A Proposed Architecture for Automated Assessment of Use Case Diagrams

Vinay Vachharajani
School of Computer Studies
Ahmedabad University
Ahmedabad, Gujarat, India

Jyoti Pareek, Ph.D.
Department of Computer Science
Gujarat University
Ahmedabad, Gujarat, India

ABSTRACT
Establishment of institutions of higher learning requires massive amount of different resources which are always in short supply. The delivery of learning material and tests to the students has become very easy with the facility of uploading the same on the web irrespective of the number of students. The assessment part could be a deterrent as far as willingness of learned faculty members to participate in the whole process is concerned. If assessment will become automated then it will be easier for any teachers to evaluate any number of students.

This paper presents a proposed architecture of automated assessment of Use – Case Diagram. The essence of this architecture is to assess large number of students very easily in short duration. This proposed work is going to be very useful for the needy students by assisting in evaluation of their performance.

General Terms
E -Learning, Syntax Analysis, Natural Language Processing, UML Diagram, Diagram, Text Processing

Keywords
Use - Case Diagram, e – Assessment, Label Matching, Structure Matching, Automated Assessment, Graph Generator, Diagram Assessment, Computer Assisted Assessment.

1. INTRODUCTION
We are living in the age of information society and majority of the work force consists of knowledge workers. Higher education has become an absolute necessity to succeed in the information age and hence there is a huge requirement of institutions of higher learning.

Establishment of institutions of higher learning requires massive amount of different resources which are always in short supply. This has created a huge gap between demand and supply of such institutions which has resulted in a large number of students not being able to fulfill their dreams of quality higher education.

With the invention of internet, web based learning has become a reality. The learning process has three major components

- Delivery of learning material to students
- Conducting tests/quizzes/ examinations
- Evaluating the answers

The delivery of learning material and tests/quizzes to the students has become very easy with the facility of uploading the same on the web irrespective of the number of students. The assessment part could be a deterrent as far as willingness of learned faculty members to participate in the whole process is concerned. The evaluation may not be such a big problem in certain situations like online objective exams or subjective evaluation of textual answers. The evaluation will be automatic in online objective examinations and it can be automated for textual answers with the help of keywords expected in the answers. The assessment becomes really difficult in case of diagrammatic answers and this could result into shortage of teachers for some of the subjects.

These days Object Oriented Frameworks are an accepted technology within the software industry. More and more people are writing use cases, for behavioral requirements, for software systems or to describe business processes [38]. There is a growing trend in diagram interpretation [33, 34, 35] and automated grading of diagrams like Use Case diagrams, Class Diagrams, Sequence Diagrams, Activity Diagrams, Collaboration Diagrams and E-R Diagrams. Keeping this in mind it is proposed to investigate the automated assessment of diagrammatic answers- particularly Use Case diagrams.

In the remainder of this paper, section 2 discusses the work done in this area, section 3 gives the outline of the system and presents the architecture, section 4 explains the benefits of the proposed architecture and in the final section of the paper we find its conclusion.

2. RELATED WORK
After an extensive survey of research papers of this area it is found that a very limited work has been done in the area of automated assessment of precise or imprecise diagrams. Some work in the area of automated assessment of E-R diagrams has been done but the scope of that work is very limited or it is under process. The area "Automated assessment of Use case diagrams or any UML diagrams" is more or less untouched.

The advances of theories and technologies in "Virtual Classroom" [1] have populated the delivery of online courses. Yi – fang Brook Wu and Xin Chen [2] reported in their paper about assessing distance learning student's performance using natural language processing approach to analyze class discussion messages. In that, they assess student's performance by considering different aspects. They derived three measurements: Keyword Density (KD), Message Length (ML) and Message Count (MC) from the online messages. Then the Performance Indicator (PI) is derived from these three measures.

Waugh, Thomas and Smith [3] have presented architecture for representation and interpretation of imprecise diagrams. They have discussed an assessment tool, developed within this architecture, for automatically grading answers to a computer architecture examination question on Entity – Relationship

Diagrams (ERDs). Smith, Thomas and Waugh have also proposed a five – stage architecture consisting of segmentation, assimilation, identification, aggregation and interpretation for interpreting imprecise diagrams. They have implemented this approach in an automatic grading tool for answers to examination questions. This tool is based on Minimal Meaningful Units (MMUs) and interprets them to yield a meaningful result [4]. Anderson and Donlon have discussed in their paper about precise diagrams such as the use of diagrams in mathematical proofs [5] and visual query interfaces to GISs [6, 7].

In another paper, Thomas, Waugh and Smith [8] have described the architecture for evaluation of Entity – Relationship Diagram by five stages. They gave scenarios to students based on which they were asked to draw ER diagrams, generate various kinds of results by comparing their mean and standard deviation, and correlate them by comparing their results by using Pearson and Spearman's correlation coefficient. They developed a "revision tool" in which students are presented with a collection of typical assessment questions on the construction of ER – Diagrams. The tool contains a diagramming tool [9] with which students draw their answers. The tool then marks an answer and provides feedback in terms of a mark and a sequence of relationship diagrams. In addition, the tool also allows the student to view an interactive version of the specimen solution. That is, clicking on a specific part of the solution causes the tool to highlight those parts of the question which relate to the chosen part of the solution.

In the extension of the work [8] Thomas, Waugh and Smith explain how patterns in diagrams can contribute to the interpretation and automatic marking of E – R diagrams [10]. They concentrate on the aggregation stage in which MMUs are combined into higher level, abstract features. In this, they discuss about different patterns and how a pattern can be viewed as a diagram with some of its details omitted. They stored reusable diagrams in a library. In one of their papers [11] they report development of an automated marking tool and the results of two large scale experiments with the tool. They gave an overview and the results of some initial, informal, evaluation of a student revision tool that they developed.

The above mentioned work has been done in the area of E-R diagrams but the scope is very limited.

The paper [20] illustrated how diagram based questions are incorporated into normal Moodle quizzes and the nature of the feedback provided. It describes the type of graph-based diagrams that can be supported by the system and outlines how the automatic marking is performed. They have reported [21, 22] that they are working on some of the NLP techniques like synonyms, hyphenation, stemming, punctuation and so forth.

Noraida, Zarinaand Sufian[12] discussed the notation extraction process from Rational Rose petal file that represents the structure for each notation of UML class diagram as a text form. They took two inputs from the user, one is the input .mdl file from student as a solution for the coursework given and the input .mdl file from instructors as a solution model for that coursework. These files were traced line by line to match each notation keyword defined. Output generated from this process will keep notation details in different tables. D. Kumar and Ratna Sanyal proposed a tool named Static UML Model Generator from Analysis of Requirements (SUGAR) [13], which generates both use-case and class diagrams by emphasizing on natural language requirements. They integrate requirement analysis and design phase by identifying use-cases, actors, classes along with its attributes and methods with proper association among classes. They discussed about the tool which generates all static UML models in JAVA in conjunction with Rational Rose.

D. Kumar describes a domain independent tool, named UML Model Generator from Analysis of Requirements (UMGAR) [14] which generates UML models like the Use Case diagram, Analysis class model, collaboration diagram and design class model from natural language requirements using efficient NLP tools. This tool implements a set of syntactic reconstruction rules to process complex requirements into simple requirements. It also provides a generic XMI parser to generate XMI files for visualizing the generated models in any UML modeling tool. In the thesis [15], authors discussed and compared how scenarios, scripts and use cases are used in different methods used to capture the functionality of a system. They developed a tool which serves as an aid in administrating the use case models and generating the object model in a project. Computer Automated Use Case Diagram Generator (CAUse) [16] is a system that allows users to automatically generate use case diagrams from English document specifications that can be used for requirements management for business applications. In this paper, the concept of NLP is being used through which the input specifications document is processed to derive potential actors, use cases, and relationships, thereby generating a use case diagram. The Use Case Driven Development Assistant (UCDA) tool [17] provides automated assistance in developing use case diagrams, writing use case specification documents and developing the analysis class models. The paper[17] introduces the UCDA tool and its application in OMCP (Object Model Creation Process). The process of automation is illustrated in a case study of an ATM System. In the paper [18], authors discussed the proposed methodology which forms the basis of the automated process designed to capture the high level system services and actors from the textual user requirements. The authors illustrated the applicability of methodology on an order invoicing case study and demonstrated it with a prototype tool. In the paper [19], authors presented an approach supported by a tool for use cases based requirements engineering.

This approach includes use case formalization, a restricted form of natural language for use cases description, and the derivation of an executable specification as well as a simulation environment from use cases.

There has been substantial previous research work in the area of syntactical and semantic matching of labels of student diagrams and model diagram.

The University of Teesside Automated Student Diagram Assessment System [33] uses manual intervention by students to manage labels. Each student is presented with a list of labels within their diagram and the model solution and requested to select which labels of their own match the labels of the model solution [33]. Jayal and Shepperd [23, 24] proposed and evaluated a new method of label matching to support e-assessment of diagrams and addressed problems of synonyms, spelling errors and differing levels of decomposition. They have mainly focused and implemented the syntactic part of e -assessment system.

In the paper [25], author proposed an approach to designing an algorithm for approximate string matching in a dictionary. In this paper, Multiple spelling errors corresponding to insert,

delete, change and transpose operations on character strings are considered in the fault model and hashing techniques has been used to avoid comparing the given word with words at large distances. The Assess By Computer system [30] by the Manchester University uses the Edit Distance Algorithm [26] for matching the labels.

WordNet has been built to be the most systematic and as close to the human level and is being applied actively in various works [32]. In paper [31] the author explores the determination of semantic similarity by a number of information sources, which consist of structural semantic information from a lexical taxonomy and information content from a corpus.

In one of the approaches [35], a diagram is treated like a graph with vertices representing various boxes (entities) and edges representing the various connectors (relationship) between them. Marking proceeds by searching for **isomorphic graphs** or sub graphs with the correct diagram. This approach has been implemented in the ABC system developed by University of Manchester and is reported to have worked well [35]. One disadvantage of this system is that other information like syntactic and semantic similarity of labels (nodes) is not considered. In MADMatch [36], authors presented many-to-many Approximate Diagram Matching approach based on an Error-Tolerant Graph Matching (ETGM) formulation. They also considered lexical and structural information.

In the paper [41], automated assessment of class diagrams has been discussed. An architecture for automated assessment of class diagrams in which UML class diagrams are considered as graphs and rule based checks for class diagrams is being implemented using graph query language.

Thus, the work has been done in the area of generating automation tools for developing Use Case diagrams but efforts for automated assessment of Use Case diagrams have not been done so far.

3. PROPOSED ARCHITECTURE

Our research objective is to explore the possibility of automating the assessment of diagrams, mainly Use – Case diagrams. This can be achieved by

- Accepting and storing the Use case data, prepared by the teacher.
- Accepting and storing the Use case data, prepared by the students, in database in the format appropriate for its assessment.
- Automating the assessment of the Use – case diagrams thus stored in the database.

The proposed architecture for the same is given in figure – 1.

3.1 Accepting and Storing Data

For accepting and storing the Use – case data, following two approaches can be used:

- First approach is to use third party tool for creation and acceptance of Use-case diagram by the students. It has already been worked on this approach and developed a tool named Use Case Extractor [39]. It fetches the Use – case diagram created in StarUML tool by the user and successfully stores the fetched diagram in the database appropriately to be used further, for the automatic assessment. This has been done by fetching Use – case data through XML file and then storing these data in database.

- **Use Case Diagram Editor:**

Second approach is to develop a user interface for creating diagram and comparing it with model use-case diagram. For that, a Use- Case Diagram Editor **(UCDE)** can be developed in which the basic components, related to Use – Cases, like oval, actor, relationship like include, extend, generalization are given so that users (particularly students) will draw the Use – Case diagram of specific problem using those components. Every component is extracted and identified from the user's diagram and stored in database for assessment.

3.2 Automatic Assessment of Use – Case Diagram

The Use-case diagram drawn by an expert is considered as model diagram for our approach. The components of Use-case diagram drawn by students are compared with this model Use-case diagram. Here assessment does not mean only right or wrong answers but quantitative as well as qualitative feedback.

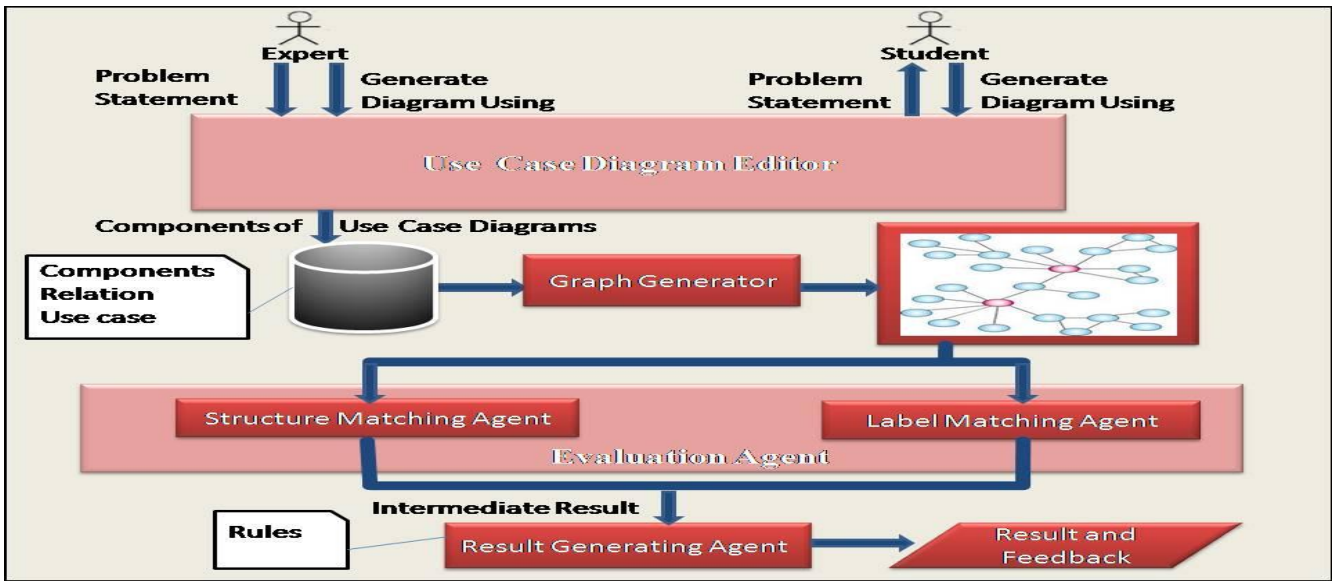

Fig – 1: Proposed architecture diagram for automatic assessment of use-case diagram

In our approach, a Use-case diagram is treated like a mixed graph with nodes representing various actors or use-cases and edges representing the various relationships between them. In one of the paper [35], a diagram is treated like a graph but the disadvantage of this approach is that they have not considered the syntactic and semantic perspective of the label. In our system, a hybrid approach will be used in which labels will be processed simultaneously in the syntactic and semantic matching stages.

This stage of architecture comprises of two sub stages.

(i) Graph generator

(ii) Evaluation Agent

3.2.1 Graph Generator

The assessment of Use-case diagram drawn by an expert and students is done by the graph isomorphism approach. The components of Use case diagrams drawn by an expert and a student is successfully stored in the database from which the graph of the components will be generated automatically by the graph generator.

3.2.2 Evaluation Agent

While working in automated assessment it is necessary to recognize both correctly and incorrectly drawn diagrams. Evaluating Use – case diagrams, automatically and correctly, is a challenging task as Use – cases have partial and sometimes inconsistent semantics.

The Evaluation Agent will work simultaneously on two stages.

(i) Label matching of student Use-case diagram with a model Use-case diagram

(ii) Structure matching of student Use-case diagram with a model Use-case diagram.

(i) Label Matching Agent:

The label matching of student Use case diagram with a model Use case diagram will be done by label matching agent. A diagram submitted by students may have labels which do not exactly match the labels in the model diagram, yet cannot be marked incorrect. Therefore, in order to automatically mark the Use – case diagrams, each and every pattern (labels) needs to be understood thoroughly (e.g. Login Authentication or Authenticate User or Login Verification etc.). All these combinations should be considered as valid combinations.

The following are the challenges in the label matching process.

- **Special symbols**: sometimes students may enter symbols like ampersand, slash, under score.
- **Misspelling**: for example, Instead of "comparison" student enters "comparision" or "comparsion".
- **Differing case, leading, trailing and embedded spaces**
- **Concatenation of words**: sometimes students concatenate multiple words in the label like "VerifyUser" or "GenerateReport" instead of "Verify User" or "Generate Report". To match this with the correct label it needs to be separated into different words.
- **Synonyms and abbreviations**: Sometimes student may use synonym or abbreviation in the label. For example, student use "Verify User" instead of "Authenticate User" or "View msg" instead of "View Message".
- **Differing number of words and order**: A label of the student diagram may contain more or less words than corresponding correct label in the model diagram. The order of the words in the label of student diagram may differ than the corresponding correct label of model diagram. For example, "Login" instead of "Login User" or "Report Generation" instead of "Generate Report".
- **Non linguistic synonym:** Sometimes label may be considered correct even if it is not a linguistic synonym of label in model use case diagram. For example, instead of "Passenger" student enters "Customer" or "User".

It is proposed to develop and implement a label matching agent for providing complete and efficient assessment.

For effective label matching these issues should be resolved. The label matching agent will resolve these issues and thus match the labels by applying different algorithms [40].

(ii) Structure Matching Agent:

In many situations Use-case diagrams drawn by the students are not précised. While working in automated assessment it is required to deal with precise as well as imprecise Use-case diagrams.

The structure matching of student Use case diagram with a model Use case diagram will be done by structure matching agent.

While working in the context of Use-case diagram, following are the challenges in the structure matching stage.

- **Opposite direction of relationship:**

Sometimes students may draw the opposite direction of the relationship like extend, include and generalization.

- **Classification of Use-case diagram**

Students may draw actor-wise diagram or functionality wise diagram.

- **Merging and separation of Use-cases**

The diagram submitted by the students may sometimes have separated Use – cases like Calculate profit, Calculate loss and sometimes Use –cases may have been merged like Calculate profit & loss as per their knowledge and understanding which cannot be marked as incorrect.

- **Different number of actors and Use-cases**

Sometimes students may draw more or less number of Use-cases or actors than the model Use-case diagram.

- **Expanded and non-expanded Use-cases**

Though a particular Use – case functionality is expanded up to certain levels in the model diagram that Use – case may be drawn in the context level by the students but may not have been expanded to other levels and vice - versa.

It is proposed to develop and implement an effective structure matching agent for solving these challenges efficiently which will work simultaneously with label matching agent. This agent will take input from graph generator stage.

A structure of model Use-case diagram generated by graph generator is compared with the structure of student Use-case diagram generated by graph generator by structure matching agent. The labels of Use-case diagrams drawn by students and an expert are processed by the label matching agent simultaneously.

3.2.3 Result Generating Agent

This stage will take input from evaluation agent. The intermediate result in the form of matching or non matching pairs will be generated by the evaluation agent. Different rules will be applied on this intermediate result to give qualitative and quantitative result.

Here assessment does not mean only right or wrong with quantitative feedback but also a qualitative feedback. The qualitative feedback will interpret the incorrectly drawn diagrams; provide a summary, suggestions or comment of errors and point out the areas of mistakes in the diagram.

4. BENEFITS OF PROPOSED MODEL

The architecture proposed in this would be useful for the instructor or teacher in evaluating large number of students online. The evaluation time would be reduced tremendously. This architecture will facilitate the institutions of higher learning to reach out to a large number of students who want to pursue higher studies from reputed institutions by encouraging learned faculty members to participate in the learning process. The needy students can get the knowledge of expert teachers online and can upgrade themselves correctly by means of automatic assessment.

5. CONCLUSION AND FUTURE WORK

This proposed work is going to be very useful for the needy students by assisting in assessment of their performance. They can get the knowledge of expert teachers online and can upgrade themselves correctly by means of automatic assessment. This proposed architecture will serve our objective to automate the assessment of Use Case Diagrams giving instant feedback to students in terms of quantitative as well as qualitative measures.

In the future, full automation in the Use – Case Diagram matching and their evaluation will be achieved. It is also planned to provide quality feedback to the students indicating the place in the question or in the diagram where the student would have made mistake or if Use-Case would be left.

Later the system will also be evaluated for other quality factors like memory requirement and execution efficiency.

6. REFERENCES

[1] Hiltz S. R. (1994). The Virtual Classroom: Learning Without Limits Via Computer Networks, Norwood NJ, Ablex

[2] Yi – fang Brook Wu, Xin Chen (2004) Assessing Distance Learning Student's Performance: A Natural Language Processing Approach to Analyzing Class Discussion Messages, In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '04) IEEE Comp. Soc. Information Systems Department, New Jersey.

[3] Kevin Waugh, Pete Thomas and Neil Smith. (2004) Towards the Automated Assessment of Entity – Relationship Diagrams, Computing Department, Open University, UK.

[4] Neil Smith, Pete Thomas and Kevin Waugh. Interpreting Imprecise Diagrams, Computing Department, Open University, UK.

[5] Jamnik M. (1998) Automating Diagrammatic Proofs of Arithmetic Arguments, Ph.D. thesis, University of Edinburgh.

[6] Anderson M., McCartney R. (2003) Diagram processing: Computing with diagrams, Artificial Intelligence 145 (1-2): pp. 181 – 226

[7] Donlon J.J., Forbus K.D. (1999). Using a geographic information system for qualitative spatial reasoning about trafficability, Proceedings of the Qualitative Reasoning Workshop, Loch Awe, Scotland

[8] Thomas, P.G., Waugh, K., Smith, N. (2005) Experiments in the Automatic marking of E- R Diagrams. In Proceedings of the 10th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2005), Monte de Caparica, Portugal, pp. 158 – 162

[9] Thomas, P.G. (2003) Drawing Diagrams in an Online Exam, In Proceedings of the 8th Annual International Conference in Computer Assisted Assessment. Loughborough University, Loughborough, UK, pp. 403 – 413

[10] Thomas, P.G., Waugh, K., Smith, N. (2006) Using Patterns in the Automatic marking of E- R Diagrams. In Proceedings of the Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2006), Bologna, Italy.

[11] Waugh K., Thomas, P.G., Smith, N. (2007) Teaching and Learning Applications Related to the Automated Interpretation of ERDs. 24th British National Conference on Databases (BNCOD' 07) IEEE computer Society. Computing Department, Open University, UK.

[12] Noraida Haji Ali, Zarina Shukur and Sufian Idris (2007). Assessment System For UML Class Diagram Using Notations Extraction. Computer Science Department, University Malaysia, Malaysia. IJCSNS International Journal of Computer Science and Network Security, VOL. 7 No. 8

[13] D. Deva Kumar, R. Sanyal (2008). Static UML Model Generator from Analysis of Requirements(SUGAR). Proc. Advanced Software Engineering and its Applications. ASEA 2008, pp. 77 – 84

[14] D. Deva Kumar, Muhammad Ali Babar (2009). An Automated Tool for Generating UML Models from Natural Language Requirements. IEEE/ACM International Conference on Automated Software Engineering

[15] Boman, T. et al. (1996). Requirements Elicitation and Documentation Using TRED. In Master's Thesis, UMNAD 159.96, Umea University. [online]. Available: http:// www.cs.umu.se/~record/ T-red/master.html. (March 20, 2004)

[16] Christobal Cayaba, Joseph Astrophel Rodil, Nathalie Rose Lim (2007). Automatic Generation of Use Case Diagrams from English Specifications Document. In the Proceedings of the 19th International Conference on Software Engineering & Knowledge Engineering (SEKE, 2007), USA. ISBN: 1 – 891706-20-9. Pp. 203 – 208

[17] K. Subramaniam, D. Liu, B. H. Far and Armin Eberlein(2004).UCDA: Use Case Driven Development Assistant Tool for Class Model Generation. In the Proceedings of the 16th International Conference on Software Engineering (SEKE' 04), Banff, Canada. ISBN: 1 – 891706 – 14 – 4. Pp. 324 - 329

[18] Shadi Moradi Seresht, Olga Ormandjieva. Automated Assistance for Use Cases Elicitation from User Requirements Text. In the Proceedings of 11th Workshop on Requirements Engineering

[19] Stephane S. Some(2005). Supporting Use Case Based Requirements Engineering. Preprint submitted to Elsevier Scinece

[20] Thomas P., Waugh K. and Smith N. (2012). Automatically Assessing Free-Form Diagrams in e-Assessment Systems. In the Proceedings of *STEM* Annual Conference 2012, 12-13 Apr 2012, Imperial College London

[21] Thomas P., Waugh K. and Smith N. (2007). Learning and Automatically Assessing Graph-based Diagrams. In the Proceedings of the 7th ALT Association for Learning Technology Conference.http://www.alt.ac.uk/ altc2007 /timetable/files/1069/ALT-C 07 Presentation.pdf

[22] Thomas P., Smith N. and Waugh K. (2009). The role of labels in the automatic assessment of graph-based diagrams. In Proceedings of the 23rd ICDE World Conference on Open Learning and Distance Education, 7-10 June 2009, Open Universiteit Nederland, Maastricht.

[23] Jayal A., Shepperd M. (2009). The Problem of Labels in E-Assessment of Diagrams. ACM Journal on Educational Resources in Computing (JERIC) Volume 8 Issue 4, Article No. 12, January 2009.

[24] Jayal, A., Shepperd, M. (2009). An Improved Method for Label Matching in E-Assessment of Diagrams. Journal on Innovation in Teaching and Learning in Information and Computer Sciences (ITALICS), Electronic journal of the UK Higher Education Academy, Volume 8 Issue 1, February 2009.

[25] Du, M.-W., Chang, S.C. (1994). An Approach to Designing Very Fast Approximate String Matching Algorithms, IEEE Transactions on Knowledge and Data Engineering, vol. 6, no. 4, pp. 620-633, Aug. 1994

[26] Navarro, G., (2001). A guided tour to approximate string matching, ACM Journal on Computing Surveys (CSUR) Volume 33 Issue 1, Pages 31-88, March 2001

[27] Faro, S., Lecroq, T.,(2013). The exact online string matching problem: A review of the most recent results, ACM Journal on Computing Surveys (CSUR) Volume 45 Issue 2, Article No. 13, February 2013

[28] Salmela, L. (2012). Average complexity of backward q-gram string matching algorithms, Elsevier Journal on Information Processing Letters, Volume 112, Issue 11, 15, Pages 433–437, June 2012

[29] Tselonis, C., Sargeant, J. (2005). Diagram Matching for human-computer collaborative Assessment. In Proceedings of the 9th CAA International Computer Assisted Assessment Conference.

[30] Wagner, R., Fischer, M., (1974). The String-to-String Correction Problem, Journal of the ACM, Volume 21 Issue 1, pp. 168 – 173, January 1974.

[31] Yuhua Li, Zuhair A. Bandar, David McLean (2003). An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources, IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 4, pp. 871-882, July-Aug. 2003.

[32] Myunggwon Hwang, Chang Choi, Pankoo Kim (2011). Automatic Enrichment of Semantic Relation Network and Its Application to Word Sense Disambiguation, IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 6, pp. 845-858, June 2011.

[33] Hoggarth G., Lockyer M.(1998), An Automated student diagram assessment system, In Proceedings of the 6th annual conference on the teaching of computing and the 3rd annual conference on integrating technology into computer science education: changing the delivery of computer science education, pp. 122 – 124.

[34] Higgins C., Symeonidis P. and Tsintsifas A. (2002), The marking system for CourseMaster, In Proceedings of the 7th annual conference on Innovation and technology in computer science education, pp. 46 – 50.

[35] Tselonis C., Sargeant J. (2005), Diagram matching for human-computer collaborative assessment, In Proceedings of the 9th CAA International Computer Assisted Assessment Conference.

[36] Segla Kpodjedo, Filippo Ricca, Philippe Galinier et al. (2013), MADMatch: Many-to-many Approximate Diagram Matching for Design Comparison, IEEE Transactions on Software Engineering, 11 April 2013. IEEE computer Society Digital Library. IEEE Computer Society.

[37] Semantic Similarity Measures (n.d.), http://marimba.d.umn.edu/similarity/measures.html [Accessed on 10 may, 2013]

[38] Alistair Cockburn (2001), Writing Effective Use Cases, Addison-Wesley

[39] Vachharajani.V., Pareek. J. (2012). Use-Case Extractor: XML Parser for Automated Extraction and Storage of Use-Case Diagram. In the Proceedings of 1st IEEE International Conference on Engineering Education Innovative Practices and Future Trends (AICERA – 2012), Kerala. IEEE Xplore. pp. 1 – 5.

[40] Vachharajani.V., Pareek. J., Gulabani S. (2012). Effective Label Matching for Automatic Evaluation of Use - Case Diagrams. In the Proceedings of 4th IEEE International Conference on Technology for Education (T4E, 2012), Hyderabad. IEEE Computer Society. pp. 172 – 175.

[41] Striewe M., Goedicke M. (2011) Automated Checks on UML Diagrams. In the Proceedings of the 16th annual joint conference on innovation and technology in computer science education (ITiCSE 2011) Darmstadt, Germany, June 27 – 29, 2011, pp 38-42