Received: November 11, 2020. Revised: December 10, 2020.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Different Approach on Automated Use Case Diagram Semantic Assessment

Reza Fauzan[1,2] Daniel Siahaan[1]* Siti Rochimah[1] Evi Triandini[3]

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
[2]*Department of Electrical Engineering, Politeknik Negeri Banjarmasin, Banjarmasin, Indonesia*
[3]*Department of Information System, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, Indonesia*
* Corresponding author's Email: daniel@if.its.ac.id

Abstract: The use case diagram is one of the diagrams commonly taught in colleges of computer science. Assessment of use case diagrams is often an obstacle for a teacher in the learning process. It is due to the interpersonal and intrapersonal problems of the teacher in assessing. Interpersonal problems are caused by the absence of an assessment standard among teachers. Intrapersonal problems are caused by the inconsistency of a teacher in assessing many diagrams of student answers. This research aims to create a semantic use case diagram automatic assessment method. Semantic assessment is divided into two kinds, namely property and relationship. All information used is a label translated from the XMI document. Similarity assessment between labels used cosine similarity, employing WuPalmer to perform WordNet searches. The results showed that the proposed method had a substantial agreement with the teacher as an expert; however, a teacher tends to look at property information rather than relationship information to assess use case diagrams.

Keywords: Automated assessment, Semantic assessment, UML similarity, Use case assessment, Use case diagram.

1. Introduction

A use case diagram is a behavior diagram in the Unified Modeling Language (UML) [1, 2]. The use case diagram describes the functional requirements of the software. Use case diagrams can be used to understand how the system should work. Therefore, the use case diagram is one of the diagrams taught in the computer science departments of various universities.

Assessment is one of the fundamental processes in teaching and learning activities [3, 4]. In this case, a teacher must carry out an assessment process in teaching use case diagrams. A teacher will assess the use case diagrams made by students based on the answer keys that have been made. The use case diagram that students build could be different from the answer key diagram. Student use case diagrams will be influenced by how well they understand the software and their vocabulary. Therefore, it can be difficult for the teacher to assess student diagrams.

The teacher's difficulty in assessing can be caused by two things, namely, interpersonal and intrapersonal problems [5]. Interpersonal problems arise when different students are assessed by different teachers, who apply different standards in assessing. The intrapersonal problem arises from the inconsistency of a teacher in assessing. It can happen any time the number of answers assessed is more than one. Internal factors such as fatigue can also play a part in assessing a teacher.

Automatic assessment is a solution for dealing with interpersonal and intrapersonal problems. The way to assess use case diagrams is to measure the similarities between the student and answer key diagrams. Apart from use case diagrams, there are several other UML diagrams which can be measured in a similar way, such as class diagram [6, 7], activity diagram [8], and sequence diagram [9, 10].

Several previous studies have tried to measure the similarity of two use case diagrams for various purposes. First, Storrle [11] measures the similarity of use case diagrams based on labels syntactically.

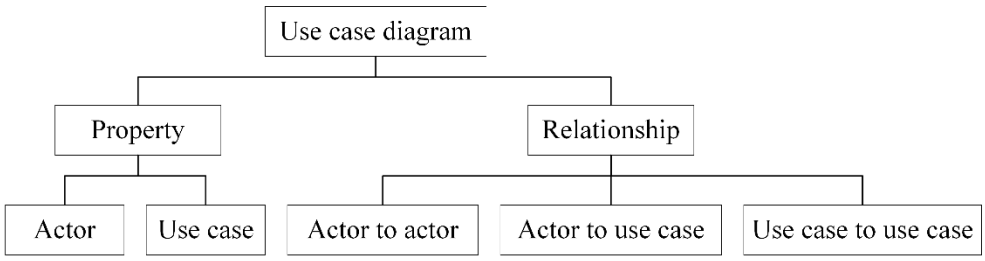

Figure. 1 Use case diagram component

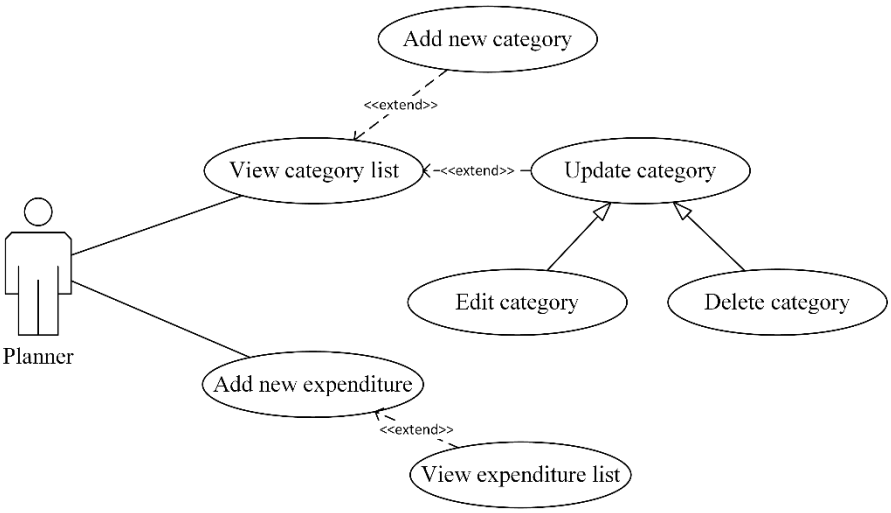

Figure. 2 Example of use case diagram (Outlay)

The label is obtained from the XMI (XML Metadata Interchange) document extracted from the use case diagram. The goal is to check plagiarism against design software. This measurement is difficult to implement because the student's answer may be correct even though it is not the same as the answer key. Second, Bonilla [12] take measurements between labels on the use case diagram semantically. They only compare the sub-components of the use case diagram. They do not compare all the components of the use case diagram. They use similarity measurements to reuse design software. Third, Vachharajani [13] measures the similarity of labels between the two use case diagrams as a whole. He auto-corrects students' answers based on the semantic answer key diagram. However, this still assumes that all components of the assessment are the same. Meanwhile, teachers may have other ways or points of view in assessing. Fourth, Fauzan [14] measures the semantic similarity between the two use case diagrams by dividing the two assessment components: property and relationships. Property consists of a collection of actor labels and use cases labels. Relationship consists of label relationships between the components of the use case diagram. Based on the previous explanation, the problem in the use case diagram assessment is that there is no necessary agreement among teachers about the assessment standards, semantically. Therefore, it is necessary to define a standard for assessing semantic use case diagrams.

This research proposes a different approach in assessing use case diagrams. The approach taken is to apply previous research by dividing diagram components into property and relationship. This research aims to use the weight of the two components to establish their levels of importance. The weight or level of importance is obtained from the results of the proposed method's assessment that has the highest agreement value with the teachers. The weights found can become the standard of semantic assessment for future use case diagrams.

The rest of this article consists of several sections: Section 2 presents the diagram translation. Section 3 presents the semantic similarity. Section 4 presents the semantic assessment. Section 5 presents the result and evaluation. Section 6 presents our discussion about the evaluation and findings. Section 7 presents the conclusion of our research.

2. Diagram translation

Before the use case diagram is assessed, we must translate the use case diagram into XMI. The label information required can be found on XMI. The information used to assess is in Fig. 1. The headline consists of properties and relationships. Property

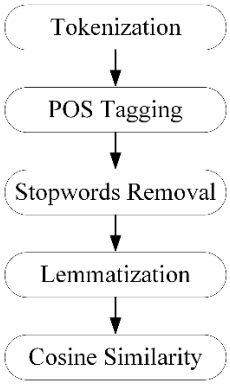

Figure. 3 Flowchart of semantic similarity

information consists of a collection of labels from actors and use cases. Information relations consist of the actor to actor (*AtoA*) relationship, actor to use case (*AtoU*) relationship, and use case to use case (*UtoU*) relationship. AtoA relationships can be in the form of generalizations of relations. AtoU relationships show the main use case from the use case diagram. UtoU relationships denote extended, included, or generalized relations.

Fig. 2 is an example of a use case diagram from the Outlay project. The Outlay project is an application for recording financial expenses. The Outlay has an actor and seven use cases. The following is a detailed description of the extraction results of the Outlay XMI use case diagram.

Property Information:

→ *actor1: planner*

→ *usecase1: view category list*

→ *usecase2: add new expenditure*

→ *usecase3: add new category*

→ *usecase4: update category*

→ *usecase5: view expenditure list*

→ *usecase6: edit category*

→ *usecase7: delete category*

Relationship Information:

→ *rel1: (actor: planner, uc: view category list)*

→ *rel2: (actor: planner, uc: add new expenditure)*

→ *rel3: (uc: view category list, uc: add new category, rel: extend)*

→ *rel4: (uc: view category list, uc: update category, rel: extend)*

→ *rel5: (uc: add new expenditure, uc: view expenditure list, rel: extend)*

→ *rel6: (uc: update category, uc: edit category, rel: generalization)*

→ *rel7: (uc: update category, uc: delete category, rel: generalization)*

3. Semantic similarity

Semantic similarity is obtained by calculating the similarity between the two labels in the use case diagram. Labels from use cases are mostly in the form of sentence fragments consisting of several words. Therefore, semantic similarities require a unique flow in their calculations. The flow of calculation for semantic similarity can be seen in Fig. 3. The entered label will be broken down by word using tokenization. Then each word will be given Part of Speech (POS) tagging using Stanford NLP [15]. Then we remove stop words. After that, we return all words to their first form using lemmatization. Finally, we compare these results against other diagram labels, using cosine similarity [16] to find semantic meaning. In cosine similarity, there is a calculation of the similarity between two words. We compare the words based on the types of words that have been informed using POS tagging. For example, "book" in the verb form will not be compared to "book" in the noun form. Their similarities should be zero. If the counted word types are the same, the similarity calculation will be continued using WordNet [17,18]. We use WuPalmer [19,20] to search for similarities on WordNet.

4. Semantic assessment

Based on our previous research [14], similarity assessment is divided into two main parts, property and relationship. Fig. 4 shows the flow of semantic similarity assessment in this research. The use case diagram semantic similarity assessment (*ucdSem*) can be calculated based on property similarity (*propSim*) and relationship similarity (*relSim*). Eq. (1) shows how to calculate *ucdSem* between use case diagram d_1 as the answer key, and use case diagram d_2 as the student answer. The level of importance between *propSim* and *relSim* is differentiated by ρ_{sem}. The value of ρ_{sem} consists of 0, 0.1, 0.2 to 1.

$$ucdSem(d_1, d_2) = (1 - \rho_{sem}) \times propSim(d_1, d_2) + \rho_{sem} \times relSim(d_1, d_2) \quad (1)$$

where
d_1 : first use case diagram,
d_2 : second use case diagram,
ucdSem : use case diagram semantic assessment,

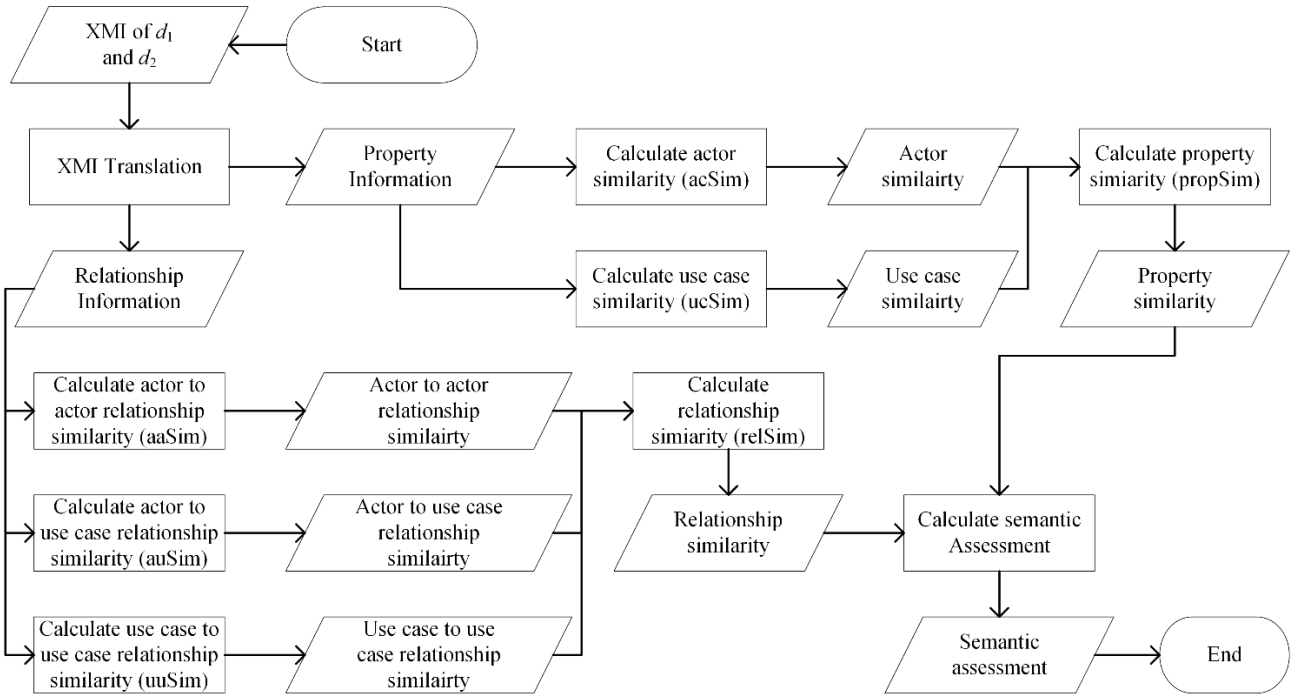

Figure. 4 Flowchart of semantic assessment

$$acSim(d_1, d_2) = \frac{2\times\left(\sum_{k=1}^{Min(|AC_1|,|AC_2|)} changePivot\left(Max(\sum_{i=1}^{|AC_1|}\sum_{j=1}^{|AC_2|} cosineSim(ac_i,ac_j))\right)\right)}{|AC_1|+|AC_2|} \quad (3)$$

where
AC_1 : actor collection of d_1,
AC_2 : actor collection of d_2,
ac_i : the i-actor of AC_1, and
ac_j : the j-actor of AC_2.

$$ucSim(d_1, d_2) = \frac{2\times\left(\sum_{k=1}^{Min(|UC_1|,|UC_2|)} changePivot\left(Max(\sum_{i=1}^{|UC_1|}\sum_{j=1}^{|UC_2|} cosineSim(uc_i,uc_j))\right)\right)}{|UC_1|+|UC_2|} \quad (4)$$

where
UC_1 : use case collection of d_1,
UC_2 : use case collection of d_2,
uc_i : the i-use case of UC_1, and
uc_j : the j-use case of UC_2.

ρ_{sem} : distinguishing semantic component importance,
propSim : property similarity, and
relSim : relationship similarity.

Then, the assessment of property similarity between the two use case diagrams is shown in Eq. (2). The property similarity assessment (*propSim*) of d_1 and d_2 was derived from actor similarity (*acSim*) and use case similarity (*ucSim*). Weight w_a is the weight of actor similarity, obtained by dividing the number of actors owned by d_1 and d_2 with the total actors and use cases owned by d_1 and d_2. Weight w_u is the weight of the use case similarity. Weight w_u is obtained by dividing the number of use cases owned by d_1 and d_2 with the total actors and uses cases owned by d_1 and d_2.

$$propSim(d_1, d_2) = w_a \times acSim(d_1, d_2) + w_u \times ucSim(d_1, d_2) \quad (2)$$

where
w_a : weight of actor similarity,
w_u : weight of use case similarity,
acSim : actor similarity, and
ucSim : use case similarity.

Eq. (3) shows how to assess actor similarity between d_1 and d_2. All actors from d_1 (AC_1) will be assessed for semantic similarity to all actors from d_2 (AC_2). The assessment details will be compared between labels separately, using cosine similarity as shown in the previous section, where ac_i is the i-th actor from d_1 and ac_j is the j-th actor from d_2. From the many results of the inter-label assessment, we use the greedy algorithm [21] to get the overall results' optimal value. The critical part of the greedy algorithm is the *changePivot* algorithm. The *changePivot* is described in Algorithm 1.

The first line in Algorithm 1 takes the point or looks for the most significant value in the matrix. The second line changes all the quantities in line x to 0. The third line changes all the quantities in the segment y to 0.

Algorithm 1: *changePivot*

Input: two-dimension matrix and pivot/coordinate maximum value (x,y)

1. Select pivot
2. M(x, :) = 0
3. M(:, y) = 0

Output: changed matrix

Eq. (4) shows an assessment of the similarity of the use case set from d_1 (UC_1) to the use case set from d_2 (UC_2). The *ucSim* assessment also uses the greedy algorithm concept by implementing *changePivot*. A detailed assessment of the similarity between two use cases is calculated using the cosine similarity, where uc_i is the i-th use case of d_1 and uc_j is the j-th use case of d_2.

Based on Eq. (1), the semantic assessment consists of property and relationship similarity. The relationship similarity assessment (*relSim*) is shown in Eq. (5). Relationship assessment is divided into three types of relationships, as previously explained. Weight w_1 is the weight for assessing *AtoA* relationships (*aaSim*). The value of w_1 is the number of *AtoA* relationships from d_1 and d_2 divided by the total of all relationships from d_1 and d_2. Weight w_2 is the weight for the assessment of *AtoU* relationships (*auSim*). The value of w_2 is the number of *AtoU* relationships from d_1 and d_2 divided by all relations from d_1 and d_2. Weight w_3 is the weight for the assessment of *UtoU* relationships (*uuSim*). The value of w_3 is the number of *UtoU* relationships from d_1 and d_2 divided by the total of all relations from d_1 and d_2.

$$relSim(d_1, d_2) = \left(w_1 \times aaSim(d_1, d_2)\right) + \left(w_2 \times auSim(d_1, d_2)\right) + \left(w_3 \times uuSim(d_1, d_2)\right) \quad (5)$$

where

w_1 : weight for assessing actor to actor relationship,
w_2 : weight for assessing actor to use case relationship,
w_3 : weight for assessing use case to use case relationship,
aaSim: assessment of actor to actor relationship,
auSim: assessment of actor to use case relationship, and
uuSim: assessment of use case to use case relationship.

$$aaSim(d_1, d_2) = \frac{2 \times \left(\sum_{k=1}^{Min(|AA_1|,|AA_2|)} changePivot\left(Max(\sum_{i=1}^{|AA_1|}\sum_{j=1}^{|AA_2|} daaSim(aa_i, aa_j))\right)\right)}{|AA_1| + |AA_2|} \quad (6)$$

where

AA_1 : actor to actor relationship collection of d_1,
AA_2 : actor to actor relationship collection of d_2,
aa_i : the i- actor to actor relationship of AA_1,
aa_j : the j- actor to actor relationship of AA_2, and
daaSim : detail assessment of actor to actor relationship.

$$daaSim(d_1, d_2) = \frac{cosineSim(srcA_1, srcA_2) + cosineSim(tgtA_1, tgtA_2)}{2} \quad (7)$$

where

$srcA_1$: source actor of a relationship in d_1,
$srcA_2$: source actor of a relationship in d_2,
$tgtA_1$: target actor of a relationship in d_1, and
$tgtA_2$: target actor of a relationship in d_2.

$$auSim(d_1, d_2) = \frac{2\times\left(\sum_{k=1}^{Min(|AU_1|,|AU_2|)} changePivot\left(Max(\sum_{i=1}^{|AU_1|}\sum_{j=1}^{|AU_2|} dauSim(au_i, au_j))\right)\right)}{|AU_1|+|AU_2|} \quad (8)$$

where
AU_1 : actor to use case relationship collection of d_1,
AU_2 : actor to use case relationship collection of d_2,
au_i : the i- actor to use case relationship of AU_1,
au_j : the j- actor to use case relationship of AU_2, and
dauSim : detail assessment of actor to use case relationship.

$$dauSim(d_1, d_2) = \frac{cosineSim(srcA_1, srcA_2)+cosineSim(tgtU_1, tgtU_2)}{2} \quad (9)$$

where
$tgtU_1$: target use case of a relationship in d_1, and
$tgtU_2$: target use case of a relationship in d_2.

Eq. (6) explains the assessment of the similarity of the *AtoA* relationships (*aaSim*). Since there can be more than one *AtoA* relationship in a diagram, the greedy algorithm is also needed to find optimal similarities amongst many results. AA_1 and AA_2 are a collection of the *AtoA* relationships from d_1 and d_2. Each aa_i will be assessed for its similarity to aa_j, where aa_i is the i-th relation of AA_1 and aa_j is the j-th relation of AA_2. The *AtoA* relationship has two components: the source actor (*srcA*) and the target actor (*tgtA*). The *AtoA* relationship cannot be directly calculated using cosine similarity. Therefore, a detailed assessment to assess between two *AtoA* relationships (*daaSim*) emerged, according to Eq. (7).

Eq. (8) explains the assessment of the similarity of the *AtoU* relationships (*auSim*). There could be more than one *AtoU* relationships in a diagram, so the greedy algorithm is also needed to find optimal similarities amongst many results. AU_1 and AU_2 are a collection of the *AtoU* relationships from d_1 and d_2. Each au_i will be assessed for its similarity to au_j, where au_i is the i-th relation of AU_1, and au_j is the j-th relation of AU_2. The *AtoU* relationship has two components: the source actor (*srcA*) and target use case (*tgtU*). The *AtoU* relationship also cannot be directly calculated using cosine similarity. Therefore, a detailed assessment to assess between two *AtoU* relationships (*dauSim*) emerged, according to Eq. (9).

Eq. (10) explains the assessment of the similarity of *UtoU* relationships (*uuSim*). Since there can be more than one *UtoU* relationships in a diagram, the greedy algorithm is also needed to find optimal similarities amongst many results. UU_1 and UU_2 are a collection of *UtoU* relationships from d_1 and d_2. Each uu_i will be assessed for its similarity to uu_j, where uu_i is the i-th relation of UU_1 and uu_j is the j-th relation of UU_2. The *UtoU* relationship has three components in it, namely the source use case (*srcU*), target use case (*tgtU*), and type of relation (*ty*) such as extend, include, or generalization. The *UtoU* relationship cannot be directly calculated using cosine similarity. Therefore, a detailed assessment to assess between two *UtoU* relationships (*duuSim*) emerged, according to Eq. (11).

5. Result

5.1 Dataset

The dataset was collected from three projects. An overview of each project can be seen in Table 1. The projects used are Outlay, QuickBill, and Restaurant Management System (RMS). The Outlay is a financial recording project. QuickBill is a point of sale project. RMS is a restaurant ordering project. The project is taken from GitHub in source code with an object-oriented approach. Each student is shown how the project is run. Then students are asked to make a use case diagram. Students create use case diagrams based on their understanding of the project and their vocabulary. The total student answers collected were 36 use case diagrams. In addition, each project has an answer key diagram. Therefore, in total, there are 39 use case diagrams.

5.2 Evaluation

Before we evaluated our proposed method, we created a gold standard as a reference for assessment. The gold standard is the average result of expert assessments of student answers based on the answer key. The expert number is twenty-one. The experts are lecturers from twelve different universities in Indonesia. Based on our observation, there is no evidence that demographics influence the assessment of UML objects. The experts questioned had software engineering educational backgrounds from several

Table 1. Overview of the collected use case diagram

Project	Answers	Actor Average	Use case Average	Relationship Average
Outlay	11	1	8	8
QuickBill	9	2	13	13
RMS	16	1	13	14

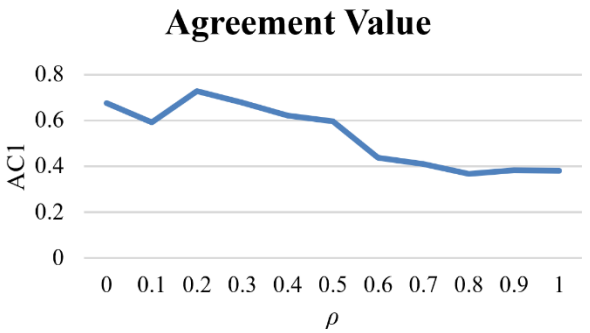

Figure. 5 Evaluation using Gwet's AC1

countries, namely Indonesia, Malaysia, Thailand, Netherlands, and Japan. We also made direct observations of expert respondents to get feedback and discussion related to the use case diagram being evaluated. Experts are lecturers who have taught and assessed use case diagrams in their lectures. The period of expert teaching use case diagrams is from one to more than twenty years. The minimum educational background for an expert is a master's degree in computer science. Seventeen experts hold master degrees in computer science and four hold doctorates in computer science. The ages of the experts varies between thirty to fifty years. Fourteen are men and seven women.

We did an evaluation using Gwets AC1 [22]. Gwet's AC1 can show the level of agreement between two experts. In this case, the first expert is the average of the experts' answers and the second expert is our proposed method. We convert the two experts' similarity values to 1 to 5, where a score of one represents an assessment result of less than twenty; two an assessment result of less than forty; three an assessment of less than sixty; four an assessment result of less than eighty; and five an assessment result is equal to or more than eighty. Fig. 5 shows the results of the gold standard agreement with the proposed method. We did some experiments by changing the value of ρ_{sem} in Eq. (1). The highest value was found at ρ_{sem} = 0.2, with an agreed value of 0.73.

6. Discussion

In building a gold standard, we test the validity and reliability of the data. For the validity test we used the Pearson coefficient [23]. The reliability test used the Cronbach Alpha coefficient. The number of diagrams used after the validity test was twenty-five student answer diagrams with a data reliability value of 0.959. Since this research aimed to establish a standardized assessment based on expert consensus, we also tested inter-rater reliability [22] amongst the experts. The average measure of interclass correlation is 0.929, based on nineteen experts.

The highest agreement value is 0.73 at ρ_{sem} = 0.2. According to Landis and Koch [24], this value indicates substantial agreement with the experts. Therefore, the proposed method can replace an expert in assessing use case diagrams. Based on Fig. 5, the best ρ_{sem} value is 0.2. It shows that experts tend to see labels from actors and use cases only, and still pay little attention to the label of relationship information. This value can become a standard of assessment so that differences in the standard of assessment between experts are resolved. However, this research is still limited to semantic assessment. This research ignores the overall structural aspects of the use case diagram.

Based on the best agreement values found, we compared our proposed method with each expert's agreement value against the gold standard. Fig. 6 indicates that each expert's average agreement value on the gold standard is 0.55. Only three of the nineteen experts had an agreement value higher than or equal to our proposed method. It shows that the majority of experts have inconsistencies in assessing. Inconsistencies may occur due to expert fatigue, which increases along with the number of assessed answers. Therefore, our automatic assessment can be better than experts because it can maintain consistency in the assessment.

We compared the previous researches [11,13] and our proposed method with ρ_{sem} = 0.2. We compared the gap against the gold standard for each student's answer. Three previous studies measured the similarity between the two use case diagrams. However, only two studies have measured the overall similarity between the two use case diagrams. They are Storrle [11] and Vachharajani [13]. Storrle [11] measures lexical similarity syntactically. Bonilla [12] measures only a portion of the use case diagram. Vachharajani [13] assessed student answer diagrams based on labels on the answer keys using label matching. If there is a label on the answer diagram similar to the label on the answer key diagram, then the answer is correct. The maximum value applied is if all the answer key labels are similar to the labels on the answer diagram and ignore different actors and use cases in the answer diagram. Fig. 7 shows the results of the gap comparisons made. Our proposed method shows a smaller gap than label matching in

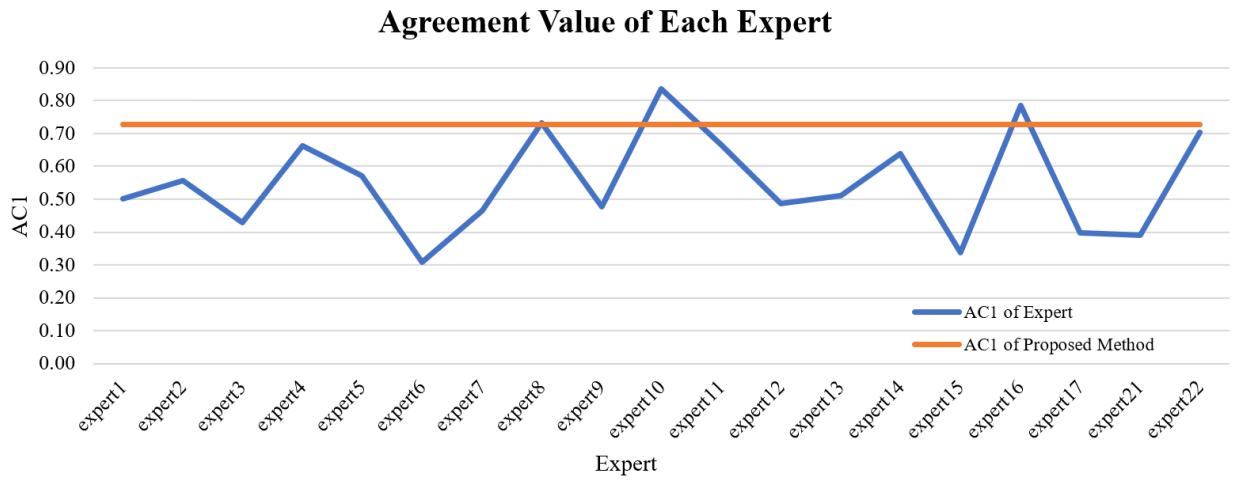

Figure. 6 Agreement value of each expert

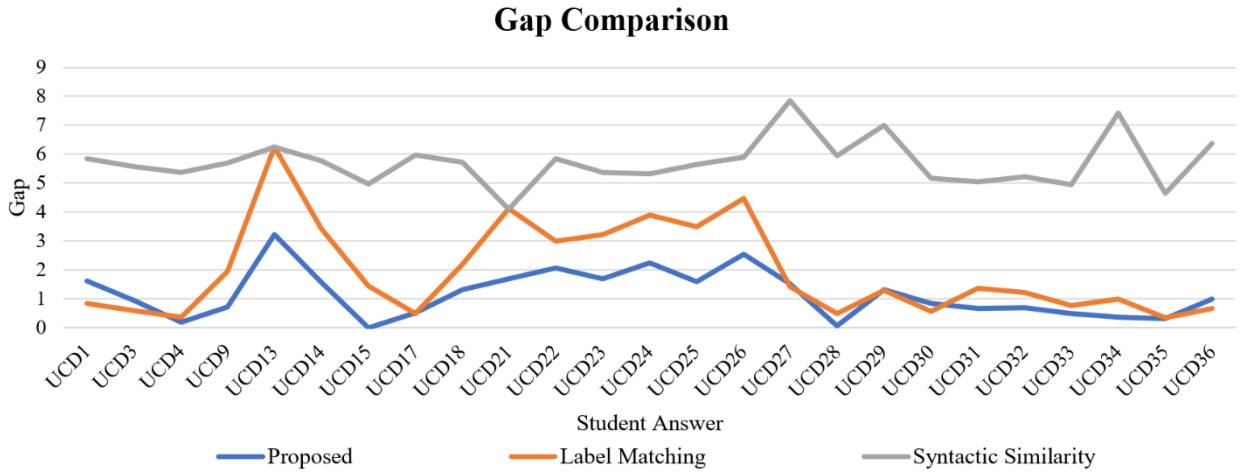

Figure. 7 Comparison with previous research

eighteen out of the twenty-five use case diagram assessments. Our proposed method also shows a smaller gap than syntactic similarity in use case diagram assessments. The average gap in our proposed method is 1.16 on a scale of 0 to 10. It is smaller than the label matching and syntactic similarity methods, which have average gaps of 1.95 and 5.72, respectively. It also shows that both semantic assessments are better than syntactic assessments.

7. Conclusion

Different approaches to semantic assessment use case diagrams can become a harmonious standard of assessment. The proposed method has a substantial agreement with experts, with a value of 0.73. In the appraisal process, experts recognize more readily the similarity of properties than the similarity of the relationship between two use case diagrams. With this automatic assessment, assessment consistency, which is a problem for experts in assessing, can be resolved.

However, this research is still limited to an assessment of the use case semantic diagram. As we know, a diagram consists of labels and structures. The structural assessment aspect may also influence the use case diagram assessment. Therefore, in the future, we can try to assess the structural use case diagram.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Reza Fauzan and Daniel Siahaan; methodology, Reza Fauzan and Daniel Siahaan; software, Reza Fauzan; validation, Reza Fauzan, Daniel Siahaan, and Siti Rochimah; formal analysis, Reza Fauzan, Daniel Siahaan, and Siti Rochimah; investigation, Evi Triandini; resources, Evi Triandini; data curation, Siti Rochimah; writing—original draft preparation, Reza Fauzan;

writing—review and editing, Daniel Siahaan and Siti Rochimah; visualization, Reza Fauzan and Evi Triandini; supervision, Daniel Siahaan and Siti Rochimah; project administration, Daniel Siahaan.

Acknowledgments

This research was funded by the Ministry of Research and Technology/National Research and Innovation Agency of the Republic of Indonesia. This research is a collaboration amongst Institut Teknologi Sepuluh Nopember, Politeknik Negeri Banjarmasin, and Institut Teknologi dan Bisnis STIKOM Bali.

References

[1] R. S. Pressman, *Software Engineering A Practitioner's Approach 7th Ed*, Palgrave macmillan, 2009.

[2] M. J. Chonoles, OCUP 2 *Certification Guide: Preparing for the OMG Certified UML 2.5 Professional 2 Foundation Exam*, Morgan Kaufmann, pp. 17–41, 2017.

[3] F. Restrepo-Calle, J. J. Ramírez Echeverry, and F. A. González, "Continuous assessment in a computer programming course supported by a software tool", *Computer Applications in Engineering Education*, Vol. 27, No. 1, pp. 80–89, 2019.

[4] J. P. Leighton, "Students' Interpretation of Formative Assessment Feedback: Three Claims for Why We Know So Little About Something So Important", *Journal of Educational Measurement*, Vol. 56, No. 4, pp. 793–814, 2019.

[5] R. Anderson, M. Thier, and C. Pitts, "Interpersonal and intrapersonal skill assessment alternatives: Self-reports, situational-judgment tests, and discrete-choice experiments", *Learning and Individual Differences*, Vol. 53, pp. 47–60, 2017.

[6] R. Fauzan, D. O. Siahaan, S. Rochimah, and E. Triandini, "Class Diagram Similarity Measurement: A Different Approach", In: *Proc. of International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 215–219, 2018.

[7] M. A. R. Al-Khiaty and M. Ahmed, "Matching UML class diagrams using a Hybridized Greedy-Genetic algorithm", In: *Proc. of International Scientific and Technical Conf. on Computer Sciences and Information Technologies*, pp. 161–166, 2017.

[8] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "Activity diagram similarity measurement: A different approach", In: *Proc. of International Seminar on Research of Information Technology and Intelligent Systems*, pp. 601–605, 2018.

[9] E. Triandini, R. Fauzan, D. O. Siahaan, and S. Rochimah, "Sequence Diagram Similarity Measurement: A Different Approach", In: *Proc. of International Joint Conf. on Computer Science and Software Engineering (JCSSE)*, pp. 348–351, 2019.

[10] A. Adamu and W. M. N. W. Zainon, "Similarity Assessment of UML Sequence Diagrams Using Dynamic Programming", In: *Proc. of International Visual Informatics Conf.*, pp. 270–278, 2017.

[11] H. Storrle, "Towards Clone Detection in UML Domain Models", *Software and Systems Modeling*, Vol. 12, No. 2, pp. 307–329, 2013.

[12] B. Bonilla-morales, S. Crespo, and C. Clunie, "Reuse of Use Cases Diagrams: An Approach based on Ontologies and Semantic Web Technologies", *International Journal of Computer Science Issues*, Vol. 9, No. 1, pp. 24–29, 2012.

[13] V. Vachharajani and J. Pareek, "Framework to approximate label matching for automatic assessment of use-case diagram", International Journal of Distance Education Technologies, Vol. 17, No. 3, pp. 75–95, 2019.

[14] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "Use case diagram similarity measurement: A new approach", In: *Proc. of International Conf. on Information and Communication Technology and Systems*, pp. 3–7, 2019.

[15] C. D. Manning, J. Bauer, J. Finkel, and S. J. Bethard, "The Stanford CoreNLP Natural Language Processing Toolkit", In: *Proc. of Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.

[16] P. Sunilkumar and A. P. Shaji, "A Survey on Semantic Similarity", In: *Proc. of International Conference on Advances in Computing, Communication and Control*, pp. 1–8, 2019.

[17] G. Majumder, P. Pakray, A. Gelbukh, and D. Pinto, "Semantic textual similarity methods, tools, and applications: A survey", Computación y Sistemas, Vol. 20, No. 4, pp. 647–665, 2016.

[18] A. Kutuzov, A. Panchenko, S. Kohail, M. Dorgham, O. Oliynyk, and C. Biemann, "Learning Graph Embeddings from WordNet-based Similarity Measures", *arVix*, Vol. abs/1808.0, 2019.

[19] Z. Wu and M. Palmer, "Verb Semantics and Lexical Selection", In: *Proc. of the 32nd annual*

meeting on Association for Computational Linguistics, pp. 133–138, 1994.

[20] S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain, "Semantic Similarity from Natural Language and Ontology Analysis", *Synthesis Lectures on Human Language Technologies*, Vol. 8, No. 1, 2015.

[21] M. A.-R. M. Al-Khiaty and M. Ahmed, "Similarity Assessment of UML Class Diagrams using a Greedy Algorithm", In: *Proc. of International Computer Science and Engineering Conf.*, pp. 19–23, 2014.

[22] S. Zepeda and A. Jimenez, "Teacher Observation and Reliability: Additional Insights Gathered from Inter-rater Reliability Analyses", *Journal of Educational Supervision*, Vol. 2, No. 2, pp. 11–26, 2019.

[23] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Pearson Correlation Coefficient in Noise reduction in speech processing*, Vol. 2. Springer, Berlin, Heidelberg, 2009.

[24] J. R. R. Landis and G. G. Koch, "The Measurement of Observer Agreement for Categorical Data", Biometrics, Vol. 33, No. 1, p. 159, 1977.