer Science
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
siti@if.its.ac.id

***Abstract*— Unified Modified Language (UML) is a modeling language standard for identifying, recording and designing a software. Reusing UML diagrams can help to accelerate the software development process. Reusing UML diagram requires a method of calculating similarities between artifacts in UML diagram. The proposed method contented of two elements of sequence diagrams, i.e. property of class and message sequence. The result of experiment showed that the property class and message sequence could be a suitable parameter in evaluating the UML sequence diagram similarity.**

Keywords— sequence diagram, measurement similarity method, UML similarity

I. INTRODUCTION

Unified Modified Language (UML) is a modeling language standard for identifying, recording and designing a software. The language is used by software engineers and software developers in developing software projects. UML helps designers to model interactions between systems and users, interaction between objects, object behavior, and implementation and logical structure of systems [1]–[3].

UML diagram development has a several problems. One problem that is often found when making UML is that it takes a long time if is it requires to make it from the beginning [4]. Reusing UML diagrams can help to accelerate the software development process. Further, reusing UML can reduce the costs and risk used [5].

Reusing UML diagram requires a method of calculating similarities between artifacts in UML diagram. The determination of similarity is an effort made in maximizing the reuse of UML diagrams. Previous research[6] has attempted to calculate the similarities between artifacts in class diagrams. Similarities in UML class diagrams are calculated from the structure of relationships between classes. Other research [2], [7], [8] do calculations on several UML diagrams.

Our contribution was proposed calculating similarity sequence diagrams using details of properties and messages. The class details in the sequence diagrams which will be calculated consisted of the class name, attributes and operations. Whereas the similarity calculated messages are source class name, method name and destination class name.

This paper proposed a method to measure similarity between two different UML sequence diagram. The method was adopted from Al-K & Ahmed [6], the result allows further reuse of software artifacts during the software development process. Thus, it enables software engineers to develop project not from scratch, but from an existing project of a similar design. The goal would be to improve efficiency within a software project. The proposed method also adopted the result of the previous research [2]. The method of measuring similarity sequence diagrams produced uses the Greedy Algebra. The previous research used simulated annealing. The results of calculation were better using simulated annealing, but it was more difficult to implement. The Greedy is simpler and this study is not to search for optimal values. This study focus on showing how to measure the similarity of two sequence diagrams.

The paper is organized as follows. The second section introduces the approach we use to measure similarity property of class and message sequence between two sequence diagrams. Then, the third section presents the test cases that we used in this study. The fourth section provides the results and analysis based on an experimentation. The last section provides the conclusion and further work

II. SIMILARITY MEASUREMENT METHOD

A. Diagram Prepocessing

Preprocessing was needed to measure similarity between 2 UML sequence diagrams. The UML sequence diagram was created by an open source UML modeling tool. The tool converts UML sequence diagram into XMI-format. XMI is supported to produce the metadata of sequence diagram.

This study provided a UML sequence diagram as an example, showed in Figure 1. The diagram preprocessing extract sequence diagram into a sequence diagram metadata. The metamodel are formed as a set of components [2]. The components can be classified as two sets, i.e. object and message. First, the object consists of a message collection. The message consists of the source object's name, source class's name, message's type, message's name, destination object's name, destination class's name. The difference from Daniel [2]] is the detail of the object. Daniel used the detail of class information as class name, attribute name, and operation name. This is more suitable for use in class diagrams. While in the sequence diagram only shows the name of the object

978-1-7281-0719-6/19/$31.00 ©2019 IEEE

and the collection of messages that pass through the object. Second, the message shows the information of message flow from the source object to the destination object. The difference from Daniel is to add additional information such as message types and object names. Message type consists of asynchCall, synchCall and reply.

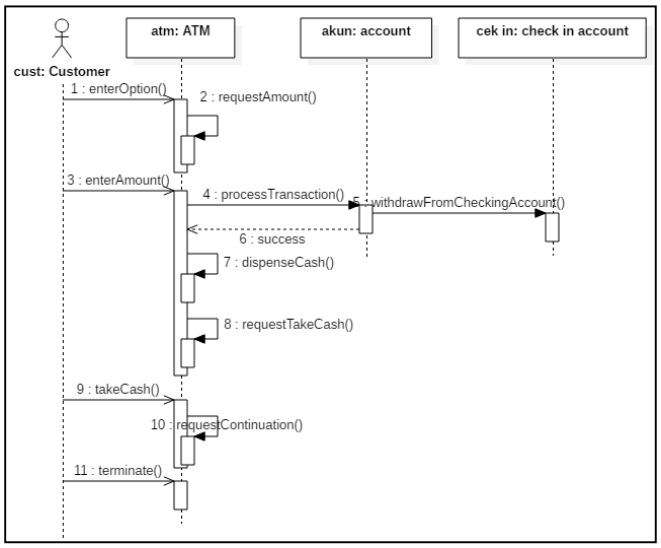

Fig. 1. UML Sequencial Diagram Example 1

Based on Figure 1, there are three objects and eleven messages. Metadata extracted from Figure 1 as following:

Object:

- → *atm: enterOption, requestAmount, enterAmount, success, dispenseCash, requestTakeCash, takeCash, requestContinuation, terminate*
- → *akun: processTransaction*
- → *cekIn: withdrawFromCheckingAccount*

Message:

- → *1: cust, Customer, asynchCall, enterOption, atm, ATM*
- → *2: atm, ATM, synchCall, requestAmount, atm, ATM*
- → *3: cust, Customer, asynchCall, enterAmount, atm, ATM*
- → *4: atm, ATM, synchCall, processTransaction, akun, account*
- → *5: akun, account, synchCall, withdrawFrom-CheckingAccount, cekIn, checkInAccount*
- → *6: akun, account, reply, success, atm, ATM*
- → *7: atm, ATM, synchCall, dispenseCash, atm, ATM*
- → *8: atm, ATM, synchCall, requestTakeCash, atm, ATM*
- → *9: cust, Customer, asynchCall, takeCash, atm, ATM*
- → *10: atm, ATM, synchCall, requestContinuation, atm, ATM*

11: cust, Customer, asynchCall, terminate, atm, ATM.

B. Measurement Similarity

The measurement similarity method used greedy algorithm for every metadata found. This paper also used the combination of cosine similarity, Wu Palmer method and Wordnet to find semantic similarity between lexical information. Wu Palmer method is a simple method and has high perfomance [9]. Several studies has used the Wu Palmer method to measure semantic similarity [10]–[12].

As previously informed, the similarity between 2 UML sequence diagrams can be measured based on their metadata. The metadata has 2 information, namely object information (*oSim*) and message information (m*Sim*). Equation 1 show how to measure the similarity between 2 UML sequence diagram (*seqSim*), i.e. d_1 and d_2.

$$seqSim(d_1, d_2) = w_o \times oSim(d_1, d_2) + w_m \times mSim(d_1, d_2) \quad (1)$$

where w_o is the weight of the object similarity from d_1 and d_2. And w_m is the weight of message similarity from d_1 and d_2. They are arbitrary weight. The next state is to measure the object similarity (*oSim*) from d_1 and d_2. This measurement is described in Equation 2.

$$oSim(d_1, d_2) = \frac{Max(\sum_{i,j=1}^{Max(|MN_1|,|MN_2|)} Co\,sin\,eSim(mn_i, mn_j))}{|MN_1|+|MN_2|} \quad (2)$$

where MN_1 and MN_2 are collections of message names belong to each object from the diagrams d_1 and d_2. Then to calculate the message similarity between two sequence diagrams (*mSim* (d_1, d_2)). How to calculate the message similarity in Equation 3.

$$mSim(d_1, d_2) = \frac{Max(\sum_{i,j=1}^{Max(|MSG_1|,|MSG_2|)} msgSim(m_i, m_j))}{|MSG_1|+|MSG_2|} \quad (3)$$

where MSG_1 and MSG_2 are the sequence of message invoked during the realization of use case as stated in sequence diagram d_1 and $d2$, respectively. Similarity of two messages, $msgSim(d_1, d_2)$, is the semantic similarity two messages as specified in Equation 4.

$$msgSim(msg_1, msg_2) = w_{osrc} \times cosineSim(osrc_1, osrc_2) + w_{csrc} \times cosineSim(csrc_1, csrc_2) + w_{mt} \times cosineSim(mt_1, mt_2) + w_{mn} \times cosineSim(mn_1, mn_2) + w_{odst} \times cosineSim(odst_1, odst_2) + w_{cdst} \times cosineSim(cdst_1, cdst_2) \quad (4)$$

where w_{osrc}, w_{csrc}, w_{mt}, w_{mn}, w_{odst} and w_{cdst} are arbitrary weight assign to object source (osrc), class source (csrc), method name (mn), object destination (odst), and class destination (cdst).The similarity message calculation is done by paying attention to the message type. If the message flow from the synchronous message type to the asynchronous message type, the similarity value is 0.8 and vice versa. However, the message flow from the synchronous or asynchronous message type, the similarity value is 0.2 and vice versa.

III. EMPIRICAL RESULT AND ANALYSIS

The purpose of this study is to present parameters and how to calculate the similarity between two UML sequence diagrams. We measured the similarity of sequence diagram pairs from six sequence diagram that has two domain, i.e. automatic teller machine and login. SQ1 to SQ3 is a sequence diagram of the automatic teller machine domain, whereas SQ4 to SQ6 are domain logins, showed in Table 1. Each sequence diagram provides a number of objects and messages.

TABLE 1. SEQUENCE DIAGRAM

Code	Sequence Diagram	Number of Objects	Number of Messages
SQ1	Withdraw money	4	11
SQ2	Withdraw money	4	13
SQ3	Withdraw money	3	12
SQ4	Login	4	6
SQ5	Login	4	7
SQ6	Login	4	7

We provided the result of calculation between the diagrams in Figure 1 (SQ1) and diagram in Figure 2 (SQ2). The similarity values of SQ1 and SQ2 were calculated based on object and message similarities.

SQ1 has tree lifeline, i.e. ATM (o1_1), akun (o1_2), and cek in (o1_3). SQ2 also has tree lifeline, i.e ATM (o2_1), akun (o2_2), and cek in (o2_3). The message of sequence diagrams also has been calculated Table 2 the result of calculating the message similarity between message in SQ1 And SQ2 using equation 4. The weight of w_{osrc}, w_{csrc}, w_{mt}, w_{mn}, w_{odst} and w_{cdst} were set experimentally to 0.125, 0.125, 0.2, 0.3, 0.125, and 0.125 respectively.

The result of two sequence diagram using equation 1 was 0.670. The weight in this study was 0.6 and 0.4 respectively. Based on this result, there is a similarity between two sequence diagram in the problem domain.

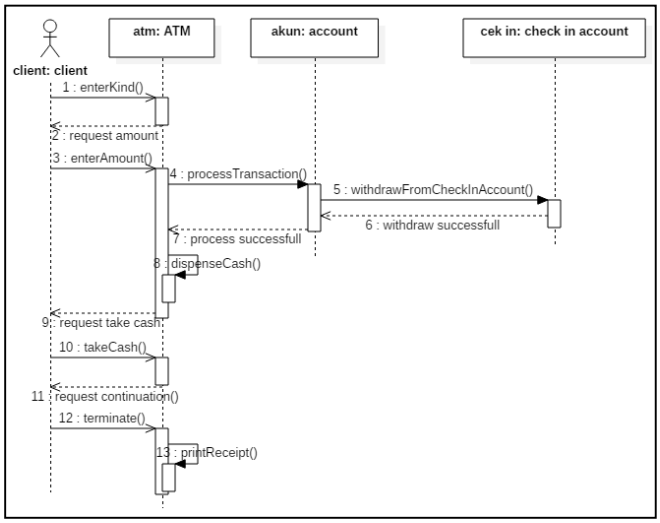

Fig. 2. UML Sequence Diagram Example

TABLE 2. OBJECT SIMILARITY BETWEEN SQD_1 DAN SQD_2

oSim	**o2_1**	**o2_2**	**o2_3**
o1_1	0.753	0.142	0.000
o1_2	0.213	0.707	0.000
o1_3	0.000	0.25	0.670

We also provided the deviation values between our proposed method with an assessment from the experts, showed in Table 3. This study used three experts to provide an assessment of similarity between two sequence diagrams. The third expert's assessment of the similarity between two sequence diagrams was then averaged.

The next step, we did some testing giving weight between properties and messages of two sequence diagrams.

TABLE 3. DEVIATION BETWEEN PROPOSED METHOD AND EXPERTS

Diagram Couples	Similarity		Average Deviation between Metode and Experts								
	Properties	Messages	(0.9,0.1)	(0.8,0.2)	(0.7,0.3)	(0.6,0.4)	(0.5,0.5)	(0.4,0.6)	(0.3,0.7)	(0.2,0.8)	(0.1,0.9)
SQ1 and SQ2	0.7105	0.7760	0.1062	0.0997	0.0931	0.0866	0.0801	0.0735	0.0670	0.0604	0.0539
SQ1 and SQ3	0.3191	0.6245	0.3770	0.3465	0.3160	0.2854	0.2549	0.2243	0.1938	0.1632	0.1327
SQ1 and SQ4	0.0000	0.2695	0.1897	0.1628	0.1358	0.1089	0.0819	0.0549	0.0280	0.0010	0.0259
SQ1 and SQ5	0.1493	0.2899	0.0734	0.0874	0.1015	0.1156	0.1296	0.1437	0.1578	0.1718	0.1859
SQ1 and SQ6	0.0000	0.2852	0.0215	0.0070	0.0356	0.0641	0.0926	0.1211	0.1496	0.1781	0.2067
SQ2 and SQ3	0.2952	0.7100	0.4100	0.3685	0.3270	0.2856	0.2441	0.2026	0.1611	0.1196	0.0781
SQ2 and SQ4	0.0861	0.2972	0.0395	0.0184	0.0027	0.0239	0.0450	0.0661	0.0872	0.1083	0.1294
SQ2 and SQ5	0.2216	0.3654	0.0040	0.0103	0.0247	0.0391	0.0535	0.0679	0.0822	0.0966	0.1110
SQ2 and SQ6	0.0000	0.3446	0.2122	0.1777	0.1433	0.1088	0.0744	0.0399	0.0054	0.0290	0.0635
SQ3 and SQ4	0.0783	0.3289	0.2233	0.1982	0.1732	0.1481	0.1231	0.0980	0.0730	0.0479	0.0229
SQ3 and SQ5	0.0471	0.4278	0.1115	0.0734	0.0353	0.0027	0.0408	0.0789	0.1169	0.1550	0.1931
SQ3 and SQ6	0.0000	0.4433	0.1290	0.0847	0.0403	0.0040	0.0483	0.0926	0.1370	0.1813	0.2256
SQ4 and SQ5	0.4458	0.5192	0.3469	0.3395	0.3322	0.3248	0.3175	0.3102	0.3028	0.2955	0.2882
SQ4 and SQ6	0.3184	0.5752	0.4692	0.4435	0.4179	0.3922	0.3665	0.3408	0.3152	0.2895	0.2638
SQ5 and SQ6	0.1643	0.6095	0.6345	0.5900	0.5455	0.5010	0.4564	0.4119	0.3674	0.3229	0.2783
Average			0.223	0.201	0.182	0.166	0.161	0.155	0.150	**0.148**	0.151

Calculation of gaps was done between expert assessments of the results of similarity calculations based on the proposed model. The next step is calculating the average gap value by testing several compositions of weight values. The smallest gap value indicates that the proposed model has been valid and can be used for further research, showed in Table 3.

IV. Conclusion

This study presented a method for measuring similarity between two sequence diagrams. The algorithm was adopted from the previous research, i.e. the greedy and cosine approaches. The proposed method contented of two elements of sequence diagrams, i.e. property of class and message sequence. This study also provided a former experimentation of the proposed method on a sequence diagrams of the same problem domain. The result of experiment showed that the property class and message sequence could be a suitable parameter in evaluating the UML sequence diagram similarity.

Further study should be carried out in order to answer several research questions. First, what would be the best weight setting to ensure the accuracy of similarity measurement of each element and parameters. Second, how well the proposed method measures the similarity between UML sequence diagrams from the problem domain. Third, an additional preprocessing method from NLP, such as deleting stopword, which will eliminate irrelevant words or find stemming words for normalization word.

Acknowledgment

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

References

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

[2] 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.

[3] 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, 2012.

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

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

[6] M. A.-R. M. Al-Khiaty and M. Ahmed, "Similarity assessment of UML class diagrams using simulated annealing," *Softw. Eng. Serv. ...*, pp. 19–23, 2014.

[7] R. Dijkman, M. Dumas, B. Van Dongen, R. Krik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Inf. Syst.*, vol. 36, no. 2, pp. 498–516, 2011.

[8] X. Yue, G. Di, Y. Yu, W. Wang, and H. Shi, "Analysis of the combination of natural language processing and search engine technology," *Procedia Eng.*, vol. 29, pp. 1636–1639, 2012.

[9] D. Guessoum, M. Miraoui, and C. Tadj, "A modification of Wu and Palmer Semantic Similarity Measure," *UBICOMM 2016 Tenth Int. Conf. Mob. Ubiquitous Comput. Syst. Serv. Technol.*, no. October 2016, pp. 41–46, 2016.

[10] C. Nuntawong, C. S. Namahoot, and M. Brückner, "A Semantic Similarity Assessment Tool for Computer Science Subjects Using Extended Wu & Palmer's Algorithm and Ontology," Springer, Berlin, Heidelberg, 2015, pp. 989–996.

[11] V. Sowmya, B. V. Vardhan, and M. S. V. S. B. Raju, "Influence of Token Similarity Measures for Semantic Textual Similarity," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 41–44.

[12] P. Sravanthi and D. B. Srinivasu, "SEMANTIC SIMILARITY BETWEEN SENTENCES," *Int. Res. J. Eng. Technol.*, 2017.