

A Programming Assessment Software Artefact Enhanced with the Help of Learners

Romeo A. Botes, Imelda Smit

Abstract—The demands of an ever changing and complex higher education environment, along with the profile of modern learners challenge current approaches to assessment and feedback. More learners enter the education system every year. The younger generation expects immediate feedback. At the same time, feedback should be meaningful. The assessment of practical activities in programming poses a particular problem, since both lecturers and learners in the information and computer science discipline acknowledge that paper-based assessment for programming subjects lacks meaningful real-life testing. At the same time, feedback lacks promptness, consistency, comprehensiveness and individualisation. Most of these aspects may be addressed by modern, technology-assisted assessment. The focus of this paper is the continuous development of an artefact that is used to assist the lecturer in the assessment and feedback of practical programming activities in a senior database programming class. The artefact was developed using three Design Science Research cycles. The first implementation allowed one programming activity submission per assessment intervention. This pilot provided valuable insight into the obstacles regarding the implementation of this type of assessment tool. A second implementation improved the initial version to allow multiple programming activity submissions per assessment. The focus of this version is on providing scaffold feedback to the learner – allowing improvement with each subsequent submission. It also has a built-in capability to provide the lecturer with information regarding the key problem areas of each assessment intervention.

Keywords—Programming, computer-aided assessment, technology-assisted assessment, programming assessment software, design science research, mixed-method.

I. INTRODUCTION

AT the North-West University (Vaal Triangle Campus), third year Information Technology learners do two database subject modules, the one building on the other. These learners have a programming background with regards to Java and/or C#. For the purpose of database programming, they need to use Standard Query Language (SQL) to code.

When merging the constantly changing computer science environment with higher education, a complex environment with techno-wise learners, a technology-driven approach to assessment is required [1]. This is even more so in the environment of teaching technology subjects. Both lecturers and learners in Computer Science (CS) acknowledge that assessment for programming subjects that are not computer-

based, needs to change to make use of the advantages provided by modern, technology-assisted assessment (TAA) tools. Learning to code programs is a time consuming activity for learners, especially from the lecturer's point of view where the provision of meaningful feedback stands central in the assessment action. Tools such as these may facilitate simulated real-life testing and feedback that is prompt, consistent, comprehensive and individualized.

A Design Science Research (DSR) process was used to develop an artefact. Initially, a pilot implementation cycle of the artefact, which allowed one program submission per intervention and was assessed upon hand-in, was implemented. The initial pilot project provided insight regarding impediments that guided the follow-up implementation. An improved second implementation cycle of the artefact allowed multiple program improvement cycles per intervention. The assessment action was initiated only when individual learners were convinced that their programs were on the highest possible standard.

The paper focuses on the continuous implementation of this artefact for feedback and assessment of this programming subject. In Section II, this paper proceeds with a discussion on literature on computer-aided assessment. This is followed in Section III where the research methodology is discussed. In Section IV a brief description about the software is described. The paper presents the results of the evaluation in Section V and reflects on the current state of the Programming Assessment Software in Section VI. Finally, the paper concludes in Section VII.

II. LITERATURE REVIEW

A. Assessment of Programming

The review of various CS courses reveals that computer programming plays an important role throughout these courses. This is also evident in the Computer Science Curricula 2013, designed in collaboration between ACM and IEEE [2]. Therefore, one of the most important skills a learner needs to acquire in information and CS courses is computer programming. Computer programming is a computing problem solving process through the formulation of executable programs, involving the design of algorithms. An algorithm may be defined as a list of clearly defined steps which take a value, or a set of values, as its input and then perform actions on it to create a new set of values as output [3]-[6], it is a tool for solving well-specified computational problems [4]. In modular programming, the problem is broken up into separate steps, and refined until the resulting steps are small enough to execute [3]. These small steps could be programming

A.R. Botes is with the North-West University Vaal Triangle Campus, Vanderbijlpark, 1910, Gauteng, South Africa (phone: 27-16-910- 3283, fax: +27-16-910-3271; e-mail: romeo.botes@nwu.ac.za.).

I. Smit is with the North-West University Vaal Triangle Campus, Vanderbijlpark, 1910, Gauteng, South Africa (phone: 27-16-910- 3294, fax: +27-16-910-3271; e-mail: imelda.smit@nwu.ac.za.).

statements such as if-else-statements, for-loops, while statements, string manipulation and mathematical calculations.

In a computer programming module, learners learn how to solve programs logically and programmatically. Software programs are written using a variety of programming languages such as C, C++, C#, Java, Python, Smalltalk, JavaScript, etc. The learners are then assessed on how well they solve a programming problem in the language used. The lecturer assesses the software programs written during the assessment intervention on their correctness and functionality.

As described in the next section, assessment can be a key factor to how learners learn [7]. Usually marks are allocated to an assessment by allocating marks after manually reviewing and executing the code. The assessment of software programs is time consuming, which often does not allow the lecturer to provide feedback concerning the quality of the TAA as discussed in Section C. TAA is used to enhance learners' learning experiences [8].

B. Computer-Aided Assessment

The common perception of CAA is the use of multiple choice questions, exploring the strengths of computers by providing consistent delivery, immediate grading and feedback; and saving academic staff time [9]. CAA, however, can be easily expanded to a range of different approaches, using existing technologies to develop new software. What is important to note about CAA is its impressive list of advantages offered for formative assessment, which include [9], [10]:

- Assessments can be repeated easily;
- Feedback to the learner is immediate;
- Evaluation results are immediately available for monitoring and adaptation;
- Results are reliable;
- The variety of evaluation is increased;
- The act of assessment is independent of evaluators;
- Assessments can occur timely and appropriately —for example weekly tests;
- Access to assessment is flexible;
- Motivation of learners may be enhanced; and
- Assessments are learner-centred — this may encourage learners to take ownership for their learning.

Three advantages provided by CAA from the above list that relates to this study, may support learners' progress to a higher level of understanding; namely its repeatability; immediate feedback, connecting the activity and its feedback in the mind of the learner; and allowing learners responsibility for their own learning.

Two advantages provided by CAA from the above list that relates to this study may support the lecturer in the teaching endeavour, including the immediate availability of marks, informing the lecturer how effective prior teaching was, and the fact that the marker does not influence the marking.

A point that may be added is that the actual marking of the assessment intervention only takes a fraction of the time compared to how long it may take to mark a non-CAA assessment intervention, allowing the lecturer to do it, instead

of an assistant. Therefore, developing CAA software that provides multiple submissions, each with immediate feedback to allow learning during each cycle, is invaluable in the programming class. CAA supports TAA which is discussed in the next section.

Sadler [11] states that assessment is “any appraisal (or judgment, or evaluation) of a learner's work or performance”. Assessment is one of the most important factors that supports and enhances learner learning [7]. It is a process that is used by academics to gather, analyse and interpret information about learners' progress – to improve teaching strategies and learner learning. By rewarding understanding and achievement, assessment is directly linked to effective teaching and learning [9]. Learners' learning efforts are most often enhanced through implementing assessment deadlines [12]. Garrison [12] states that effective academics “use assessment activities strategically to motivate learners to engage successfully in productive learning activities”. This strategic use of assessment may motivate learners with the following a list of some of the reasons for the use of assessment [7]:

- Feedback is provided;
- Learners are graded;
- Learners are allowed to correct mistakes;
- Learning is consolidated;
- It allows the application of abstract ideas in a practical setting;
- Learner potential may be identified;
- Supplying lecturing staff feedback on the effectiveness of their teaching; and
- Providing statistics for decision purposes.

Two reasons for the use of assessment provided above that relates to this study, may support learners' progress to a higher level of understanding; namely to provide (meaningful and constructive) feedback to the learner and to enable learners to correct errors. At the same time, the lecturer may be supplied with feedback on how well learners understand concepts taught. In the teaching of programming subjects this could be accomplished using CAA – as opposed to the traditional written feedback, that provide advice and guidance on how to improve.

Since the first semester database subject module is a prerequisite for the second semester subject module, CAA may be valuable to inform both the learner and the lecturer about progress.

C. Technology-Assisted Assessment

Technology is increasingly leveraged by lecturers to enhance learners' learning experiences [8]. Using various technologies for creating vivid, playful, interactive learning environments that support multimedia presentations, adaptive online exercises, and virtual discussions with greater learner control of learning and pacing is known to be technology-assisted learning (TAL). It supports vocabulary learning more than face-to-face learning but is comparatively less effective in developing listening comprehension skills [8]. Still, it has a lasting impact on education.

In the context of this paper, the use of TAA supports TAL. Learners learn about content; their mistakes are highlighted and they get prompts regarding potential corrections through the guidance provided by the feedback from the software.

D. Programming Assessment Software

In this paper, Programming Assessment Software, referred to as PAS is suggested. Although it is similar to software described as CAA and TAA in the sense that PAS makes strategic use of assessment to provide feedback and grading, it is utilised in a subject module where learners already know how to program. They are third year learners expected to apply their programming skills, as well as their programming knowledge in a new environment, that of coding SQL programs in the MySQL environment. PAS is only used to assess third year learners. For this reason, PAS treats the learner as a programmer who is on the verge of entering the job market – where they will be expected to produce programs that are well written and tested.

Typically, learners do not test their code thoroughly; they do not enter different values and they do not accommodate all types of input. This is where PAS is key, since it tests functions, procedures and methods to check whether all inputs produce the correct results. With PAS, learners should get to a point where they understand that although their code can execute successfully it is possibly not robust enough to accommodate different input values and types.

The next section will elaborate on the research methodology followed in this study.

III. METHODOLOGY

A. Design Science Research

Using DSR as a research paradigm and the DSR process as a methodology, a model was developed to assess programs and supply feedback to learners. It is defined within the IS discipline as the construction of a wide range of socio-technical artefacts such as decision support systems, modelling tools, governance strategies, methods of IS evaluations and IS change interventions [13]. It also analyses the performance of a designed artefact in order to understand and improve the artefact [14] and is primarily the creation and evaluation of an artefact used to acquire the solution to the identified organizational problem through understanding thereof [14], [15]. The evaluation of these artefacts could be subject to quantitative and/or empirical and qualitative methods [15]. There are several approaches available to guide researchers in performing DSR research [13], [14], [16]-[18]. The approach by Vaishnavi and Kuechler [17] is preferred for the design of the artefact and consists of five phases: awareness of the problem, suggestions, development, evaluation and conclusion.

The five phases suggested by Vaishnavi and Kuechler [17] form the main outer cycle of a DSR study. The development phase may be subdivided into inner cycles of repetitive phases. The first and main outer cycle presents the overall objective of the development of the artefact and the second

and inner cycle presents the detailed steps of creating the artefact.

Using DSR supports the focus of this paper, which is to develop software to assist the lecturer in the assessment and feedback in programming subjects. The following sections do not focus on the traditional DSR cycles format, but rather on the artefact itself, therefore the model of the artefact is described in Section IV.

After one repetition of the DSR inner phase was completed, it was implemented. The PAS development team then decided to expand this pilot development to include questionnaires to obtain feedback data from the learners using this software. This would allow an informed second development repetition of the DSR inner phase

B. Mixed Method Research

There are various techniques available for a researcher to collect data. Some of the common techniques include: interviews, questionnaires, observations and documents [19], [20]. The data collected through the application of these techniques may be grouped into quantitative data and qualitative data.

Seaman [21] argues that “qualitative data is richer than quantitative data” with data collected using qualitative methods contain more information than data collected using quantitative methods [21]. At times, the researcher may combine data collection techniques. The combination of qualitative and quantitative data is known as mix-method techniques and referred to as triangulation [19]. This paper incorporates a mixed method methodology, by utilising both quantitative and qualitative data from a questionnaire given to the learners.

The next section provides background to the artefact and its development and implementation.

IV. BACKGROUND OF THE PAS ARTEFACT

Evaluating programming code is problematic and a dreaded task for some higher education educators since programming code may be written in various formats, it uses varied syntaxes and different types of codes. These differences pose the problem of how to evaluate programming code effectively without wasting time; while at the same time providing feedback on how to improve the code. Using the theory provided earlier regarding Input-Process-Output (see Fig. 1), PAS uses the input and output as a means of evaluating the programming code. If a certain input is given, a certain output is expected, regardless of how it was processed. To evaluate programming code one must therefore focus on these two aspects as they remain constant while the processing may vary based on the learner's view and programming skills. One learner may for example use ten programming statements (lines of code) to complete a task while another could use more or less.

For the assessment of programming code using PAS, three important pre-requisites or rules must be adhered to; the format of input is provided for learners to use, the format of the output is provided for learners to use, and the correct

naming conventions is provided for learners to use. Fig. 2 shows a PAS example of the input-process-output format suggested in Fig. 1; the programming code processes the input and provides output, it then evaluates whether the output is correct for the given input. Marks are assigned and feedback is given. In some instances; based on the type of error produced, learners may be awarded part of the marks for that program, since program functions, procedures or methods are used to allocate marks. On the occasion where the output provided by the programming code is incorrect, the error messages from the compiler are used to provide feedback.

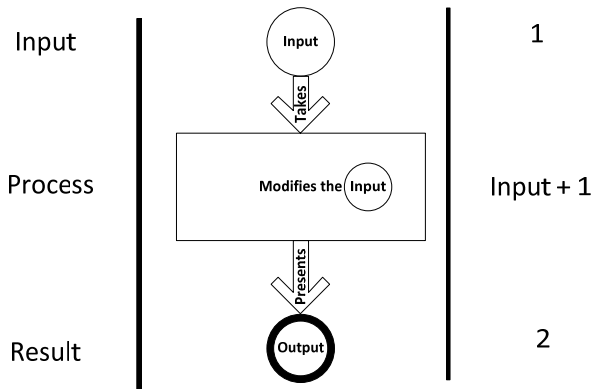


Fig. 1 Input-Process-Output

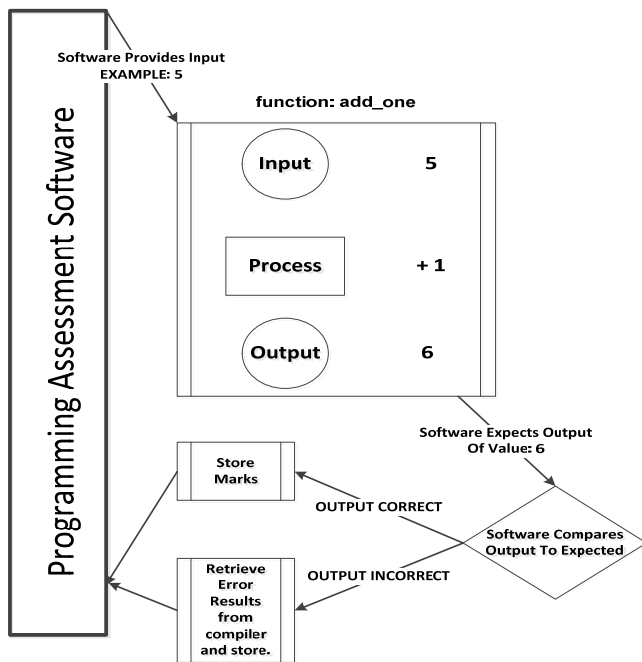


Fig. 2 Assessment logic of PAS

A restriction of this model is that the name of the function, procedure or method in PAS needs to be exactly the same as that of the function, procedure or method the learner wrote. It is therefore very important to provide these details to the learners whose programs are being evaluated. It is also important to test the function, procedure or method with

different values to eliminate the possibility of learner programs providing hardcoded output.

Fig. 3 shows an example of a learner assignment where two procedures need to be written (with line numbers 1-4 indicating the required procedure name, parameters, output and objective). It is important to note that a learner may get no marks if the program is not functional, but the mark allocation is more refined than simply allocating no marks or full marks, as can be seen from the rubric example supplied to learners. Each function, procedure or method is individually assessed – to allocate a part of the possible marks to be earned.

Evaluation

Use your database (dXXXXXXXX) schema and the created 'largeco' on the university MySQL Server to address the following:

Procedure 1

- 1 Procedure name: eval1_employee_salary_history
- 2 Procedure parameters: emp_num_search int, from datetime, to datetime
- 3 Procedure output: EMP_NUM, EMP_FNAME, EMP_LNAME, SAL_AMOUNT
- 4 Procedure objective: Find all salaries for an employee between the selected dates as a criteria.
If emp_num_search = 'all' then all records must be selected for employees between those dates, if not then selected employee records must be returned.

Procedure 2

- 1 Procedure name: eval1_find_all_products_customer
- 2 Procedure parameters: cust_code_search int
- 3 Procedure output: CUST_FNAME, CUST_LNAME, PROD_SKU, PROD_DESCRIP, PROD_PRICE.
- 4 Procedure objective: Find all product/s purchased by the customer.

Notes:

- A note an automatic event will fire after submission date to back up your databases; as well as mark your assignments on the server. Therefore no late submission will result in a 0 mark allocation.
- Not using the procedure names and parameters as stated above will also result in a 0 mark allocation.
- Since different parameter values will be utilised during the evaluation for the purpose of mark allocation.

Rubric:

| | Procedure | Run | Description | Mark |
|---|-----------|-----|-------------------------------------|------|
| 1 | 1 | 1 | Test for specific employee & dates. | 6 |
| 2 | 1 | 2 | Test for all employees. | 6 |
| 3 | 2 | 1 | Test for specific customer. | 6 |
| 4 | 2 | 2 | Test for another customer. | 4 |
| | | | TOTAL | 22 |

Fig. 3 Example of a programming assignment with the accompanying assessment rubric

After PAS assessed learners' code, learners may expect output similar to that shown in Fig. 4, where results found to be correct are assigned a mark, while those found to be incorrect, are supplied with feedback using the generated output of the program being assessed, as well as the error messages supplied by the compiler.

```

23658002
(5) >>>> OK >>>> P1R1 FOUND
(2) >>>> OK >>>> P1R2 FOUND
(3) >>>> OK >>>> P1R2 FOUND
(X) >>>> Procedure 2 Run 1 >>>> Unknown column
'employye.emp_id' in 'field list'>>>>
(X) >>>> Table Creation >>>> Table
'd23658002.tes1b_employee' doesn't exist>>>>
(X) >>>> Insert_Data >>>> Table 'd23658002.tes1b_employee'
doesn't exist>>>>INSERT INTO `tes1b_employee` VALUES ('888',
'Sam', 'Rosenberg', '0115024083', 'NA', 'NA', '12', 'msc');
(X) >>>> Insert_Data >>>> Table 'd23658002.tes1b_employee'
doesn't exist>>>>INSERT INTO `tes1b_employee` VALUES ('777',
'Diane', 'Cruz', '0123214561', 'NA', 'NA', '9', 'msc');
(2) >>>> OK >>>> Average Calculated Wrong
(X) >>>> Insert_Data >>>> Table 'd23658002.tes1b_employee'
doesn't exist>>>>INSERT INTO `tes1b_employee` VALUES
('1232', 'Rene', 'Ruffchild', '0115024083', 'NA', 'NA', '18', 'msc');
(2) >>>> OK >>>> Max Calculated Wrong
TOTAL MARK: 14

```

Fig. 4 Example of assessment output generated by PAS

A. PAS Artefact – Once-off Time Dependent Submission

The initial goal of the pilot iteration of PAS, was to assess the programming code of learners, allocate marks accordingly, and provide feedback. The pilot artefact allowed one submission of programming code. After learners coded their programs, the code was stored on a server to be assessed later.

At a pre-scheduled time PAS would execute an event and allocate marks and feedback to the submitted programs. This process is shown in Fig. 5. This pilot PAS was used in the subject module since 2012 to assess program assignments, tests, as well as the programming sections of examinations.

During 2012 57, 2013 84, in 2014 107 and during 2015 106 learners were assessed using PAS. Informal feedback from learners on this pilot artefact revealed that they were unhappy about the rigid way of marking and complained that the feedback given to them was inadequate. Initially it was felt that PAS should be the adequate since it is used to assess third year learners, but with time the PAS development team realised that the implementation of the pilot PAS is possibly too far removed from what learners are used to and some “softer” approach may be necessary to help learners to adhere to the rigours PAS requires from them. At this point the PAS developing team decided to compile a questionnaire to involve the learners using PAS in improving the artefact. Table I lists the questions asked to the learners; it also indicated the reasoning behind the question.

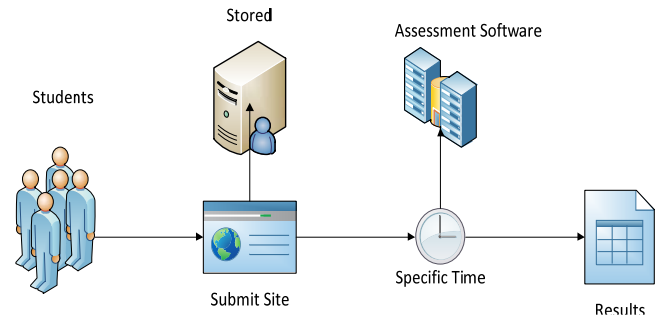


Fig. 5 Once-off Time Dependent Submission

TABLE I
PAS QUESTIONNAIRE FOR DATABASES I

| Nr. | Question | Answering options | Reason |
|-----|--|---|--|
| 1 | Did you earn good marks for full assessments or parts of assessment interventions that was marked by PAS? Indicate your mark on average: Additional comments: | Y N < 40 40-60 > 60 Open-ended | Learners' answers may explain why a particular learner responds in a positive or negative way when answering the questionnaire. |
| 2 | Did you find the feedback you received from PAS to be valuable? If your answer is YES: In essence PAS uses the compiler available to you to assess your programs. What are the reasons for you still receiving valuable feedback? If your answer is NO: How can PAS be improved to supply you with valuable feedback? Please explain your answer in full. | Y N Follow-up; open-ended Follow-up; open-ended | Determine whether PAS is of use to learners in their database learning, especially in SQL-programming (MySQL). Determine whether learners realise that they do not perform well when they have no strategy to test their program with test data sets. Possibly learners may come up with workable ways to improve PAS, this question aimed to find out if and how. |
| 3 | Do you think PAS is a fair assessment tool? Please motivate your answer. | Y N Open-ended | Informal feedback to the PAS development team indicated that learners perceived PAS to be an unfair assessment tool This question aimed at determining how many learners felt his way and why. |
| 4 | State what you perceive to be a fair way to assess third year programs. | Open-ended | With the answer to this question learners had the opportunity to come up with alternatives, especially those who feel that PAS is not utilizing a fair assessment strategy. |

The questionnaire was introduced to learners with an explanation with regards to what the questionnaire aims to do.

V. RESULTS AND FEEDBACK

From a class of 106 learners in 2015, 36 learners completed the questionnaire at the end of 2015. Although the participation rate is less than one third of the full class, it can

be seen from the answers that learners who performed well, as well as those who did not perform well in the programming component, participated. Learners who indicated that they earned good marks for full assessments or parts of assessment interventions that were marked by PAS, number 20 learners (55.5%), while 11 learners (34.4%) indicated that they did not perform well and five learners (13.9%) did not answer the

question. The highest number of learners, namely 17 (53.1%) said that they an average mark of 40-60%, 12 learners (37.5%) above 60% and five learners (13.9%) indicated a mark lower than 40%. From the eight additional comments completed on this question, three was positive and five negative. Meaningful answers are included in Table II.

TABLE II
 ANSWERS WITH A POSITIVE AND THOSE WITH A NEGATIVE POINT OF VIEW TO THE QUESTION "DID YOU EARN GOOD MARKS FOR FULL ASSESSMENTS OR PARTS OF ASSESSMENT INTERVENTIONS THAT WAS MARKED BY PAS?"

| Positive | Negative |
|---|---|
| awesome | the program is not fair sometimes because it is biased |
| it's quick and gives feedback | it does not recognise some of the things such as PK |
| the software was fair although it had some mistakes | the fact that it is sensitive is really a problem because we tend to fail because you didn't write a capital letter somewhere |
| | more time was needed for tests and examinations |

Although less positive replies were received than negative ones, some feedback may be disregarded since it does not

TABLE III
 ANSWERS TO TWO QUESTIONS ASKED

| In essence PAS uses the compiler available to you to assess your programs. What are the reasons for you still receiving valuable feedback? | How can PAS be improved to supply you with valuable feedback? |
|--|---|
| The system helped us (learners) to be more cautious in coding code implementation | it is unfair at times, the marking is unfair |
| It's straight forward regarding errors and gives the cause and it can be restructured | it should be more flexible since one can code in many different ways |
| Because when we received feedback the PAS the feedback would indicate where you lost marks and why | PAS should mark the code not only the end product, because we sometimes do not know where the error is, when the program does not show the results installed PAS and ensure that it make code line to line instead of marking the whole end product |
| The feedback focused on valuable points that are relevant to what we worked with in our project | it should be able to detect wrong spelling and supply auto correct |
| Personally, I believe that My SQL indicates to you when you've messed up | it should at least give you feedback and reasons where you went bad and at least give you tips on how to improve |
| The feedback that I have been receiving has been fair though not transparent | When you use data integrity options such as the cascaded delete options it requires that the table be populated in a specific order if inserted in the wrong order you get a zero |
| It indicates clearly where faults were | feedback unfair |
| Provides an indication of marks and the criteria it's looking for | one small mistake could cause you to fail. It does not test your knowledge or understand of MySQL |
| It's informative and helps to learn from mistakes | make space or reservations for programs that will not look exactly the same as that of the lecturer but still get same output. |
| It saved a lot of time | NO. it can show what I did wrong |
| | it can tell us where we went wrong |
| | did not any feedback at all |
| | it did not give feedback and if it did give feedback it was technical |

The answers listed in Table III highlights that the feedback PAS provides is limited and may be improved. This, to some extent, it true since feedback is taken directly from the compiler as would be in a real environment – which is the same environment learners have access to. It is encouraging that some learners find PAS feedback to be sufficient to help them to be more “cautious” when coding.

When asked whether learners think PAS to be a fair assessment tool 13 learners (36.1%) found it to be fair, 22 learners (61.1%) found it to be unfair and one learner (2.8%) supplied no answer. A number of 10 answers were received that may be perceived as positive and 20 being negative. Table

relate to the features of PAS or it is not contributing; including “it does not recognise some of the things such as PK”, and “the program is not fair sometimes because it is biased”.

Although the three negative comments are vague, it may relate to the fact that the programming skills and knowledge of learners are not as well developed as it should be on third year level – which may explain why time is seen as limited and “sensitivity” – referring to the rigour of programming – is an issue. It is encouraging to note that some learners realise the value of PAS; the fact that feedback is received quickly, and feedback is given.

When asked whether learners find the feedback received from PAS to be valuable, 12 learners (33.3%) found it of value and 21 (58.3%) did not find it to be of value, three learners (8.3%) did not supply an answer. Table III includes the meaningful open-ended answers to two follow-up questions asked based on their initial yes-no answer.

IV includes the meaningful open-ended answers to this question.

It is encouraging to see that some learners see the value of PAS. Learners who indicated it as being of value, realised that “databases need no errors in the real world” and “it actually prepare you for the workplace”, PAS is used collaboratively with a rubric – “it matches our programs to the criteria”, and “correct answers are correct”.

Learners who indicated PAS as an unfair assessment tool, mentioned the fact that it is “case sensitive”, the fact that “it does not look at the actual coding of the program”, and “it marks final output and nothing in between”. One learner even

indicated that PAS is “it is not a valuable assessment tool, it tests your program not your knowledge”.

TABLE IV
 ANSWERS WITH A POSITIVE AND THOSE WITH A NEGATIVE POINT OF VIEW TO THE QUESTION ASKED “DO YOU THINK PAS IS A FAIR ASSESSMENT TOOL?”

| Positive (Yes) | Negative (No) |
|--|--|
| it keeps you on your toes and you can't afford even one mistake; databases need no errors in the real world, so it is relevant | the marking is case sensitive, which is a disadvantage |
| it marks all the programs the same way and give relevant feedback | it's not, since sometimes one can use the same program in a different way compared to PAS. |
| it matches our programs to the criteria | it does not look at the actual coding of the program |
| it is not biased, and it marked transparently and fairly | it doesn't mark the actual coding at the product |
| it focuses on the relevant information or errors that one in the program but it also oversees a couple of errors | there are no criteria that the user can see |
| I think it is because you avoid making unnecessary mistakes and it actually prepares you for the workplace. | absolutely not, if it could consider for e.g. our coding rather than just output because there is knowledge |
| I saw the reports that generate once an assessment is complete | it is good for self-evaluation maybe, but one way or the other you need a lecturer's feedback for a detail explanation |
| yes, it removes the human aspect of marking code so no favourites could be found | NO, at all it is not because it is so sharp-sharp if it was a human it should have been fair. |
| I believe it's fair because even though it makes a few mistakes, the lecturer was able to review and fix those | it is not a valuable assessment tool, it tests your program not your knowledge |
| correct answers are correct | if you make a small error (human error) the whole thing is wrong |
| | because you get wrong answers if your code is not identical to that of the lecturer |
| | NO, I did everything on the program but still got a 70 |
| | it does not mark fairly where mistakes may have been made |
| | No, it marks final output and nothing in between |
| | is too rigid and unfair |

The answers to the question on what learners perceive to be a fair way to assess third year programs can be categorised into six groups:

1. The biggest group of learners, in total 12 (33.3%) suggested that their programs should be marked manually by the lecturer, one answer reflects what learners feel: “a lecturer must at least sit down and inspect each learner's program to see difference in programming, understanding & styling”.
2. Three learners (8.3%) preferred PAS.
3. Four learners (11.1%) suggested an improved PAS – their answers include “something more reliable”, “it may be fair if it shows what the learner did wrong so that it helps improve the program”, “it could subtract marks where mistakes could have been made”, and “to distribute the report of PAS results to the learners”.
4. There were five learners (13.9%) who suggested that PAS should be used, but in combination with the lines of code being marked manually.
5. One learner (2.8%) suggested peer assessment.
6. One learner (2.8%) suggested that a model answer should be supplied.

Ten learners did not answer the question.

Reflecting on the results provided by PAS and the feedback from learners, the PAS development team came to the conclusion that most learners code only enough to either get their programming code executing, regardless of whether it is working properly or not. This is possibly why learners would prefer hand marked programming code rather than have a software program mark it. They rely on the sympathy of the lecturer marking who may give them just enough marks to pass. Here it is important to note that the value of PAS is to

highlight the fact that learners tend to stop the development of their programs before it is fully functional - due to the fact that they do not do proper testing. With the use of proper test data, error messages which did not occur when the learner runs the program with insufficient test data, occurs during assessment.

Learners made some valuable suggestions:

1. Improve PAS's sensitivity to spelling mistakes and case sensitivity.
2. Review individual parts of the coding and not just the input versus the output.
3. The inclusion of corrective measures in the implementation of PAS.
4. The utilization of PAS for self-assessment, it seems that the single submission opportunity does not make sufficient provision for learning.

The first point listed above, may be addressed in future by making a special effort to make learners aware of the fact that, in a business environment, newly developed programs use existing functions, procedures, and methods, as well as existing databases with its associated tables and attributes. Making human errors, such as spelling mistakes will result in a program not fulfilling its function. It therefore is of utmost importance to code with care.

With the second point, learners should be made aware of the value of a test data set – compiled with the design (and in real life the analysis as well) of a program. The purpose of such a test data set is to ensure that all possible options are tested – which implies that all included functions, procedures and methods are tested. It is also important to make learners aware of developing their own coding fingerprint.

With point three learners should be made aware that their value as programmers is lying in the fact that they produce a

program that solves a problem. They need to find the solution to the problem they are faced with. The internet provides quite an extensive library of solutions to the types of problems learner receive from the compiler. Another valuable source may be to use peers (in this case they are in the same situation having to solve the very same problem) to support reflection on problems. More than one person working on a problem may support the process of finding a solution to a problem (especially when used in conjunction with the last suggestion). Learners should be encouraged to search for the solution to the set problem. By doing this, learners extend their knowledge in problem solving by making use of available resources.

The fourth suggestion concerning utilising PAS for self-assessment is feasible. The lecturer realised that the one-time submission and feedback version of PAS is not sufficient to assist learners in learning what is expected from them at the third year level. A PAS that allows more than one time-independent submission may be helpful in allowing learners to understand what is expected from them when receiving PAS feedback; they need to correct code based on PAS feedback. This suggestion allowed the PAS developers to improve PAS to allow multiple time-independent submissions. The following section provides a brief description of this new improved implementation.

VI. CURRENT STATE OF PAS

After reviewing the results from the pilot evaluation of PAS, the PAS development team decided to adapt the system, to allow multiple time-independent submissions. The following section gives a brief overview of the second iteration of PAS.

The goal of the second iteration of PAS, is to allow multiple time-independent program submissions of code, each with immediate feedback on the learner's submission. For this iteration, programming code is stored on the server and the server would execute events based on submission status. Learners may use their program's submission feedback to correct their mistakes and submit again. No limitation is placed on the number of submissions. Fig. 6 shows this process.

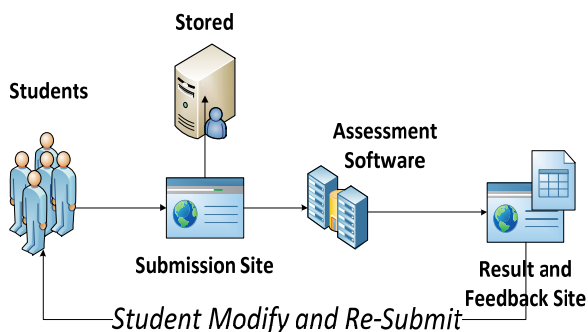


Fig. 6 Multiple Time-Independent Submissions

The second iteration of PAS was tested with a class of 132 third-year learners. This evaluation of the second iteration forms part of future research to enable on-going development

of PAS. At this point learners have received and completed two programming assignments. They were also assessed summatively.

For the first assignment 78 learners handed in their assignment after using PAS 268 times. It is interesting to compare their average mark, which is 56%, to that of a similar assignment in 2015, where the assignment average was 17%. For the second assignment 116 learners handed in their assignment after using PAS 751 times. Comparing their average mark, which is 85% to that of a similar assignment in 2015, where the assignment average was 34%. During the second assignment cycle PAS revealed that learners start to accommodate different input to thoroughly test their programs – to produce correct output. The increased use of PAS since the first assignment cycle is encouraging. Fig. 7 shows the result of using the improved PAS. It is important to note that for formal assessment purposes the first version of PAS, where a once-off time dependent submission is allowed, is still used.

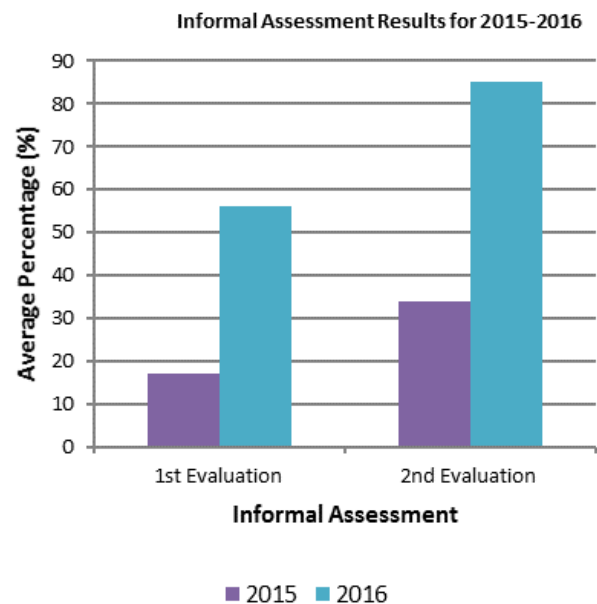


Fig. 7 Graph comparing informal assessment results from 2015 (use of PAS once-off time dependent submission in preparation) to that of 2016 (use of PAS multiple time-independent submissions in preparation)

It is true that one now expects an improved mark since many learners went through more than one feedback cycle. Unfortunately, summative assessment did not result in improved marks. This is represented in Fig. 8.

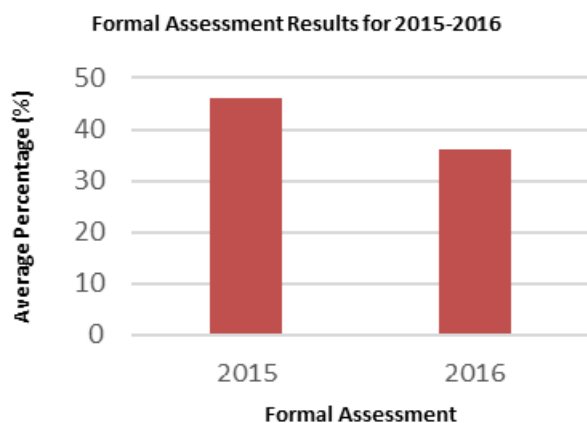


Fig. 8 Graph comparing formal assessment results from 2015 (use of PAS once-off time dependent submission in preparation) to that of 2016 (use of PAS multiple time-independent submissions in preparation)

VII. CONCLUSION

In this paper, CAA and TAA literature were discussed to provide supportive theoretical background to the development of a technology artefact, called PAS. The PAS is a system developed for the assessment of programming code. In the current context, PAS evaluates third-year SQL code as part of a database subject module. It is a TAA tool which enables a TAL environment for learners regarding programming. The pilot implementation of PAS allowed for a once-off time dependent submission. This once-off time dependent submission PAS were evaluated by means of questionnaires distributed to the learners at the end of 2015. The feedback provided by these questionnaires was valuable in the evaluation of PAS. It showed that learners appreciate the value technology may add in facilitating feedback in programming and it gave an indication of limitations. These limitations surfaced as the main obstacles in learners' use of PAS. The impediments regarding some of the improvements suggested, as well as the view that the lecturer has regarding the learning that needs to take place, has been discussed. With this in mind, the researchers looked at actions regarding awareness that may help future learners to use PAS effectively with its perceived limitations, while looking at the use of PAS from a vantage point that learners need to prepare themselves for the job market. One limitation presented an opportunity to improve PAS. The second iteration of implementation enabled multiple time-independent submissions of programs.

It is with anticipation that the researchers continue with this research project. The current usage of PAS is encouraging. Future research will focus on how well the new measures implemented is received and whether it makes a difference to the standard of programs produced by learners, especially when a more encompassing plan to support learners are put in place during 2017. A similar questionnaire to the one learners completed in 2015, will be completed by current learners at the end of 2016. It is hoped that learners will direct further development with their inputs. It would also be interesting to

see whether their attitude towards PAS changed with the implementation of both versions.

REFERENCES

- [1] Staley, D.J. and D.A. Trinkle, The changing landscape of higher education. *Educause Review*, 2011. 46(1): p. 16-33.
- [2] Draft, S., *Computer Science Curricula 2013*. 2012.
- [3] Attaway, S., *Matlab: A Practical Introduction to Programming and Problem Solving*. 2009: Elsevier Science.
- [4] Cormen, T.H., et al., *Introduction to Algorithms*. 2001: MIT Press.
- [5] Deitel, H.M. and P.J. Deitel, *Visual Basic 2005: How to Program*. 2006: Pearson/Prentice Hall.
- [6] Deitel, P.J. and H.M. Deitel, *Visual C# 2012: How to Program*. 2013: Pearson Education, Limited.
- [7] Brown, S.A. and A. Glasner, *Assessment matters in higher education: Choosing and using diverse approaches*. 1999: McGraw-Hill International.
- [8] Hui, W., et al., Technology-assisted learning: a longitudinal field study of knowledge category, learning effectiveness and satisfaction in language learning. *Journal of Computer Assisted Learning*, 2008. 24(3): p. 245-259.
- [9] Jenkins, M., *Unfulfilled Promise: formative assessment using computer-aided assessment*. *Learning and Teaching in Higher Education*, 2004. 1(1): p. 67-80.
- [10] Charman, D., *Issues and impacts of using computer-based assessments (CBAs) for formative assessment*. S. Brown, J. Bull and P. Race (Eds.), *Computer-Assisted Assessment in Higher Education*, 1999: p. 85-94.
- [11] Sadler, D.R., *Formative assessment and the design of instructional systems*. *Instructional science*, 1989. 18(2): p. 119-144.
- [12] Garrison, D.R., *E-learning in the 21st century: A framework for research and practice*. 2011: Taylor & Francis.
- [13] Gregor, S. and A.R. Hevner, *Positioning and presenting design science research for maximum impact*. *MIS Quarterly*, 2013. 37(2): p. 337-355.
- [14] Hevner, A.R. and S. Chatterjee, *Design Research in Information Systems*. *Integrated Series in Information Systems*, ed. R. Sharda and S. Vob. Vol. 22. 2010, New York Dordrecht Heidelberg London: Springer. 320.
- [15] Hevner, A.R., et al., *Design Science in Information Systems Research*. *MIS Quarterly*, 2004. 28(1): p. 75-105.
- [16] Peffers, K., et al., *A Design Science Research Methodology for Information Systems Research*. *Journal of Management Information Systems*, 2008. 24(3): p. 45-77.
- [17] Vaishnavi, V. and K. Kuechler, *Design Research in Information Systems*. 2004 16 August 2009 (cited 2013 14 June); Available from: <http://desrist.org/design-research-in-information-systems/>.
- [18] Iivari, J., *A paradigmatic analysis of information systems as a design science*. *Scandinavian Journal of Information Systems*, 2007. 19(2): p. 39.
- [19] Saunders, M., P. Lewis, and A. Thornhill, *Research methods for business students*. 5th ed. 2009, Harlow England: Pearson. 614.
- [20] Oates, B.J., *Researching Information Systems and Computing*. 2006: SAGE Publications.
- [21] Seaman, C.B., *Qualitative methods in empirical studies of software engineering*. *Software Engineering, IEEE Transactions on*, 1999. 25(4): p. 557-572.