### SAIEE Africa Research Journal

<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Computer Based Assessment System for Undergraduate Electrical Engineering Modules</td>
<td>P.A. van Vuuren, A. Alberts and L. Hager</td>
<td>126</td>
</tr>
<tr>
<td>An Improved Fundamental Design Influencing Methodology in a Constrained Project Management Environment</td>
<td>A. Wessels and L. Pretorius</td>
<td>131</td>
</tr>
<tr>
<td>Spatial-Spectral Watermarking Scheme for JPEG Steganography</td>
<td>B. Marques, B. Sim and L. Cheng</td>
<td>154</td>
</tr>
</tbody>
</table>

---

**SAIEE AFRICA RESEARCH JOURNAL EDITORIAL STAFF**

IFC
A COMPUTER BASED ASSESSMENT SYSTEM FOR UNDERGRADUATE ELECTRICAL ENGINEERING MODULES

P.A. van Vuuren *, A. Alberts † and L. Hager ‡

* School of Electrical, Electronic and Computer Engineering, North-West University, Private Bag X6001, Potchefstroom, 2520, South-Africa. E-mail: Pieter.VanVuuren@nwu.ac.za
† E-mail: Andreas.Alberts@nwu.ac.za
‡ E-mail: Louw.Hager@nwu.ac.za

Abstract: Good assessment is an essential component of good learning. Marking test and exam papers for large classes is however a time-consuming and repetitive task. This paper reports on the design and implementation of a computer based assessment (CBA) system suitable for undergraduate engineering modules. This prototype CBA system is capable of automatically grading relatively open-ended analysis and design problems and is also relatively user friendly. This CBA system consists of two separate subsystems, namely a test delivery subsystem and an automatic grading subsystem (which is the contribution of this paper). Automatic grading of the student answer papers is performed in Matlab®. The grading subsystem can automatically grade typed student answers containing symbolic mathematical expressions as well as numeric answers. Grading is performed accurately, consistently and very fast compared to a human lecturer. Students are given partial credit for partially correct answers and feedback is given in a very intuitive form quite close to the traditional manner in a pen-and-paper test. The system has been successfully used to take final exams in a second year engineering module on basic circuit analysis.

Key words: Computer based assessment, engineering education, automatic grading.

1. INTRODUCTION

Large classes are common in modern university level education. This is especially true of first and second year engineering modules. This situation places enormous pressure on lecturers to give timeous feedback to students after tests or exams have taken place. Despite being an onerous task grading of test papers is very important, since good assessment (including detailed feedback) forms one of the pillars of good learning.

Traits of a good test or exam are: accuracy, consistency and speed. The accuracy of a test refers amongst other things to the correctness of the grading of the student’s answers [1]. Tests should also be marked consistently for all students in the class. Lastly, students should also receive speedy feedback on the correctness or quality of their efforts. The speed with which students receive their grades has a huge impact on the effectiveness of the assessment in the learning process [2]. This is obviously becomes a significant problem in large classes.

One solution to this problem is through the use of multiple-choice questions. Computer-based assessment (CBA) systems making use of multiple-choice questions can give near-instantaneous feedback, but students are often critical of the inability of multiple-choice type assessments to give partial credit for the method that they used [3]. Large problems requiring multiple stages of calculation furthermore aren’t suited to multiple-choice questions. This limitation can be circumvented by dividing a large problem into smaller sub-problems which in turn can be assessed by means of multiple-choice or fill-in-the-blank type questions [4]. It is however our opinion that such an approach carries the risk of guiding students through the solution of larger problems.

Despite their limitations, tests consisting of so-called "teacher-supplied answers" (e.g. multiple-choice, fill-in-the-blank and true/false questions) still form the mainstay of most modern CBA systems e.g. OASIS [5], I-ASSESS [6] and QuestionMark’s Perception [7]. Modern CBA systems are however capable of much more than mere multiple-choice questions as shown in [1]. In their paper Scalise and Gifford propose a taxonomy of e-learning assessment on the basis of the degree to which the problem is constrained and also the relative complexity of the problem [8]. From this taxonomy one can conclude that modern CBA systems can assess almost anything ranging from true/false questions to written essays [1]. It should come as no surprise that CBA systems have been used in a wide variety of disciplines ranging from hydraulic engineering [9] to pedagogical psychology [10].

One approach to extend the capabilities of CBA systems entails making use of existing technology to automate certain aspects of the grading process. Harnessing the power of symbolic mathematics engines, some CBA systems allow students to give answers in the form of mathematical formulae. Determining whether the student’s answer is mathematically equivalent to the memorandum answer is performed by means of software routines. Examples of such CBA systems are AiM [11], CABLE [12] and STACK [13]. These CBA systems have been applied successfully to assess first year calculus and algebra modules. It should however be noted that the type of questions used in these systems are still relatively simple convergent problems. Another disadvantage of STACK is
that it assigns partial credit to student answers by means of so-called "potential response trees" (directed graphs) that are cumbersome for a lecturer to create.

Another example of CBA systems capitalizing on existing software technology is the use of computer language compilers to automate the grading of computer programming tests. Since modern compilers can give quite detailed feedback on syntax errors and scripts can be written to automate the testing of working programs, it is no small wonder that even students’ computer programs can be semi-automatically [14] or automatically [15] graded by modern CBA systems. In fact, it is even possible to automatically assess not only whether a student’s program is syntactically and semantically correct, but also whether the student’s program exhibits more advanced programming skills (such as meeting execution time and/or memory constraints) [16].

An alternative approach is to endow the computer with a measure of artificial intelligence to allow it to recognize numerous variants on the correct answer(s). One such example is a CBA system designed to assess practical IT skills of students [17]. This system makes use of a rule-based expert system to give feedback on the student’s efforts. Rule-based grading has also been applied for the automatic assessment of free-body diagrams that are interactively drawn in a biomechanics module [18]. Bayesian networks have been used to automatically assess the performance of students performing a practical exercise in a virtual electronic laboratory environment [19]. The behaviour of the students is captured during their interaction with the virtual laboratory (by monitoring their mouse clicks and keyboard strokes). Each student’s knowledge, skills and insight in the particular experiment is then deduced by the Bayesian network from the observed behaviour and converted into a grade for the particular student.

Although only indirectly related to either formative or summative assessment the work of Hoeft et al. is noteworthy for their automatic assessment of computer drawn concept maps (in order to assess student’s knowledge representations) [22]. Another graph-based approach for the assessment of students’ structural knowledge can be found in [23] where quantitative features are extracted from student-drawn pathfinder graphs with the objective of predicting the students’ performance in exams taken at a later date. Recently, an interactive computer-based assessment system was presented that has been used for formative assessment of computing and technology students’ argumentation ability by means of argument maps [24]. In this system students are presented with a blank argument map which has to be populated from a list of options. This system is implemented by means of OpenMark [25] in Moodle [26].

A common denominator of a number of the above mentioned examples is that advanced CBA systems (CBA systems that can automatically grade questions that allow students more freedom to compose their own answers to the questions) are often limited in scope. Applying a CBA system that was designed with one particular subject in mind to another subject is sometimes difficult. Another limitation of advanced CBA systems is the ease of use for the lecturer. Providing a computer with the necessary alternative solutions to empower it to automatically grade highly variable answers often becomes a very time-consuming task.

In view of the above mentioned limitations we decided to design and implement a CBA system suitable for second year engineering modules. This system had to be capable of automatically grading relatively open-ended analysis and design problems and also be quite user friendly (requiring minimal additional skills from the students). The design process for this CBA system was guided by the following criteria:

1. Accuracy. Answers should be graded correctly and reliably.
2. Consistency. All students should be treated the same.
3. Ease of use. The prerequisite skills for all users of the CBA system (both students and lecturers) should be minimal. Furthermore, the feedback given by the automatic grading software should also be intuitive.
4. Speed.
5. Partial credit. Students should be given credit for the process that they followed to arrive at an answer, even if that process itself is only partially correct.
6. It must be possible for the student to write a closed-book test on his/her own computer.

The CBA system which is the subject of this paper consists of two subsystems: a test delivery subsystem and an automatic grading subsystem. The test delivery subsystem is responsible for delivering the test to the students and collecting their answers. The automatic grading subsystem is responsible for grading each student’s answers and presenting the results in a format that gives useful feedback to both the student and lecturer.

The main contribution made by the above mentioned CBA system is that students are allowed to answer relatively open-ended questions, which are then graded automatically.
In this paper the spotlight falls on the automatic grading subsystem. Section 2 briefly summarizes the structure of the entire system. More detailed discussions on the test-delivery subsystem and the grading subsystem are given in respectively sections 3 and 4. The entire CBA system was implemented in a second year engineering module during the first semester of 2012. Section 5 evaluates the performance of the automatic grading subsystem during its trial implementation. Recommendations for future work form the topic of the last section of this paper.

2. CBA SYSTEM OVERVIEW

The test delivery subsystem (figure 1) consists of a central server that communicates to remote computers via a computer network. A web-page interface is used to deliver the test to the students. Each question in the test consists of a field showing the question and a text box in which the student can type his/her own answer. When the student is done with a particular question or the entire test he/she clicks on a "submit" button which causes the student’s answers to be sent back to the server where it is stored in a database.

Currently, the student answers are graded offline. After the test has been completed, the automatic grading subsystem (figure 2) queries the database for each student’s answers and then grades them by means of the memorandum supplied by the lecturer. After each student’s entire set of answers have been graded, the answers are saved in a LaTeX file, which can be compiled into a PDF file showing both the student’s answers as well as the traditional red tick-marks indicating correct responses. When the entire class has been processed, an Excel spreadsheet is generated showing how each student fared in each question and also giving the final grade of the student.

As we can all appreciate, grading exam papers is largely a repetitive task. Depending on the nature of the question, some knowledge and understanding of the subject matter might be required to make sense of the students’ answers. Examples of such questions are open-ended analysis or design problems. Therefore, the fundamental hypothesis upon which the automatic grading subsystem of this CBA prototype is based is as follows:

Provided that the question, answer and memorandum are sufficiently detailed and structured, automatic grading doesn’t require true knowledge or insight into the specific subject being assessed.

The above mentioned hypothesis can only be valid in a real-life examination scenario if the potential variability in student answers can be constrained to be within acceptable limits. The variability in the student answers to a question can be limited in three different ways:

(a) By constraining the question. This topic is briefly addressed in the discussion on the test-delivery subsystem in section 3.2.

(b) By including different options in the memorandum. This tactic is often used in traditional pen-and-paper exams, where the lecturer beforehand knows which different strategies will probably be followed by the students and then accordingly includes all of the options in the memorandum.

(c) By endowing the automatic grading program with a measure of simulated intelligence. By simulated intelligence is meant a program consisting of a number of conditional statements that has the appearance of intelligent behaviour on very small set of tasks.

3. THE TEST-DELIVERY SUBSYSTEM

The test delivery subsystem is built on web based technology, in order to ease its deployment in modern heterogenous computing environments. It consists of two major parts, namely the web site on which the test is administered and the environment in which the test is delivered.

The test delivery software is built using a traditional LAMP (Linux, Apache, MySQL, PHP) stack. Written in PHP, it dynamically renders web pages showing the test questions and accepts form data from the clients containing the student’s answers to the test questions.

Modern web environments are very open by default, posing significant risks when summative tests are delivered using it as a medium. We overcome this obstacle by delivering the test in a secure environment, using a live
booth Linux client, configured with only the necessities for the test and nothing else. This way, otherwise insecure systems can be made more secure by deploying a non-intrusive temporary environment on the client machine which is not under the end user’s control.

More details on the various components of the test-delivery subsystem are discussed in the following few subsections.

### 3.1 Test-delivery software

The CBA test system is hosted on a Linux server running the CentOS distribution. CentOS is a freely available community driven rebuild of the sources of the commercial Red Hat Enterprise Linux distribution, widely renowned for its reliability and stability. We use the Apache web server, the PHP scripting language and the MySQL database server in our test system deployment.

Tests are encoded in text files using a very simple markup style, in order to code and deploy them easily. Students are registered in a database table containing their user IDs and passwords, so that only authorized users can log into the test. When a student successfully logs into the test program, the directory containing the encoded tests is parsed to determine which tests are available to the user. This information is written to a table that keeps track of session-specific parameters, such as the time the test has started, the amount of time left on the tests and the amount of times the test has been submitted. The available tests are then presented to the user.

When a user clicks on a test, the encoded text file containing the test is parsed and converted into a HTML page that is displayed in the browser. Currently, the system allows for auto-shuffled multiple choice, as well as free form text style questions.

Answers are saved when the user clicks on the submit button at the bottom of the test. An enhancement that can still be made is to submit answers in the background by means of AJAX queries as soon as they are entered, so that they are immediately persisted and to prevent the user from having to submit periodically [27].

At this point, the student must copy and paste the LaTeX code in the bottom window of the equation editor to the testbox in the test browser. The numerical answers and LaTeX code typed in by the student will then later be graded by the automatic grading subsystem.

In order to limit the potential variability in the students’ answers, each question is accompanied with a list of permissible variables to be used in the answers. The LaTeX versions of these symbols are also given in the test browser. These symbols can then be copied and pasted into the equation editor to construct the expressions that the student desires. In this manner the process of typing math by means of an equation editor can be speeded up significantly. In fact, is our experience that the entire process of typing in answers by means of the equation editor has only increased the time required for students to complete a test by a factor of at most $\frac{3}{2}$.

### 3.2 The equation editor

The *lingua franca* of most engineering and scientific disciplines is mathematics. It is therefore also very important to allow students to express their thoughts in a test or exam by means of mathematical expressions. Whereas writing math by hand is quite easy, typing mathematical expressions is a different story all together.

Various standards exist for typesetting mathematical symbols e.g. MathML (mathematical markup language) and LaTeX. Of the available standards LaTeX is probably the most compact and human-readable representation of mathematical expressions. Unfortunately learning to type math in LaTeX represents quite a steep learning curve which would be unreasonable to expect from students enrolled in e.g. a module on electric circuits. This problem can however be addressed by means of an equation editor which allows the user to construct WYSIWYG mathematical expressions by means of a mouse and drop-down menus and also convert it to the corresponding LaTeX code. One such equation editor is EqualX [28]. EqualX is a freeware equation editor primarily designed for the Linux operating system. This allows it to be relatively easily incorporated in the Ubuntu live boot disk used for this CBA test delivery subsystem.

An example of EqualX is shown in figure 3. Numerous mathematical symbols and templates can be accessed via mouse from the dropdown menus at the top of the screen. When anyone of the buttons in the dropdown menus is pressed, the corresponding LaTeX code is shown in the bottom textbox. The keyboard can then be used to construct any conceivable mathematical expression. The purpose of the centre window is to show the final typeset version of the LaTeX code (in human-readable form). The centre window therefore also serves the important purpose as quality control of the code that the student has "typed". It is actually quite simple: if the expression in the centre window looks correct (i.e. if it looks similar to what the student intended to write by hand), then the underlying LaTeX code is also correct.

In order to lock down the workstations the tests are being displayed on without installing any software or altering them in any way, we deployed a live boot image of the Ubuntu Linux distribution containing a minimal set of software packages. The live boot distribution loads itself from a DVD-ROM into a RAM disk on startup, and therefore does not interfere with the configuration of the computer in question in any way. Users writing tests in this stripped down environment only have access to a minimalistic browser and the equation editor software, and do not have root access to alter this configuration. This solution is similar to the zip disk approach proposed by Ko and Cheng [29] in that students are issued with
the live boot DVD prior to a test. Our solution however
allows all student answers to be conveniently stored on a
central database by means of a tightly controlled network
connection.

3.4 Authoring a CBA test

At this point it might be instructive to take a look behind
the scenes at how a test is authored by the lecturer in
this CBA system. The test question and any necessary
figures are produced by the word processing software of
the lecturer’s choice. The test delivery subsystem however
requires the test in a text file format, which precludes
the use of figures, tables and mathematical expressions
as in e.g. a normal Word-file. This limitation can be
circumvented by including each question in its entirety
as a picture file. The result is a text file with numerous
references to picture files, each containing a separate
question.

4. THE AUTOMATIC GRADING SUBSYSTEM

4.1 Subsystem layout

Grading a test or exam paper by means of a memorandum
is in essence a pattern recognition problem. It entails
comparing a student’s answers with the information given
in the memorandum. If a match occurs, then suitable credit
should be assigned to the student. To facilitate feedback
to the student, tick marks are traditionally used to indicate
both correct answers as well as the number of marks earned
by each correct answer.

The process of reading a student’s answer and comparing it
to a memorandum can be easily automated. Figure 4 shows
the main sequence of events in the automated grading
subsystem.

At first, the computer reads the memorandum encoded in
a text file and stores the correct answer(s) to each question
separately. Secondly, Matlab queries the MySQL
database containing the student answers to determine the
class list and determine the number of students in the class.
Then a main loop is engaged which cycles through all of
the students in the class. For each student, the program
retrieves his/her answer to each question separately. The
answer is then converted into a format suitable for later
LaTeX compilation. Finally, each answer is compared with
the corresponding part of the memorandum. This process
is denoted by the red rectangle in figure 4 and consists of
further operations which will be discussed shortly. After
all of the students in the class have been processed in this
manner, a final summary Excel report is generated. This
report indicates how each student fared overall as well as
in each question individually.

Obviously, the heart of the entire grading subsystem is the
component responsible for comparing a student’s answer

to an individual question to the corresponding section in the memorandum. Highlighted in red in figure 4,
this component itself consists of a loop calling different subroutines. This loop entails that each line in the memorandum is read, interpreted and suitably used to grade the student’s answer.

With the exception of the preamble, each line in the memorandum has the following structure:

\textbf{GRADING COMMAND:} (N) answer. The meanings of the three different components in the line are as follows:

- \textbf{GRADING COMMAND} - This is a reserved word that indicates how the computer must treat the text given in \textit{answer} and how the student is given credit. There are currently only six grading commands namely: \textit{SYMBOLS external}, \textit{SYMBOLS internal}, \textit{TICK}, \textit{TickNumEq}, \textit{ANSWER} and \textit{TickEq}. Each of these commands will be introduced shortly.

- (N) - An optional input is the number of marks allocated to a correct answer. The default value is one.

- \textit{answer} - A free-form string, number or LaTeX mathematical expression which indicates a correct answer.

As an example, consider the following memorandum line:

\texttt{ANSWER: (3) R_1 = 10 \text{\textOmega}}.

The above line instructs Matlab to search through the student’s answer to question \(x\) for a statement in which the student has calculated that \(R_1 = 10 \text{\textOmega}\). This correct answer is worth three marks.

For each line in the memorandum answer to a question, the following process is then followed:

1. Read the grading command, the value (N) of a correct answer as well as the correct answer.

2. If \texttt{GRADING COMMAND = SYMBOLS external} then a list is compiled of all allowable variable symbols that students might use in answering this question. This list contains numerous variations on spelling of the variable names given in the test paper as a safeguard against student typing errors.

3. If \texttt{GRADING COMMAND = SYMBOLS internal} then a list of aliases for the above list of variable names is compiled. Two objectives are attained with this command. Firstly, all variable names are encoded into a format suitable for Matlab’s Symbolic Toolbox.
Secondly, all the different variable symbols that can be used by the students are translated into a single list of variables for use by Matlab.

4. If GRADING COMMAND = TICK then Matlab searches through the student’s answer for an exact match to the string given in answer. This grading command is of limited use, since it is extremely sensitive for typing variations. In future versions of the CBA system, this grading command should be improved to be robust for typing and spelling errors.

5. If GRADING COMMAND = TickNumEq then Matlab searches through the student’s answer for a numerical value close enough to the given value in answer. If the correct answer is e.g. 15 μF, then any numeric value in the student’s answer that lies within a prespecified tolerance from $15 \times 10^{-6}$ is regarded as being correct. TickNumEq doesn’t require that the student use the correct unit, it is only sensitive for the correct value scaled by the SI prefix used for the unit.

6. If GRADING COMMAND = ANSWER then Matlab searches for a numeric value close to the given answer. This command is similar to TickNumEq with the exception that a few additional requirements have to be met before the student receives credit for his/her correct answer. These additional requirements are specified in answer. As an example the following answer: $L_1 = 20 \text{ mH}$ implies that the student must use (a) the correct variable (namely $L_1$) as well as the correct unit (Henry). This grading command is useful in design and analysis questions where the student has to determine specific component values or specific quantities in a circuit.

7. If GRADING COMMAND = TickEq then Matlab searches through the student’s answer for a mathematical expression that is equivalent to the one given in answer, e.g. $\frac{V - 220V}{12}$.

8. Repeat the loop until the end of the memorandum for this question is reached.

At present, the grading subsystem suffers from the limitation that no logical line of reasoning is enforced by the automatic grading software. This limitation can however be addressed relatively easily by means of pointers progressing through both the memorandum and a student’s answer. In this manner a student will only be given credit for correct answers if they are in the same sequence as in the memorandum.

Of the six grading commands introduced in section 4.1 only three warrant additional explanation, namely: TickNumEq, ANSWER and TickEq. (The Matlab code of the entire automatic grading subsystem is freely available from the authors.)

4.2 The logic of ‘TickNumEq’

The purpose of this grading command is to search through a student’s answer for a number which is within a prescribed tolerance from a given number in the memorandum. The following sequence of events occurs when TickNumEq is called in the memorandum:

1. The required (complex-valued) number is extracted from the memorandum. This step is actually quite involved since both the memorandum and student answers are encoded in text files. Strings therefore have to be read and converted into numeric values where applicable. Regular expressions can be used to great advantage to perform this step.

2. As soon as the memorandum number is extracted, the program checks whether the number is accompanied with a unit and SI-prefix (e.g. kilo). All units are discarded. If an SI-prefix is detected, then the memorandum number is re-scaled accordingly.

3. Now the attention of the program shifts from the memorandum to the student’s answer. All numbers are extracted from the string containing the student answer. For each of these numbers the following steps are performed:

   (a) Search for a unit and SI-prefix. If found, re-scale the number accordingly.

   (b) Subtract the found number from the memorandum number.

   (c) If the difference is within the prescribed tolerance, then the required number of tick marks are placed at the end of the relevant line in the student’s answer. The current sub-routine is also summarily ended and the attention of the automatic assessor is diverted to the next line in the memorandum.

4.3 The logic of ‘ANSWER’

This command is similar to the command TickNumEq with the exception that in addition to a correct numeric value the student also has to supply at least a correct variable and sometimes the correct unit as well. As one could expect, the logic of ANSWER is also quite similar to that of TickNumEq. Evaluating the command ANSWER entails the following procedure:

1. Translate all variable names in the student’s answer to valid Matlab variable names. This is done by making use of the lookup table compiled during the evaluation of the commands: SYMBOLS external and SYMBOLS internal as discussed in section 4.1.

2. Find the specific variable mentioned in the memorandum answer associated with the current grading command.

3. Find the correct numeric value of this variable from the memorandum answer. This numeric value is also rescaled appropriately if an SI-prefix is detected.
4. Find the unit specified (if any) in the memorandum answer.

5. Search for the next numeric value in the student’s answer string.

6. Search for any valid variable name immediately prior to the number extracted in the previous step.

7. Search for any valid unit and SI prefix immediately after the number extracted in step 5.

8. Test whether the student’s number is sufficiently close to the memorandum number. Also test whether the correct variable name and unit (if any) are present in the student’s answer. If both (or all three) of these conditions are met, then the student gets his/her due credit and the subroutine is discontinued. Otherwise, the search carries on until all numbers in the student’s answer have been evaluated in this manner.

4.4 The logic of ‘TickEq’

This command directs Matlab to search through the student’s answer string for a mathematical expression which is equivalent to the one supplied in the memorandum answer.

The basic idea upon which the software of this grading command is based is to convert a student’s LaTeX expression into a format acceptable for Matlab’s Symbolic Toolbox. Testing for mathematical equivalence in the Symbolic Toolbox environment then becomes a mere formality. The entire procedure can be summarised as follows:

1. Extract the LaTeX equation in the memorandum answer.

2. Convert the LaTeX expression into Symbolic Toolbox format. Regular expressions once again will simplify this process considerably. At present however the conversion from LaTeX to Symbolic Toolbox format is done by means of string operations. As in the case of ANSWER (step 1), all variable names both in the memorandum as well as in the student’s answer are translated to the correct internal format. The program responsible for converting LaTeX into Symbolic Toolbox expressions is obviously a work in progress. At present it is limited to fractions and algebraic equations.

3. Search through the student’s answer for the next candidate equation. A candidate equation is any line containing at least one equals (=) sign.

4. Parse the candidate expression into its main terms.

5. Convert each term into Symbolic Toolbox format and test for mathematical equivalence. Two expressions are equivalent if their difference is zero. This test can be easily performed in the Symbolic Toolbox. If the student’s answer is equivalent to the memorandum expression, then credit is allocated and the loop discontinued.

Although the actual program code of the automatic grading subsystem is quite complicated, the underlying principles are simple and intuitive. In its present form the grading software is generic and forms a useful foundation for further development. As the next section shows, the initial results that were obtained during the exams are encouraging.

5. RESULTS

This section gives an example of the feedback that the system can give to students. Thereafter the performance of the automatic grading subsystem is evaluated on the basis of the results obtained by students who wrote one of their mid-year exams in the CBA system, as well as on the basis of qualitative feedback obtained from the same students.

As mentioned earlier, the CBA system was implemented in a second year module on basic electrical circuits. This module is compulsory for all students in the Faculty of Engineering at the North-West University, but is presented separately to electrical and electronic engineering students on one hand (78 students in 2012) and chemical and mechanical engineering students on the other hand (217 students in 2012). The quantitative test results reported in this paper are those obtained by the latter group of students due to its greater size.

5.1 An example of a graded answer

Space limitations and ethical considerations prohibit the publication of an entire graded test paper. An example of a graded answer to a single question is however given in figure 5. This question required the students to calculate a specific voltage in a circuit by means of the well-known node-voltage method. Each red tickmark(√) indicates one mark (as is tradition). All of the marks obtained for a particular line in the student’s answer are placed either at the end of that line or immediately below it.

The design and development of the prototype CBA system was guided by the list of criteria given in section 1. How the CBA system measures up to each of these criteria will now be discussed in turn.

5.2 Accuracy

No errors should be made by the automatic grading subsystem. More specifically, correct answers shouldn’t be overlooked and neither should incorrect answers receive any credit. Quality in this aspect of the CBA system is ensured by the following three step process: (i) spot checks performed by the lecturer; (ii) internal moderators; and (iii) the students themselves.

An indirect measure of the accuracy of the CBA grading subsystem is the distribution of the marks obtained by the
students for the various assessments during the course of the semester. Two semester tests were taken by means of the traditional pen-and-paper method. Experience gained in presenting the module over the past few years has shown that the performance in the semester tests is a good indicator of the performance in the exam. This holds for both the individual student as well as the class overall. If this trend also holds true after the introduction of the CBA system, then it is safe to conclude that the automatic grading system can accurately perform its task.

Figure 6 shows the results of these two handwritten semester tests as well as the first CBA exam paper. These results reflect the performance of the 207 active students in the class of 217 Chemical and Mechanical Engineering students. These distributions show that there isn’t much difference between the results of the two handwritten semester tests and the CBA exam. Consequently, one can conclude that the introduction of the CBA system didn’t have a significant impact on the students’ performance. Furthermore, the general shape of the distributions seem to be roughly the same for both the handwritten tests and the CBA tests and is no cause for alarm.

5.3 Consistency

From a computer’s perspective each student reduces to nothing more than a file that has to be processed. Consistency is therefore an inherent trait of any CBA system, since it is impossible for an inanimate piece of hardware to exhibit a preference for one student over another.

The truth of the matter is one thing, but it is also important to take the perceptions of the students into account when evaluating the CBA system. With this in mind, an attitudinal survey was conducted amongst the students three months after the first semester exams. Lickert scale questions were used to measure their opinions and experiences on a wide variety of aspects of the CBA system. Only four response categories were used in the Lickert scale, namely: strongly disagree, disagree, agree and strongly agree. Unfortunately only 44 of the grand total of 294 second year students participated in the survey. This does cast some doubts on how representative the results of this survey are of the true general feeling amongst the students.

Two questions were posed in the survey to specifically measure the students’ perception on the fairness of the automatic grading subsystem. These questions were:

Q1 Do you feel that the computer is too rigid? In other words, would you prefer it if a person can read your answer and judge its merit?

Q2 Do you feel that the computer was truly consistent in terms of treating all students exactly the same?

Figure 7 summarizes the results obtained by these two questions. It is quite clear that all of the respondents felt that the grading system is too rigid. Unfortunately their preference for a human assessor that can judge their answers on merit unavoidably introduces subjectivity into the grading process. Whereas it is undoubtedly true that a human will often have mercy on a student due to his/her situation (which constitute external factors to the test), this does have a detrimental effect on the quality of the assessment.

With respect to the second question (namely whether the CBA system treated all students equally), the results in figure 7 indicate that the majority of respondents felt that the system did indeed treat all students exactly the same. Why 43 % of the students didn’t agree with this statement is puzzling and should be investigated more deeply. All of the above mentioned results should however be interpreted by taking into consideration that the CBA system was introduced during the last two weeks of the semester. The students therefore only had three weeks to get accustomed to the CBA system before writing the exam.
5.4 User friendliness

Any CBA system has two primary users, namely the lecturer and the students. The benefits of a CBA system for the lecturer in terms of grading test papers are obvious. Although test papers have to be converted in the correct format to be displayed in the test-delivery subsystem, this is indeed a small price to pay in return for having the grading performed accurately, consistently, quickly and above all automatically.

To a large extent the students’ experience on the user friendliness of the entire CBA system is determined by the test-delivery subsystem.

In terms of the user friendliness of the test-delivery subsystem, the following five questions were posed in the attitudinal survey:

Q1 Although the CBA test-delivery system was introduced only three weeks prior to the date on which I wrote the final exam, I was relatively confident with the CBA test delivery by the time that I had to write the exam.

Q2 Students in general will be able to use the test delivery system with confidence given enough time to get to know the system.

Q3 I prefer to write traditional pen-and-paper tests.

Q4 I will not mind writing tests or exams on CBA systems.

Q5 I found the CBA test delivery system user friendly.

The results obtained from these five questions are summarized in figure 8. The response to question 1 clearly indicate that the students felt that they had too little exposure to the system before they had to write their exam. The fundamental user friendliness of the system is however revealed in the response to question 2. A small majority of the class were of the opinion that they would be quite confident in the usage of the system, given sufficient exposure.

The typical resistance to paradigm shifts is quite evident in the response to question 3 (namely that the vast majority of the class prefers pen-and-paper tests). A significant minority of the class (29.5% to be exact) however wouldn’t mind to write tests or exams on the CBA system in the future. This is an important result, since it shows that the system isn’t fundamentally flawed, only introduced unwisely. Although the fifth question clearly shows that the majority of the respondents felt that the test-delivery subsystem was unfriendly, this opinion should be balanced by slightly more than a third of the respondents who experienced the system as user friendly.

In balance, the students felt that the CBA system is not user friendly; a feeling that can only partly be attributed to the late introduction of the system. Any prototype has significant room for improvement. Specific aspects of the test-delivery subsystem that need attention in future were identified from three open-ended type questions in the survey. These questions asked the students respectively what they regarded as the disadvantages and advantages of the system, as well as any suggestions for future improvement.

A large contingent of the respondents remarked that they had to perform draft calculations on paper before typing in the important parts of their answers. This two-stage process inconvenienced them, wasted a lot of time and was a cause of uncertainty (since they had to choose which parts of their calculations to type in and which not to). The last drawback mentioned by the students is very interesting, since it highlights the fact that the CBA system forces students to put extra thought into how they communicate their answers. One could therefore postulate that the CBA system has the added benefit of fostering logical and systematic thought in students. Support for the latter hypothesis can be found in the following interesting statement from one of the students in the survey: “The CBA system did not make it any more difficult to pass the module in the sense that the format is exactly the same as when the equations were written on paper. Instead it helped me to avoid writing down unnecessary mathematics and calculations.”
Surprisingly, a number of students also mentioned that typing *per se* was a problem for them. Being competent with a keyboard is however essential for professional engineers. It is therefore important for students to get even more exposure to electronic communication technology (even during tests and exams).

Lastly, a number of students implicitly recommended that the equation editor be embedded in the web-browser. This will undoubtedly improve the user friendliness of the system and ultimately increase the levels of user acceptance.

To measure the quality of feedback given by the automatic grading system, the following question was posed in the attitudinal survey: "Are you of the opinion that the graded answer papers delivered by the computer (the pdf report with red tick marks in it) gives informative feedback so that you can learn from your mistakes?"

The feelings of the students on this matter are summarized in figure 9. It is disappointing that only 39 % of the students regarded the graded answer papers as being useful feedback. The viewpoint of the majority of the students can probably be attributed to the fact that tickmarks are only placed at the end of a line, rather than in between the text as in a traditional pen-and-paper test.

5.5 Speed

The time required by an average student to complete a CBA test obviously depends on the student’s proficiency in the system. Seasoned LaTeX users can produce mathematical expressions with minimal delay, while novices will take a long while to type in an equation by means of a mouse.

Our practical experience with the system has been that it takes an average student approximately two minutes for each mark in a test. A typical three hour exam can therefore only have a maximum of 90 to 100 marks. This compares favourably to the three hours and 120 marks allocated for previous handwritten exams in the module. One can therefore conclude that the CBA system only has a small impact on the time required by students to answer tests.

This conclusion is supported by the results of the student survey. Figure 10 reflect the students’ experience of how long it took to complete a CBA test. The results of two questions are combined in figure 10, namely:

Q1 I found that I took longer to answer questions in the CBA system than in handwritten tests and exams.

Q2 I could answer all of the questions that I knew the answers of in the time provided to do the exam on the CBA system.

Clearly, all of the students felt that it took longer to complete a CBA test, than a traditional pen-and-paper test. This result should however be seen in the perspective of the results of the second question, from which one can conclude that only a small majority of the class felt that they didn’t have enough time to communicate the answers that they knew in the CBA system.

The issue of the time required to answer a CBA test can therefore be easily addressed by the following measures. Firstly, students should be given exposure to the system from as early in a semester as possible. Furthermore the contents and length of a CBA test should be adjusted to fit within the constraint of three hours.

The savings incurred during the automatic grading of answer papers is obviously enormous. The mere fact that this part of the work is performed automatically already implies that the lecturer can continue with research during office hours and leave the grading of tests over to the computer while he/she is otherwise occupied. Still, the computer does perform the task of grading answer papers significantly faster than humans can. More precisely, it takes on average 60.5 seconds to automatically grade an entire exam paper of a student. To place these numbers in perspective, it took approximately 23 minutes in previous years to mark a single exam answer paper by hand. The relatively small amount of additional effort required from the lecturer to compile a machine-readable memorandum, is therefore handsomely compensated for by the subsequent automatic grading process.
5.6 Partial credit

As we’ve seen in the previous section, the automatic grading subsystem can grade a full length exam paper in approximately one minute. Whereas this speed is impressive compared to humans, it is still far from the near-instantaneous feedback offered by multiple-choice tests. The difference lies in the detailed feedback and partial credit offered by the prototype CBA system.

It should be clear from the example given in this paper (figure 5) that the system does indeed give students partial credit for what they did correctly in a question. The grading commands comprising the core of the automatic grading subsystem (see sections 4.1 to 4.4) allow lecturers numerous creative ways to allocate marks in a test. This makes it possible to test more than one aspect of a module in a single question and yet give the student credit for each part that he/she could answer correctly.

It is interesting to observe that once again the students didn’t share the above mentioned viewpoint in the attitudinal survey. Two questions on the theme of obtaining partial credit were included in the survey, namely:

Q1 Do you think that the computer system can give credit for more than one possible approach to answer a question?

Q2 Do you feel that the computer gave you credit for following the correct process in certain questions even though your final answers were sometimes incorrect?

The histograms in figure 11 summarize the student’s feelings on the topic of the ability of the grading system to assign partial credit. Clearly, the majority of the respondents are of the opinion that the system can’t assign partial credit to their answers. This majority does however correlate quite well with the general negative disposition of the class towards the CBA system (due to its late and forced introduction in a module that is highly unpopular amongst the majority of the students in the faculty).

6. CONCLUSIONS AND RECOMMENDATIONS

The prototype CBA system was primarily developed with an introductory course on electrical circuits in mind. Even though it is more suitable for senior undergraduate engineering subjects than other CBA systems currently available, it is still quite limited in its capabilities. These limitations should however be seen as fertile soil for further research and development, rather than critical flaws.

- Grading is performed offline once the test has been completed by all of the students. User acceptance of the system will obviously be boosted if grading could be implemented online as well.

- No automatic correction of spelling and typing errors is performed. The grading software is therefore very sensitive for spelling errors by the students. This however didn’t pose much of a problem in the modules in which this system has been implemented, due to the mathematical nature of both modules.

- Graphical input can’t be graded at present. Only numerical answers and mathematical expressions can at present be reliably graded.

The current version of the system is limited to modules with a significant mathematical content. Work is however underway to extend its abilities to automatically grade textual answers (e.g. paragraph-style questions) in other Engineering modules.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the help and advice given by Gustav Otto, Celeste du Preez, Steyn Geldenhuys as well as the anonymous reviewers who improved the quality of our paper.

REFERENCES


AN IMPROVED FUNDAMENTAL DESIGN INFLUENCING METHODOLOGY IN A CONSTRAINED PROJECT MANAGEMENT ENVIRONMENT

A. Wessels* and L. Pretorius**

* Doctoral student, Department of Engineering and Technology Management, Graduate School of Technology Management, University of Pretoria, Lynnwood Road, Pretoria, 0002, South Africa, E-mail: ariewessels@gmail.com
** Professor, Department of Engineering and Technology Management, Graduate School of Technology Management, University of Pretoria, Lynnwood Road, Pretoria, 0002, South Africa, E-mail: leon.pretorius@up.ac.za

Abstract: The aim of this paper is to develop conceptually from literature an improved design influencing concept in a constrained project management environment. The project management and systems engineering interface pertaining to project cost and schedule has been researched. Design as part of Systems Engineering (SE) and Project Management (PM) is an important process for product development. The interaction between SE and PM can influence the success of a product development project. A model has been developed to better understand why design iterations are fundamental to the design process. This model has been expanded into a constrained design influencing model that provides a better understanding of the influence of project management in the design process. This model shows that the project manager, particularly if he is under unrealistic constraints, can force a premature design release for integration to the next system level. This model helps to provide a fundamental understanding of the design process.

Key words: Design, project, management, system, engineering, iterations

1. INTRODUCTION

The aim of this paper is to develop conceptually from literature an improved design influencing concept in a constrained project management environment.

SE is an interdisciplinary approach encompassing the entire technical effort and processes of a project in order to develop a successful system that will satisfy a customer’s needs [1, 2]. Design as part of the systems engineering process is an iterative and dynamic process and is a fundamental part of the systems engineering process for the development of the individual system components [3]. The individual system components when integrated will function as a whole to provide the required system functionality.

The systems engineering process is well documented in the systems engineering handbooks [1, 2]. Design iterations are fundamental to the systems engineering process, primarily due to the design influencing needed to drive the design to maturity. This will be further discussed in section 4. A design is successively refined until it is mature and acceptable for further integration into the system as shown in Figure 1.

According to NASA [2], successive refinement involves a recursive and iterative design loop driven by a set of stakeholder expectations where a draft architecture/design and the derived requirements are developed. Each step also involves an assessment of potential capabilities and potential pitfalls identified through experience-based review of the data associated with lessons learned from other projects [1, 2].

PMBOK [4] defines a project as a temporary endeavour in that it has a defined beginning and end in time, and
therefore has scope and resource constraints. Project management is the application of knowledge, skills and techniques to execute projects effectively and efficiently [4]. Project management is a structured milestone driven process.

The SE process is a “static” process in the sense that the processes have no schedule constraints. The SE process does not place any constraints on either the activity time, or a resource requirement on the individual process steps [1, 2].

A system cannot be developed using the systems engineering process by itself. A systems development project requires both the SE and PM processes. Figure 2 illustrates the two processes and areas of overlap. SE requires PM amongst others, to coordinate and manage the schedule as well as the consumption of resources to ensure ultimate project success, [5].

In order to develop an improved design influencing model, the effects of both the SE and PM processes should be taken into consideration, illustrated in Figure 2. This paper identifies the fundamental mechanisms that result in design iterations and the influence that management has on this process.

In order to develop an improved design influencing model, the effects of both the SE and PM processes should be taken into consideration, illustrated in Figure 2. This paper identifies the fundamental mechanisms that result in design iterations and the influence that management has on this process.

Figure 2: Systems Engineering environment [5].

2. PROBLEM STATEMENT

Since a process always functions within another process, the interface between the two processes can have a distinct influence on the project’s performance.

A SE design team’s primary objective is technical compliance with the design requirements. To achieve this, a design generally goes through a number of iterations until all the requirements have been achieved. The number of iterations required for a specific design cannot always be accurately determined at the start of the project. At best the number of iterations can be estimated from past experience depending amongst others on the maturity of the selected technology.

The project manager’s primary focus is project cost and schedule. However, the project management process requires that all activities and resource expenditure must be accurately planned and managed at the start of a project. Iterations are not inherently supported by the PM process unless defined and planned [4].

The consequence is that indeterminate loops which are an essential part of the SE process for design optimisation are not supported by the PM process.

To further exacerbate the problem, teams behave in accordance to how they are measured [6]. The metric for design team success is primarily technical performance and compliance of a design to requirements. The metric for project management team success is primarily scope achievement, cost and schedule. Both criteria are necessary for the success of a project and customer satisfaction.

Checkland and Scholes [7], recommend that for the smooth and efficient running of a complex systems development project a soft systems methodology be used. Such a process cannot always be quantified and measured and depends entirely on the cooperation and team spirit of the individual members of the project team.

The consequence of the difference between the two processes for the development project is that the project manager endeavours to curtail the number of design iterations. This creates a conflict situation with the design team.

The negative impact of the conflict situation can be mitigated by optimising the effectiveness of the design team as will be discussed later.

3. RESEARCH OBJECTIVES

The aim of this paper is to determine the fundamental mechanisms that give rise to design iterations. Once these mechanisms are fully understood it will be possible to optimize design influencing and determine the effect that PM has on the system development process.

Design Science Research (DSR) methodology has been followed. DSR is to observe, analyse and understand the design process [8].

4. DESIGN INFLUENCING

Engineering design often involves a very complex set of relationships amongst a large number of coupled problems. The complex coupling leads to iteration among the various engineering tasks [9].

Buede [10] states that design influencing is a process to improve the future status of the product, and one that culminates in the allocation of resources to affect the chosen change. The objective of design influencing is to accelerate design optimisation with the aim of driving the design to maturity. The earlier this is addressed in the design process, the lower the cost impact of a design change will be [10, 11].
Before design influencing can be considered in detail, and a model developed, it is necessary to have a clear perspective of the most basic requirements of the design in question. A good design must, amongst other factors, function properly within its design parameters and environments and be cost effective. These environments are external influences and are at best predictions that cannot be controlled by the designer. To ensure that a design always behaves in a controlled and orderly fashion the designer must also consider the design’s behaviour for out of specification conditions. A good example would be a software module processing the data from an external sensor. If the sensor provides data that is erratic and/or out of specification, the software must behave in an orderly manner and must not hang-up, but elevate the condition to the next system level [22].

Therefore, a good SE design team must not only focus on the technical requirements of a design but also on the constraints and external conditions which are inherently imposed on the design. This requires two different and almost opposing mindsets which are very difficult to vest in one design team alone.

Design influencing can be made more objective and repeatable by the application of influence diagrams and decision trees [10, 12]. Design influencing can be further refined by applying success frame and failure frame considerations to a design, [12].

The studies by Kim and Kang, [13] and Kuhn and Poole [14], found that teams that developed integrative conflict management styles made more effective decisions than teams that utilized confrontation and avoidance styles. They also found that teams that never developed a stable style were less effective than teams with integrative styles. Also Kim and Kang [13], found that cross-functional cooperation between teams in new product development had a positive impact on product development performance. A topic under review by different mind-set work groups will be looked at from all aspects and not just from one aspect as would be the case for a homogeneous mind-set group [13, 14], thus producing better results.

Applying this principle, the effectiveness of a design team can be improved by dividing the teams into two diverse groups addressing different aspects of the design process. One group to focus on the functional requirements and another group to focus on the non-functional requirements.

To achieve the functional requirements, the design team must focus on design success. The design team must focus on all aspects to make the design work. The mindset of the team focussing on compliance with the functional requirements therefore works in the design success domain.

To address the non-functional requirements, the design team must focus on how the design can fail and how it must behave under those conditions to achieve the requirements. The mindset of the team focussing on the non-functional requirements can therefore be said to be working in the failure domain.

Such a division would lead to a Success Domain (SD) and Failure Domain (FD) team. The SD design team would then focus on the functional requirements whilst the FD design team would focus on the non-functional requirements. This will create a constructive conflict design environment. Now it is possible to develop a model to study the interaction between the two domains.

Applying these principles to improve team interaction and effectiveness, the two opposing but complementary design teams can be constituted by utilising the following two groups:

- A system/subsystem development team, referred to as the Success Domain (SD) team.
- A logistics engineering development team referred to as the Failure Domain (FD) team.

The Success/Failure domain concept is shown in the Figure 3:

![Success/Failure domain concept](image)

**SUCCESS DOMAIN**

**FAILURE DOMAIN**

Figure 3: Success/Failure domain concept [12].

5. DEVELOPMENT OF AN IMPROVED DESIGN INFLUENCING MODEL

Dividing the design team into FD and SD groups will ensure that a design is thoroughly analysed and evaluated from all aspects before being released for integration at the next system hierarchy level. This will reduce the risk of an unexpected forced design chance during the system integration phase that will consume unplanned project resources.

5.1. Success domain team (SD)

The “Success Domain” design team must strive for design success. In other words the mindset of the SD team is: “what is the minimum acceptable success?” The mind-set of the SD team comprising systems engineering, subsystem development teams and design engineers are therefore set in the “Success Domain”. This team’s objective is to get the system, subsystems and associated software working in compliance with the requirements and development specifications.
5.2. Failure domain team (FD)

The “Failure Domain” team must identify design weaknesses. In other words the mindset of the FD team is: “what is the maximum tolerable failure and what are the weaknesses in the design?” The mind-set of the FD team is failure mitigation of the design. The whole objective is to analyse the system, subsystems architecture and designs to determine what makes them fail and what the maximum tolerable failures are.

5.3. SD-FD team interaction

A system can only be developed in a project management environment, since project management provides the time function (schedule) to the system development project, [5].

Placing a project management time function on the Success Domain (SD)-Failure Domain (FD) requirements and constraints, a dynamic design influencing model can now be developed shown in Figure 4. This model makes the static design influencing processes illustrated in Figures 3 dynamic. The model in Figure 4 shows the iterative design influencing process between the SD and FD teams. The objective of both teams is a successful compliant design.

Figure 4: Interaction between the SD and FD teams

5.4. Application of the SD-FD design influencing model

One team is responsible for the development and architecture of the system whilst the other team is responsible for the design analysis.

Eisner, [15], states that if there is no coherent design, there is nothing to analyse. This implies that the SD team must first provide a concept design before it can be analysed by the FD team. Only when the Success Domain (SD) team makes a draft design available, can it be analysed by the Failure Domain (FD) team and feedback provided to the Design Review Board (DRB). In practice this is an informal iterative process between the SD and FD teams with short iterative cycles.

Expanding Figure 4 showing the interaction between the SD and FD teams, an unconstrained design influencing model can now be developed. Once the SD team has prepared a concept design, it can be analysed by the FD team and submitted to the DRB. The DRB will then order another design iteration if the concept design deviates from the design requirements. The design iterations will be repeated until all the design requirements have been satisfied. Once the design is acceptable, the design baseline is fixed and released for further integration into the system.

The DRB functions as a gate, similar to the Stage Gate model proposed by Markeset and Kumar [16]. This process effectively results in design iterations until the design is optimised and acceptable as illustrated in Figure 5.

Expanding the SD block in Figure 5, the logistic engineering analysts, as part of the FD team, analyse this draft design for the “ility” performance requirements against the specification. The Design Review Board (DRB) refers any shortcomings or deviations from the requirements back to the SD team for another design iteration. This iterative design process continues until the design complies with all the requirements and the design configuration is frozen and placed under configuration control in preparation for the next level of system integration. The number of iterations required is generally determined by the maturity of the technology selected and the technical complexity of the design, [9]. The FD team can only perform the analysis after a concept design has been provided by the SD team. In other words design influencing is an “effect-to-cause” process.

This process, although at configuration item (CI) level, agrees with the successive refinement process in Figure 1. Again the question remains “when is the design acceptable?” This question is not trivial since a number of the design requirements such as reliability can only be verified after extensive qualification Test-Analyse-and-Fix (TAAF) testing [17]. Experienced design review teams normally take a calculated risk based on past experience with similar technologies and designs to expedite the release and baseline of a design.

5.5. Real world design influencing model

1 Any of the engineering “ilities” (e.g., reliability, testability, producibility, supportability) [2].
The SE process by itself cannot bring a system into being. It requires the PM process to structure and manage the systems engineering activities and the consumption of resources, thereby ensuring the delivery of the system to the client on time and within budget. The two processes therefore cannot be separated and must function in an integrated harmonious manner.

5.5.1. Project management team (PM)

A project is “a unique temporary endeavour, with a set beginning and end” and “the application of knowledge, skills, tools and techniques to a broad range of activities in order to meet the requirements of a particular project”,[4].

The project management must satisfy the requirements of the project stakeholders [4]. Therefore the development team objectives amongst others are:

- Successful project within cost and schedule
- Satisfied client
- Satisfied company management

The developed unconstrained design influencing model shown in Figure 5 can be expanded to incorporate the influence of project management.

As discussed in the previous section, PM objectives are different from those of SE and as such can place additional constraints, in particular those of cost and schedule on the design process.

5.6. Development of the real word design influencing model

Expanding Figure 5 and introducing the project management gate, a constrained design influencing model can be developed and is shown in Figure 6. The model adds project management to the design process. Project management is now formally represented on the DRB and can apply its influence to the design process.

Whereas the systems team reviews a concept design from a pure requirements and technical perspective, the project management team reviews a proposed design from a project cost and schedule perspective as well.

Again in the constrained design influencing model, the SD team prepares a concept design, to be analysed by the FD team and submitted to the DRB. The DRB identifies any deviations of the concept design from the specification and if acceptable, the design baseline is fixed and released for further integration into the system, similar to the unconstrained design influencing model in Figure 5.

The iterative design for the constrained process design influencing model is identical to the unconstrained design process, but with the addition of a gate in the iterative design process controlled by the project manager. The project manager, depending on his constraints, generally cost and schedule, can allow design iteration or force a premature design release. The design may therefore not be fully optimised and mature to the satisfaction of the SD and FD teams.

If deviation of the concept design from the specification is identified by the DRB, project management has the final decision whether to allow another design iteration or to force a release of the design for the next level of integration. This increases the risk that problems may occur at the next level of system integration as a result of the prematurely released design.

5.7. Risk mitigation

A premature design release of a component due to PM constraints illustrated in Figure 6 can increase the risk that a latent design defect may surface later in the project. Components with inherent latent design defects very often only surface during the system integration and testing phase. This can be very detrimental to the project cost and schedule since other functionally coupled components may also be affected and forced to change as well.

Design review checklists can be used to mitigate these risks [1]. Design review checklists must be dynamic and must be regularly updated from company management information systems such as a Problem Reporting and Corrective Action System (PRACAS). The checklists must be universal and not project specific. The checklists must be developed to incorporate the lessons learned from not only the present system but also other systems under development as well as experience gained from field data.
The “Stage Gate” model was developed to reduce system development project risk, [10]. The gates ensure that the next phase of the program is not entered before the objectives of the first one have been achieved, confirming the validity of the developed models shown in Figures 5 and 6. The gate ensures that the next step is achievable and the risk of proceeding is acceptable, [1]. This also agrees with the findings by Sommerville, [18].

6. CONCLUSIONS

Underfunding and applying overly stringent and unrealistic schedules to a development project exacerbate the project risk. The general literature view is that apart from minimising the technical risks, ensuring that a project is not under budget and realistic timescales have been set, can reduce the risks of a system development project [2, 19, 20].

The literature also cautions against project underfunding [2, 19, 20]. The rationale for this caution can be deduced from the developed constrained design influencing model shown in Figure 6. A project manager under unrealistic cost and schedule pressure may be forced to take very high risks and release an otherwise unacceptable design.

In practice all that happens is that the problem is shifted to the next level of integration, where the resources required for corrective action become considerably more expensive, primarily due to the ripple effect of the corrective action throughout the system hierarchy.

Underfunding and unrealistic timescales can sometimes lead to the total failure of an otherwise promising system development project, [19].

A model has been developed to better understand why design iterations are fundamental to design. This model has been expanded to a constrained design influencing model that provides a better understanding of the influence of project management in the design process.

The model agrees with the discussed literature and also addresses some of the main shortcomings in the design process. The influence of PM on the design process is very often not taken into account. PM influence on the design process can have a distinct influence on design quality, [21].

This model shows that the project manager, particularly if he is under unrealistic constraints, can force a premature design release for integration to the next system level. The developed model provides a fundamental understanding of the design process.

7. RECOMMENDATIONS

The aim of this study was to develop improved design influencing conceptually from literature. Further evaluation is required with more case studies and cause effect analyses of the SD-FD design influencing concept in a constrained PM environment.

8. REFERENCES


A CCH CYCLE ADAPTIVE ALGORITHM FOR IEEE 802.11P MAC PROTOCOL IN SAFETY APPLICATIONS

L. Miao*, K. Djouani†, B. Van Wyk‡ and Y. Hamam§

*French South African Institute of Technology (F‘SATI), Tshwane University of Technology, Pretoria 0001, South Africa E-mail: miaolusheng@hotmail.com
†French South African Institute of Technology (F‘SATI), Tshwane University of Technology, Pretoria 0001, South Africa E-mail: djouani@gmail.com
‡French South African Institute of Technology (F‘SATI), Tshwane University of Technology, Pretoria 0001, South Africa E-mail: VanWykB@tut.ac.za
§French South African Institute of Technology (F‘SATI), Tshwane University of Technology, Pretoria 0001, South Africa E-mail: yskandar@hamam.ws

Abstract: The IEEE 802.11p standard, especially the 802.11p MAC protocol, has attracted much attention as part of the WA VE protocol in VANETs. Safety applications are very challenging for the design of MAC protocols due to their low latency and high reliability requirements. The CCH interval is also a key parameter for the 802.11p MAC protocol since it can affect the performance of delivering safety messages significantly. In this paper, a simulation based evaluation is proposed firstly to evaluate the performance of the 802.11p MAC protocol with various vehicle densities and CCH interval settings. Based on the simulation results, a CCH cycle adaptive algorithm for the 802.11p is then proposed. In this, the CCH interval is adjusted according to the vehicle density. The simulation results show that the proposed algorithm outperforms the IEEE 802.11p MAC protocol in terms of channel utilization and reliability in safety applications.

Key words: Vehicular ad-hoc Networks, IEEE 802.11p; MAC; safety, delay; collision; reliability, throughput.

1. INTRODUCTION

Vehicular ad-hoc Networks (VANETs) have attracted much attention owing to many present day societies’ transportation problems such as traffic congestion, traffic accidents, lack of mobility and accessibility etc. VANETs present a challenging environment for protocol and application design due to their low latency and high data rate requirements in a high mobility environment. The IEEE 1609 working group has defined the first version of the protocol stack IEEE 802.11p/1609.x protocols [1], also known as WA VE (Wireless Access in a Vehicular Environment). The WA VE protocols are designed for the 5.850-5.925 GHz band, the Dedicated Short Range Communications (DSRC) spectrum band in the United States (US), known as the intelligent transportation systems radio service (ITS-RS). This 75 MHz band is divided into one central control channel (CCH) and six service channels (SCH) as depicted in Fig. 1. The IEEE 802.11p standard [2] defines the physical (PHY) and medium access control (MAC) layers based on earlier standards for Wireless LANs (Local Area Networks). The IEEE 802.11p uses the enhanced distributed channel access (EDCA) MAC sub-layer protocol designed based on the IEEE 802.11e with some modifications, while the physical layer is OFDM (Orthogonal Frequency Division Modulation) as used in IEEE 802.11a.

According to the IEEE 1609.4 coordination scheme [4], as illustrated in Fig. 2, the channel time is divided into synchronization intervals with a fixed length of 100ms, consisting of 50ms (including 4ms guard interval) alternating CCH and SCH intervals. All vehicles stay in the control channel during the CCH period and switch to one of the six service channels during the SCH interval.

However, as stated in [5], this is a significant concern if BSMs (Basic Safety Messages) are constrained to be sent on the CCH during the 50ms CCH interval, since there could be hundreds of devices in a given area and the collision rate could be very high. A special safety Channel

Figure 1: The set of channels defined in the WA VE trial standard [3]

Figure 2: Channel interval
On the other hand, the 50 ms CCH interval could be too long and therefore wasted in a low vehicle density environment. Certain concepts for adapting the intervals of CCH and SCH are proposed in [7] and [8]. In those works, the CCH interval is reduced in order to improve the SCH service, but the authors do not consider extending the CCH interval for a high vehicle density environment in order to reduce the collision probability. In our previous work [10], the performances of the IEEE 802.11p MAC protocol in safety applications with various CCH intervals are evaluated based on INET-2.0.0 [11]. However, due to the limitation of INET-2.0.0, the CCH/SCH switching is not considered in [10]. The main purpose of this paper is consequently to propose a CCH adaptive algorithm for the 802.11p MAC protocol in terms of the safety applications in VANETs, in which the CCH interval is adapted according to the vehicle density environment. In order to obtain a more accurate CCH adaptation threshold, the IEEE 802.11p MAC is reevaluated based on Veins [12] in terms of reliability and delay.

The rest of this paper is organised as follows. In Section II, the related work is discussed. The IEEE 802.11p MAC is reevaluated and the CCH interval adjustment threshold is obtained in Section III. In Section IV, the proposed CCH adaptive algorithm for the 802.11p MAC is presented and evaluated. Finally, the conclusions and possibilities for future studies are provided in Section V.

2. RELATED WORK

The effects of time allocations on CCH and SCH in IEEE 802.11p are analysed in [7] and [8]. In [7], the effect is analysed while the CCH/SCH duty cycle is changing. The results obtained indicate that the performances on CCH and SCH alter significantly following the changing of the CCH/SCH duty cycle. Three various CCH intervals (9, 27 and 45 ms) are considered, and the trade-off between the numbers of users on CCH and SCH are described. However, the effect is not analysed when the interval of CCH is greater than 45 ms. In [8], an algorithm is proposed to improve the channel access scheme by adapting the intervals of CCH and SCH. However, the proposed algorithm cuts off CCH in order to extend SCH; hence it only improves the service channel utilisation without considering that of the control channel.

In [9] a variable CCH interval (VCI) multichannel MAC Scheme is proposed. The VCI MAC scheme adopts a new coordination mechanism to provide contention-free SCHs by a channel reservation on CCH. However, the CCH interval reserved for safety message delivery is not calculated well. Firstly, in the VCI MAC scheme, only the safety message transmission time is counted in the CCH interval calculation. The backoff time should also be counted; hence the time duration needed by the safety message delivery will be much longer than the result obtained in [9]. Secondly, the safety message frequency is considered as 2 per second which is too low to meet the safety application requirement. In addition, in the analytical model only the saturation scenario is considered, which is not suited to the realistic network environment.

It can be observed from the above discussions that no work has been conducted on considering both the increasing and decreasing of the CCH interval according to the vehicle density. In our previous work [10], the performances of the IEEE 802.11p MAC protocol in safety applications with various CCH intervals were evaluated. The 802.11p MAC protocol was evaluated in terms of collision, reliability, delay and throughput with the model of INET-2.0.0 [11]. However, the CCH/SCH switching is not supported by INET-2.0.0. The 4 ms guard interval is also not considered in the INET simulation model. As a result, the simulation results obtained from [10] may not be very accurate for some scenarios. Hence, in this work, the IEEE 802.11p MAC protocol is reevaluated in order to determine a more accurate CCH interval adjustment threshold.

3. PERFORMANCE EVALUATION

In this section, the performance of the IEEE 802.11p MAC protocol in safety applications is evaluated. Its performance with various CCH intervals is focused on.

3.1 Evaluation Configurations

To build an accurate and realistic simulation model, the latest released Veins 2.0-rc2 [12] is used, which is based on MiXiM 2.2 [13] working on OMNeT++ [14] platform and features the new models of IEEE 802.11p and IEEE 1609.4 DSRC/WAVE protocols. In Veins 2.0-rc2, the HCF (Hybrid coordination function) backoff process [2], WSM (Wave Short Message) application [15] and CCH/SCH switching function are all properly considered. Compared with our previous work in [10], the Veins model is more accurate and realistic for the performance of the 802.11p.

In the simulations conducted, as shown in Fig. 3, a bi-directional highway segment of length 2000 m is considered. A number of vehicles varying from 10 to 100 are deployed on the highway randomly. The speed of vehicles ranges from 60 to 120 km/h. Each vehicle will return once it reaches the border of the field in order to make sure that every vehicle remains in this highway segment. In this scenario, assume a vehicle (the red vehicle) is stopped right at the middle of this segment due to an accident. The accident vehicle broadcasts one emergency message every 100 ms in AC3. Each of the other vehicles sends one status safety message every 100 ms in AC3. The safety messages are generated randomly during [0, T_{CCH}] for each vehicle, where T_{CCH} denotes the CCH interval. The T_{CCH} is set as [20, 30, 40, 50, 60, 70, 80, 90, 100] respectively in order to evaluate the performance of 802.11p MAC at various CCH intervals. As defined in [16], the message size for the BSM part I is 39 bytes and for Part II, the VehicleSafetyExtension frame of the BSM, which is less than 100 bytes. Hence, the emergency and status safety message sizes are set as 139 bytes (100+39) and 39 bytes respectively. The communication range R is set as 1000m since the expected
radio range for a highway is up to 1000m [17]. The value of parameters used in the simulation is depicted in Table 1.

3.2 Evaluation Results

Fig. 4 depicts the average status safety message delivery delay for various contention vehicles and CCH intervals. It can be observed that the delay increases according to the increase in number of contention vehicles and a decrease of the CCH interval. The delivery delay $T_{\text{Delay}}$ is the time duration from when the safety message is generated till the time that the message is received by a vehicle. The $T_{\text{Delay}}$ is determined as

$$T_{\text{Delay}} = \frac{S_{\text{Status}} + S_{\text{MAC}}}{R_{\text{Data}}} + T_{\text{Queue}} + T_{\text{Backoff}} + \delta,$$

where $\delta$ is the propagation delay, and $T_{\text{Queue}}$ and $T_{\text{Backoff}}$ denote the Queue time and Backoff time respectively. It can be observed that the $T_{\text{Delay}}$ is mostly determined by $T_{\text{Queue}}$ and $T_{\text{Backoff}}$ since the other parameters such as $S_{\text{Status}}, S_{\text{MAC}}$ and $R_{\text{Data}}$, are constant. $\delta$ is also a approximate constant in this short range communication scenario. The increase in numbers of contention vehicles as well as the decrease in the CCH intervals will cause more contention together with a longer Queue and Backoff time. As a result, a lengthier delay is caused.

Fig. 5 illustrates the average emergency safety message delivery delay for various contention vehicles and CCH interval. It can be observed that the delay does not change much according to the contention vehicles and CCH intervals. Since only one emergency safety message is broadcast every CCH interval in AC3, neither the contention vehicles nor the CCH interval affect the delay significantly. The delay is quite short and stable since the emergency safety message, which is transmitted in AC3, always has the highest priority to access a channel.

Fig. 6 portrays the average status of the delivery reliability of safety messages for various contention vehicles and CCH intervals. It can be observed that the reliability decreases according to an increase in numbers of contention vehicles and a decrease in CCH intervals. The reliability is strongly related to the collision probability. Since each vehicle sends one status safety message every CCH interval, the increase in contention vehicles means that more vehicles will contend for the idle slots. As a result, more collisions will be caused. The decrease of CCH intervals will also cause more collisions since the total idle slots are decreased. As a result, the probability that more vehicles will contest an idle slot at the same time increases. More collisions will cause worse reliability. According to the requirements for safety applications in VANETs [18] [19], the delay should be less than 100ms. However the reliability requirement is not defined precisely. In this work, considering a specific reliability requirement, the network capacity represented by the number of vehicles that can be accommodated in a network can be derived from Fig. 4 and Fig. 6. The results obtained are listed in Table 2. It is
Table 1: Parameter settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status safety message access Class AC</td>
<td>AC0</td>
</tr>
<tr>
<td>Emergency safety message access Class AC</td>
<td>AC3</td>
</tr>
<tr>
<td>aSlotTime</td>
<td>13 $\mu$s</td>
</tr>
<tr>
<td>Size of emergency safety message $S_{\text{emergency}}$</td>
<td>140 bytes</td>
</tr>
<tr>
<td>Size of status safety message $S_{\text{status}}$</td>
<td>39 bytes</td>
</tr>
<tr>
<td>Size of MAC header $S_{\text{MAC}}$</td>
<td>32 bytes</td>
</tr>
<tr>
<td>Data rate $R_{\text{data}}$</td>
<td>6 Mbps</td>
</tr>
<tr>
<td>Communication range $R$</td>
<td>1000 m</td>
</tr>
<tr>
<td>Maximum backoff time $k$</td>
<td>5</td>
</tr>
<tr>
<td>Message sending frequency</td>
<td>10/s</td>
</tr>
<tr>
<td>Simulation time</td>
<td>10 s</td>
</tr>
<tr>
<td>Maximum transmission power</td>
<td>760 mw</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>-82dBm</td>
</tr>
<tr>
<td>Mobility model</td>
<td>LinearMobility</td>
</tr>
<tr>
<td>Propagation model</td>
<td>SimplePathlossModel</td>
</tr>
</tbody>
</table>

Fig. 7: Reliability of emergency message
evident that the reliability is much more critical than the delay. Hence, the reliability constitutes the bottleneck for the IEEE802.11p MAC protocol in meeting the safety application requirements in VANETs. Compared with our previous work in [10], it can be observed that the network capacity achieved in this work is mostly less than the results obtained in [10] because in the Veins simulation model, the CCH/SCH switching is considered and hence, the 4ns guard interval is included in the $T_{\text{CCH}}$. As defined in [2], no message should be transmitted during the guard interval. However, in our previous work, the guard interval is not considered. As a result, the network capacity obtained in this work is less than our previous result in [10].

Fig. 7 indicates the average delivery reliability of emergency safety messages for various contention vehicles and CCH intervals. It can be observed that the reliability decreases according to any increase in contention vehicles and decrease in CCH intervals. As discussed before, the reliability is closely related to the collision probability. Although there is only one emergency transmitted for every CCH interval, the probability of collision with the status safety message will increase according to an increase in contention vehicles and a decrease in CCH interval. As a result, more collisions cause worse reliability. However, the emergency safety messages delivery reliability is always 100% for the scenarios presented in Table 2.

In conclusion, in VANETs safety applications, the CCH interval can be adjusted according to the number of contention vehicles in order to improve the channel utilisation ratio. In the mean time, the safety application requirement should be satisfied since the safety application is very critical in VANETs.

4. CCH ADAPTIVE ALGORITHM

In the former section, it is concluded that the channel utilisation ratio can be improved via adjusting the CCH interval with guaranteed, certain reliability. In this section, a CCH adaptive algorithm is proposed, in which the CCH interval is adjusted according to the number of contention vehicles.

4.1 Proposed Algorithm

As shown in Fig. 8, the basic concept of the proposed algorithm in this work is to adjust the CCH interval parameter according to the average number of contention vehicles in a highway segment. The number of contention vehicles is recorded by each RSU and collected at
Table 2: Accommodated number of vehicles

<table>
<thead>
<tr>
<th>$T_{CCH}(\text{ms})$</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of vehicles ($P_{\text{Reliability}} \geq 99%$)</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>Number of vehicles ($P_{\text{Reliability}} \geq 95%$)</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3: Accommodated number of vehicles [11]

<table>
<thead>
<tr>
<th>$T_{CCH}(\text{ms})$</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of vehicles ($P_{\text{Reliability}} \geq 99%$)</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>30</td>
<td>30</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>Number of vehicles ($P_{\text{Reliability}} \geq 95%$)</td>
<td>25</td>
<td>35</td>
<td>45</td>
<td>55</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>125</td>
</tr>
</tbody>
</table>

**Algorithm 1** Contention vehicle recording algorithm

1: switch to CCH interval
2: $N_i \leftarrow$ the number of contention vehicles
3: if received a safety message successfully then
4: $N_i = N_i + 1$
5: else if failed to receive a message due to collision then
6: $N_i = N_i + 2$
7: end if
8: switch to SCH interval
9: send $N_i$ to upper layer

Contention vehicle recording: In this algorithm, the CCH interval is set according to the number of contention vehicles in a VANET. Hence, the contention vehicle number $N_i$ needs to be recorded by each RSU, where $N_i$ denotes the number of contention vehicles recorded by RSU $i$ during one CCH interval. In this scheme, each RSU records the number of vehicles according to the safety messages received for each cycle. The detail of the contention vehicle recording algorithm is shown in Algorithm 1. In order to simplify the algorithm, the collision is assumed to occur only between two vehicles. As a result, the number of contention vehicles is recorded as 2 for a collision. The $N_i$ will be finally sent to the control centre in order to calculate the proper CCH interval parameter according to the average number of contention vehicles in a highway segment.

CCH interval calculating:

In this algorithm, as shown in Fig. 8, the control centre is responsible for calculating the CCH interval according to the number of contention vehicles. As discussed in the former section, the $N_i$ recorded by RSU $i$ for each CCH interval is collected in the control centre periodically. Suppose that a highway segment contains $j$ RSUs; the average number of contention vehicles $N_{\text{ave}}$ can then be determined as

$$N_{\text{ave}} = \frac{1}{j} \sum_{i=1}^{j} N_i$$

(2)

The CCH interval parameter $T_{CCH}$ can then be determined according to $N_{\text{ave}}$ and Table 2. Suppose the reliability requirement is 95%; the details of the CCH interval calculating algorithm are shown in Algorithm 2.

**Algorithm 2** CCH interval calculating algorithm

1: $N \leftarrow 0$
2: $N_{\text{ave}} \leftarrow 0$
3: $j \leftarrow$ the number of RSUs
4: for $i = 1; i \leq j; i++$ do
5: receive $N_i$ from RSU $i$
6: $N = N + N_i$
7: end for
8: $N_{\text{ave}} = \frac{N}{j}$
9: if $N_{\text{ave}} \leq 20$ then
10: $T_{CCH} = 20$
11: else if $20 < N_{\text{ave}} \leq 25$ then
12: $T_{CCH} = 30$
13: else if $25 < N_{\text{ave}} \leq 30$ then
14: $T_{CCH} = 40$
15: else if $30 < N_{\text{ave}} \leq 40$ then
16: $T_{CCH} = 50$
17: else if $40 < N_{\text{ave}} \leq 50$ then
18: $T_{CCH} = 60$
19: else if $50 < N_{\text{ave}} \leq 60$ then
20: $T_{CCH} = 70$
21: else if $60 < N_{\text{ave}} \leq 70$ then
22: $T_{CCH} = 80$
23: else if $70 < N_{\text{ave}} \leq 80$ then
24: $T_{CCH} = 90$
25: else if $N_{\text{ave}} > 80$ then
26: $T_{CCH} = 100$
27: end if
28: broadcast $T_{CCH}$ to all RSUs

CCH interval resetting:

As stated in [15], in WAVE, each vehicle is expected to...
Algorithm 3 CCH interval resetting algorithm

1: start timer \( t \leftarrow T_{CCH} \)
2: receive a WSA
3: \( T_{CCH} \leftarrow \text{CCH interval parameter in WSA} \)
4: if \( t=0 \) then
5: switch to SCH
6: end if

join a WBSS (WAVE Basic Service Set), which is a unique identifier for each communication zone. Vehicles must associate with only one WBSS at a time. To establish a WBSS, a RSU periodically broadcasts the WAVE Service Advertisement (WSA) on the CCH. WSA contains the necessary information for the users to join the WBSS, such as the WBSS identifier, the availability of a service, the selected SCH, synchronisation timing information etc. In the meantime, all the vehicles have to listen to the CCH during the CCH intervals. In our proposed CCH adaptive algorithm, the WSA is extended to contain a CCH interval parameter \( T_{CCH} \). The \( T_{CCH} \) is calculated by the control centre and distributed to every RSU every 100 ms. Each vehicle will reset its CCH interval according to the \( T_{CCH} \) contained in WSA. However, the new CCH interval parameter will be activated during the next cycle since the vehicle is already on a CCH channel. The detail of the CCH interval resetting algorithm is depicted in Algorithm 3.

4.2 Validation and Discussions

The proposed CCH adaptive algorithm is evaluated utilising the same simulation configuration as discussed in Section 3.1. The CCH adaptive algorithm is evaluated for two reliability requirement settings respectively. As discussed in Section 3.2, for emergency messages, the delay is quite short and stable. In addition, the reliability is always 100% for the scenario in Table 2. Hence, only the status messages are discussed in this section.

Fig. 9 illustrates the CCH interval settings for various contention vehicles. In the IEEE 802.11p, the CCH interval is a constant with 50 ms. In the proposed CCH adaptive algorithm, the CCH interval is adjusted according to the number of contention vehicles in a network. According to Table 2, the CCH adjusting thresholds are different for different reliability requirements. It can be observed that the CCH interval for the adaptive algorithm increases according to the number of contention vehicles in order to meet the reliability requirement. In the scenario where the network density is low, a shorter CCH interval (< 50 ms) is sufficient for the status safety messages to meet a certain reliability requirement. As a result, a longer SCH interval (> 50 ms) is obtained to offer a better service to the customer. The SCH interval can be extended up to 60%. On the other hand, in the scenario where the network density is high, the network needs a longer (> 50 ms) CCH interval to deliver the status safety messages in order to meet the reliability requirement.

Figure 9: CCH interval

Figure 10: Reliability of status message

Figure 8: CCH interval adaptive example
Fig. 10 depicts the average delivery reliability status of safety messages for various contention vehicles. It can be observed that the CCH adaptive algorithm outperforms the 802.11p in terms of reliability when the number of contention vehicles is greater than a specific threshold. The threshold is $N_{\text{Threshold}} = 15$ and $N_{\text{Threshold}} = 30$ for the two reliability requirements ($\geq 99\%$ and $\geq 95\%$) respectively. On the other hand, the 802.11p outperforms the CCH adaptive algorithm in terms of reliability when the number of contention vehicles is less than the threshold $N_{\text{Threshold}}$. It can be noted that the threshold $N_{\text{Threshold}}$ is just the point that the $T_{\text{CCH}}$ is switching to be $50\text{ms}$ according to Table 2. However, the reliability of the CCH adaptive algorithm is always better than the reliability requirement when the number of contention vehicles does not exceed the maximum network capacity $N_{\text{Max}}$, which is $N_{\text{Max}} = 55$ and $N_{\text{Max}} = 100$ for the two reliability requirements ($\geq 99\%$ and $\geq 95\%$) respectively.

In Fig. 11 the average status safety message delivery delay for various contention vehicles is illustrated. It can be observed that the delay shows a similar distribution to that of the reliability. The CCH adaptive algorithm records a slightly longer delay than the 802.11p when the number of contention vehicles is less than the threshold $N_{\text{Threshold}} = 15$ and $N_{\text{Threshold}} = 30$ for the two reliability requirements ($\geq 99\%$ and $\geq 95\%$) respectively. On the other hand, the CCH adaptive algorithm has a slightly shorter delay than the 802.11p when the number of contention vehicles is greater than the threshold $N_{\text{Threshold}}$. It can also be observed that all the delays are much shorter than the latency requirement ($< 100\text{ms}$).

In conclusion, the CCH adaptive algorithm can improve the channel utilisation ratio for a low traffic scenario (up to 60%) and increase its reliability for a heavy traffic scenario via adjusting the CCH interval. In the meantime, the average delivery delay is much shorter than the latency requirement in the scenarios considered.

5. CONCLUSIONS

In this study, a CCH adaptive algorithm for the IEEE 802.11p MAC protocol is proposed, in which the CCH interval is adjusted according to the average number of contention vehicles in a VANET. In comparison with the existing works in [7] [8] [9], the safety application’s reliability, rather than that of the SCH service, is considered and guaranteed a higher priority. In order to establish a more accurate and realistic CCH adjustment threshold, the 802.11p MAC protocol is evaluated based on the latest released Veins 2.0-rc2 model. The network capacity with a certain reliability requirement is derived, which is taken as the CCH adjustment threshold in the CCH adaptive algorithm. The simulation results demonstrate that the CCH adaptive algorithm can improve the channel utilisation ratio (up to 60%) for a low traffic scenario and increase the reliability for a heavy traffic scenario via adjusting the CCH interval.

Some of the future studies will include: (1) proposing a analytical model to evaluate the performance of the IEEE 802.11p MAC protocol with different CCH/SCH interval settings; and (2) evaluating the network capacity of IEEE 802.11p standard for SCH interval.

ACKNOWLEDGMENT

The authors would like to thank Dr. David Eckhoff (Department of Computer Science, University of Erlangen, Germany) and Dr. Christoph Sommer (Institute of Computer Science, University of Innsbruck, Austria) for helping to develop the simulation model based on Veins.

REFERENCES


SPATIAL-SPATIAL WATERMARKING SCHEME FOR JPEG STEGANOGRAPHY

B. Marques, B. Sim and L. Cheng *

* School of Electrical and Information Engineering, University of the Witwatersrand, Private Bag 3, Wits. 2050, Johannesburg, South Africa.
Emails: bradley.marques@students.wits.ac.za; benjamin.sim.za@ieee.org; ling.cheng@wits.ac.za

Abstract: Steganography is the art of concealing the presence of a communication, across an open channel, from undesired parties. This is achieved by the encoding of message data in innocuous cover objects, such as JPEG digital images. Many JPEG steganography algorithms exist; most perform the encoding in the spectral domain of the image. A new steganographic method, deemed the Spatial-Spectral Watermarking Scheme (SSWS), has been developed and utilises both the spatial and the spectral domains for the encoding process. A developed mathematical model of the scheme predicts a maximum gain in terms of steganographic capacity of 4.9 over an existing algorithm, JSteg. Experimental data obtained from a software prototype confirms this result. Furthermore, the experimental data confirms that the securities in the spatial domain and the spectral domain are inversely proportional to the information amounts embedded in these two domains.

Key words: Data hiding, Information theory, JPEG, Steganography, Steganalysis, Watermarking.

1. INTRODUCTION

Steganography is a field of information theory involving the concealment of the presence – not merely the content – of communication between two parties from an outside observer [1–5]. In modern digital steganography, this is achieved by hiding the message data within innocuous cover media such as image, audio or video files. The Joint Photographic Experts Group (JPEG) digital image format has become a de facto standard in the field of digital photography and is prevalent on the World Wide Web [6]. It thus offers a perfect platform for digital image steganography. The applications of JPEG steganographic systems include covert communications, embedding of image metadata within the image itself, and digital image watermarking for copyright protection.

Existing JPEG steganographic schemes – such as JSteg, OutGuess and J3 – utilise the spectral domain of the JPEG file to conceal secret message information [7, 8]. A new method of JPEG steganography, the Spatial-Spectral Watermarking Scheme (SSWS), which utilises both the spectral and the spatial domains of a JPEG image has been designed, developed and investigated. The linking of the two domains is made possible through the use of Forward Error Correction (FEC), allowing errors introduced by the dual encoding to be recovered. By utilising both domains of the image, the capacity and security of the encoding process is increased.

The remainder of this paper is organised as follows: relevant background information on JPEG steganography is presented in Section 2. In terms of steganographic concepts, existing JPEG steganographic systems and steganalysis. Concepts of FEC utilised in SSWS are also highlighted. The design of the SSWS is then examined in Section 3. A mathematical model describing the SSWS is presented in Section 4. Experimental data collected to verify this model in terms of capacity and security are also presented in this section. Finally, a conclusion to the paper is presented in Section 5.

2. PRELIMINARIES

2.1 Steganography and Steganalysis

Steganography is the art of hiding a secret message in an innocuous medium, deemed the cover object. Steganography focuses on the concealment of the presence of communication - not necessarily the content thereof, as is the case with cryptography [1,3,5]. The message data are embedded into the cover to produce the stego object [3]. The strength of digital image steganographic systems is defined by three main criteria: capacity, security and robustness. Capacity is a measure of the amount of information encodable within a cover image. Security represents an algorithm’s resistance to a steganalytic attack [1, 3, 9]. Such an attack may be targeted or blind: targeted techniques use method-specific limitations or features that are introduced during encoding to detect the presence of a message: blind, or universal, techniques use statistical attacks, such as the chi-squared test, to detect message presence [1, 2, 10]. Other techniques, deemed quantitative staganalysers, use statistical methods not only to detect the presence of the message but to also estimate its length [7, 9]. Steganographic robustness defines a technique’s ability to recover a message after tampering. Such tampering may be active – intentionally altering the content of an embedded message to disrupt the steganography – or passive – such as cropping or blurring an image [11]. However, robustness is not a factor of the encoding algorithm alone, but is also dependant on any FEC techniques applied to the data prior to embedding. Thus, it will not be considered further in this paper.
2.2 Existing Steganographic Techniques

Existing JPEG steganographic techniques typically perform the encoding process in the frequency domain of the image [3]. Thus, changes are elicited in the spectral domain – or the Discrete Cosine Transform (DCT) coefficients – of the JPEG image. Additionally, many techniques utilise the concept of Least Significant Bit (LSB) encoding [2, 3]. This minimizes the change to the image, since only the selected DCT coefficient’s LSB is altered, not the entire value. For instance, the JSteg algorithm sequentially replaces each DCT coefficient’s LSB with the required message bit [2, 3]. The OutGuess 1.0 algorithm relies on a similar scheme as JSteg, but the selection of DCT coefficients in which to embed is more advanced; the algorithm performs a pseudorandom walk over the entire image spectral domain, embedding the message at the visited locations [3, 7, 12]. The seed of the pseudorandom number generator thus becomes a private key for the communication [13]. To increase algorithm security against statistical blind steganalysis, methods such as OutGuess 2.0 and J3 attempt to maintain histogram neutrality during the embedding process [3, 8]. This is achieved by introducing counteractive changes for every change that is introduced during embedding.

2.3 Forward Error Correction

FEC is a type of error correcting code used in the field of telecommunications, and involves the addition of procedurally generated redundant information to a message for the purpose of noise and data loss insensitivity [14]. The redundancy is procedurally generated from the message data and enables the receiver to recover the sent message in the event of induced errors or erasures in the received message [14, 15]. Various FEC techniques exist which offer varying levels of error and erasure correcting capabilities.

3. THE SPATIAL–SPECTRAL WATERMARKING SCHEME

The existing algorithms discussed in Section 2.2 define the actual process of data embedding. The developed SSWS, however, does not. Rather, it defines a scheme to improve the performance of any existing steganographic algorithm. SSWS defines a protocol for the encoding process whereby message data are encoded within both the spectral and the spatial domains of the image. Any existing or new technique may be used as an underlying algorithm.

The SSWS encoding process, illustrated in Figure 1, initially interleaves the message. The interleaved message is divided into two sections, \( m_1 \) and \( m_2 \) defined by a splitting factor, \( e \in [0, 1] \). Since the SSWS embeds in both spatial and spectral domains, FEC encoding is required to facilitate error recovery. The \( m_1 \) message section undergoes FEC encoding, which adds parity information to facilitate error recovery in the event of the spatial embedding introducing errors into the spectral message section. The FEC technique used was a block code with the code rate \( r \). The encoded \( m_1 \) is then embedded (by the underlying algorithm) within the spectral domain. The spectral cover image is losslessly decompressed to generate a spatial image. The remainder of the message, the \( m_2 \) message section, is then embedded into this spatial image by the same underlying steganographic algorithm.

By utilizing both domains, the SSWS can offer improved capacity and security (see Section 4.). The capacity is increased as it uses two separate domains into which the information can be embedded. Furthermore, SSWS offers a two-fold improvement to security. Firstly, by splitting the message content between two domains, the message content (in bits) is reduced within each domain for a given payload. Thus, steganalytic attacks targeting either domain have less information to detect. Secondly,
by performing the double encoding, the first layer of information is effectively obscured by the second encoding process. Thereby, improving the systems resistance to spectral based steganalysis techniques. Note that this scheme can be easily modified and applied to a McEliece public-key cryptosystem, thanks to the FEC process used to bridge between the two domains.

4. RESULTS AND ANALYSIS

4.1 Mathematical Model of SSWS Capacity

In order to develop a mathematical model of the SSDDW scheme, it is necessary to define the capacity gain, \( g \), that the technique offers over its underlying algorithm as

\[
g = \frac{m' - m}{m},
\]

where \( m' \) is the maximum length of embeddable message by the SSWS, and \( m \) is the maximum length of embeddable message using the standard algorithm. The message \( m' \) is split by the ratio \( e \) into two smaller messages, as shown in Figure 1. As mentioned in Section 3., the component embedded in the spectral domain undergoes FEC encoding with a code of rate \( r \). The two message sections, \( m_1 \) and \( m_2 \) are thus written as

\[
m_1 = \frac{em'}{r}
\]

and

\[
m_2 = m'(1 - e)
\]

Two independent constraints are present in the system: the capacity of the spatial domain and the capacity of the spectral domain. It is evident that for large images, using LSB embedding techniques, a maximum of one bit per image pixel can be embedded in the spatial domain. Hence,

\[
m'(1 - e) \leq p,
\]

where \( p \) is the number of pixels in an image (including all three colour channels). Similarly, it is known that for the standard algorithm, the maximum embeddable message \( m \) is

\[
m = p\alpha.
\]

From experimental results, it was found that in the spectral domain there is a linear relationship between image size, \( p \), and algorithm capacity. This was found to hold true for all investigated embedding techniques, as may be seen in Figure 2. The relationship between \( p \) and spectral capacity is thus a constant value, \( \alpha \) (in bits per pixel) for a given embedding algorithm. The constraint thus placed on the spectral domain capacity is:

\[
\frac{em'}{r} \leq p\alpha.
\]

By solving (1) for \( m' \), substituting into (4), and using (5), a maximum bound is found for \( g \) as follows

\[
g \leq \frac{p}{\alpha(1 - e)} - 1.
\]

Similarly, by using (6), the following constraint is found

\[
g \leq \frac{r - e}{e}.
\]

Combining these two constraints in (7) and (8) leads to the upper bound of the capacity gain, \( g \), of the SSWS system as

\[
g \leq \min \left( \frac{p}{\alpha(1 - e)} - 1, \frac{r - e}{e} \right).
\]

As mentioned in Section 2.1, capacity is a defining characteristic of steganographic systems. The upper bound of (9), was used with a known \( \alpha \) (for the JSteg technique) to produce the surface plot of Figure 3, which shows the theoretical maximum gain as a function of the message split percentage \( e \) and FEC code rate \( r \). As may be seen, the estimated maximum gain is around 4.9 at a message split ratio of 0.16. This theoretical maximum gain is obtained with an FEC code rate of 1 (no parity information added). However, this is impractical since this theoretical maximum assumes that no errors are introduced by embedding in the spectral domains (see Section 3).
Experiments were conducted to validate the model. The maximum embeddable message length (capacity) was determined as a function of the message split ratio \( e \) for a given technique and code rate. The JSteg technique, which has an \( \alpha \) value of 0.203 bits per pixel (refer to Figure 2) was used with a Reed-Solomon code of rate \( \frac{4}{7} \).

The empirical data collected in addition to the predicted bounds of the model (9), are shown in Figure 4. As may be seen, the experimental results are in accordance with the bounds of the theoretical model. As shown, with 7% of the message embedded in the spectral domain, the SSDDW scheme provides a maximum gain of 4.4 over the standard JSteg algorithm. This gain in capacity is offset by a decrease in algorithm security, which is examined in the next subsection.

![Figure 3: Predicted results for capacity gain](image)

Figure 3: Predicted results for capacity gain

As mentioned in Section 2.1, the security of embedding – the degree to which the technique resists steganalysis – is a defining characteristic of steganographic systems. A few measures, targeting both the spectral and spatial domains, were used to test the security of the SSWS. Since the method utilises both spectral and spatial domains, security must be tested in both of these domains.

The Kullback-Leibler (KL) distance is a metric for steganographic security which quantifies the difference between the cover and stego image histograms. It is thus a measure of the change induced in the spectral domain during message embedding. An experiment was performed in which the KL distance was measured for a stego image produced from embedding a message of fixed length (10 kbit) into a test cover image, under a varying percentage split \( e \). The obtained KL distances are inverted (since lower KL distances indicate a more secure system) and are expressed as a percentage improvement over the inverse KL distance obtained with the standard underlying algorithm, JSteg. The experimental results obtained may be seen in Figure 5. As may be expected, the security of the technique decreases with increasing \( e \). This is the case since, as more message is encoded into the spectral domain, steganalysis techniques targeting this domain are more likely to detect message presence.

![Figure 4: The gain in capacity of the SSWS system, using JSteg, for different message split ratios \( e \).](image)

Figure 4: The gain in capacity of the SSWS system, using JSteg, for different message split ratios \( e \). The theoretical bound detailed in Section 4.1 are shown to accurately predict the measured behaviour.

![Figure 5: The gain in spectral security of the SSWS, using JSteg, for different message split ratios \( e \).](image)

Figure 5: The gain in spectral security of the SSWS, using JSteg, for different message split ratios \( e \).

The field of LSB steganalysis is a mature field, with algorithms with proven performance [16–18]. Thus, security testing in the spatial domain was performed using the Sample Pairs (SP) analysis as presented in [18]. A set message size and set message content was used for the experiment. The message split \( e \) was varied, and measurements of the estimated message length taken. These results were then compared to the message length estimated by the SP analysis for a stego image produced by JSteg under the same conditions. The relative gain in security afforded by the SSWS was thus determined. A linear regression trendline of the obtained data, as illustrated by Figure 6 was determined. As may be seen,
the SSWS has poor security in the spatial domain. The results indicate that the spatial security increases as more data is embedded in the spectral domain, as may be expected.

Thus, whilst the SSWS increases both the capacity and spectral domain security of this steganographic technique, the encoding of data in the spatial domain is highly vulnerable to steganalysis techniques such as the SP analysis. The choice of message splitting ratio $e$ must thus depend on the application, and whether capacity or security is important.

![Figure 6: The gain in spatial security of the SSWS, using JSteg, for different message split ratios $e$.](image)

5. CONCLUSION

Steganography is the concealment of secret information in innocuous cover media such that the very presence of the communication is hidden. This paper presents a new technique deemed the Spatial-Spectral Watermarking Scheme (SSWS) which uses a spatial-spectral embedding process to improve steganographic performance. The message is split across both the spatial and the spectral domains of the image and is embedded by an underlying stenographic algorithm. FEC is used to recover from errors introduced in the embedding process. The algorithm was tested in terms of its capacity and security relative to its underlying technique. Through the development of a mathematical model and experimental testing, the capacity gain afforded by use of the SSWS was shown to be 4.4 over the existing JSteg algorithm. This result was obtained with 7% of the message embedded in the spectral domain, and the remaining 93% embedded in the spatial domain. The splitting technique of SSWS improves the steganographic security since the experimental results (refer to Figure 5 and Figure 6) show that the securities in both domains are inversely proportional to the information amount embedded in either domain.

ACKNOWLEDGEMENT

The authors would like to acknowledge the CeTAS group and Prof Ivan Hofsajer of the University of the Witwatersrand as well as Phil Sallee [19], Rick Rosson [20] and Jessica Fridrich [21] for various MATLAB functions used in the prototype software.

REFERENCES


