The Problem of Labels in E-Assessment of Diagrams

AMBIKESH JAYAL and MARTIN SHEPPERD
Brunel University

In this article we explore a problematic aspect of automated assessment of diagrams. Diagrams have partial and sometimes inconsistent semantics. Typically much of the meaning of a diagram resides in the labels; however, the choice of labeling is largely unrestricted. This means a correct solution may utilize differing yet semantically equivalent labels to the specimen solution. With human marking this problem can be easily overcome. Unfortunately with e-assessment this is challenging. We empirically explore the scale of the problem of synonyms by analyzing 160 student solutions to a UML task. From this we find that cumulative growth of synonyms only shows a limited tendency to reduce at the margin despite using a range of text processing algorithms such as stemming and auto-correction of spelling errors. This finding has significant implications for the ease in which we may develop future e-assessment systems of diagrams, in that the need for better algorithms for assessing label semantic similarity becomes inescapable.

Categories and Subject Descriptors: K.3.1 [**Computer Uses in Education**]: Computer-managed instruction—*experimentation*

General Terms: Experimentation

Additional Key Words and Phrases: E-assessment, diagrams

ACM Reference Format:
Jayal, A. and Shepperd, M. 2009. The problem of labels in E-assessment of diagrams. ACM J. Educ. Resour. Comput. 8, 4, Article 12 (January 2009), 13 pages. DOI = 10.1145/1482348.1482351. http://doi.acm.org/10.1145/1482348.1482351.

1. INTRODUCTION

There has been growing interest within the e-learning community of automated marking of student coursework [Brown et al. 1997]. This is generally referred to as electronic assessment or e-assessment. The motivation for this is threefold. Automated marking is hoped to be more consistent than human-based marking. Second, there are potential resource benefits arising from more economical marking, and third, there is the opportunity for more timely feedback for the students. Timeliness is of particular value for formative assessment.

Authors' address: A. Jayal and M. Shepperd, Department of IS & Computing, Brunel University, Uxbridge, UB8 3PH, UK.

Assessments are diverse in form as well as content. Some forms, such as multiple choice, clearly lend themselves to easy automation. One area that is important but rather challenging is assessment that involves diagrammatic notation. These are commonplace in subjects such as computer science where diagrams generally have semi-formal semantics.

The challenge derives from the need to deal with partial and even incorrect semantics. It is not usually appropriate to give a partially correct solution zero marks, hence such solutions must also be understood. Moreover, typically much of the meaning of the diagram resides in the diagram labels. For example, consider a state model (used by the popular UML software specification and design notation [Larman 2002]); without meaningful labels the model cannot be interpreted yet, in general, the choice of label is unbounded so there are problems of synonyms, homonyms, abbreviations, misspellings, and so forth. And of course the problem is compounded by the fact that labels comprise variable numbers of words.

Thus the challenge is not merely one of syntactic equivalence; consequently approaches such as searching for graph isomorphisms are seldom adequate. Therefore we need to consider the extent of the problem of matching semantically equivalent labels to those employed in the correct solution. So in practice, what is the range of labels that might be used and how might we attempt to reduce the number of synonyms? It is this particular problem that our article addresses.

The authors are unaware of any other empirical research that has explored the question of diagram labeling directly. As we have stated, the problem of assessing diagrams with labels give rise to two questions. How effective are preprocessing techniques in reducing the number of synonyms that may be encountered? What is the impact of scale? In other words, as we successively increase the number of pieces of coursework does this reduce the number of synonyms previously unencountered? If this is the case, we might expect economies of scale.

The remainder of this article is organized as follows. We review existing approaches to automated assessment of diagrams and in particular approaches adopted for matching labels. Next we describe our empirical study based upon 160 computer science undergraduate student solutions for a UML design problem. We present our results and conclude with a discussion of their potential significance for the problem of e-assessment of diagrams.

2. RELATED WORK

There has been previous research work to automatically mark coursework. This has targeted objective type questions [Clark 2002; Feng et al. 2006], free response text-based questions [Valenti et al. 2003; Perez et al. 2005; Kerejeta et al. 2005], mathematics based questions [Beevers 2002; Pollock 2002], and computer programming questions [Ala-Mutka 2005]. There has also been growing interest in diagram-based questions [Hoggarth and Lockyer 1998; Higgins et al. 2002; Thomas 2004; Tselonis and Sargeant 2005] but results to date have been limited. The diagrams are difficult to automatically mark

Table I. Tools for E-Assessment of Diagrams

Reference	Tool name	Source
eA1	University of Teesside (UK) Automated Student Diagram Assessment System	Hoggarth and Lockyer 1998
eA2	Nottingham University (UK) CourseMaster	Higgins et al. 2002 Higgins and Bligh 2006
eA3	Open University (UK) DEAP Diagram Tool	Thomas 2004 Thomas et al. 2007
eA4	Manchester University (UK) Assess By Computer (ABC)	Tselonis and Sargeant 2005
eA5	Loughborough University (UK) Diagram Tool	Batmaz and Hinde 2006
eA6	Canterbury University (NZ) KERMIT Tool	Suraweera and Mitrovic 2002

Table II. E-Assessment Tool Details

Ref.	Marking technique	Label similarity technique	Label diversity problematic?
eA1	Object Oriented Metrics	Manual intervention	Unknown
eA2	Object Oriented Metrics	Unknown	Unknown
eA3	Local Metrics (label, type)	Edit Distance algorithm, Synonyms, abbreviations, punctuation, hyphenation and stemming	Yes
eA4	Graph Isomorphism, Local Metrics (label, type)	Edit Distance algorithm	Yes
eA5	None, human marking	None	Unknown
eA6	None	Manual intervention	Unknown

because of problems such as the diagram being malformed or possessing missing or extraneous features [Smith et al. 2004]. In addition there are problems of topology where equivalent diagrams can be laid out differently and semantic problems where topologically identical diagrams have different meanings due to different labellings.

We conducted a systematic literature review for the e-assessment software systems for diagram-based coursework and this yielded a total of six different systems (see Tables I and II). The systematic search of the literature was conducted on five bibliographic databases.[1]

Table II indicates some diversity in technique for automated marking ranging from a graph theoretic approach (eA4) to use of object-oriented (OO) metrics (eA1 and eA2) which are limited to situations of OO software development. Likewise there are a number of techniques for assessing label similarity to the specimen solution. Edit distance is the most common but it suffers when words are used with similar meanings but very different spellings or when comparing phrases of differing lengths.

Interestingly only two of the six studies identified, the Open University DEAP Diagram Tool (eA3) and Manchester University ABC tool (eA4) mention

[1]The bibliographic databases IEEE/IET Electronic Library, ACM Digital Library, ScienceDirect, Scopus, and SpringerLink were searched in May 2008 using the search term "E-assessment AND diagram". Alternative terms used for E-assessment were "computer aided assessment" and "computer-assisted assessment". This retrieved almost 900 articles that were then hand-checked for relevance.

the problem of the diversity of labels used by students in a diagram. McGee Wood et al. [2005] encountered a high degree of variation in the labels used by students for an objective two-word phrase in a technical domain. The study further mentions that it would be alarming to make speculations on the variation in the labels used by students in an open book test with no time pressure. Clearly the diversity of labels used by students in a diagram will significantly impact the complexity of the automatic marking process. Therefore we decided to empirically investigate the extent of this problem, particularly as we are unaware of any other empirical study to quantify the extent of this problem.

3. METHOD

This observational study was conducted after the coursework assessment process had been completed, which means that it was uninfluenced by our research and we did not interfere with the process in any way.

Recall that we wished to empirically explore the diversity of labels used by students. We selected coursework from a second-year undergraduate computer science course on software engineering methodology from Brunel University in London. The students were required to draw UML activity diagrams as part of their solution. They were free to draw the diagrams either at home or in the labs over a period of a month. The coursework description consisted of three paragraphs of text explaining the requirements for a bus travel card system and required students to draw an UML activity diagram for this problem. The coursework and the model solution provided by the lecturer are presented in Appendix A.

The students had created a project for the UML activity diagram [OMG 2007a] in the Borland Architect CASE tool [Together 2008] and submitted the complete project folder as a single compressed file. Initially we received 193 compressed files each containing a Borland architect case tool project. Unfortunately some of the compressed files could not be opened because they were corrupted and some did not contain the UML activity diagram so after removing all such files we were left with 160 compressed project files for the study. We then uncompressed each of these files, opened them in Borland Architect and extracted the UML XMI [OMG 2007b] [Frankel 2003] file using the Borland Architect export utility. We also wrote a Java program to parse the XMI files and generate reports about the labels present in these diagrams. This processing was convenient for our situation but the details will obviously depend upon the local circumstances of those conducting the research.

We analyzed the cumulative effect of adding ten new diagrams at a time which were randomly selected without replacement from the pool of 160 diagrams. Since the order in which the sets of 10 diagrams were selected from the pool of all diagrams might be influential, the randomization and cumulative analysis was repeated 30 times.

Having extracted the labels we applied three sequences of text processing. These are summarized in Figure 1, where the ellipses denote specific text transformations such as trimming. However, there are ordering issues so the combination of transformations are referred to as transformation sequences

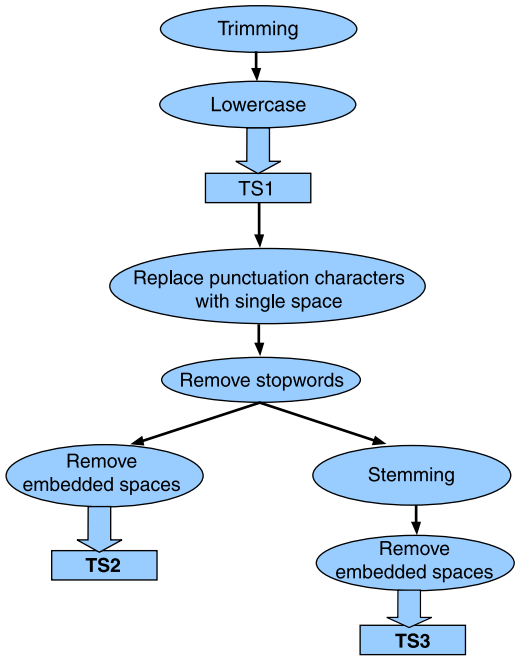

Fig. 1. Text transformation processing.

(TS) and these are denoted by rectangles. Table III gives examples of each individual text transformation.

—Do nothing: No processing of the labels extracted from the diagrams.

—Transformation Sequence 1 (TS1): This involved trimming the labels and converting them to lowercase, so for example, the terms "Update Balance" and "Update Balance" would be transformed into "update balance".

—Transformation Sequence 2 (TS2): This involved first replacing the punctuation characters such as underscore with a single space, then removing the stopwords and then finally removing the embedded spaces.[2] Stopwords are very common words such as "to" and "the" that can be ignored while comparing labels Wikipedia [2008]. So "update_balance" is first converted to "update balance" and then to "updatebalance", "display the charge" is first converted to "display charge" and then to "displaycharge" and "process card" is converted to "processcard".

[2]Embedded spaces were removed because we needed to identify synonyms but the student labels did not always contain spaces between words, for example "invalidbeep". To avoid treating "invalidbeep" and "invalid beep" as different labels we removed the embedded spaces. Another solution would be to split "invalidbeep" into two separate words but for this we would require the automatic correction of words using a spell checker. Unfortunately spell checkers are not always accurate and can lead to over-correction, for example the Open Office spell checker auto corrects the label "cardreader" to "car dreader". Hence we decided to remove the embedded spaces from all labels.

Table III. Text Transformations

Text Transformation	Before	After
Trimming	"Update Balance"	"Update Balance"
Lowercase	"Update Balance"	"update balance"
Replace punctuation characters with single space	"update_balance"	"update balance"
Remove stopwords	"display the charge"	"display charge"
Remove embedded spaces	"process card"	"processcard"
Stemming	"processing"	"process"

Table IV. Basic Data

Item	Count
Total number of students	160
Total number of labels	2013
Mean number of labels per assessment	12.58
Number of labels in the "correct" solution	8
Mean number of words per label	3.06

—Transformation Sequence 3 (TS3): This differs from TS2 in that the stemming text processing must be performed prior to removing embedded spaces. This is because the stemming algorithm which reduces a word to its root form cannot deal with concatenated words, hence embedded spaces are essential to delineate each word.

4. RESULTS

Since we wished to explore the relationship between scale (number of coursework items submitted) and the number of unique labels, we looked at the problem cumulatively. To do this we took 10 items and determined the number of unique labels by performing a string match.[3] Then a further 10 items were successively added until all 160 items were included. The ordering was determined randomly. In order to smooth out any random effects the study was repeated 30 times and the mean from 30 runs was used for our analysis.

Table IV summarizes the raw data of our study and reveals that there is a tendency for students to provide more extensive state models than the "correct"[4] solution. It also indicates the prevalence of multi-word[5] labels, thus we see a typical label has slightly in excess of three words. This compounds the problem of determining whether diagram labels are synonyms.

Table V indicates the impact of our text manipulation strategies to reduce the number of labels. As can be seen, the most effective of these strategies is TS3 which includes word stemming. This has the positive effect of reducing the number of unique labels by nearly 75%, however, in practice this still leaves

[3]We perform a string match and only count the exact matches so that if a misspelling occurs we end up with two different terms. So for example the terms "valid bep" and "valid beep" are considered as two different terms.

[4]Of course a complication is that there isn't a single correct solution, even setting aside the problem of synonymous labels. An area that is particularly manifest is the problem of differing levels of decomposition. We discuss this later when we consider how the correct synonyms are distributed.

[5]A word is defined as any sequence of alphabetic or numeric characters bounded by a space or punctuation character or by the end of the string.

Table V. Text Transformation Impact upon Label Count

Text Transformation Sequence	Count	%
Total number of labels	2013	100%
(Do Nothing) Total number of unique labels	773	38.4%
(TS1) Case and space trimming	638	31.7%
(TS2) Punctuation and stop words	571	28.4%
(TS3) Stemming	537	26.7%
Total number of unique correct label synonyms from TS3	358	17.8%

Table VI. The Impact of Spelling Mistakes upon Synonym Proliferation

Text Processing Strategy	Count	%
Total number of correct label synonyms	358	100%
Misspelt correct synonyms	35	9.2%
Auto-correctable synonyms	20	5.2%
Matching auto-correctable synonyms	10	2.6%

us with 537 unique labels, which has a considerable impact if these must be examined manually. Unfortunately there is no evidence that this task can be ignored since approaching 20% of the overall total (358) are in fact correct, that is, synonymous—as judged by the human marker—with a label in the model solution. This means in the absence of automation the human marker must deal with a total of 358 variants of the eight labels in the model solution (see Figure 4 contained in Appendix A).

Next we consider the impact of spelling errors upon synonym proliferation since these can be potentially dealt with automatically by a spelling corrector. Table VI reveals that a hand search of the correct synonyms finds less than 10% are the result of spelling errors. By applying the auto-corrector from Open Office we were able to correctly repair just over half of these spelling errors but unfortunately in half of the cases we merely generated a different unique synonym, albeit correctly spelled. Thus the impact of repairing spelling errors is to reduce the number of correct unique labels by 2.6%, that is a somewhat marginal impact.

The line plot in Figure 2 explores the question of economies of scale. This is an interesting question since one driver for automated assessment is economic. They indicate the number of new unique labels added per 10 students so as to present the effect of increasing the number of students. In our case we had a total of 160 students so we simulated the process of growing the number of students by randomizing the order and successively adding groups of 10 students. The three line plots represent our three levels of text processing. We can see that the number of new unique labels added tends to decrease as the number of students is increased. This is not particularly surprising since we might expect increasing numbers of label collision, that is, picking a non-unique label. However, there also appears to be a tendency to flatten out from about 90 students onwards. The disconcerting issue here is that even after Level 2 text processing there is little evidence that new unique labels are being added at a rate of less than 30 per 10 additional coursework answers. Nor does this rate appear to be declining. Since this is potentially quite a hurdle for e-assessment of diagrams we examine this problem a little more closely.

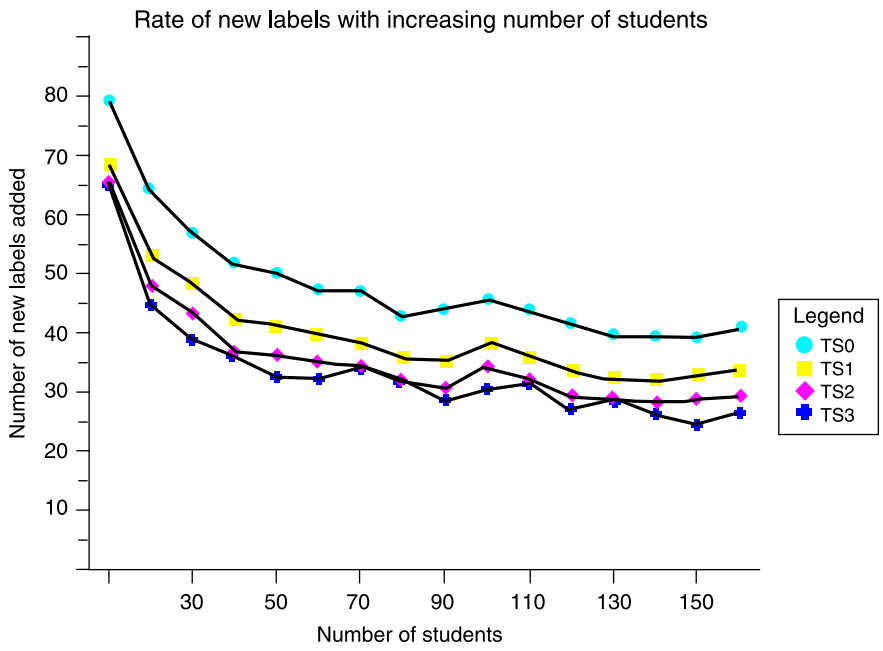

Fig. 2. Rates of new labels with increasing numbers of student coursework.

Table VII indicates the huge variability in the number of synonyms per label that range from six to 258. Unsurprisingly "start state" and "end state," which are defined within the UML notation, have the lowest synonym counts. The reason these have six and seven variants respectively is that the software tool (Borland Together) forces unique labels, and on occasion students, as a side effect of editing, may have had more than one such label. The consequence is an automatically appended integer for the purposes of differentiation. Subsequently the student may have deleted the original rather than the differentiated state leading to a proliferation of synonyms. The obvious solution of removing trailing integers from otherwise identical labels, although would work for this assignment, is more generally problematic since there are other labels, for example, "deduction £1" where the trailing integer could be significant.

Worryingly, "read card" which is a state taken from the partial solution provided, resulted in 40 correct variants. This indicates that even a modification type task still leads to an explosion of valid synonyms. Lastly there is the question, why should the update total yield so very many valid alternatives, more than four times more than the next largest count? In part the answer lies in differing levels of decomposition adopted by the students such that some solutions provided more detail than the model answer with the consequence that there is a subsumption relationship between one label in the model solution and a set of many labels in the student solution. Here the human expert will give marks for those lower level labels in the student answer which when composed add up to form the concept (and label) in the model answer.

Table VII. Distribution of Synonym Counts for the Correct Labels

Label	Number of Synonyms
check card type	11
check time	14
end state	7
invalid warning	61
read card	40
start state	6
update total	258
valid beep	30

5. DISCUSSION

To summarize, we analyzed 160 student diagrams for a real-world assessment and from this observational study we obtained following results:

—We found that even a simple diagrammatic task in which the students were asked to extend a given model led to an explosion of labels and many synonyms of the correct labels (those provided by the lecturer performing the assessment).

—While basic text processing such as trimming spaces and removing stopwords were important in reducing the number of synonyms they are not in themselves sufficient. In our particular example this still left us with 358 variants of the eight correct labels, that is, an average of almost 45 synonyms per label.

—The rate at which new unique labels are added does decrease with new students but only to a point, so that even after processing 100 student assessments we still find new (i.e., previously unencountered) labels at a rate of three per student. Over the range of students we studied this trend shows no sign of diminishing.

So what are the implications of these results? First, we believe this study suggests that the problem of labels is substantial and cannot be easily avoided for the e-assessment of at least some classes of diagram. Of course our approach to text processing is relatively naïve. For example, we do not use autocorrection from a spelling checker nor we explore alternative word orderings for multi-word labels. The reason for this approach is we are interested in assessing the scale of the problem of labels, not the efficacy of different solutions. Obviously more sophisticated text processing may improve our ability to deal with labels but outstanding challenges include the prevalence of multi-word labels and differing levels of decomposition. Incidentally the latter problem also impacts simple syntactic approaches such as searching for graph isomorphism. Second, the economies of scale effect that one might hope to encounter is only present to a limited degree at least over the range of student numbers we studied. Therefore we argue that there are a number of technical challenges that need to be addressed before e-assessment of diagrams can become a practical reality and in Section 6 we discuss this research agenda in more detail.

Finally, we consider what threats may exist to the validity of this study. First, how typical is this coursework? UML is a widely used and taught design and development method in computer science. The coursework we selected had

already been given to the students and subsequently marked. If anything the task is somewhat small with a mere eight states so we may be underestimating the scale of the problem. The question also arises how typical are our students? Clearly one issue is their facility with language. Less than 10% of the cohort were overseas students.

So we conclude that while we only consider one coursework task in this study many of the threats may lead us in the direction of optimism rather than pessimism.

6. FUTURE WORK

We see three areas that may repay further research if e-assessment of diagrams is to become a practical proposition.

First, we need to explore and develop more effective text processing techniques to better automate the process of discovering synonyms for the diagram labels. To do this we need to look to the NLP community. These techniques may include effective syntax-matching algorithms, semantic-matching algorithms, and ontologies. We intend to carry out further research in this area of enhancing techniques for matching labels for marking diagrams.

Second, we have only conducted a single empirical study based upon 160 students. It would be useful to see this work replicated by other researchers using different groups of students and diagrammatic coursework.

Last and certainly not least, we need to further study those factors that influence the student's choice of a label particularly with a view to reducing the search space. Apart from the personal factors of the student such as behavioral traits, background, level of knowledge of the subject, etc. over which the lecturer has no control, the student's choice of labels in a diagram may be also affected by some other factors over which a lecturer has control such as the text used in the coursework question or by presenting to students a large pool of labels and instructing them to use the labels from this pool only. Since this observation study was conducted post coursework assessment process and we did not interfere with the assessment process in any way, we could not measure the effect of such factors on the choice of labels used by the students in a diagram. So there is a scope for a future study to measure the effect of various factors like student's behavioral traits, background, subject knowledge, text of the coursework question, etc. on the choice of labels used by students in a diagram.

APPENDIX A

Case Study: Problem Specification

The Cockle Card System: Chipolata Buses of Marlin on Sea plan to invest in a new bus card system. In addition to a travel card (monthly, weekly, daily), there is a prepay card (pay-as-you-go) where customers can purchase credit in advance, and a concessions card allowing free off-peak travel for certain groups of people. Each bus is to be fitted with a card reader which will read the card, update the amount of credit (for prepay) or check it is valid (travel cards or concession cards). Different fares are charged for peak and off-peak services.

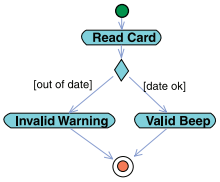

Fig. 3. Part of question.

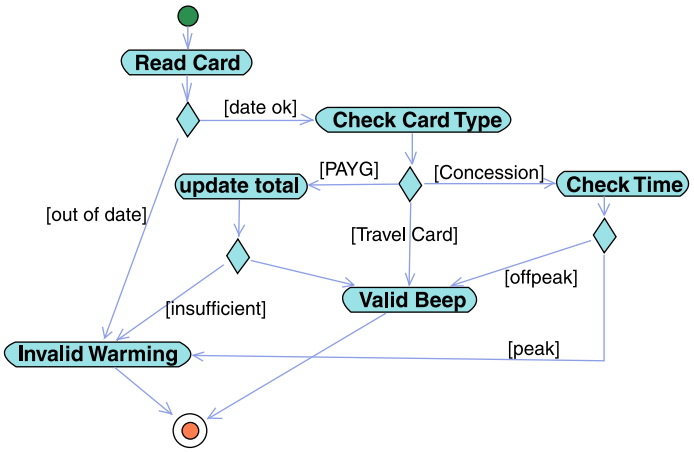

Fig. 4. Model answer.

If the card is not valid for some reason (e.g., out of date, no credit, or can't be used at peak times) the reader should give an audible warning to prompt the driver to read the display and take appropriate action. The reader should also give a valid beep so that the driver and passenger know that the card has been read.

The prepay card needs to be debited each time it is used. However, there is a daily cap so that it never exceeds the amount that would be charged for a daily travel card. There is a flat fare for each journey, but peak journeys (before 9:30 a.m.) cost more than off-peak journeys. The amount charged is displayed.

Current costs:

(1) Daily travel card £6
(2) Weekly travel card £35
(3) Monthly travel card £120
(4) Single peak journey £2
(5) Single off-peak journey £1

The following activity diagram [refer to Figure 3] only partially models the requirements in the case study. Complete it.

Case Study: Model Solution

The Model Solution, as devised by the lecturer who marked the coursework, is presented in Figure 4.

ACKNOWLEDGMENTS

We would like to thank Dr. Michelle Cartwright for her assistance in analyzing the student coursework. We would also like to thank Dr. Pete Thomas, Dr. Tracy Hall, Dr. Annette Payne, and Gernot Liebchen for reviewing an earlier version of this article. Finally we wish to thank the anonymous reviewers for their careful reading of this article and constructive comments.

REFERENCES

ALA-MUTKA, K. 2005. A survey of automated assessment approaches for programming assignments. *Comput. Sci. Educ. 15*, 2, 83–102.

BATMAZ, F. AND HINDE, C. 2006. A diagram drawing tool for semi-automatic assessment of conceptual database diagrams. In *Proceedings of the 10th International Computer Assisted Assessment Conference (CAA'06)*, *4*, 71–82.

BEEVERS, C. 2002. The SCHOLAR programme in Scottish education. In *Proceedings of International Conference on Computers in Education (ICCE'02)*, 490–491.

BROWN, G., BULL, J., AND PENDLEBURY, M. 1997. *Assessing Student Learning in Higher Education*. Routledge.

CLARK, J. 2002. A product review of WebCT. *Internet Higher Ed. 5*, 1, 79–82.

FENG, M., HEFFERNAN, N., AND KOEDINGER, K. 2006. Addressing the testing challenge with a Web-based E-assessment system that tutors as it assesses. In *Proceedings of the 15th International Conference on World Wide Web (WWW'06)*, 307–316.

FRANKEL, D. 2003. Model driven architecture. Wiley, New York.

HIGGINS, C. AND BLIGH, B. 2006. Formative computer based assessment in diagram based domains. In *Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'06)*, 98–102.

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 (ITiCSE'02)*, 46–50.

HOGGARTH, G. AND 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 (ITiCSE'98)*, 122–124.

KEREJETA, M., LARRANAGA, M., RUEDA, U., ARRUARTE, A., AND ELORRIAGA, J. 2005. TOKA: A computer assisted assessment tool integrated in a real use context. In *Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, 848–852.

LARMAN, C. 2002. Applying UML and patterns. Upper Saddle River, NJ: Prentice Hall.

MCGEE WOOD, M., SARGEANT, J., AND JONES, C. 2005. What students really say. In *Proceedings of the 9th International Computer Assisted Assessment Conference (CAA'05)*.

OMG. 2007a. UML Version 2.1.2. Tech. rep., Object Management Group.

OMG. 2007b. XMI 2.1.1. Tech. rep., Object Management Group.

PEREZ, D., GLIOZZO, A., STRAPPARAVA, C., ALFONSECA, E., RODRIGUEZ, P., AND MAGNINI, B. 2005. Automatic assessment of student free-text answers underpinned by the combination of a Bleu-inspired algorithm and latent semantic analysis. In *Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference (FLAIRS'05)*, 358–362.

POLLOCK, M. 2002. Introduction of CAA into a mathematics course for technology students to address a change in curriculum requirements. *Int. J. Tech. Design Educ. 12*, 3, 249–270.

SMITH, N., THOMAS, P., AND WAUGH, K. 2004. Interpreting imprecise diagrams. In *Proceedings of the 3rd International Conference in the Theory and Application of Diagrams (DIAGRAMS'04)*, 22–24.

SURAWEERA, P. AND MITROVIC, A. 2002. KERMIT: A constraint-based tutor for database modeling. In *Proceedings of the 6th International Conference on Intelligence Tutoring Systems (ITS'02)*, 377–387.

THOMAS, P. 2004. Drawing diagrams in an online examination. In *Proceedings of the 8th International Computer Assisted Assessment Conference (CAA'04)*.

THOMAS, P., WAUGH, K., AND SMITH, N. 2007. Learning and automatically assessing graph-based diagrams. In *Proceedings of the 7th Association for Learning Technology Conference (ALT'07)*.

TOGETHER, B. 2008. Borland Together. Tech. rep., Borland Together.

TSELONIS, C. AND SARGEANT, J. 2005. Diagram matching for human-computer collaborative assessment. In *Proceedings of the 9th International Computer Assisted Assessment Conference (CAA'05)*.

VALENTI, S., NERI, F., AND CUCCHIARELLI, A. 2003. An overview of current research on automated essay grading. *J. Inform. Tech. Educ. 2*, 319–330.

WIKIPEDIA. 2008. Stop words. Retrieved from http://www.wikipedia.org.